جامعة بيرزيت
BIRZEIT UNIVERSITY
Faculty of Graduate Studies and Research كلية الدراسات العليا والأبحاث

Department of Applied Statistics and Data Science

Video Popularity Duration Prediction for Videos Posted on YouTube using Deep Learning Algorithms: Case Study USA Popular Videos.

Supervisors:

Dr. Ismail Khater **Dr. Mohammed Hussein**

Submitted by: **Student ID. No.:**

Natalie Abdallah **1205313**

This thesis was submitted in partial fulfillment of the requirements for The Master's degree in

Applied Statistics and Data Science

June/2023

Video Popularity Duration Prediction for Videos Posted on YouTube using Deep Learning Algorithms: Case Study USA Popular Videos.

Submitted by:

Natalie Abdallah

This thesis was defended successfully on …… and approved by:

Committee Members	Signature
1. Supervisor Name: Dr. Ismael Khater	……..
2. Co-supervisor Name: Dr. Mohammad Hussien	……..
3. Internal Examiner Name:	……..
4. External Examiner Name:	……..

توقع مدة شعبية الفيديو لمقاطع الفيديو المنشورة على الYouTube باستخدام خوارزميات التعلم العميق: دراسة حالة مقاطع الفيديو الشعبية في الولايات المتحدة

إعداد

نتالي سعيد عبدالله

نوقشت هذه الرسالة بتاريخ: ..... وأجيزت.

أعضاء لجنة المناقشة:

أعضاء اللجنة التواقيع

د. إسماعيل خاطر مشرفًا ورئيسًا .....

د. محمد حسين مشرفًا ورئيسًا .....

..... ممتحنًا داخليًا .....

..... ممتحنًا خارجيًا .....

Declaration

I hereby declare that the thesis entitled “Video Popularity Duration Prediction for Videos Posted on YouTube using Deep Learning Algorithms: Case Study USA Popular Videos”, under the supervision of Dr. Ismail Khater and Dr. Mohammad Hussein is an authentic work of my own, except for quotations and citations of the-state-of-the-art work as referenced in the thesis.

Date: 13-06-2023

Name: Natalie Abdallah Signature:

Abstract

Every day 3.7 million videos are posted on YouTube for the world to watch. Right after Google, YouTube is ranked as the second most viewed website where many content creators, stakeholders, businesses, marketers, and platform administrators strive to benefit from trending videos that are channeled through this platform worldwide. Therefore, understanding the dynamics of YouTube videos and their popularity lifecycle is an important challenge. This thesis seeks to provide a machine learning and statistical framework for understanding the key features behind videos going viral after being published on YouTube. Namely, a variant of the recurrent neural network models known as gated recurrent unit (GRU) has been trained, validated, and then tested on a dataset of 120k instances of metadata and text data that was feature engineered into numeric, categorical, natural language processed features and temporal features, to well-predict the number of days a popular YouTube video will remain trending. This GRU model showed a goodness of fit (R-squared) equal to 0.767 and a mean squared error (MSE) equal to 0.755 day^2 and root mean squared error (RMSE) of 0.869 day (e.g., the predicted popularity duration using this model has an RMSE error of 20.76 hour error which is effectively less than a day of error). A comparative analysis has also advocated for this GRU model to surpass other enhanced/classical models (e.g., XGBoost, gradient boosted decision trees, random forest, support vector regression (SVR) and linear regression) by means of scoring a better R-squared, MSE, root mean squared error (RMSE), and mean absolute error (MAE). Furthermore, applying dimensionality reduction on the engineered feature set demonstrated how the video trending lifecycle is fundamentally influenced by engagement gains and gain rates derived from the daily change in the video views, comments, and likes, as well as total engagement, likes ratio and trend lag. Overall, this thesis enhanced our understanding of YouTube's data dynamics and demonstrated for content creators the key potential of the GRU model in predicting the trending duration of their videos with 0.869 of a day error.

الملخص

في كل يوم يتم تحميل ونشر ما يقارب 3.7 مليون مقطع فيديو على موقع YouTube بحيث أنه أصبح هذا الموقع يعتبر ثاني أكثر المواقع الإلكترونية زيارة بعد موقع Google مباشرة. مما يجعل صانعي المحتوى، ورجال الاعمال، والمروجين والمالكين أكثر التفافا لمثل هذا المحتوى الذي يتم طرحه بصور متداولة وبكثرة خلال زمن الانتشار. لهذا إن الرسالة تعنى بتقديم إطار عمل إحصائي وآلي لفهم سلسلة حياة مقاطع الفيديو الشعبية. وعليه فإن الخوارزمية المقترحة لتوقع عدد الأيام المتبقية لمقطع فيديو أن يظل شعبيا مبنية على أسس الشبكات العصبية المتكررة والتي تسمى اختصارا (GRU). أن نتائج تدريب هذه الخوارزمية على معلومات كتابية ورقمية والأخذ بعين الاعتبار التسلسل الزمني أدى إلى الحصول على خوارزمية قادرة على تفسير 76.7%(R squared) من التغيير في قيم الأيام المتبقية للشعبية. حين مقارنتها بالخوارزميات الأخرى أظهرت أعلى مؤشر ملائمة (R squared) بينما كانت قيمة متوسط تربيع الخطأ بالمسافة (MSE) لهذه الخوارزمية يساوي 0.755 يوم^2 وجذر تربيع الخطأ بالمسافة (RMSE) مساوٍ الى 90.86 يوم. مما يجعل الخطأ في أداء الخوارزمية بتوقع عدد الأيام المتبقية للشعبية أقل من يوم واحد. بالإضافة إلى ذلك، عند القيام بتطبيق آلية تقليل عدد المتغيرات التي يتم اعتبارها في تدريب وتحسن الخوارزميات واختيار أفضلها، فإنه قد تبين أن أعلى علاقات تربط بين عدد أيام الشعبية المتبقية والمتغيرات هي للمتغيرات التي تختص بعرض كمية التفاعل المكتسب للمشاهدات والتعليقات والاعجاب على حد سواء، يليها كمية التغيير في التفاعل المكتسب لهذه المتغيرات أيضا. نهاية، هذه الرسالة تهدف إلى تحسين فهم تفاعلات موقع YouTube لمقاطع الفيديو المتداولة الشعبية وإظهار امكانية الشبكات العصبية المتكررة GRU في توقع عدد الأيام المتبقية للشعبية.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AML – Advanced Machine Learning

AI – Artificial Intelligence

CML – Classical Machine Learning

CNN – Convolutional Neural Network

GRU – Gated Recurrent Unit

KSC – K- Spectral Clustering

RNN – Recurrent Neural Network

LSTM – Long Short Term Memory

NER – Named Entity Recognition

NLP – Natural Language Processing

NLTK – Natural Language Toolkit

SVM – Support Vector Machine

SVR – Support Vector Regressor

TF-IDF – Term Frequency – Inverse Document Frequency

Chapter One

Introduction

According to Cisco Annual Internet Report, the expected total number of Internet users by 2023 is 5.3 billion users, which equates to two thirds of the earth's current population. (Cisco, 2020). According to the latest update on January 2023, Semrush showed that YouTube holds the second most viewed website with 86.9 billion visits per month, coming shortly after Google. (Semrush, 2023). In addition, according to Hayes in the article "YouTube Stats: Everything You Need to Know In 2023!" the total number of active monthly users on YouTube is 2.2 billion users with 19 minutes of average watch time per visitor per day. As for the uploaded content, the article states that 3.7 million new videos are uploaded daily on YouTube to add up to the existing 800 million videos on the platform (Hayes, 2022).

According to Google support, popular YouTube videos are defined as videos on YouTube that are found to be interesting by a wide range of viewers and thus are transmitted to trending tab. In addition, this list of trending videos is geographically dependent, thus from one area to the other this list may differ (Google Support, n.d.).

As a result, content can be overshadowed by the amount of uploaded videos, which ultimately makes content creators and providers exert massive efforts to harbor interactions to make their videos popular. Thus, in the last decade or so, huge efforts were directed towards finding machine-learning models that can provide insights on popularity for various platforms including YouTube.

In particular, back in 2010, Asur and Huberman published the most influential paper regarding popularity of content posted on Twitter, and was able to produce accurate popularity predictions on real life data several hours in advance. As a result, this paper became the seminal work in the field of social media analytics, and became the groundwork for all subsequent studies with 3309 citation times (Asur and Huberman, 2010).

Since then, published work on popularity predictions has taken two main routes: regression and classification. Regression models focus on delivering continuous values of numeric predictions and popularity quantifications for most targeted attributes such as shares, views and comments. However, the classification models delivers discrete values for the predictions or classification

task. As for classification methods, the presented work focuses mainly on two main binary classes either predicting content as popular or unpopular (De Sa et. al., 2021).

Additionally, popularity predictions use a variety or a combination of classical and advanced machine learning models to predict outcome. For such models, the content type and attributes may differ depending on the scope of the study. Thus, the prediction models can be further divided based on content type such as videos, images and news or a combination of them. It can also be divided based on attributes and used features. For example, features can be of textual, visual nature and metadata or a combination of them (De Sa et. al., 2021).

As a result, the model in hand deals with numeric, categorical, textual and visual data that are preprocessed and feature extracted to prepare a feature set that is used to feed into variant of recurrent neural network named gated recurrent unit to tackle the regression prediction model of acquiring the days in which a video remains viral on YouTube.

1.1 Problem Statement

Since 2010 with the release of the Asur and Huberman paper highlighting the capability to predict viral tweets several hours in advance, many statistical and AI-based models have emerged in the literature to accurately predict the popularity of different published content, yet so far, the state-of-the-art literature lacks the following:

1. Most of the existing studies have used statistical and AI-based models to predict popularity of news headlines, while little efforts were made to predict popularity of other content types such as images or videos (Asur and Huberman, 2010; Bao et. al., 2013; Cai and Zheng, 2022; Ma et. al., 2013; Rathord et. al., 2019; Saeed et. al., 2022; Shang et. al, 2022; Zhao et. al., 2015).
2. The aforementioned state-of-the-art studies predicted popularity of news, images and/or videos using either classical approaches or cascades to interpret the popularity of such content in terms of the expected view count, popularity rate or user engagement (Haimovich et. al, 2022; Nisa et.al, 2021; Massimiliano et.al, 2021; Sibo et. al, 2021 and Tang et.al, 2017).

The prediction of trending videos on social media platforms is an appealing problem to many parties and require attention as it can lead to maximizing profits, minimizing cost, and achieving success (Haimovich et. al, 2022; Nisa et.al, 2021; Massimiliano et.al, 2021; Sibo et. al, 2021 and Tang et.al, 2017).

The exploratory data analysis in this thesis will later demonstrate how 35-40 YouTube videos go viral in the USA per day. Thus, making the chances for these videos to go viral very miniscule with respect to ~3.7 million videos uploaded globally on YouTube per day (Iqbal,2022). However, from same analysis, the total trending period of viral videos range from 1 day to 35 days, with an average of 6 days of trending. These observations and variability within the trending period make ignoring such feature hard and wasteful as it is concerned with videos that are so unique people kept viewing and interacting with.

Thus, trending period reflects exposure and a video trending for prolonged period will effectively impact larger group of people and can create/redirect social trends towards the content displayed in that viral video. Those social trends can have a huge impact on certain businesses, industries, and even influence high stake political elections (e.g., as high as the election to the presidential office of superpower countries).

This thesis exploits the recurrent neural networks (RNN) to build an AI-based model that can predict how many days a viral YouTube video will remain trending given the video data for at least 3 past consecutive days. To our best knowledge, this thesis is the first to interpret the video popularity problem by means of predicting the number of days a viral video stays trending. This prediction is made by training a regression model rather than classification to inference the remaining number of trending days in the viral video lifecycle. For baseline comparison, other classical and enhanced models from the literature will be fitted and their performance compared with the trained RNN model.

1.2 Questions of Study

This study aims to answer the following questions:

1. Is it possible to train a deep-learning model that can produce daily predictions of the remaining number of days an already popular video on YouTube will continue to trend for after three consecutive days of trending?
2. Can the proposed model outperform baseline models?
3. What are the effects of temporal, engagement and textual effects on popularity?

1.3 Study Objectives

The goals of the study are:

1. Design a deep learning model that can perform daily predictions for videos with at least three consecutive trending days to predict the number of days left for these videos to remain trending.
2. Extract new features by handcrafting and NLP extraction methods.
3. Explore effects of temporal, engagement, textual effects on popularity
4. Reduce features based on feature selection and tune model parameters.
5. Evaluate the performance of the model using accuracy measures.
6. Compare evaluation metrics with baseline models.

1.4 Importance of Study and Motivation

According to Business of Apps, YouTube has generated a revenue of $28.8 billion in 2021, which accumulates to a year on year increase equal to 46% (Iqbal, 2022). Such revenue makes YouTube a hot platform for content creators due to the sole desire of a stable income and having broader exposure.

Previous studies have primarily addressed the challenge of predicting content popularity through either classification or by inferencing future engagement metrics. However, most of these approaches often rely on classical methods for video data while reserving deep neural networks for textual data such as tweets. (Haimovich et. al, 2022; Nisa et.al, 2021; Massimiliano et.al, 2021; Sibo et. al, 2021, Trzciński et. al, 2017, Cai and Zheng, 2022 and Tang et.al, 2017).

As a result, this thesis is motivated to fill this literature gap by exploring neural networks, and sequencing for video data. Specifically by predicting time in terms of trending days left for a video to remain viral to help content creators and business owners to swiftly capitalize on trending content by using metadata and textual content of videos to make predictions.

1.5 Scope of Study

The proposed data for this thesis initially uses an open source dataset that is acquired from YouTube API and shows all the popular videos in the United States (USA) (Sharma, 2022).

The start date of this data is 12th of August 2020, while the end date is he 26th of September 2022, with a total of 155k video instances of repeated ids based on the number of days a video was trending. The total number of unique video ids is 28.9 k unique video with various feature types that can be grouped as metadata and engagement data as shown in **Table 1** (Sharma, 2022):

Table 1: Meta Data and Engagement Feature Description.

Feature Name	Feature Type
Video ID	Categorical
Title	Text
Published At	Numeric
Channel ID	Categorical
Category ID	Nominal
Trending Date	Numeric
Tags	Text
View Count	Numeric
Likes	Numeric
Dislikes	Numeric
Comment Count	Numeric
Thumbnail Link	Image
Comments Disabled	Dichotomous
Ratings Disabled	Dichotomous
Description	Text

Chapter Two

Literature Review

Machine learning (ML), is a division of artificial intelligence (AI) that involves the use of algorithms and statistical models to enable computer systems to improve performance on a specific task with experience (Hastie et. al., 2009).

The core benefit of employing ML is its ability to learn patterns and make predictions from data without explicit programming. This ultimately enables ML to have many real-world applications, such as regression, classification, clustering, image and speech recognition, natural language processing and recommendation systems (Hastie et. al., 2009).

2.1 Machine Learning in Perspective to Classical and Advanced Approaches

Machine learning algorithms that depend primarily on statistical methods and linear algebra to analyze and model data are considered as part of classical machine learning (CML). For example, classical algorithms include techniques such as linear regression, logistic regression, decision trees, and support vector machines (SVM), and are typically deployed for supervised and unsupervised learning tasks such as classification, clustering, and regression (Hastie et. al, 2009).

On the other hand, advanced machine learning (AML) is a newer field that focuses on models that are more complex in nature. These models, require larger amounts of data and are suitable for tasks similar to image and speech recognition, natural language processing (NLP), recommendation systems and intelligent decision making for predictive models. Such models include convolutional neural networks (CNNs), recurrent neural networks (RNNs) and long short-term memory (LSTM) variation of the RNN model (Sarker, 2021).

While CML and AML both have their strengths and weaknesses, AML has shown significant promise in recent years, especially in areas where large amounts of data are available. However, AML models can also be computationally expensive and require large amounts of computational resources and specialized hardware in order to acquire the desired accuracy (Sarker, 2021).

2.2 A State-of-the-Art Review of Predictive Features and Diffusions

In terms of ML application for popularity prediction on social media, the proposed methodologies can be clustered into three categories: feature-based, deep learning–based and generative methodologies. For example, feature-based methodologies use predictive features that are handcrafted and identified manually using feature engineering to acquire temporal features (Pinto et. al., 2013; Szabo & Huberman, 2011) , structural features (Bao et. al., 2013; Weng et. al., 2014), and content features (Ma, 2013; Tsur, 2012).

For such methodology, CML models are highly used for predictions. This type of approach generally requires heavy feature engineering and the performance is influenced by the relevance of the extracted features (Ma, 2013; Tsur, 2012).

As stated in the previous section, more efforts are directed into exploiting deep neural networks in ML. Such exploitation is explored by cutting down on the time needed for manual feature engineering as well as increasing the prediction accuracy of the newly proposed models, by imploring techniques acquired from image and natural language processing procedures (Gao et. al., 2021; Gao et. al., 2022; Shang et. al., 2021).

It is worth mentioning that generative methodologies are probability-based and are used in prediction by imploring models such as epidemic and point processes models to provide probabilities of early diffusion of online content, which highly affect the popularity of this content. That being stated, the efficiency of such models depend on the assumptions made for such diffusions and may be considered a huge setback of such predictive approaches if the initial assumptions do not represent the actual spreading pattern of the content (Lin et. al., 2013; Shen et. al., 2014; Zhao et. al., 2015).

Based on a survey that was published in 2021, De Sa et. al. discuss the most frequently observed features in literature review for popularity prediction models for classification and regression problems across various content type and found the following features to be of most importance as shown in **Table 2** (De Sa et. al., 2021):

Table 2: Features Observed in Literature (De Sa et. al., 2021)

Feature	Feature	Feature	Feature
Category	Number of Keywords	GIST	Thumbnail Contrast
Author or Source	Frequency of Positive Words	Output of CaffeNet	Number of Tweets/ Retweets
Title Subjectivity	Frequency of Negative Words	Output of ResNet	Number of Shares
Content Subjectivity Score	Number of Words in Title	Video's Length	Number of Views in the first day
Number of Friends /Followers of Author	Number of words in Content	Video's Resolution	Number of Views
Number of Named Entities	HOG	HUE	

2.3 A State of the Art Review of Predictive Popularity Models

With the rise of social media outlets, almost all content creators and businesses rely on their accounts to post videos, photos and statuses for the world to see. These posts eventually can become a solid source of income in many platforms. To Viola Massimiliano and coworkers, posting the precise viral content consistently can lead to a noteworthy increase in interactions, follower number and ultimately sales and income for these businesses and content creators. As a result, the state of the art techniques for predicting popularity in the literature vary from feature based models, generative models to deep learning models (Massimiliano et.al., 2021).

For example, with the high regard for videos on YouTube, Facebook and Instagram more researchers focus their work on predicting this particular data, as it contains valuable information alongside numeric features. For example, Nisa et. al., directed their research efforts towards creating a model that can effectively predict the popularity of YouTube videos by predicting a popularity score for a specific video using XGBoost technique and using a combination of static

and temporal features extracted from the data itself to produce higher prediction accuracy (Nisa et. al., 2021).

On the other hand, Haimovich et. al., considered accessing the popularity of Facebook videos using a generative model named as the Excited Hawken's Point Process model. Much like Nisa et. al., Haimovich et. al., use a model that depends on Gradient Boosted Decision trees to predict the view count of a video for a predefined time periods, as well as provide a growth rate for the views. (Haimovich et. al., 2022; Nisa et. al., 2021). It is worth mentioning, that unlike Nisa et. al., Haimovich et. al., also focused on computational time as part of the research goal to reach the desired predictions (Haimovich et. al., 2022).

As for the use of neural networks in popularity predictions, both Massimiliano et. al., and Tang et. al., used deep learning neural networks models to inference popularity of videos. However, in Massimiliano's approach, the study focused on classifying videos into high popularity and low popularity using temporal features extracted by pre-trained CNN and CNN-RNN models for images and videos acquired from Instagram platform (Massimiliano et. al., 2021; Tang et. al., 2017).

In addition, Massimiliano's approach focused on dealing with individual creators as research units to find the popularity and relative popularity from one post to the other for that unit, while Tang's et. al., goal was to produce real time prediction of viral videos on the Facebook platform as a whole. Thus, the network requires a continuously updated dataset, and large GPU powers as well as work force to produce lists with popular videos that are then reduced and ranked into a final list with 1 million entries for most popular videos across the Facebook platform in an attempt to reduce the high quality streaming costs (Massimiliano et. al., 2021; Tang et. al., 2017).

Unlike Massimiliano temporal features; Tang et. al., models depended on metadata features and the interaction on the videos to feed the CHESS algorithm, which is one application of neural networks, to derive the popularity lists (Tang et. al., 2017).

As for feature based models, Sibo et. al., worked on producing models to predict number of views for YouTube using multiple linear regression model. While, Figueiredo worked on clustering time series data for YouTube videos using K-Spectral Clustering (KSC) and randomized ensemble trees

to inference the view count and popularity trends respectively. Both of these approaches require static features and high computational time. (Sibo et. al., 2021; Figueiredo, 2013)

Similarly, Rathord et. al., compared the performance of classical ML models for news popularity prediction and proposed methodologies to improve the performance of random forest, SVM, AdaBoost, KNN, linear regression, logistic regression, Naïve Bayes and genetic algorithm. His findings show that random forest yields the highest accuracy among all tested models (Rathord et. al., 2019).

In 2021 De Sa et. al., published a paper concerning the state of the art of popularity prediction models for news and videos. In it, a documentation of the state of the art features is presented based on the most frequented features used in model predictions. Moreover, the paper showcased an implementation of popularity classification on a streaming service video dataset using a hybrid approach of NLP feature extraction and classical random forest model (De Sa et. al., 2021).

Despite the growing popularity of YouTube, however, a large portion of the recently published papers seem to concentrate on news predictions. For example, Saeed et. al., Cai and Zheng both exploited deep neural networks in predicting news popularity using different variations of neural networks on extracted textual features, metadata and engagements for either regression or classification purposes. For example, Cai and Zheng explored a GRU neural network regression prediction model to predict news popularity while Saeed et. al., used deep neural networks in the form of temporal propagation patterns to classify news popularity (Cai and Zheng, 2022; Saeed et. al., 2022).

In their papers, Cai and Zheng's model performance have a RMSE value of 0.109 and goodness of fit equal to 0.6 while Saeed's et. al., model has an F-score equal to 92% (Cai and Zheng, 2022; Saeed et. al., 2022).

It is worth mentioning that the GRU neural network is a modification of LSTM neural network, that outperforms the later for smaller datasets as it requires less training time with considerably similar results to LSTM (Cai and Zheng, 2022).

In a total different approach, Shang et. al., posted in 2022, a paper that discuses popularity in terms of social influence and homophily features. The findings of the study shows the importance of the effects of social group representation on early users in popularity prediction accuracy, and

concluded that it should be added as a feature in the prediction of early popularity (Shang et. al., 2022).

That being stated, the question of popularity is still a very important question and more work is still being introduce to exploit the full potential of advanced models and trying hybrid techniques between classical and advanced in order to acquire better performance for predictions both before and after content publishing.

Finally, this thesis aims to explore the regression aspect of the prediction problem of video popularity duration by employing a multi- feature data centric approach on a model named GRU that is a variant of the LSTM recurrent neural network. This model in particular seems to provide high accuracy values for prediction models that are concerned with classification of news. Thus, this thesis aims at exploring the power of a GRU model in predicting popularity on different content, i.e., videos on a different content outlet, i.e., YouTube.

It is important to note that Appendix shows a Table comparison between all the discussed papers in this section.

Chapter Three

Methodology

In the last few years, the presented approaches towards tackling prediction problems with large datasets, in general, were directed towards deep neural networks. These networks provide unique framework architecture by merging the various processing layers such as input, hidden and output layers in order to learn from data and make predictions accordingly (Xin et al., 2018; Han, 2015).

3.1Recurrent Neural Networks (RNN)

RNN is one type of feedforward neural networks that gained popularity due to having a recurrent hidden state (h_t) that its' activation at time *t* is dependent on the previous time instance *t-1* (Graves et. al., 2013).

In specific, generative recurrent neural networks produce probability distributions as output for a current state $h_{t,}$ over the upcoming element of the sequence. However, such generative models are found to be hard to train due to long-term dependency that either vanishes or bursts (Bengio et. al., 1994).

This ultimately causes the gradient descent optimization based approaches to fail due to effects of short and long-term dependencies. As a result, some of the significant efforts that worked towards enhancing the activation function into more intricate form using gating units are LSTM and GRU models, which are considered as gated RNNs (Hochreiter and Schmidhuber, 1997; Cho et. al., 2014).

3.2 LSTM Neural Network

LSTM, also known as long short- term memory model, was introduced in 1997 by Hochreiter and Shcmidhuber, and proposed a modification to original RNN by modifying the nonlinear activation function to include a memory *c* at time *t* (Hochreiter and Shcmidhuber, 1997).

This model, contains multiple gates which are input, forget and output gates denoted as f_t , i_t and O_t respectively, as shown in **Figure 1**, however the most important are the forget gate (f_t) and input gate (i_t). These gates are used to modulate the amount of memory for content exposure that goes into the input gate as well as modulate the extent to which the existing memory can forget in the forget gate (Hochreiter and Shcmidhuber, 1997; Graves, 2013)

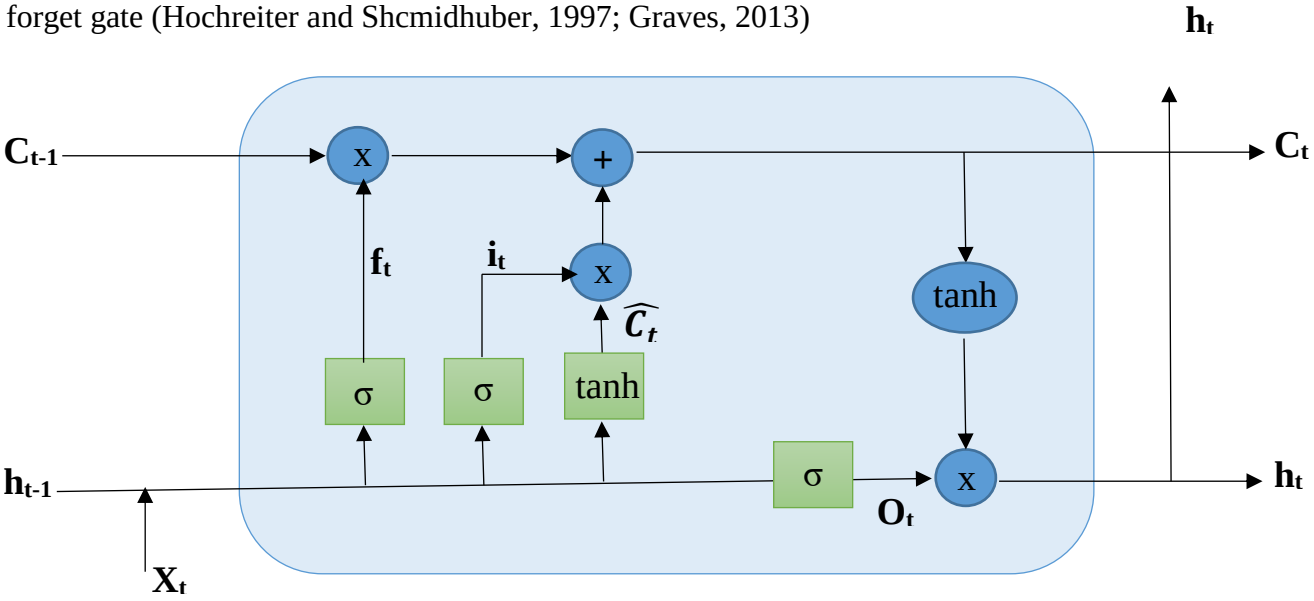

Figure 1: LSTM Architecture Diagram (Hochreiter and Shcmidhuber, 1997)

By introducing the discussed architecture, the LSTM overcomes the shortcoming of traditional RNN models that overwrite content at each time step t_i. In addition, early in the training whenever the LSTM model detects an important feature from the input sequence, it can intuitively carry the feature information, and thus capture the data's long-term dependencies (Gaves, 2013).

3.3 GRU Neural Network

In 2014, Cho et. al., proposed a novel approach, which was denoted as gated recurrent unit (GRU). The paper discusses that although LSTM solves the long- dependencies of the traditional RNN model. It is still internally a complex structure, which results in a larger number of parameters to attend to and longer training time (Cho et. al., 2014).

GRU is similar to LSTM in its' gating mechanism to solve long- dependencies, however the GRU uses two gates only which are the reset gate and the update gate. It is worth mentioning that this network abandons the output gate, combines the cell state with the hidden state, and modulates the

information flow inside a unit without having a separate memory cell. (Cho et. al., 2014; Cai and Zheng, 2022).

More importantly in **Eq (1)**, the activation function (h_t) in a GRU model is of linear interpolator nature between previous activation h_{t-1} and h_t (Chung et. al., 2014):

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t \qquad \textbf{Eq (1)}$$

Where, $\odot$ represents the operational rule of element wise multiplication and the update gate z_t determines how much the unit updates its activation and content using **Eq (2)**, (Chung et. al., 2014):

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \qquad \textbf{Eq (2)}$$

Where, σ represents the logistic sigmoid functions, x_t is the input while W_z and U_z represent the weights that are learnt (Cho et. al., 2014).

Additionally in **Eq (3)**, Cho 2014 showed that the estimated activation $\hat{h}_t$ is similarly computed to that in the recurrent network (Cho et. al., 2014):

$$\widehat{h_t} = \tanh(\mathrm{W}x_t + \mathrm{U}(r_t \odot h_{t-1})) \qquad \textbf{Eq (3)}$$

Where, W and U represent weights as discussed in **Eq (2)**, r_t is a set of reset gates and $\odot$ represents the operational rule of element wise multiplication, thus, when r_t is relatively small and close to zero, the reset gates allow the unit to forget the previously computed state. It is worth mentioning that the reset gate can also be computed according to the equation of the update gate i.e. **Eq (2)** (Chung et. al., 2014).

As a result, the mechanism of a GRU network, is shown in **Figure 2**, and its' simpler structure allows for fewer parameters as discussed previously which ultimately makes this network more favorable over traditional structures of RNN (Cai and Zheng, 2022).

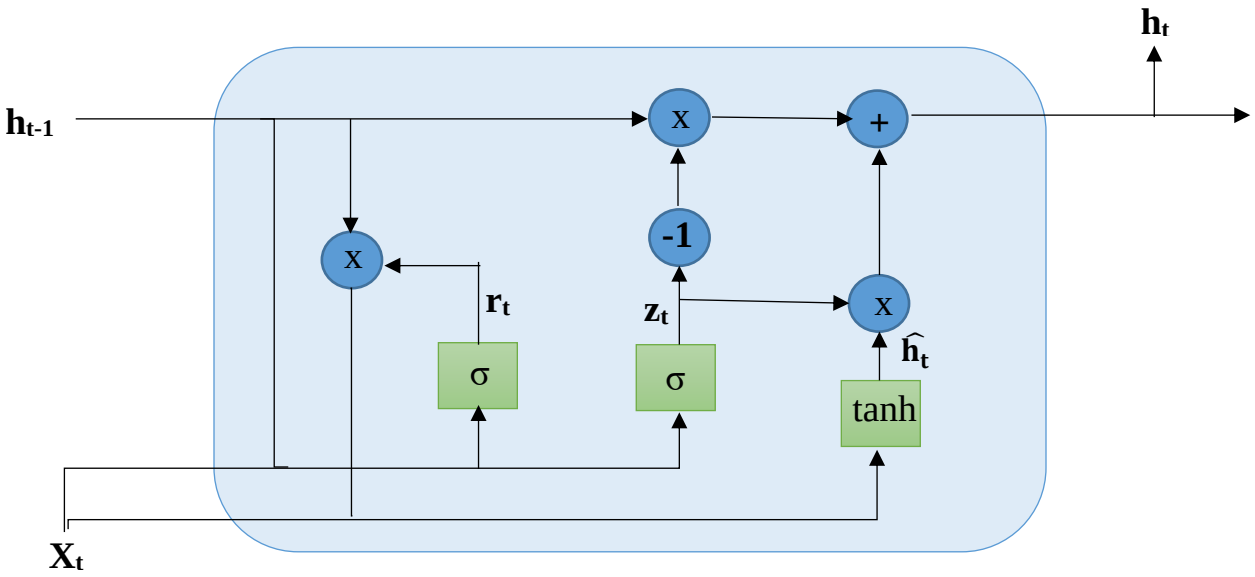

Figure 2: GRU Architecture Diagram (Cho et. al., 2014)

3.4 Video Popularity Duration Model based on GRU Neural Network

The main framework process to achieve video popularity duration prediction is in **Figure 3** as the following:

1. Exploratory Data Analysis and Visualization: to understand data and to have general overview of key variables with basic feature engineering.
2. Data Preprocessing and Feature Engineering: the dataset will be segmented based on its' type numeric, categorical and textual data. Each type will be preprocessed based on the feature extraction needs as the following:
    a. Check data soundness and remove noisy data from rows by inspecting for Nan values, removing data with incomplete trending days and trending gaps.
    b. Perform feature engineering on numeric data such as assigning video id as key column, converting dates into date time series to calculate target variable which is trending days, and calculating other independent variables such as trending lag, engagement gains, engagement gains rate and engagement ratios. In addition to removing any non- interpretable values such as negative and positive infinities and none values from feature-engineered data.

c. Perform text data preprocessing and sentiment analysis by employing natural language processing algorithms. Preprocessing will include text preparation such as lower casing, punctuation removal, stop words removal, lemmatization, non-English words removal and repeated letters removal to perform sentiment analysis using Vader lexicon and NLTK library to create new sentiment features for text data, as well as finding text lengths and keywords to length text ratios.
d. Perform categorical variable mapping on originally categorical data, and feature engineered data such as sentiment data, and date time data so to be able to encode it and understand it mutually. Encoding will be done by applying get dummies function on these data

3. Feature set fusion: the acquired features from step 3 will construct the final feature set after conducting feature selection and then splitting the data into train, test and validation sets using 60:20:20 ratio from best practice standard (Baheti, 2023).
4. Feature Extraction: the aim of this step is to acquire the most relevant variables to reduce computational power and not over fit the resulting model by using random forest feature selection from scikit-learn library and median threshold to select the most important features from the feature-engineered training dataset to prevent data leakage.
5. Normalization of the feature selected dataset based on the training set will be done using min-max scaler from scikit-learn library to maintain the underlying distributions and due to the importance of the zero value in the target variable.
6. Model training: the model will use the simplified GRU network that is derived from LSTM, and will take mean squared error (MSE) as loss, Adam optimizer, and softplus activation function.
7. Final Model: based on this framework, the multi-feature extraction and fusion of videos will be realized, and the GRU structure will be used to train a regression prediction model to predict YouTube video popularity duration in days and tuned using the validation set.
8. Model Evaluation: the model will be evaluated based on evaluation metrics shown in section **3.6** and will be compared with baselines from the state of the art models as shown in section **3.7** to provide comparison about performance and accuracy.

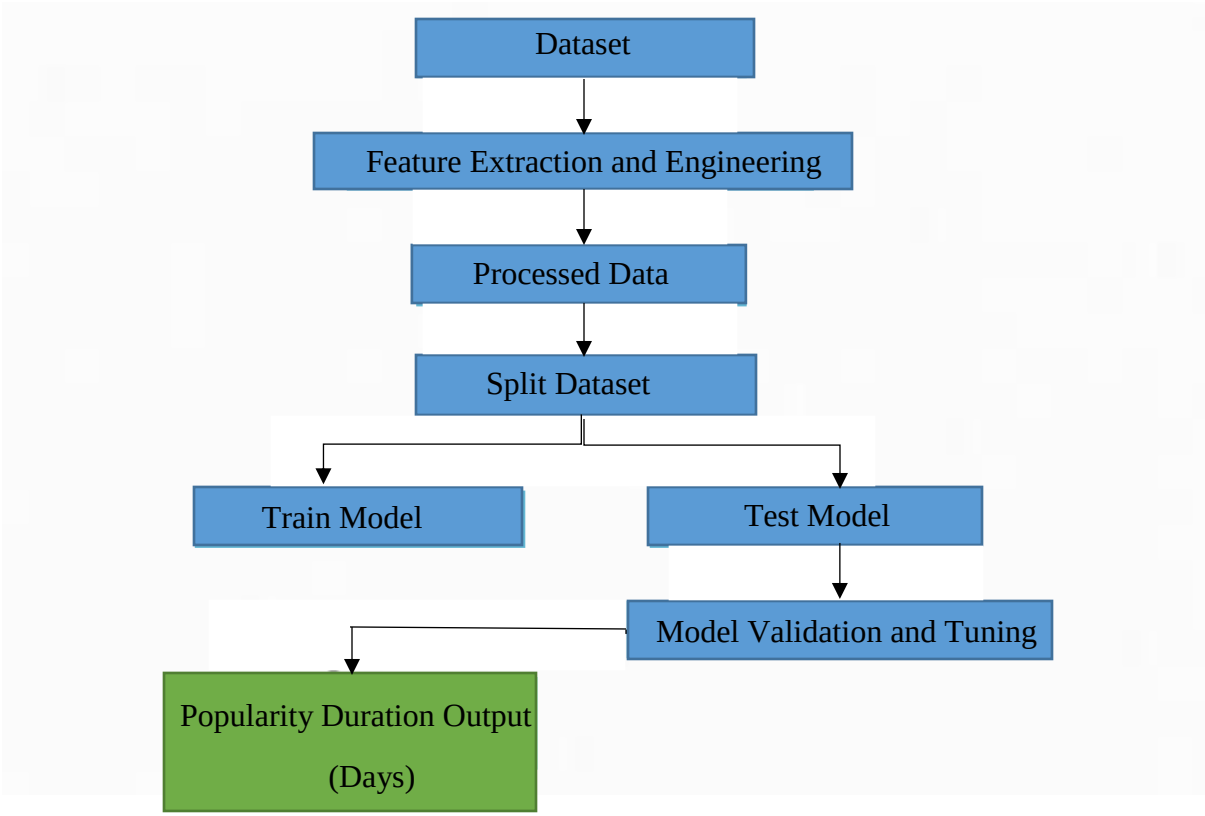

Figure 3: Proposed Model Workflow.

3.5 Dataset

As shown in section **3.4**, this thesis aims to describe a popularity duration model that is regression based to predict the number of days a video stays popular. For such a use case, the data was acquired from YouTube API and shows all the popular videos in The United States starting from 12th of August 2020 to the 26th of September 2022 as described in **Table 1** from **section 1.5**.

3.5.1 Raw Data:

The total number of raw instances is 155k instances from which approximately 28.9k are of unique video ids as shown in **Table 3** (Sharma, 2022)

Table 3: Original Dataset from YouTube API Description.

Total Video Count	Total Unique Video Count	Number of Features	Trending Area
155k	28.9k	15	USA

The video instances in the dataset can be mapped into the video categories based on the supporting document that was downloaded with the dataset as shown in **Table 4**.

Table 4: Video Category Labels.

Category Number	Category Label
1	Film & Animation
2	Autos & Vehicles
10	Music
15	Pets & Animals
17	Sports
18	Short Movies
19	Travel & Events
20	Gaming
22	People & Blogs

Category Number	Category Label
23	Comedy
24	Entertainment
25	News & Politics
26	How to & Style
27	Education
28	Science & Technology
29	Nonprofits & Activism
30	Movies

For better visualization of the numeric features – views, likes, dislikes and comments-, the raw data was grouped based on the unique video ids to acquire the variation in the values within the trending period by examining the first and last days of trending counts. Additionally, since the data has high ranges, as shown in the data description in **Table 5**, the visualization of the boxplots was plotted as log of values in order to make it easier to interpret and identify outliers, as shown in **Figure 4**.

Table 5: Numeric Features Description.

	1st Day Views	Last Day Views	1st Day Likes	Last Day Likes	1st Day Dislikes	Last Day Dislikes	1st Day Comment	Last Day Comment
Mean	$1.257802e^{6}$	$2.718954e^{6}$	$8.601305e^{4}$	$1.345023e^{5}$	919.000	1946.00	$7.387750e^{3}$	$1.059994e^{4}$
Std	$2.644334e^{6}$	$6.824920e^{6}$	$2.301464e^{5}$	$3.921189e^{5}$	4480.00	10589.0	$4.523660e^{4}$	$8.359994e^{4}$
Min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
25%	$3.191470e^{5}$	$5.604718e^{5}$	$1.428600e^{4}$	$1.997400e^{4}$	0.000000	0.00000	$1.028250e^{3}$	$1.376000e^{3}$
50%	$5.990550e^{5}$	$1.140095e^{6}$	$3.169300e^{4}$	$4.745700e^{4}$	184.000	296.000	$2.297000e^{3}$	$3.07600e^{3}$
75%	$1.269420e^{6}$	$2.509931e^{6}$	$7.560675e^{4}$	$1.194028e^{5}$	681.000	1213.00	$5.346000e^{3}$	$7.136500e^{3}$
max	$8.589037e^{7}$	$2.777917e^{8}$	$7.110071e^{6}$	$1.602153e^{7}$	405329	879354	$3.400291e^{6}$	$6.738537e^{6}$

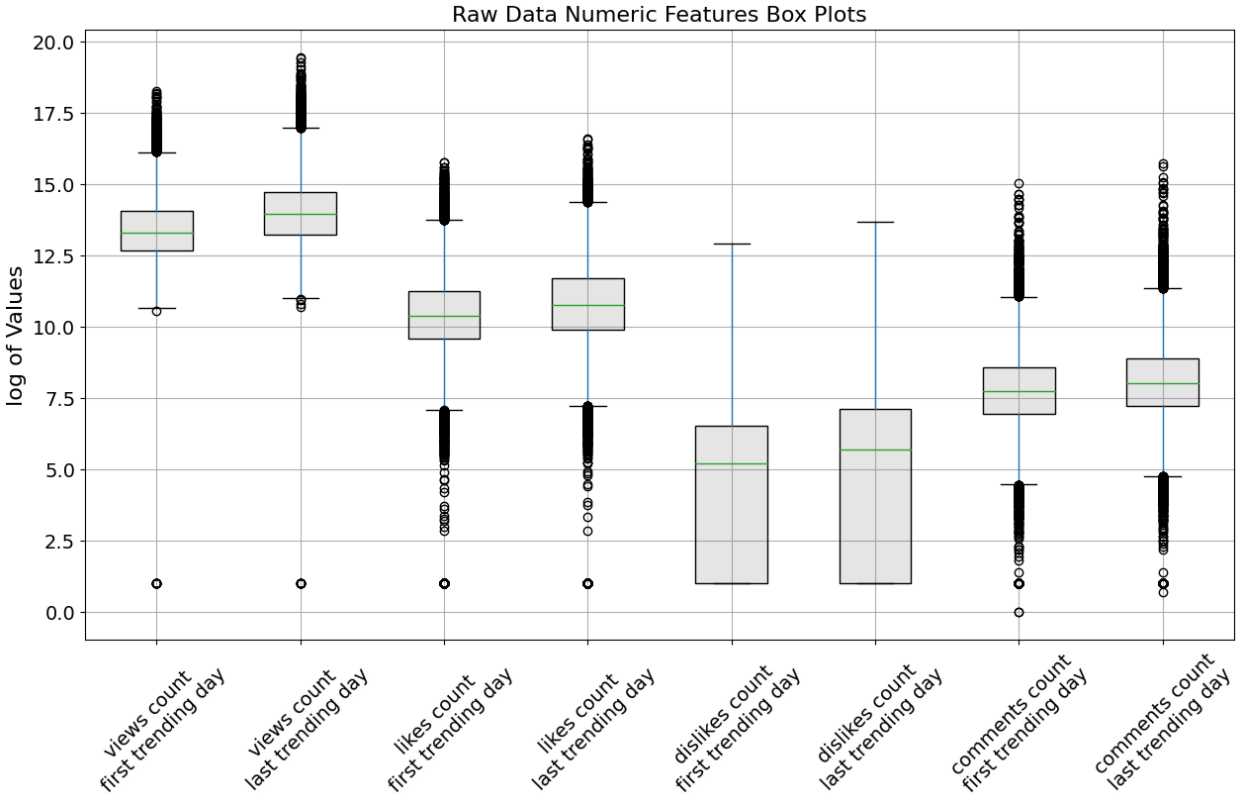

Figure 4: Numeric Features Box Plots from Raw Dataset.

The box plots in **Figure 4** show excessive outlier data points inside the dataset, these outliers are of significant importance as they belong to videos that have exceeding popularity than other posted

videos which made them trend for a specific period on the platform. Thus, keeping these points in analysis is essential for the popularity prediction problem.

3.5.2 Initial Exploratory Data Analysis:

In order to transform the raw data into a preprocessed and feature engineered dataset, initial exploratory data analysis was done by first grouping the data based on video id and doing basic feature engineering to better understand the underlying trends and viewer interactions on YouTube platform.

To understand the distribution of the total trending periods (in days) in the dataset, feature engineering was applied to calculate the new initial feature in **Eq (4)** " total trending days" was acquired by subtracting the trending start from the trending end.

$$Total\ Trending\ Days = Trending\ End - Trending\ Start \qquad \textbf{Eq (4)}$$

The plotting of the distribution of "total trending days" feature shows that it is heavily concentrated between 0 and 15 days with highest frequency related to 6 days of total trending time and slight positive skewness towards the rarely occurring values such as 36 days as shown in **Figure 5**. By conducting the Shapiro test to determine normality using scipy library, the resulting p-value is equal to zero which is less than 0.05. Since p-value ≤ 0.05, then the null hypothesis is rejected and the feature "total trending days" significantly deviates from the normal distribution as shown in **Figure 5** (Shapiro and Wilk, 1965).

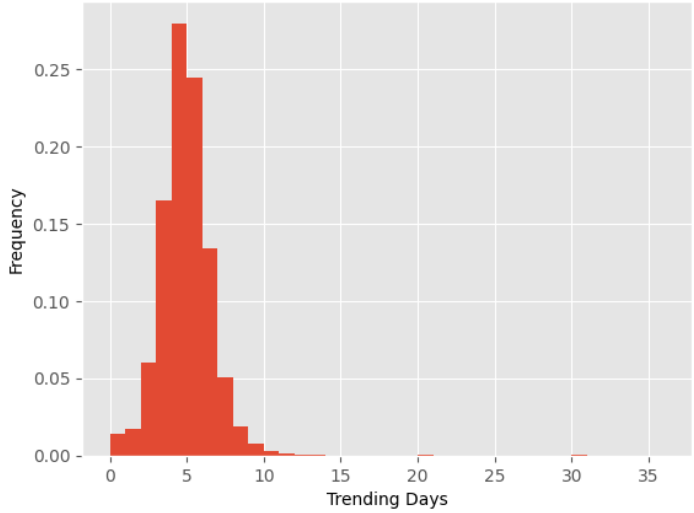

Figure 5: Total Trending Days Feature Distribution.

It is worth mentioning that for the actual model building, the target feature is a modification of the total number of trending days by acquiring the remaining number of trending days

Furthermore, the mean of the total trending days was plotted against various time intervals to understand the variation of the feature during the day, months and years as shown in **Figure 6**, **7** and **8**, respectively. It is worth mentioning that plotting the week-day against the mean of total trending days showed that the variation in the mean is very minimal with slightly highest means on Fridays and Saturdays.

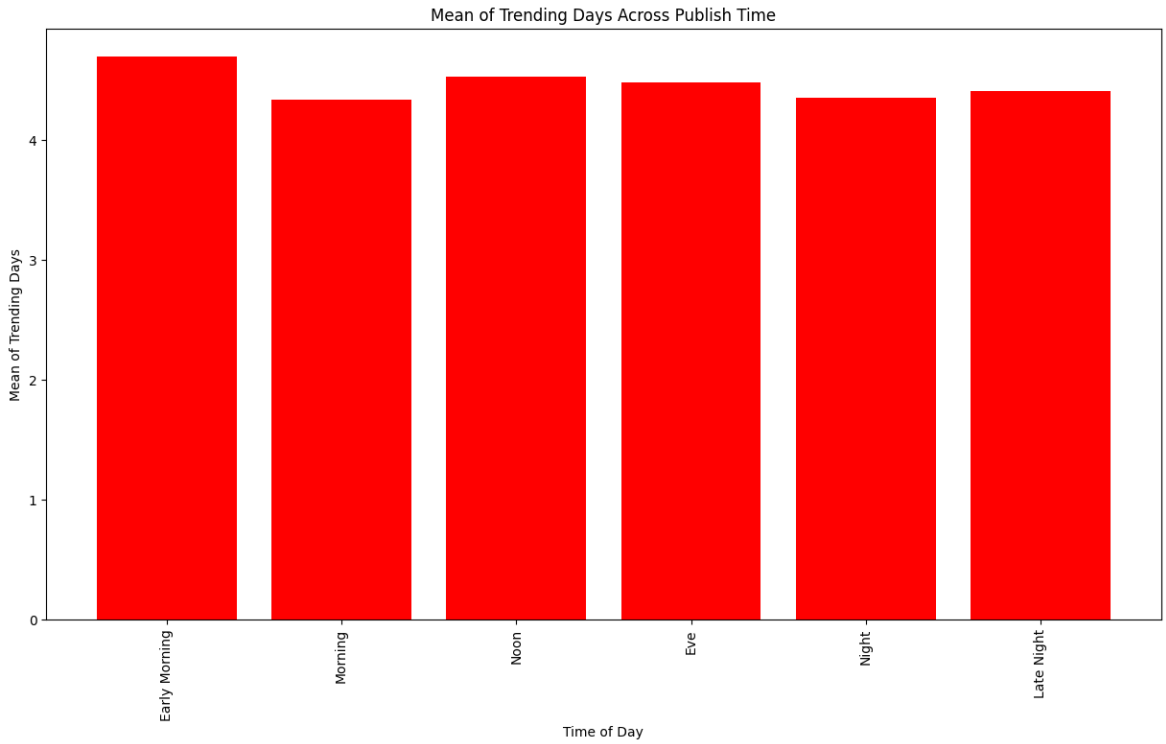

Figure 6: Mean of Total Trending Days across Publish Time during The Day.

From **Figure 6**, it can be seen that the effect of publish time on the mean value of trending days is highest for videos published early in the morning followed by noon. However, all hours seem to be associated with trending days higher than 4 days.

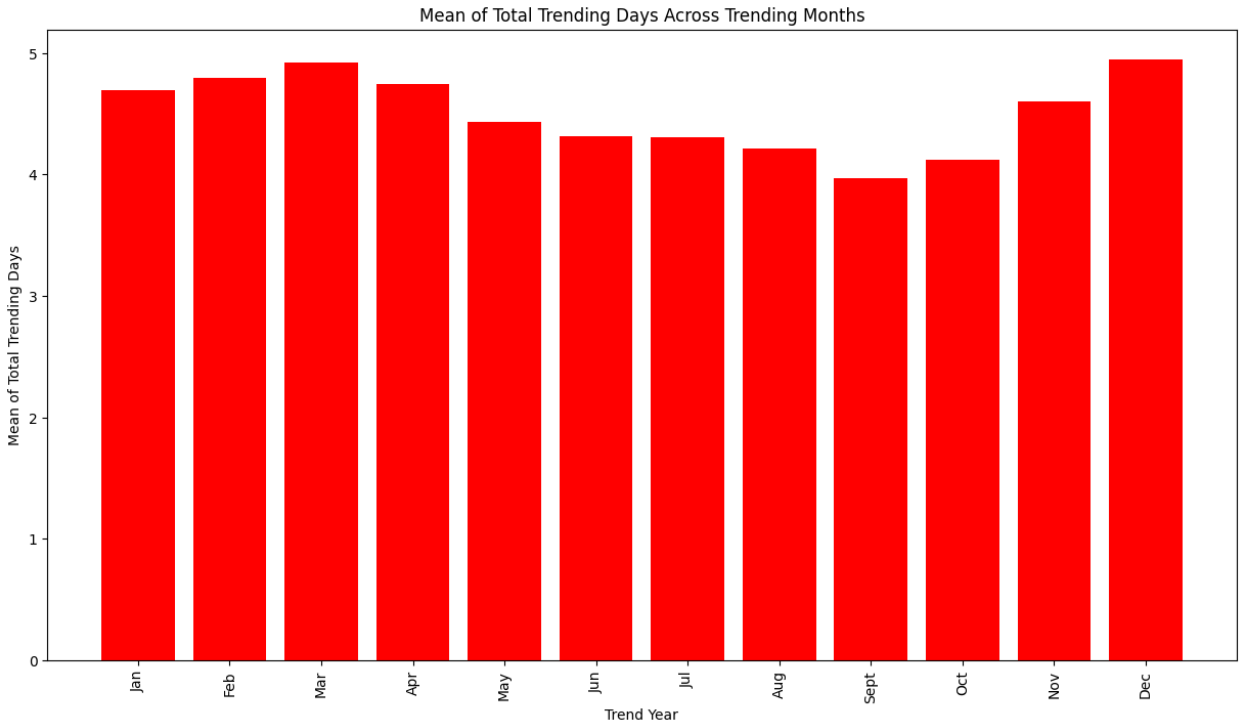

Figure 7: Mean of Total Trending Days across Trending Months from all Years.

From **Figure 7**, December seem to gather the highest mean value for total trending days, closely followed by March – approximately 5 days-, while September is associated with the least mean value of total trending days- approximately 4 days-. That being stated, the variation between months seem to be more evident than the variation of publish hours in **Figure 6**.

As for **Figure 8**, the years 2020 and 2021 seem to have almost similar mean values of trending days; however, 2022 seem to be lower -with approximately 4.1 days-. This decrease in the mean value of total trending days for 2022 can be the direct result of the dataset duration. Meaning, from **Figure 7** the highest mean value was accounted for December while the lowest for September and this dataset ends on the 26th of September, which is the month of lowest trending mean. Adding that December is not included in the 2022 year, the decrease in the mean value of total trending days become reasonable.

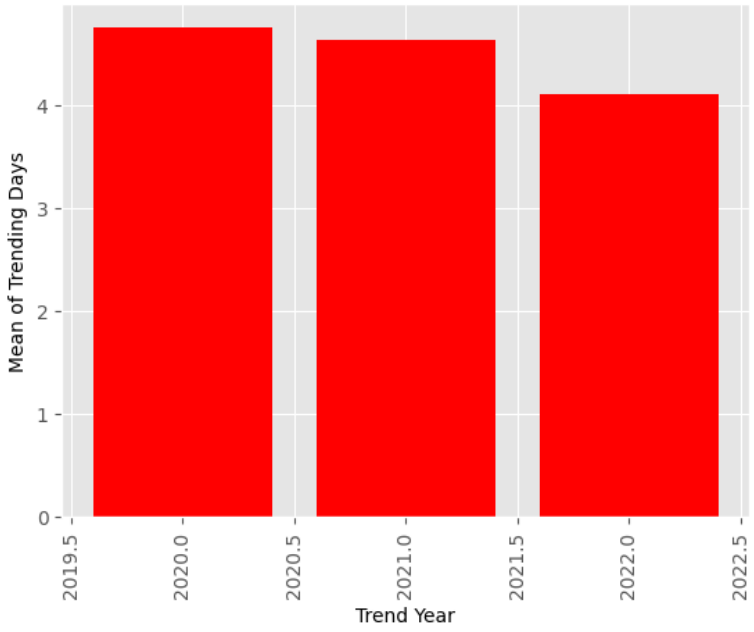

Figure 8: Mean of Trending Days across Years.

In an attempt to understand YouTube dynamics between publication day and first trending day, a new feature in **Eq (5)** was formulated as trend lag by subtracting the publish date from the trending start in days as the following:

$$Trend\ lag\ = Trending\ Start - Publish\ date \qquad \textbf{Eq (5)}$$

The plotting of histogram of the new feature "trend lag" shows that the feature values range from -1 to 5 days. Where a -1 day lag means that the video became trending in less than 1 day. In addition, the zero day lag is associated with the highest frequency indicating that most of the trending videos become popular after 1 day of posting, and in some rare cases the trending starts after 5 days of publishing as shown in **Figure 9.**

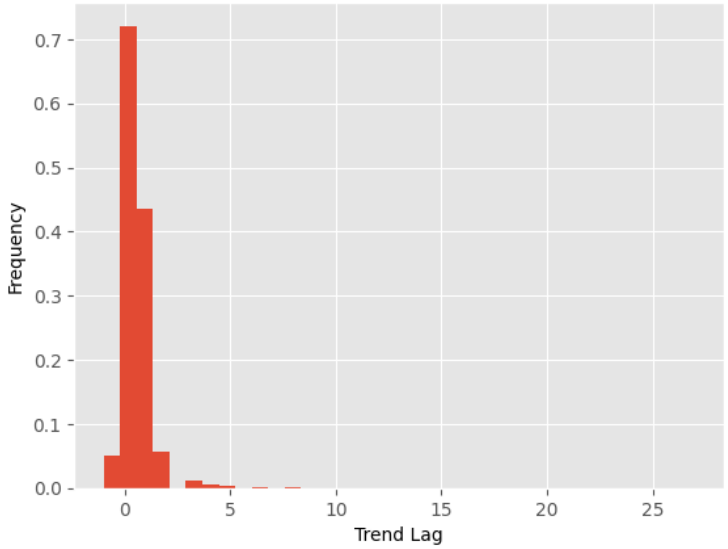

Figure 9: Trend Lag Feature Distribution.

Moreover, in order to understand the volume of published and trending videos per day on the YouTube platform, a counter for published videos per day as well as counter for trending videos per day were acquired and plotted to better understand the transition between published and trending data.

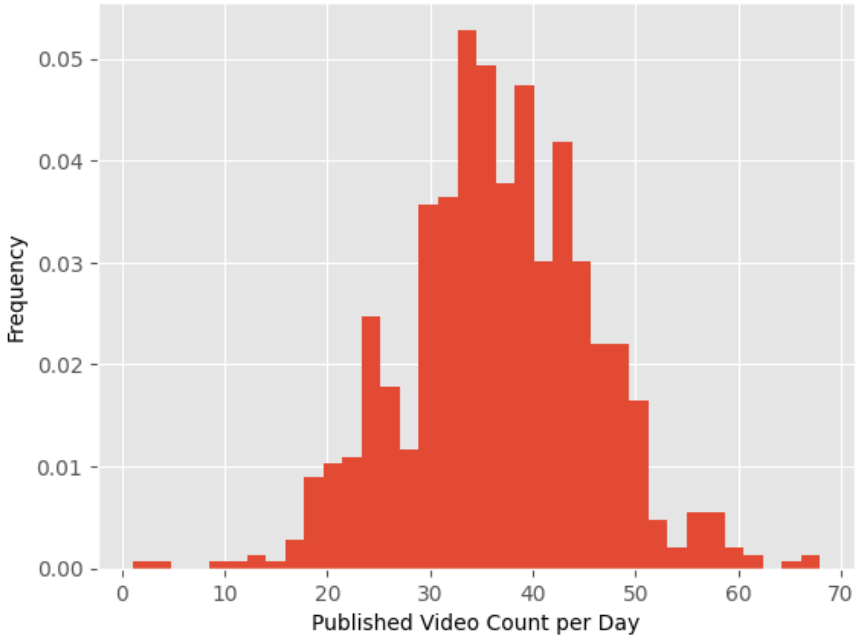

Figure 10: Published Video Count per Day Distribution.

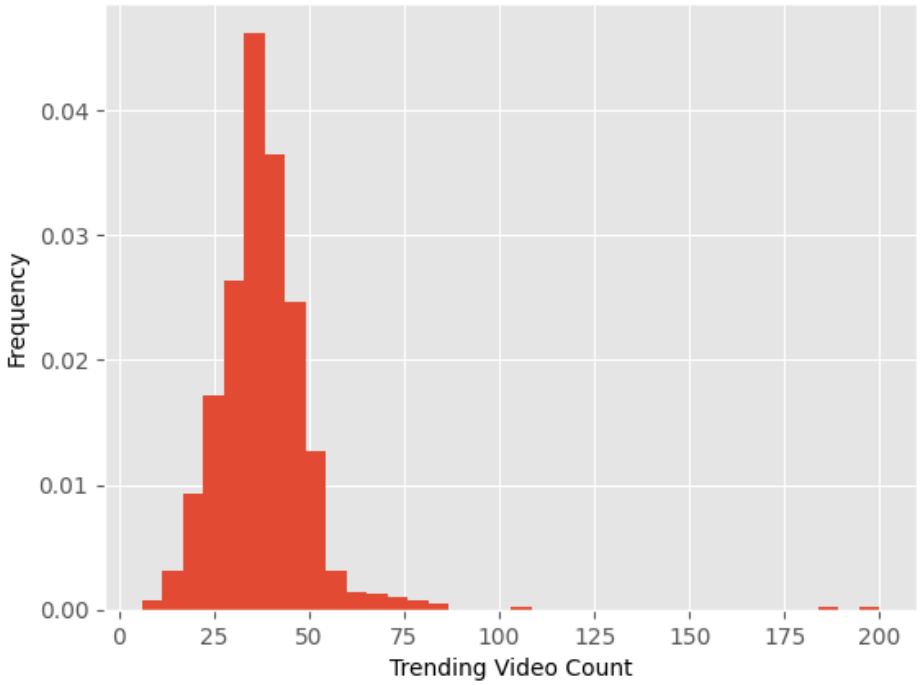

Figure 11: Trending Video Count per Day Distribution.

From **Figure 10** and **11**, it is visible that the two distributions are not normally distributed, and by conducting the Shapiro test of normality to both variables, the resulting p-values from both tests were below 0.05 significance with values equal to 0.0365 and $1.192e^{-31}$, respectively. Thus, the Shapiro test further indicates that the two features do not follow the normal distribution.

In addition, trending video count have highest frequency around 35 and 40 trending videos per day in USA, In comparison to the total uploaded videos, these values seem minimal. Even in the maximum number of trending videos, which is 200 videos per day, it is greatly lower than the total published videos in the USA.

By visualizing this issue, the importance of why video popularity and content popularity in general remain a very hot topic to explore can be understood and derived.

Additionally, by plotting the counts of published and trending videos against time as a series as shown in **Figure 12** and **13**, respectively, the resulting plot in **Figure 12** shows a somehow variable plot for the published trending videos, as you cannot control user behavior for upload. However, **Figure 13** displays a more constant (stochastic) trend against the 35 video line, this variation between these two figures can be attributed to YouTube's internal policies regarding trending data

by directing the trending list towards such constant ranges using its' recommendation system and any other constraints on the number of viral content per day.

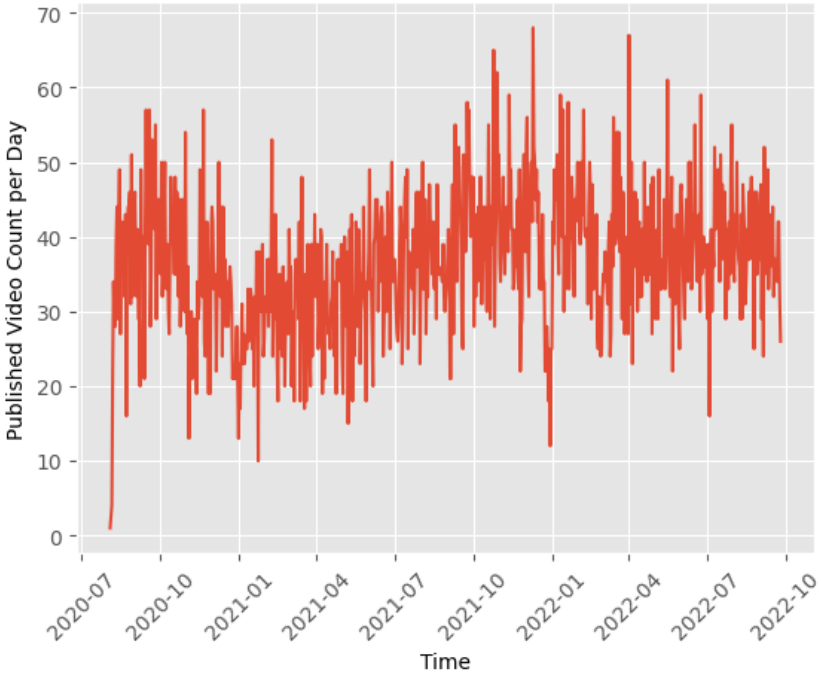

Figure 12: Published Video Count per Day Series.

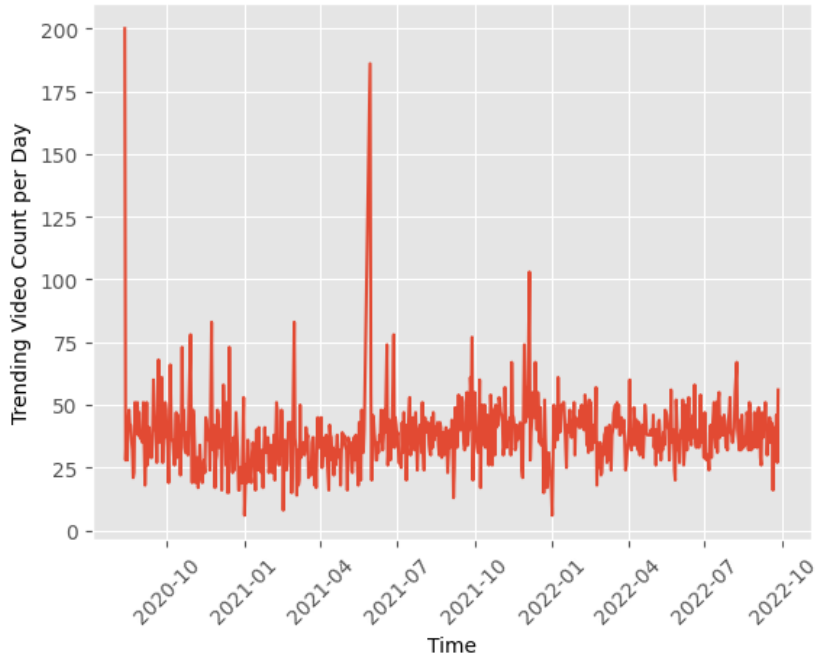

Figure 13: Trending Video Count per Day Series.

Moreover, the trending video count across hours, months and years are shown in **Figure 14**, **15** and **16**, respectively. While the mean of trend lag across hours, months are shown in Figure **17**, **18** and **19**, respectively.

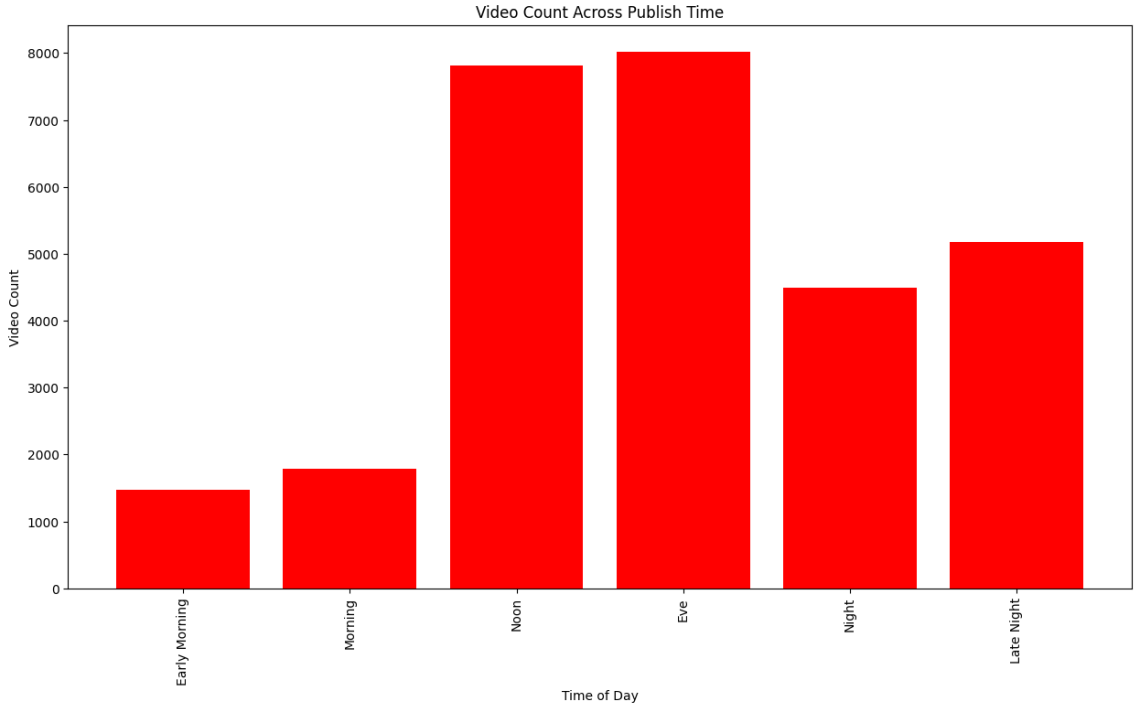

Figure 14: Trending Video Count across Publish Time during the Day.

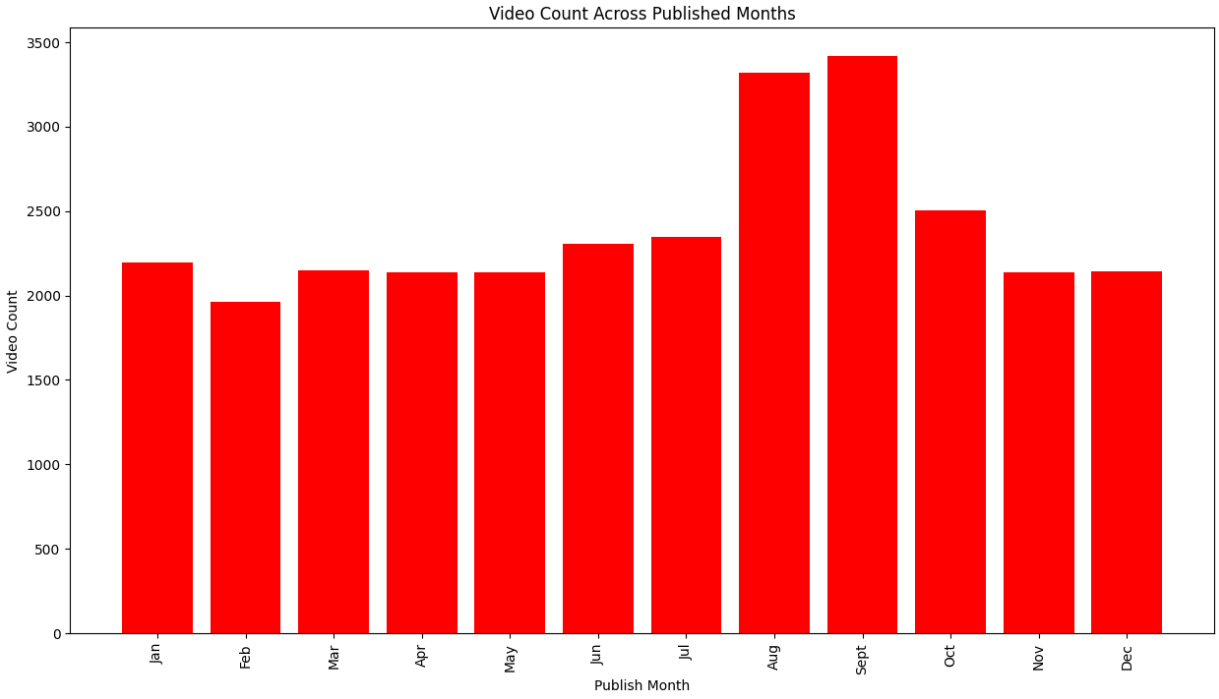

Figure 15: Trending Video Count across Months from all Years.

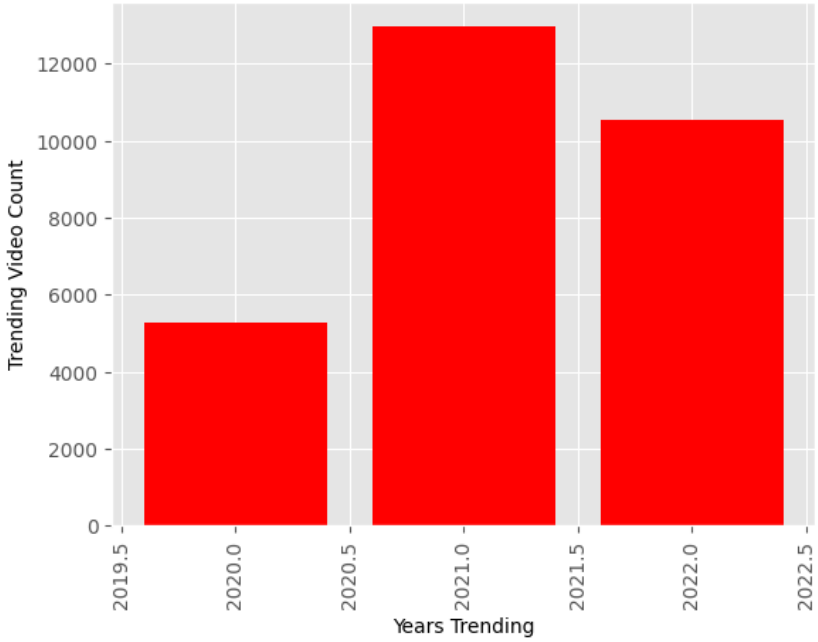

Figure 16: Trending Video Count across Years.

From **Figure 14**, it can be seen that the highest video counts are associated with noon and eve, and unlike the mean total trending days, shown in **Figure 6**, the highest was associated with early morning posting. As for the months in **Figure 15**, August and September seem to have the highest video counts; this can be explained because the start and end are at August and September, respectively. As for the years shown in **Figure 16**, the high variation in the total number of trending is consistent with the collected data duration in each year. Meaning, 2020 year data relates to approximately 4 and a half months of data, while 2021 relates to a full year data and 2022 relates to approximately 9 months of data.

As for **Figure 17**, the mean of trend lag shows lowest value for videos posted late night. In addition, **Figure 18** shows that the highest mean of trend lags occur in May followed by August while January account for the lowest mean of trend lag.

As for the mean of trend lag across years that is shown in **Figure 19**, the mean of the trend lag seems constant for years 2021 and 2022. However, it shows lower mean for data posted in 2020, this lower value can be due to 2020 containing data from August to December only, which is significantly less than the other two years in comparison. Thus, strong verdicts about what caused this decrease in the mean may be misleading.

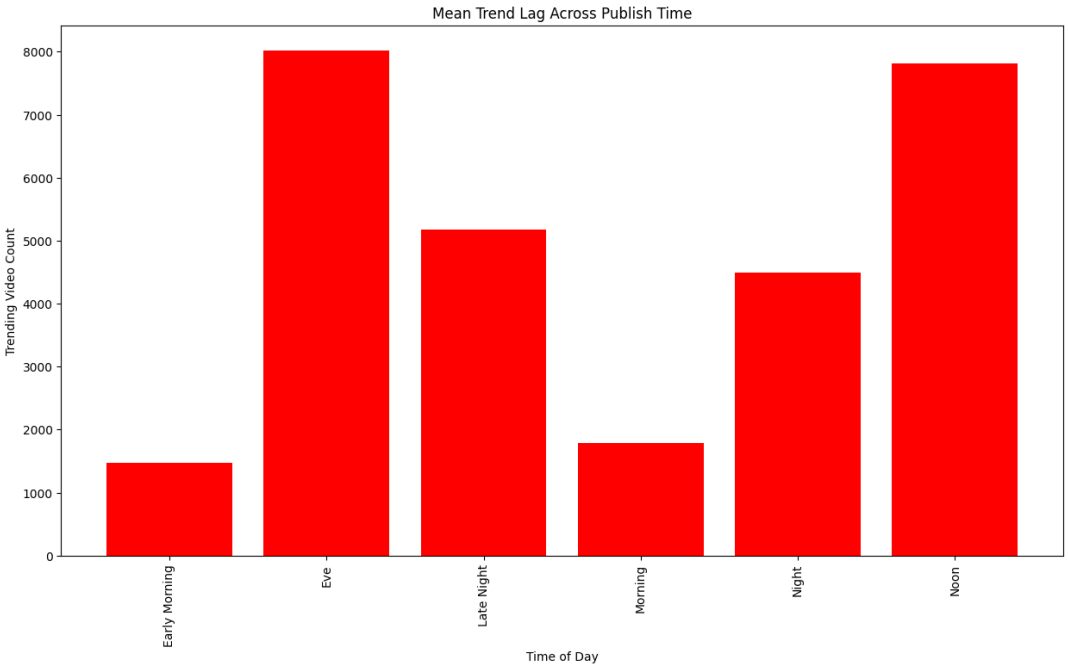

Figure 17: Mean of Trend Lag across Publish Time during the Day.

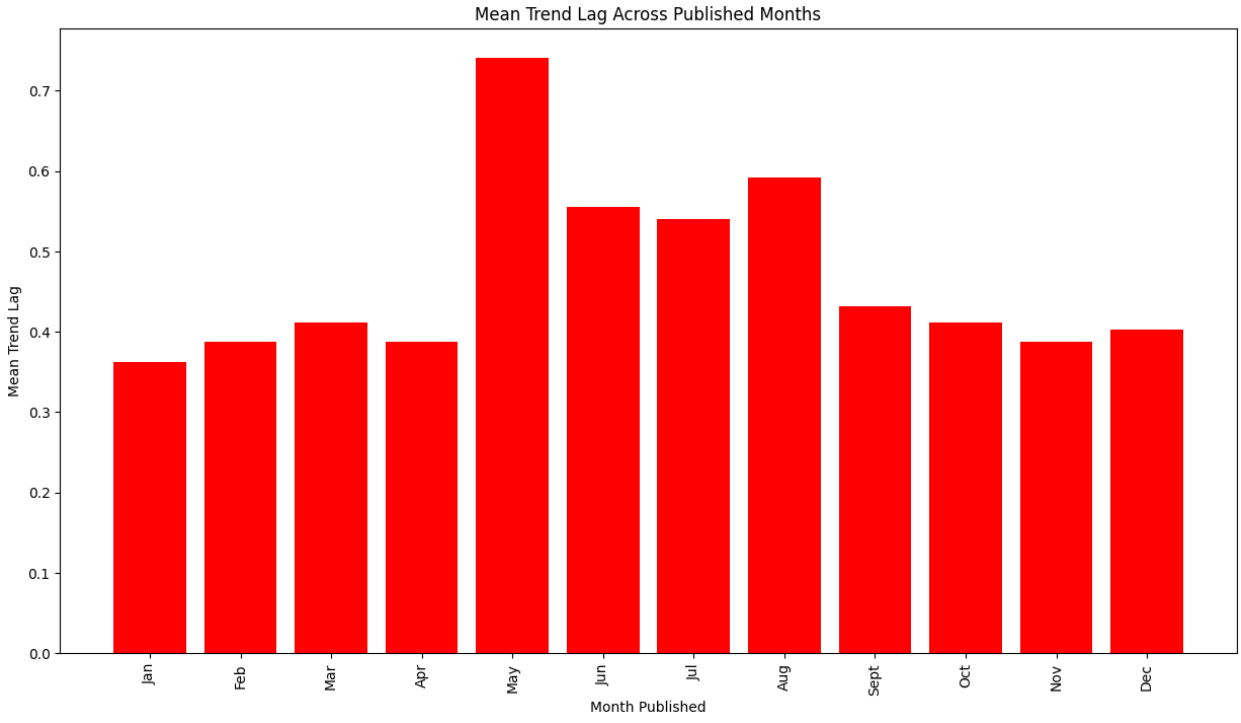

Figure 18: Mean of Trend Lag across Months from all Years.

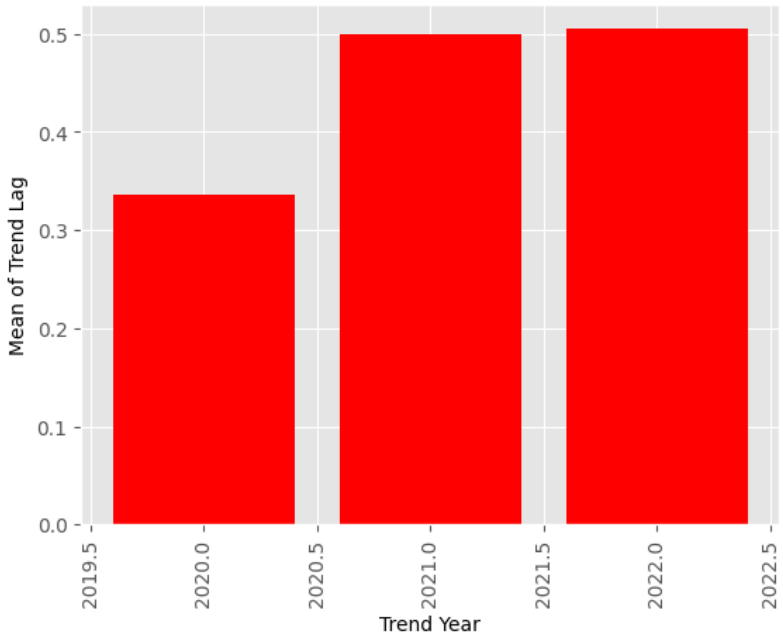

Figure 19: Mean of Trend Lag across Years.

As for engagement features, the data set is aggregated to find the initial values of likes, dislikes, comment count and views as well as the final values of these features during the trend period. For each of these features a new feature gain was calculated following **Eq (6)** (Orishko, 2020)

$$Feature\ Gain\ = Feature\ Max - Feature\ Min \qquad \textbf{Eq (6)}$$

The distribution of views gains shows heavy positive skewness in the data as shown in **Figure 20**, thus, the logarithmic transformation was introduced to the views features to better visualize the set as shown in **Figure 21**.

The mean of log of view gain across years in **Figure 22**, shows from first glance that the log of view gains in 2020 and 2021 are equal. However, from the number of months included in each year, it can be concluded that from August to December in 2020 harbor same mean view gain as a 12 month year in 2021. On the contrary, for a 9-month interval in 2022, the view gain of 2021 is also very high but lower than 2020.

Note, in the actual model, the concept of gains is taken and introduced as difference between two consecutive dates and as difference between current and feature minimum value for each video id.

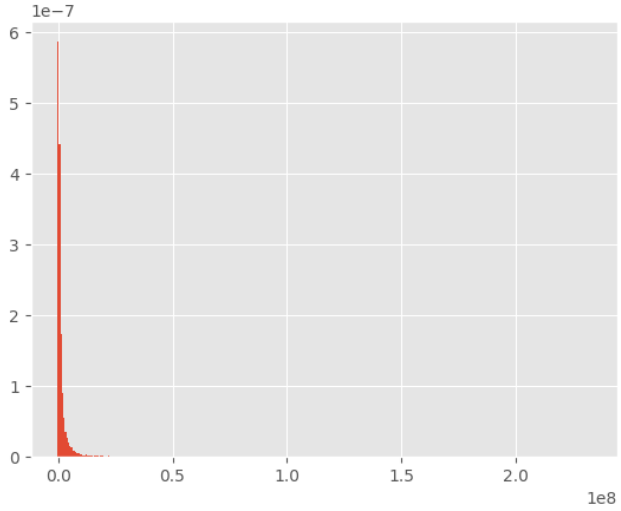

Figure 20: Views Gain Feature Distribution.

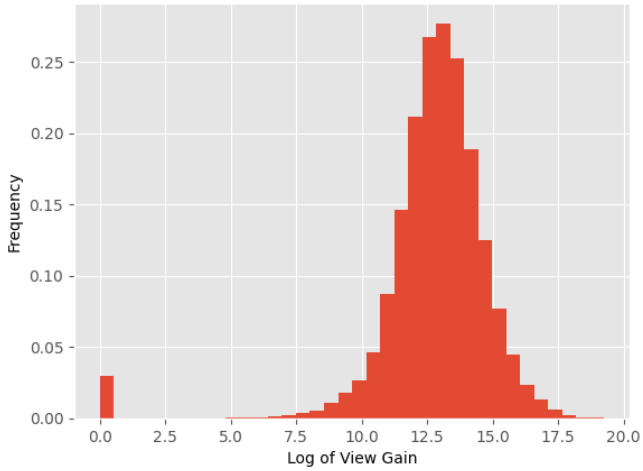

Figure 21: Log of View Gain Feature Distribution.

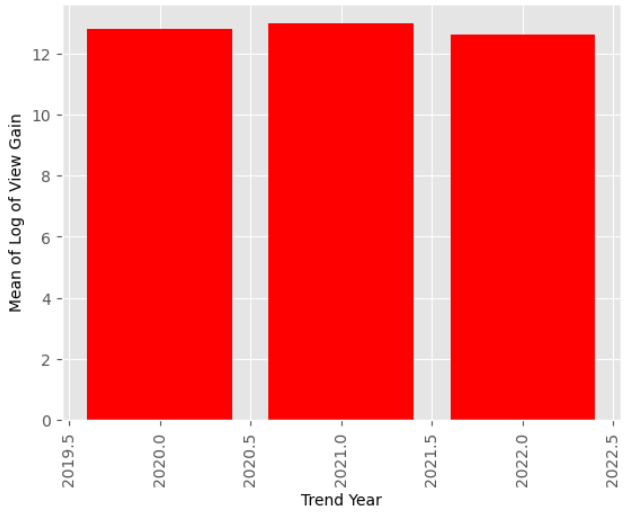

Figure 22: Mean of Log of View Gain across Years.

Similarly, across the years, mean of like gain, mean of comment gain and mean of dislike gain shown in **Figure 23**, **24** and **25**, respectively, seem to be the highest in 2020 and decrease accordingly when compared with their corresponding months. It is worth mentioning that the mean of dislike gain in 2022 in **Figure 24** dropped to zero, Since likes continued to be harbored in the same year, then the sudden drop can't be attributed to disabled ratings, and thus, YouTube users seem to exert less negative emotions on the platform, which makes this particular feature important for the popularity prediction problem.

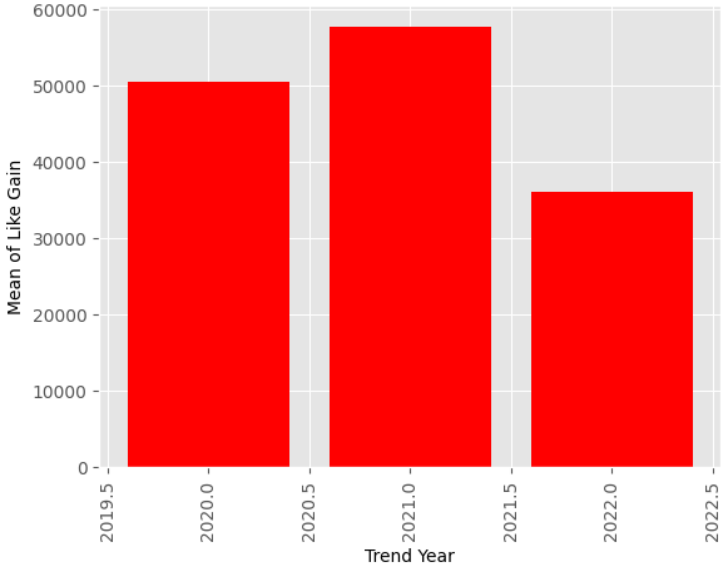

Figure 23: Mean of Like Gain across Years.

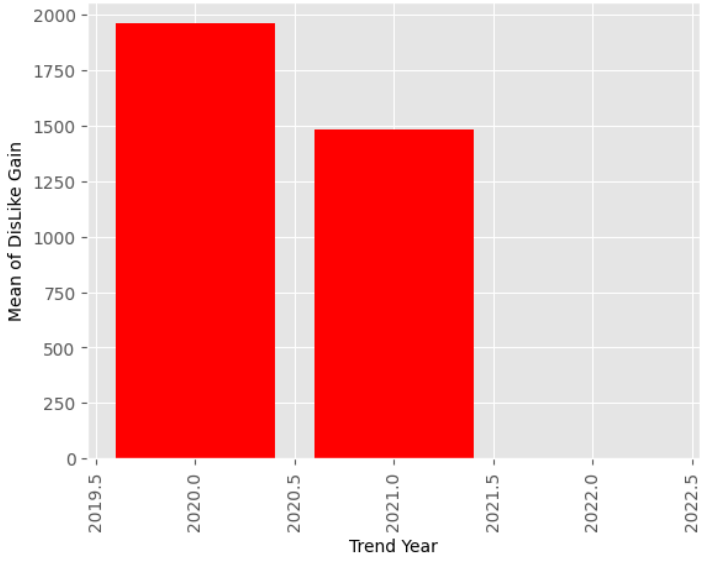

Figure 24: Mean of Dislike Gain across Years.

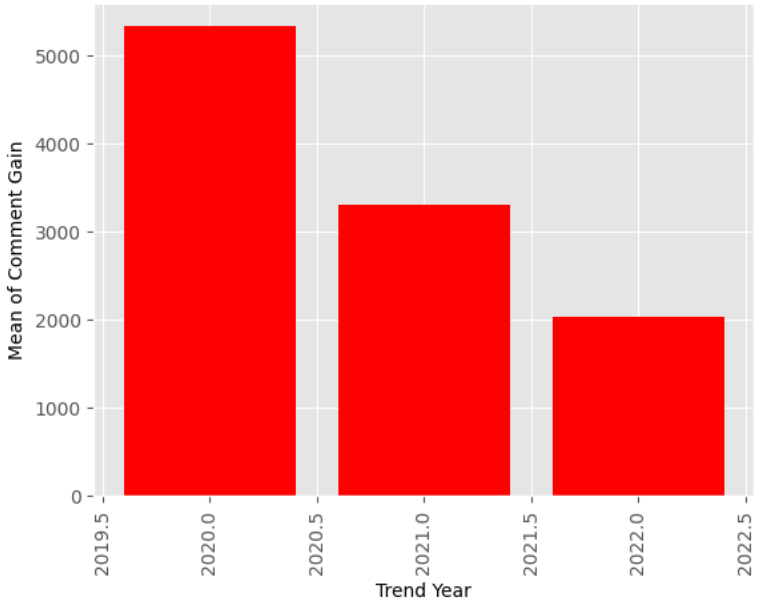

Figure 25: Mean of Comment Count Gain across Years.

As for video category, engagements can be affected by the category of the video, thus likes, views, comments and dislikes as well as the total number of trending days can have varying effects on videos from one category to the other. It is worth mentioning that comments are continuously losing power from one year to the other as shown in **Figure 25**.

Figure 26 shows the mean of trending days across categories, the highest trending days account for the nonprofits and activism category and then music with almost 5.5 days and 4.8 days respectively, while sports possessed the lowest mean of approximately 3.9 days.

In **Figure 27**, the highest mean of log view gain is associated with music and lowest with news and politics. Moreover, in **Figure 28**, **29** and **30**, music continues to be associated with highest like, dislike and comment gains.

In **Figure 28** comedy, education and entertainment seem to garner similar mean like gains and directly come after people and blogs in positive user response while news and politics seem to garner the lowest positive response from viewers.

As for **Figure 29**, as the music category is subjected to the highest viewer gains, it is expected to account for the highest negative mean responses, however travel and events seem to harbor the lowest dislike gain of zero, i.e. only positive responses from viewers while **Figure 30** shows lowest comment gain for sports category.

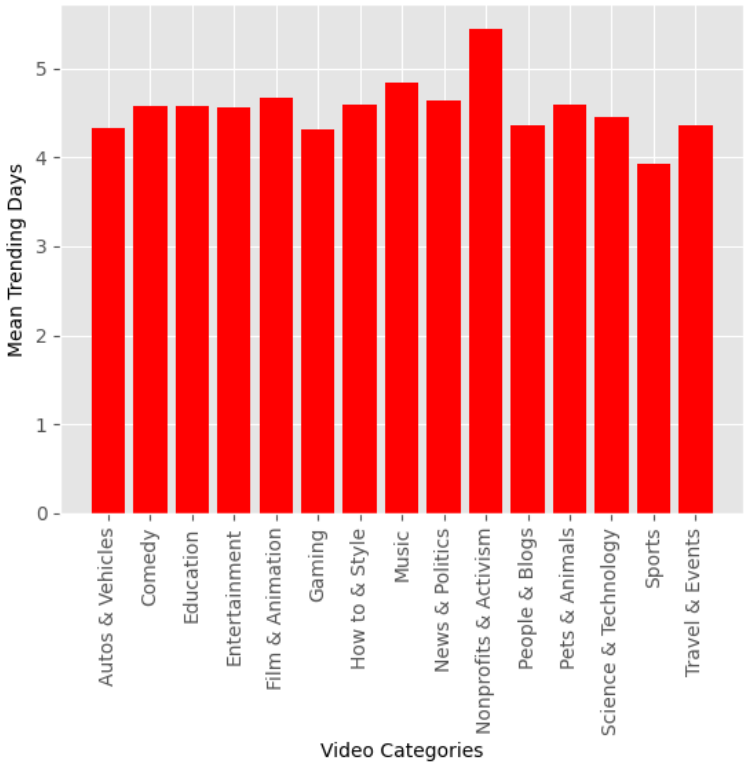

Figure 26: Mean of Trending Days across Video Categories.

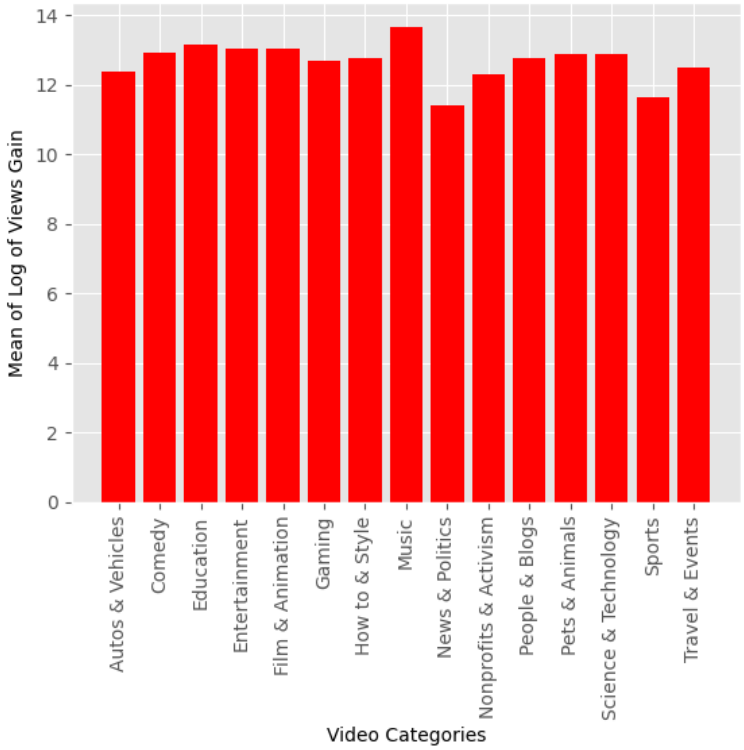

Figure 27: Mean of Log of View Gain across Video Categories.

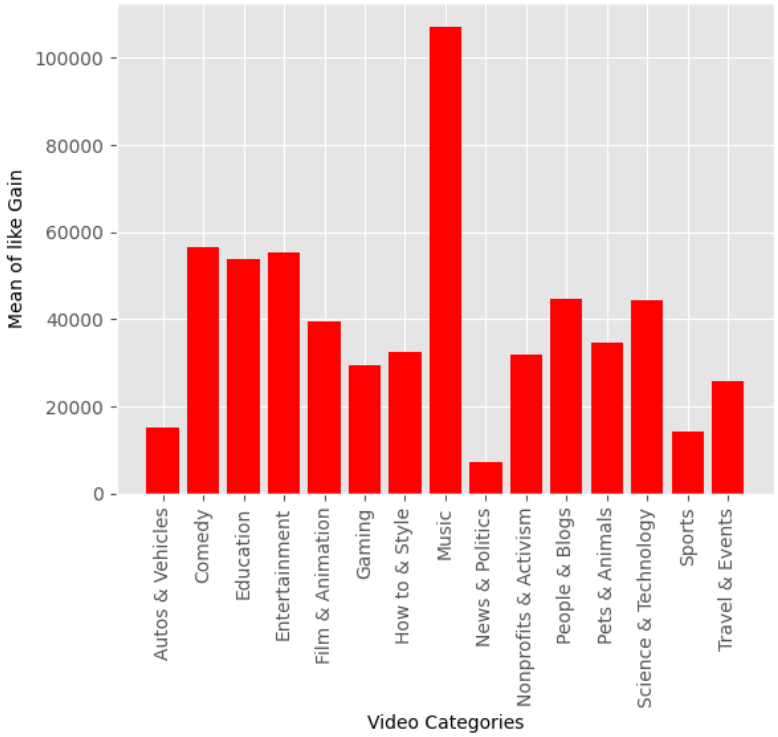

Figure 28: Mean of Like Gain across Video Categories.

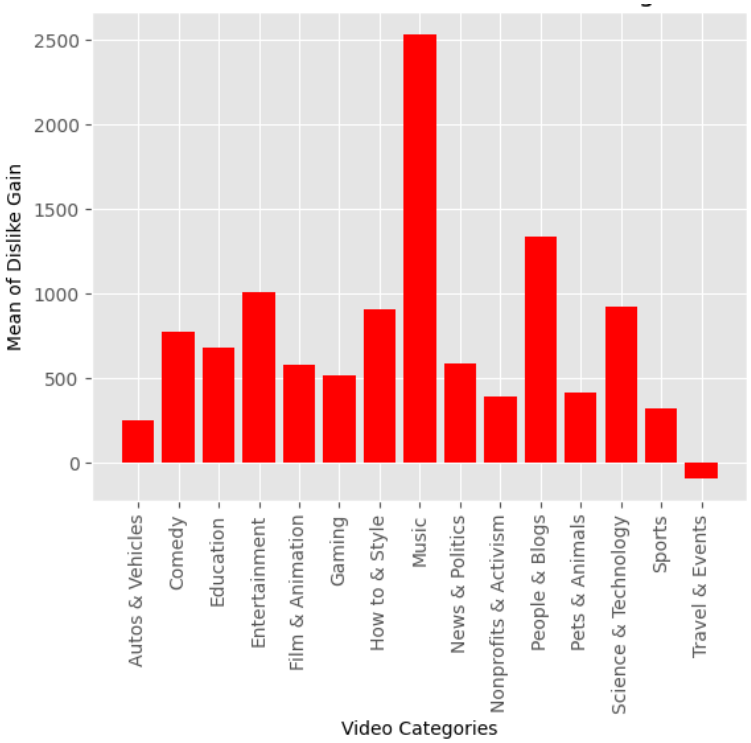

Figure 29: Mean of Dislike Gain across Video Categories.

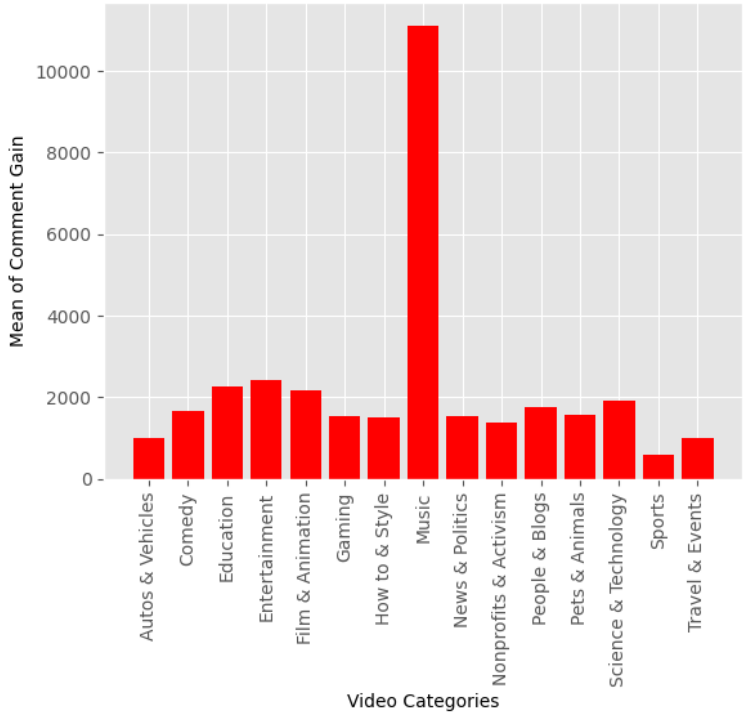

Figure 30: Mean of Comment Gain across Video Categories.

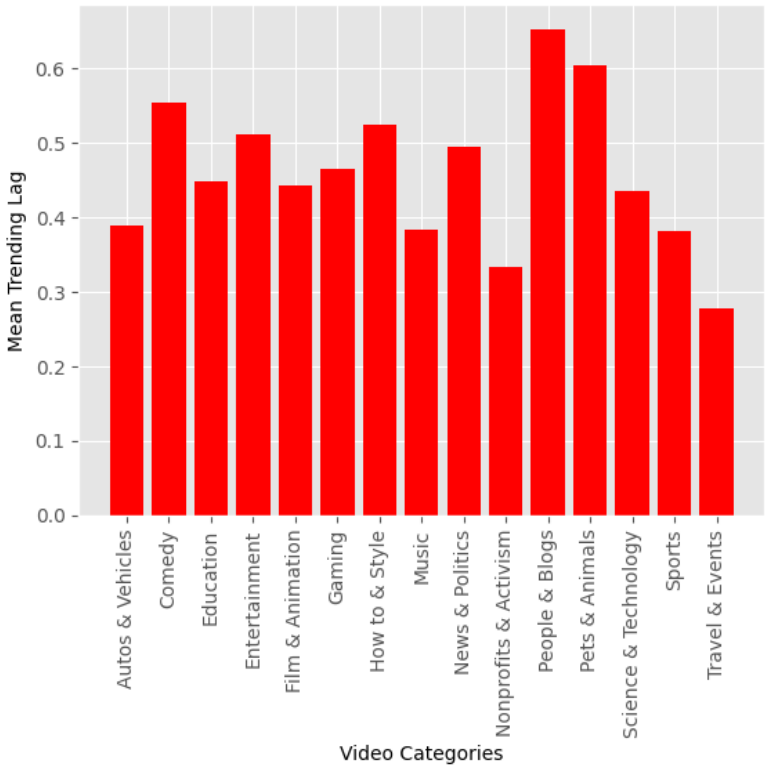

Figure 31: Mean of Trend Lag across Video Categories.

Figure 31 shows that the highest mean of trend lag occurs for people and blogs category followed by pets and animals category, while the lowest mean of trend lag is associated with travel and events category.

Additionally, a new feature named total engagement was initialized from the total values of likes, dislikes and comment count for last trending day of a video and was divided by the total number of views for that video to see the overall engagement on last day using **Eq (7)** (sehl and Tien, 2023).

$$Total\ Engagement = \frac{Likes + Dislike + Comment}{Views} \qquad \textbf{Eq (7)}$$

It is worth mentioning, that for visualization purposes, the log of views was taken to make **Figure 32** more interpretable as the total views can reach up to millions of views. Moreover, this feature will be calculated for each trending date of a video in order to capture the variation in day-to-day total engagement.

The total engagement feature measures how a user acts after viewing the video, however, there is an internal bias in this feature as a user can watch the same video more than once and not necessarily engage in each view time.

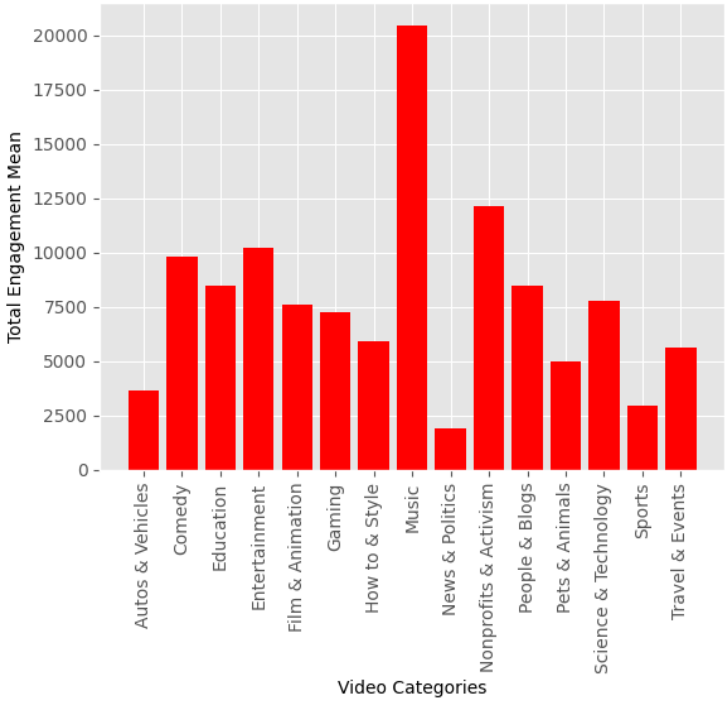

Figure 32: Mean of Total Engagement across Video Categories.

According to Sehl and Tien in an article that was published in February 2023, an engagement rate between 1 and 5% is considered to be good engagement metrics of a social content and any engagement higher than 5% is considered as very high engagement (Sehl and Tien, 2023).

Moreover, a new feature named likes to total likes and dislikes ratio was calculated from the total of the maximum values of likes and divided by the total of maximum values of likes and dislikes using **Eq (8)**

$$Likes\ to\ Total\ Likes\ and\ Dislikes\ Ratio = \frac{Likes\ Max}{Likes\ Max + Dislike\ Max} \qquad \textbf{Eq (8)}$$

The reason behind initializing this feature was to test the general user response towards videos and categories as a sort of engagement metric.

Thus, **Figure 33** indicates that all categories generate positive responses except for news and politics.

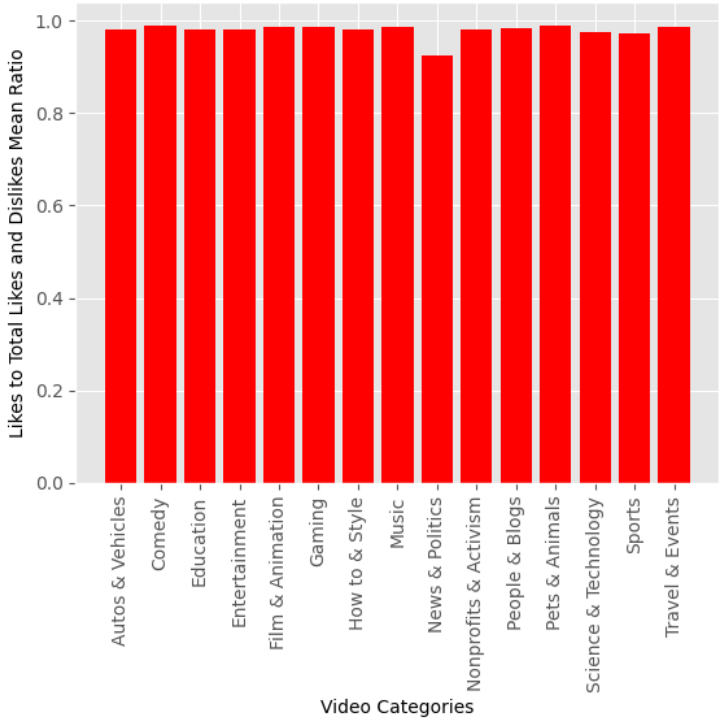

Figure 33: Mean of Likes to Total Likes and Dislikes Ratio across Video Categories.

From the provided insights about the features, it can be concluded that there should be implementation based on categories as they can convey intangible information regarding human engagements towards the various genres.

As for missing data in the dataset, Nans were only present in the description feature, as people can publish videos without writing anything in the description box and keeping them in the dataset. Thus they should not be discarded.

As for the textual features initial handling such as video title, tags and description using string library punctuation was removed from all the textual features, as well as lowercasing and emoji removal.

After the previously mentioned preprocessing of textual data, tokenization of the data was done using NLTK library. In addition, from the NLTK corpus English stop words were imported and removed only from the description and tags but were kept in the title as they hold valuable information to the user. Finally, lemmatization to reduce the word to their actual origin was done for all textual features, these were all done so to reduce the processing time and retrieve the important information from the text.

In an attempt to understand sentiment effect on trending data, initial Named entity Recognition (NER) implementation using en_core_web_lg pre-trained NER model for English language acquired from spaCy library was done on titles in order to identify key subjects and personnel that may influence the trending behavior. After counting the entities, the first 50 most frequent entities were acquired from the title text and are shown in **Table 6**, were a rank of 1 shows highest frequency and rank of 50 shows lowest frequency.

Table 6 showed what people are actually interested in viewing; this instance triggered finding the channels that people resort to find trendy subjects. Thus, The 50 most frequent channel titles was introduced and shown in the **Table 7**. This table shows the 50 most dominant channel titles that have highest recognition in trending data. It is worth mentioning that using NER from Spacy library with en_core_web_lg enabled the retrieved data to maintain the actual entity names which is quite impressive as the names of these channels are challenging for NLP.

Table 7 further addresses the hidden behavior of viewer selective engagement, interaction and popularity and will be taken into account in the engineered feature for proposed models. It is worth mentioning that rank equal to 1 show highest frequency and rank of 50 show lowest frequency.

Table 6: 50 Most Frequent Entities Found from Titles.

Rank	Entity	Rank	Entity	Rank	Entity
1	first	22	christmas	43	nba k
2	nfl	23	two	44	paris
3	one	24	night	45	august
4	hour	25	tonight	46	russian
5	nba	26	second	47	starlink
6	hermitcraft	27	september	48	sec
7	friday	28	celtic	49	summer
8	nbc	29	minecrafts	50	ford
9	wa	30	tnt		
10	week	31	russia		
11	gta	32	ukraine		
12	season	33	american		
13	cbs	34	billie eilish		
14	stephen	35	million		
15	snl	36	year		
16	minute	37	taylor		
17	lil drunk	38	winter		
18	ucl	39	israel		
19	hbo	40	june		
20	netflix	41	tokyo		
21	america	42	justin bieber		

Table 7: 50 Most Frequent Channel Entities Found from Channel Title.

Rank	Entity	Rank	Entity	Rank	Entity
1	nba	24	zhc	47	jeffreestar
2	nfl	25	tonight	48	wadzee
3	espn	26	dazn boxing	49	mandjtv
4	mrbeast	27	veritasium	50	moriah elizabeth
5	cbs	28	dantdm		
6	nbc	29	bwf		
7	sssniperwolf	30	lazarbeam		
8	ssundee	31	tom scott		
9	first	32	hbo		
10	spacex	33	bt sport		
11	saturday	34	pokémon youtube channel		
12	night	35	videogamedunkey		
13	shannon	36	warner bros picture		
14	abc	37	mrwhosetheboss		
15	ufc	38	brownlee		
16	ryan trahan	39	japan		
17	youngboy	40	hybe		
18	tnt	41	calebcity		
19	bein sport usa	42	america		
20	smtown	43	markiplier		
21	bangtantv	44	today		
22	jyp entertainment	45	sony		
23	matt	46	grian		

As for the tags, TF-IDF vectorizer was used to acquire the most 50 frequent words as shown in **Table 8**, where a tag of rank 1 has highest frequency and a tag with rank 50 has lowest frequency.

Table 8: Most Frequent Tags Found from Tags.

Rank	Tag	Rank	Tag
1	among	29	nba
2	audio	30	new
3	baby	31	nfl
4	best	32	night
5	challenge	33	official
6	day	34	oficial
7	ep	35	one
8	episode	36	part
9	every	37	reacts
10	feat	38	real
11	final	39	season
12	first	40	short
13	fortnite	41	show
14	friend	42	sport
15	ft	43	teaser
16	full	44	time
17	game	45	trailer
18	get	46	update
19	got	47	video
20	highlight	48	week
21	home	49	win
22	house	50	world
23	life		
24	lil		
25	live		
26	minecraft		
27	music		
28	mv		

3.5.3 Preprocessed and Feature Engineered Dataset:

From the initial exploratory analysis of the data that was shown in **section 3.5.2**, and based on the GRU needs to including sequencing of video id trending data, the input dataset was handled, preprocessed as the following:

3.5.3.1 Data Preprocessing and Numeric Feature Engineering

Numeric data preprocessing can be divided into two workflows; data cleansing and computed numeric features.

Numeric features include publish date trending date, tags, view count, likes, dislikes and comment count. For noise removal, the publish and trending dates were changed to Pandas date time to perform the following noise reduction steps for each video ID:

1. Remove data that was trending before the first day of data collection, which is 12th of August 2020.
2. Remove data that continued to trend to last date of data collection, which is 26th of September 2022.
3. Remove inconsecutive trending dates of a specific video id by inspecting the gaps between trending dates, any value higher than one; the row is discarded.

In this way, the noisy data regarding incomplete total trending days and history were eliminated. In addition, Video ID data was taken as key column in order to track the data throughout the various model phases. Moreover, for engagement metrics, the following features were calculated for each of the following numeric data for each video ID: view count, likes, dislikes and comment count to include:

1. Engagement threshold: is the value that caused the video to become trending.
2. Engagement gain: which is the difference in engagement for current engagement and engagement threshold.
3. Engagement gain rate: is the change in engagement gain for two consecutive trending dates.

Two more engagement metrics were calculated from combining view count, likes, dislikes and comment count:

1. Total engagement: is the summation of likes, dislikes and comment count over the number of views for each trending date.
2. Likes ratio: is the number of likes to the total number of likes and dislikes.

For these engagement features, negative infinity values, infinity values and none were expected to occur. Thus, bad values were replaced by zero as zero was the direct cause of these values.

In addition to that, panda date time values were handled by extracting from each publish and trend date features the following features:

1. Target feature remaining number of trending days 'trending days': is calculated by subtracting the current trending date from the total number of trending days for each video id.
2. Trend lag: is the difference between publish time and first trending date for each video id.
3. Year: year of publication and year of trend.
4. Month: month of publication and month of trend.
5. Week: day of week of publication and day of week of trend.
6. Time of day: time of day of publication.

The resulting values were either numeric or categorical features. Time of day was numeric feature and required further mapping into categorical variable that translated the 24 hours of the day into: early morning, morning, noon, eve, night and late night.

These variables can be treated as categorical variables for further processing. It is worth mentioning that trending dates did not have trending time thus the time of day feature was not applicable.

It is worth mentioning, that in order to maintain effective sequencing for GRU model, the model needs at least 3 previous video id rows of trending data in order to inference the remaining number of days. Thus, video IDs with total number of trending days less than 3 were discarded. In addition, video IDs with total trending days higher than 9 had bad GRU representation due to the network forgetting history as well as having low video count for the network to learn from.

After this step, the original publish and trend dates features were discarded and were replaced by the new engineered features and the new number of numeric feature set is 22.

.

3.5.3.2 NLP Preprocessing and Sentiment Analysis Feature Engineering

The text features include title, tags and description, and text preprocessing was done as the following using NLTK library:

1. For hashtags in tags features: they were converted into points to separate these values instead of tokenization as the sentiment analyzer takes strings as input.
2. Cleaning the text by removing links, lower casing, removing punctuation and emoji's, remove repeated letters, lemmatization and removing non English words.
3. Preparing strings of paragraphs to insert it into the sentiment intensity analyzer from NLTK library.

For sentiment analysis, Vader lexicon was used with the sentiment intensity analyzer to perform sentiment analysis as the following:

1. For nan values in description and [none] values inside tags: which means empty descriptions and tags; the sentiment was translated to neutral.
2. Compound score: the sentiment score was calculated from sentiment intensity, which is a combines positive, neutral and negative scores into a single value.
3. These polarity scores were then mapped into categorical features as very negative, negative, weakly negative, neutral weakly positive and positive.
4. Keywords ratio: the keywords ratio to total length of text was done using **Eq (9)** for each textual feature by calculating the length of stop words inside the text then subtracting it from the total text length and dividing it on the total text length.

$$Keywords\ Ratio = \frac{total\ text\ length - stopwords\ length}{total\ text\ length} \qquad \textbf{Eq (9)}$$

The resulting sentiment variables for each textual feature can be now treated as categorical variables for further processing, while keywords ratio as numeric feature.

3.5.3.3 Categorical Features Mapping, Encoding and Hashing

Existing categorical features include category ID, channel title and channel ID, comments disabled, ratings disabled. The last two mentioned features, which are Comments and ratings disabled were mapped from true and false to zero one without any encoding as they are dichotomous in nature . However, these two features are highly unbalanced towards being enabled which are expected to have low interpretability on the model.

Moreover, category ID was first mapped from numbers into actual categorical names provided from YouTube's API to make visualization of data easier and then was one hot encoded.

For channel ID and channel title, channel id was taken into account so to avoid channel title name changes. Since channel ID feature has high cardinality meaning the feature possesses large number of classes, making one hot encoding problematic due to computational complexity in model. As a result, hashing of Channel ID was introduced. The concept of hashing is to mask the categorical feature and transform it into a single numerical variable with iterable values. Note that hashing was done using feature hashing function from scikit-learn library without further encoding.

Lastly, all the categorical features that were computed in **sections 3.5.3** were then dummy encoded into new columns dichotomous features and replaced the original un-encoded data. The encoding and hashing of data is crucial in order to incorporate non-numeric data without introducing bias into the model.

As a result, 12 categorical features were mapped, encoded and hashed into 85 dichotomous new feature set.

3.5.3.4 Feature Engineered Dataset

Sections 3.5.3.1-3.5.3.3 showcase the details of feature engineering that was applied to the original 15 feature set acquired from YouTube API as shown in **section 1.5**, **Table 1**.

The resulting complete dataset contains 109 feature – including target feature ' trending days'- as shown in **Table 9**, where the description of each feature is thoroughly explained in the previous sub-sections of **section 3.5.3**.

Table 9: Feature Engineered Dataset.

Feature Name	Feature Name	Feature Name
Trending lag	Published day Friday	Trending month February
Views gain	Published day Monday	Trending month January
Likes gain	Published day Saturday	Trending month July
Dislikes gain	Published day Sunday	Trending month June
Comments gain	Published day Thursday	Trending month March
Views gain rate	Published day Tuesday	Trending month May
Likes gain rate	Published day Wednesday	Trending month November
Dislikes gain rate	Published day phase early morning	Trending month October
Comments gain rate	Published day phase eve	Trending month September
Total engagement	Published day phase Late Night	Trending year_2020
Likes ratio	Published day phase Morning	Trending year_2021
Title length	Published day phase Night	Trending year_2022
Title keywords ratio	Published day phase Noon	Title sentiment negative
Description length	Published month April	Title sentiment neutral
Description keywords ratio	Published month August	Title sentiment positive
Tags length	Published month December	Title sentiment strongly positive
Tags keywords ratio	Published month February	Title sentiment very negative
Views threshold	Published month January	Title sentiment weakly negative
Likes threshold	Published month July	Title sentiment weakly positive
Dislikes threshold	Published month June	Description sentiment negative
Comments threshold	Published month March	Description sentiment neutral
Comments disabled	Published month May	Description sentiment positive
Ratings disabled	Published month November	Description sentiment strongly positive
Category id autos & vehicles	Published month October	Description sentiment very negative
Category id comedy	Published month September	Description sentiment weakly negative
Category id education	Published year 2020	Description sentiment weakly positive
Category id entertainment	Published year 2021	Tags sentiment negative
Category id film & animation	Published year 2022	Tags sentiment neutral
Category id gaming	Trending day Friday	Tags sentiment positive
Category id how to & style	Trending day Monday	Tags sentiment very negative
Category id music	Trending day Saturday	Tags sentiment weakly negative
Category id news & politics	Trending day Sunday	Tags sentiment weakly positive
Category id nonprofits & activism	Trending day Thursday	Channel id
Category id people & blogs	Trending day Tuesday	Trending days
Category id pets & animals	Trending day Wednesday	
Category id science & technology	Trending month April	
Category id sports	Trending month August	
Category id travel & events	Trending month December	

3.5.3.5 Dimensionality Reduction and Feature Selection

The dataset shown in **Table 9** is split into train, test, and validation sets using 60:20:20 ratio, resulting in approximately 72k, 24k and 24k entries for training, validation and testing, respectively. Although the preprocessing removed many of the data entries, however, it improved the quality of the data while still maintaining a good dataset.

In addition, it is important to acquire features that can explain the underlying trends and behavior of the target feature. Thus, the total feature engineered dataset may potentially have features that have low target variable interpretability and thus only adds to the complexity of the model and increased computational power without introducing significant interpretational power to the model.

As a result, dimensionality reduction of the dataset was done using random forest on the training dataset to acquire the most important features to incorporate in the model. As a result, random forest regressor and select from model function from scikit-learn library were used to perform feature selection.

The feature selection criteria is a threshold equal to median of importance feature scores. Meaning, after fitting the random forest regressor with training dataset, the feature importance scores were acquired from select from model function, sorted in a descending manner and the median score was calculated from all the feature scores in the dataset. Consequently, features with scores higher than median threshold were selected and features below were discarded.

It is worth mentioning that the random forest dimensionality reduction technique was chosen because it can handle mixed datasets that contain numeric and categorical data. In addition, random forest algorithm can handle unscaled data as it is not affected by noise in the dataset due to performing regression using trees trained on subsets of instances and features (Ho, 1995)

Thus, the corresponding feature scores are computed from all the trained trees to acquire final importance scores, which makes the accuracy of such scores more reliable. In addition, features acquire higher scores when their absence cause significant loss in model accuracy. As a result, **Table 10** and **11** show the selected features with above median scores.

Table 10: Selected Features using Random Forest with Importance Scores

Feature Name	Importance Score	Feature Name	Importance Score
Views gain	0.323908383	Published month December	0.003384503
Comments gain	0.107346865	Published month April	0.002847608
Views gain rate	0.093964096	Published month May	0.002546141
Comments gain rate	0.093429659	Trending day Wednesday	0.002271354
Likes gain rate	0.026486746	Trending day Thursday	0.002239316
Likes gain	0.024054408	Published month February	0.002235239
Comments threshold	0.023243785	Trending day Tuesday	0.00222542
Trending lag	0.020550919	Published month September	0.002117745
Total engagement	0.017277832	Published day phase eve	0.002116167
Views threshold	0.017181796	Published day phase night	0.002103911
Likes threshold	0.016934663	Trending day Saturday	0.001982721
Description length	0.015968039	Trending day Friday	0.001962335
Dislikes gain rate	0.015522338	Published day phase noon	0.001960798
Description keywords ratio	0.015305821	Trending day Monday	0.001955804
Likes ratio	0.012756657	Trending month May	0.001922314
Tags length	0.011757541	Published day phase late night	0.001909623
Dislikes threshold	0.011236154	Trending day Sunday	0.001881525
Title length	0.00917191	Published day Saturday	0.001812713
Dislikes gain	0.009109591	Channel id	0.001811332
Title keywords ratio	0.009107867	Published day Friday	0.001780084
Tags keywords ratio	0.008181655	Trending year 2021	0.00175218
Published month January	0.005531052	Trending month October	0.001669238
Trending month March	0.004887059	Trending month December	0.001665391
Published year 2022	0.0039828	Trending month November	0.001657889
Trending month January	0.003821515	Category id Music	0.001635009

Table 11: Cont. Selected Features using Random Forest with Importance Scores

Feature Name	Importance Score
Published year 2021	0.001594819
Trending month June	0.001553377
Category id Entertainment	0.001546124
Published month June	0.001538917
Title sentiment neutral	0.001515922

In order to account for predictive accuracy optimization, iterative evaluation of the model was done based on different subsets of features. These features were selected cumulatively based on the importance scores to calculate the goodness of fit (R-squared) and MSE values of trained random forest models.

These values were plotted as shown in **Figure 34** and based on these plots, it can be shown that the features selected with the median which is 55 features provide the optimized number for accuracy in both plots as these plots seem to flatten after this point and minimize interpretational power.

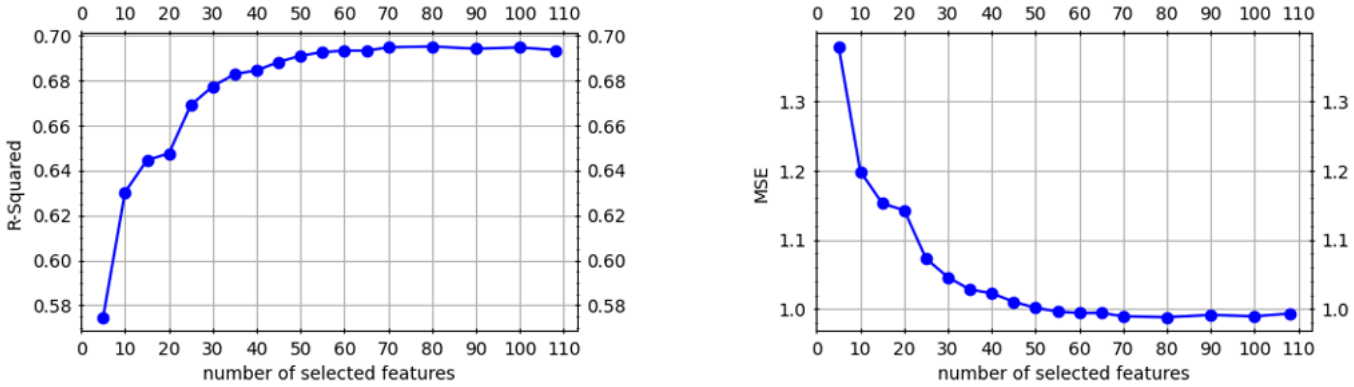

Figure 34: R- Squared and MSE vs. Number of Selected Features.

Lastly, other dimensionality reduction techniques were not used such as principle component analysis (PCA) due to only being effective on numeric data. Since the provided dataset contains encoded data, PCA is not recommended as this technique is built on computing variance. In addition, PCA transforms original data into new data with lower dimensions, which makes the new features less interpretable.

3.5.3.6 Sequencing, Normalization and Tensors

After concatenating the feature-selected dataset of 120k instances and 55 features, the data is normalized using min max scaler from scikit-learn library based on the train set to prevent any data leakage from test data into the trained model. In addition, min max scaler is used to preserve zero values in the dataset as they hold significant meaning.

In addition, sequences for each data was done to allow for sequential modeling, which was created by producing multiple sequences for same video id depending on the number of trending days the data is trending for. For example, a video ID that was trending for 4 consecutive day, the data for this video id would be added into the video sequences incrementally. For example, sequence of length 1 only includes one row of the data, which is data from previous trending day and sequence of length 2 only includes two rows from previous two days of trending and so on until the video becomes untrendy and there are no more sequences to produce.

By preparing the data this way, it allows for daily increment of video data after the video trends for three consecutive days.

Lastly, the data is converted into tensors and loaded into sets to be fed into the model via data loaders. It is worth mentioning, that for baseline models, the same created train, test and validation sets for GRU model were used, however, sequences were removed and were scaled using same scaler to prevent any potential bias in the trained baseline models stemming from improper data handling.

3.6 Algorithm Evaluation Indices

The prediction problem that this thesis deals with is of a regression nature. Thus, measures such as mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE) are used to measure the proposed framework performance as well as the discriminant coefficient (R^2) to measure the goodness of fit.

In general, lower values of MAE, MSE and RMSE indicate higher accuracy and lower prediction errors while a higher value of R^2 shows higher interpretability of the proposed framework.

3.7 Baseline Model

In order to evaluate the performance of the proposed GRU model, benchmarking the model's evaluation scores with baseline models and state-of-the-art models are needed. The proposed comparative are random forest, gradient boosted decision trees, XGBoost, linear regression and SVR with Gaussian radial basis function models, which are discussed in **section 2.3** from the literature review chapter of this thesis (Nisa et.al, 2021; Haimovich et. al, 2022; Sibo et. al, 2021; Trzcinski and Rokita, 2017).

Chapter Four

Results and Discussion

The following discussion displays data experimentation on machine learning models, by training GRU models with full feature set and a subset of selected features and fine tuning the hyper-parameters of both models in order to acquire the best performing model from each feature sets and lastly comparing the performance of these models against baseline models as discussed in **section 3.4**.

4.1 Selected Feature Set in Comparison with Full Feature Set

As discussed in **section 3.5.3.5**, the feature engineered dataset contains 109 feature including the target feature which is the remaining number of trending days 'trending days'. This set was split into train, validation and test sets and the train set was used to extract important features.

It was important to use the training set to extract the key features and not the complete dataset in order to prevent data leakage into the predictive model. Thus, it can be concluded that feature selection algorithms are sensitive to the data provided inside these features and the key features can differ depending on the available instances.

As a result, selecting the random forest feature selection technique provides a robust approach to data sensitivity as it uses random number of features and random subsets of the instances for each tree inside the forest and then calculates the average importance scores from these trees to acquire the final score. The robustness of the technique was further ensured by training the model on various datasets and plotting the resulting MSE and R-squared against number of features used, as shown in section 3.5.3.5, **Figure 34** to ensure that 55 features provide optimal prediction accuracy.

The highest importance scores were associated with views, comments and likes gains and gain rates. While dislike gains and gain rates showed lower importance to the previously mentioned gains and rates.

In addition, thresholds of views, comments, likes and dislikes were found to be of importance to popularity prediction as they directly influence the video by introducing into the trend list. Moreover, among these thresholds comments threshold was the most important. This finding is consistent with **Table 2** in **chapter 2,** which states the features observed in the literature.

Moreover, the period between publication and becoming trending, which is denoted as trend lag showed to be important for the popularity problem. As for engagements, the total engagement, which is a measure of interaction on a video in relevance to the view count, was found to be important and more important than the like's ratio, which measures the count of likes in reference to total rating count i.e., likes and dislikes.

As for keywords ratio for title, tags and description, which measures how many informative words are inside a sentence or a paragraph, the highest importance was attributed to tags followed by title and lastly by description keywords ratio. This feature is complementary to 'number of words in title', which is also observed in literature **Table 2**.

As for publishing and trending data, publishing in January, December, April, May, February, September and June were relevant to the popularity problem using the median selection criteria while trending in March, January, May, October, December and November, were not based on the same criteria.

In addition, publishing on Saturday and Friday showed importance while other publishing days did not. However, for trending days, all days of the week showed importance for the popularity prediction problem. Moreover, posting videos in Eve, night, Noon and late night were found to be important and the remaining period of the day were not found to be relevant based on the selection criteria.

Moreover, channel id was found to be important as it entails hidden information about the specific characteristics of the channel itself. From category id, only music and entertainment were found of importance to the popularity problem based on the selection criteria. This is also complimentary to **Table 2**, as author or source of the item is frequently used for prediction.

Lastly, from sentiment analysis, only title sentiment neutral was found to be of importance to the prediction problem based on the selection criteria. It is worth mentioning that the selected features are consistent with the observed features in the literature as in **Table 2**.

4.2 GRU Model Architecture and Implementation:

The GRU is built using Pytorch library and implemented with torch.nn Module to initialize and instantiate the model. The architecture of the GRU model takes input, hidden and output sizes as hyper-parameters. The input size of the model references the number of independent features that are used to inference and predict the target feature, while the output size denotes the size of the target feature. Since, the target feature is of numeric nature with single value prediction then the output size is one.

As for the hidden size, in order for the GRU model to capture complex representations found inside the dataset, hidden layers are needed. Thus, the hidden size of the GRU model leverages the model with higher ability to learn and capture complex relationships within the data; however, increasing the hidden size affects the model by introducing higher model computational complexity (Goodfellow et. al., 2016).

In addition to hidden size in capturing complexity, the number of GRU layers inside the model also affect the capacity of the model to capture complex data patterns. Thus, balancing between the number of layers and number of neuron inside the hidden layers of the model is of crucial importance in the tuning phase of the model (Goodfellow et. al., 2016).

More importantly, in order to capture non-linearity in the network, activation function is used. This function is applied to the output of the neurons of the layers. The choice of the activation function depends on the data, problem and desired behavior. Thus, since popularity data contains large values related to views, comments thresholds and contains some negative inputs related to hashed channel id, trend lag and many other normally ranging values. Then using soft plus activation function that is aimed at such a dataset is validated (Zheng et. al., 2015).

In **Eq (10)** , the soft plus activation function is essentially exponential with log transformation that constraints the output to always be positive (Zheng et. al., 2015):

$$softplus(x) = \log(1 + e^{x}) \qquad \textbf{Eq (10)}$$

Moreover, L2 regularization is introduced by calculating L2 regularization loss and adding it to the validation loss in order to produce a penalty on the Model to prevent it from overfitting to the

training data and learning noise. The L2 regularization loss is computed using **Eq (11)** (schmidhuber, 2015):

$$L2\ regularization\ loss = \lambda * ||W||^2 \quad \textbf{Eq (11)}$$

Where, λ is the regularization parameters = 0.0001, and $||w||^2$ is the squared L2 norm of weight vector.

In addition to L2 regularization, dropout is also used to prevent the model from overfitting. Drop out probability works by randomly eliminating portion of the input units or neurons inside layers to help the model not depend on specific input values when making predictions (Goodfellow et. al., 2016).

Moreover, the loss function used in GRU is mean squared error (MSE) loss function with Adam optimizer. Adam optimizer works by adaptively computing and changing value of the learning rate for each parameter so to ensure optimal convergence for the model (Goodfellow et. al., 2016).

As for the training phase, it depends on factors such as learning rate and number of epochs. The learning rate is responsible for the step size that causes the model to update its' parameters thus the higher the value of the learning rate the faster the model converges. However, faster convergence does not necessarily mean getting better results. Thus, a change in the learning rate will directly affect the number of epochs used for training (Goodfellow et. al., 2016).

The benefit of training the model using epochs and batch sized is that this way allows the model to see the data using random batches multiple times which can make the model perform better. The idea with epochs is that it allows the training set to be passed through the network to create gradient updates. However, the data is passed to the model using data loaders and batch sizes, which can serve in introducing subsets of the data in each iteration while efficiently lowering memory utilization as well as making the training set more resilient to outliers as the batches are randomly shuffled after each batch selection (Goodfellow et. al., 2016).

For the implementation of the GRU model, experimentations were done by training and tuning the model using the feature selected set and the complete feature set that are shown in **Table 10**, **11** and **Table 9**, respectively.

Thus, for each GRU model, the model hyper-parameters were chosen based on the tuning process to acquire the highest performance evaluation metrics, by expanding the model capacity without excessively increasing the computational complexity of the predictive model nor overfitting the model to the training data.

4.2.1 GRU Hyper-Parameter Selection and Evaluation across Popular Days Left

In order to acquire the most efficient GRU model, hyper-parameter experimentation must be done as an initial step to make sure that these parameters serve to help the model learn how to predict the remaining days of popularity by understanding the underlying complexity of the data.

As a result, for the selected feature set, the initial model was trained on various input sizes, such as 16, 32, 64 and 128 with a number of GRU layers including 1, 2, 3 and 4 layers and using learning rates between 0.0015 and 0.002. As for epochs, epoch numbers of 100 and 150 were used with the following batch sizes 16, 128, 512, 2048, 4096, 8192, 32768 and 65536.

As a result, after some experimentation with same hidden size of 32 and 3 GRU layers; batch size 2048 seemed to provide the highest R-squared value and lowest RMSE value as shown in **Figure 35**.

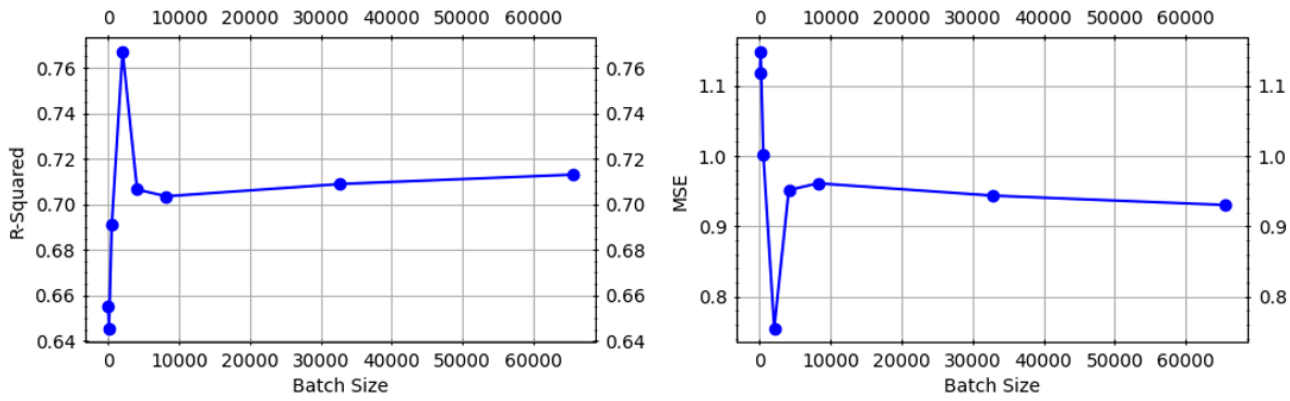

Figure 35:R- Squared and MSE vs. Batch Size for GRU Model

It is worth mentioning that after further experimentation with variations of input size, learning rates, GRU layer numbers and epochs; the produced models performance evaluation is shown in **Table 12**.

Table 12: Performance Evaluation of GRU Models Trained with Batch Size 2048.

	Model Hyper-Parameters	R-Squared	MSE	RMSE	MAE
1	Hidden size 32, layers 3, learning rate 0.0016 and epoch 100.	0.708	0.947	0.973	0.740
2	Hidden size 16, layers 4, learning rate 0.0018 and epoch 100.	0.753	0.801	0.895	0.690
3	Hidden size 32, layers 4, learning rate 0.0018 and epoch 100.	0.765	0.763	0.874	0.669
4	Hidden size 64, layers 4, learning rate 0.0018 and epoch 100.	0.747	0.817	0.904	0.687
5	Hidden size 32, layers 3, learning rate 0.0018 and epoch 100.	0.767	0.755	0.869	0.666
6	Hidden size 32, layers 3, learning rate 0.0018 and epoch 150.	0.702	0.966	0.983	0.742
7	Hidden size 32, layers 3, learning rate 0.002 and epoch 100.	0.711	0.934	0.967	0.735
8	Hidden size 64, layers 3, learning rate 0.0018 and epoch 100.	0.758	0.784	0.885	0.670

From **Table 12**, model 3, 5 and 8 possess closest R-squared, MSE and RMSE values with different hyper-parameters. By reflecting on these hyper-parameters, higher hidden size should provide the model with better interpretability of relationships in the dataset. However, increasing the number of hidden size from 32 to 64 while keeping other parameters the same had negligible effect on the performance and only increased the computational complexity as shown between model 5 and 8. As a result, model 8 is disregarded.

As for Model 2 and 5, the only difference in hyper-parameters is in GRU layers with four and three layers, respectively. Usually increasing GRU layers provides the model with higher capacity for capturing complexity of data. However, adding another layer to model 2 also had negligible effect

on performance metrics and only increased the computational complexity of the model. As a result, the selected model is model 5.

4.2.2 GRU Model Trained on Feature Selected Set

The first proposed GRU model is model 5 from section 4.2.1 and is trained based on the feature-selected set and has the following tuned hyper-parameters:

1. Input size: the input dimensions which is 55
2. Hidden size: 32
3. Output size: 1
4. N layers: 3
5. Learning rate: 0.0018
6. Number of epochs: 100
7. Batch size: 2048
8. L2 lambda: 0.0001
9. Dropout probability: 0.4

For the provided model, the MSE loss across epochs was plotted for training and validation sets as shown in **Figure 36**. From the shape of the graph it can be seen that the curve of the training and validation losses match and follow an almost smooth curve that converges.

As for the model's performance metrics shown in **Table 13** and **Figure 35** show the convergence of R^2, MSE, RMSE and MAE, across epochs for Training, validation and test sets. It is worth mentioning that testing the model on the test set was done for visualization purposes and was not included in the update of parameters like the validation set. Thus, the proposed GRU model trained on selected feature set is a good fit model that can interpret 76.7% of the target feature with a root mean squared error in prediction equal to 0.869 days. Meaning the error in prediction in the model is 0.869 of a day, which is roughly 20 hours and 52 minutes error.

In addition, since the GRU model was trained to predict multiple remaining days, the performance of the model in terms of MSE and RMSE was plotted against each remaining day, from 1 to 8 remaining trending days as shown in **Figure 37**. From this figure, the remaining number of days 5 and 6 seem to have the highest root mean squared error equal to approximately 1.15 day.

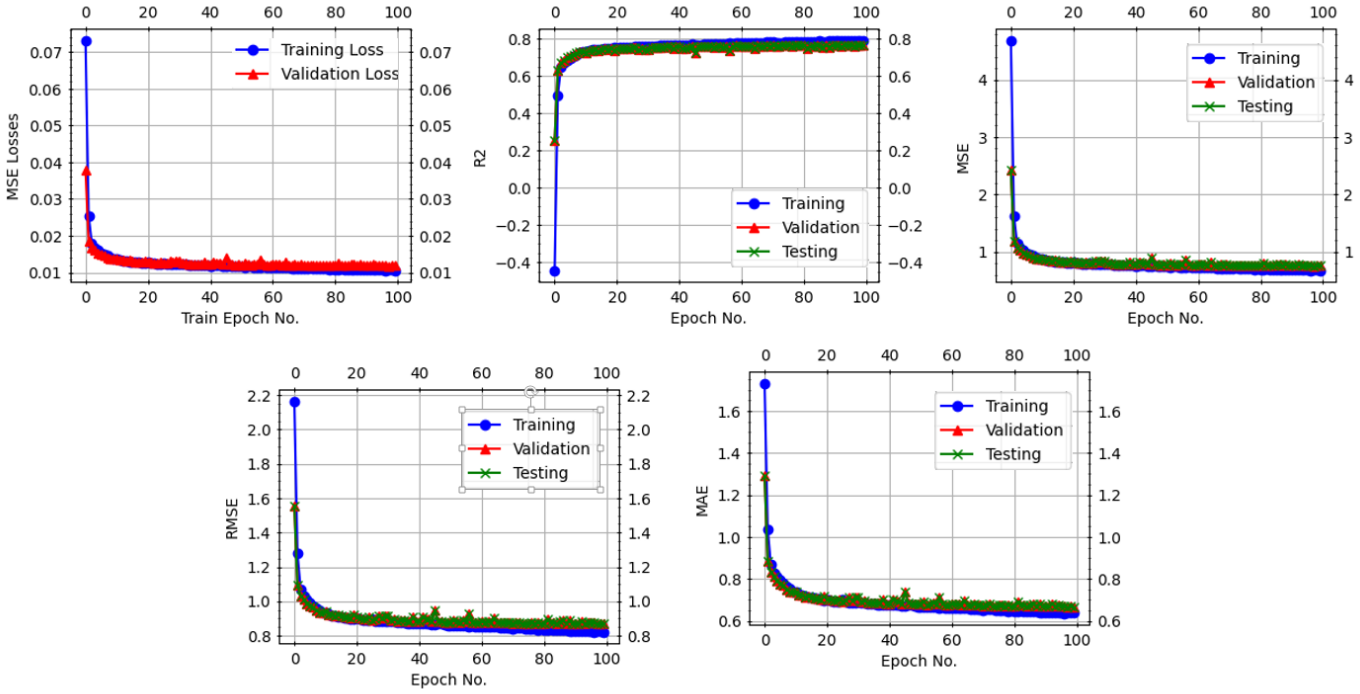

Figure 36: Visualization of MSE Loss and Performance Metrics across Epochs for GRU model on Selected Feature Set.

Table 13: Performance Metrics for GRU Model on Selected Feature Set.

Metric	GRU
R^2	0.767
MSE	0.755
RMSE	0.869
MAE	0.666

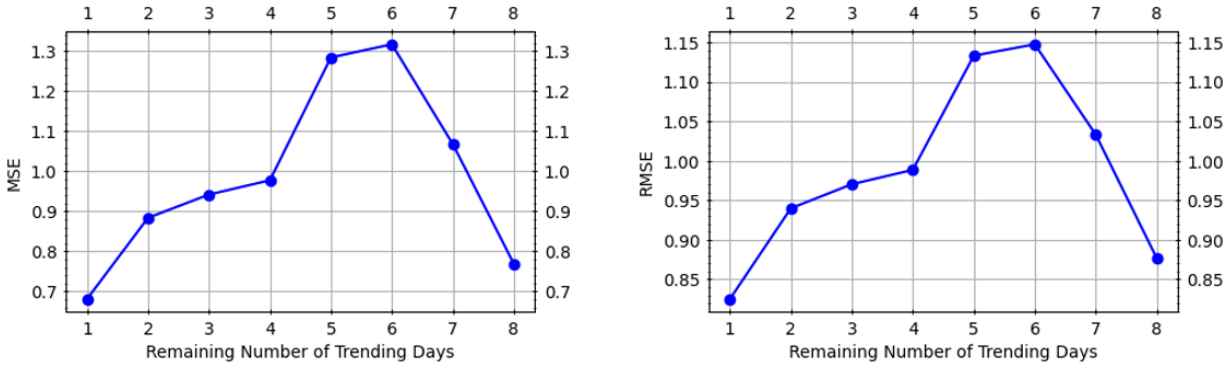

Figure 37: MSE and RMSE of GRU Model Trained on Selected Features across Remaining Trending Days.

4.2.3 GRU Model Trained on All Feature Set

The second proposed model is with all feature set training based on the tuned hyper-parameters of the converged model in section 4.2.2. This is done in order to visualize the effect of input size on the model.

From **Figure 38**, the MSE loss across epochs for validation and training sets show that the model converges. However, in this case the input size is approximately doubled when compared to the previous model making the chances of acquired correlated and redundant features more probable and effectively reducing the ability to interpret the relationships in the data and make it more prone to overfitting.

This finding is further backed by the visualization of performance metrics in **Figure 37** and **Table 14**. Where, the trained model seem to have substantially lower R^2 of 0.687 and a higher RMSE value of 1.006 day error in prediction.

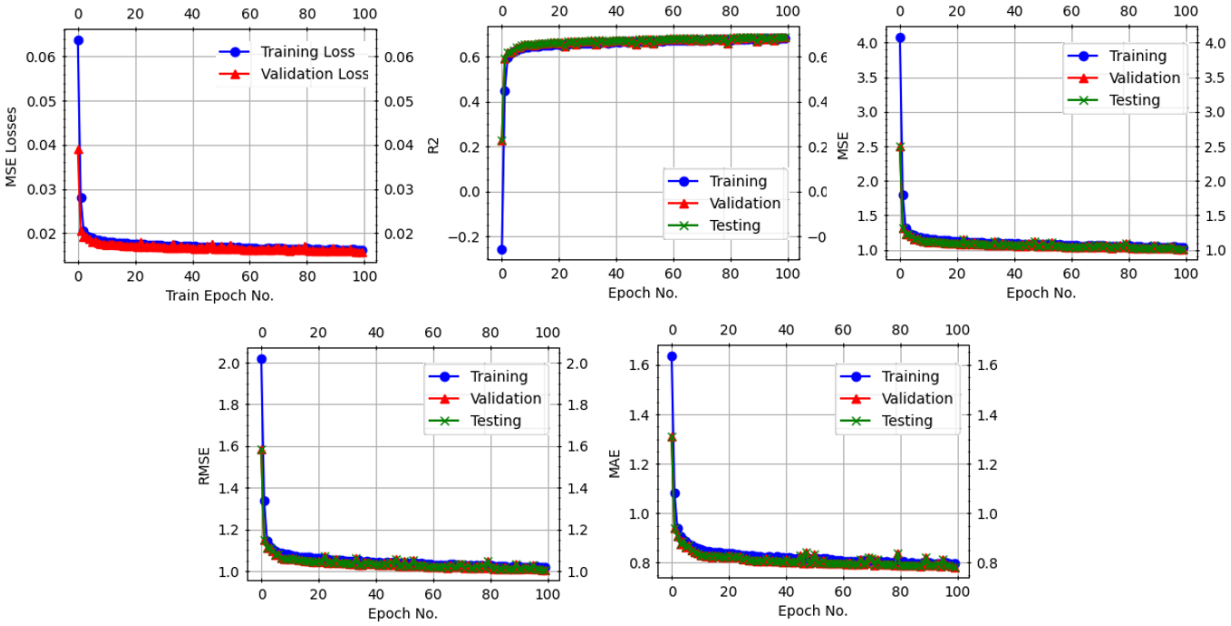

Figure 38: Visualization of MSE Loss and Performance Metrics across Epochs for GRU model Trained on All Feature Set based on Previous GRU Hyper-Parameters.

Table 14: Performance Metrics for GRU Model on All Feature Set with Previous GRU Hyper-Parameter.

Metric	GRU
R^2	0.687
MSE	1.012
RMSE	1.006
MAE	0.783

Performance comparisons between the first and second GRU models in **sections 4.2.2** and **4.2.3** can be done as both were trained using the same model architecture with only difference in feature input size as shown in **Table 13** and **14**. In theory, adding significant variables should help improve the model's explanatory power, however, when adding variables with low explanatory power, it can cause the model to have worse performance and potentially over fit to the training data.

As a result, the proposed model in section 4.2.2 with 55 features, and 2084 batch size, 32 hidden size, 3 GRU layers, 0.0018 learning rate outperforms all other model experimentations, including the one with the full dataset.

4.3 Baseline Models Comparison with Proposed GRU Model

In order to further understand the chosen model performance, comparisons with state of the art models and baselines must be done. These comparative models are trained on the same dataset with the same train, validate and test sets in order to make referenced performance comparisons.

It is worth mentioning that the comparative models that were taken from literature regarding popularity problems have not taken the remaining number of trending days as a target variable but rather focused their work on classification and engagement metrics predictions such as views as discussed in **chapter 2 section 2.3**. Thus, **Table 15** shows the metric performance of GRU model, XGBoost, Gradient boosted Decision Trees, Random Forest, Linear Regression and support vector regression (SVR) trained on the feature-selected dataset, which is the dataset that resulted in the best GRU model performance.

Table 15: Evaluation Metrics for GRU with Selected Feature Set and Comparative Models

Metric	GRU	XGBoost	Gradient Boosted Decision Trees	Random Forest	Linear Regression	SVR
R^2	0.767	0.728	0.665	0.6953	0.078	0.325
MSE	0.755	0.880	1.08	0.996	2.989	2.187
RMSE	0.869	0.938	1.04	0.998	1.729	1.478
MAE	0.666	0.724	0.828	0.784	1.425	1.144

From **Table 15**, it is evident that the chosen GRU model trained on the selected feature set outperforms the other comparative models in all performance metrics, R^2, MSE, RMSE and MAE. Followed closely by XGBoost, random forest, gradient boosted decision trees, SVR and lastly linear regression.

XGBoost shows the closest behavior to the GRU model, with R^2 equal 0.728 and root mean squared error equal to 0.938 day, which means that the XGBoost root mean squared error of prediction is 22.5 hours, varying from GRU by 2 hours.

In addition, random forest possess 69.5% explanatory power of the target feature with root mean squared error equal to 0.996 day, which is approximately 1 day of prediction error. In addition, gradient boosted decision trees show explanatory power of target feature equal to 66.5%, and a root mean squared error equal to 1.08 of a day.

Additionally, the XGBoost is an enhanced variation of the Gradient Boosted Decision Trees model by applying intricate L1 and L2 regularization, thus from previous experiments and theory, the XGBoost is expected to outperform the Gradient Boosted Decision Tree which is further affirmed in this implementation.

As for SVR model with radial basis function kernel, it exhibits significantly lower explanatory power of the target feature with R^2 equal to 32.5% and root mean squared error equal to 1.5 days. Moreover, the linear regression model showed the least goodness of fit with R^2 equal to 0.078, and root mean squared error of 1.7 days. When testing for multicollinearity in linear regression using VIF from Scikit-learn library, 22 features out of the selected 55 showed moderate to high multicollinearity.

Lastly, it is worth mentioning that exploring non-linear regression was not possible due to requiring large memory.

Figure 39, **40**, **41**, **42, 43** and **44** show the predicted values against actual values for Models proposed in **Table 15**. It is worth mentioning that since GRU model contained sequences, the resulting sample that was plotted contains different point of the dataset.

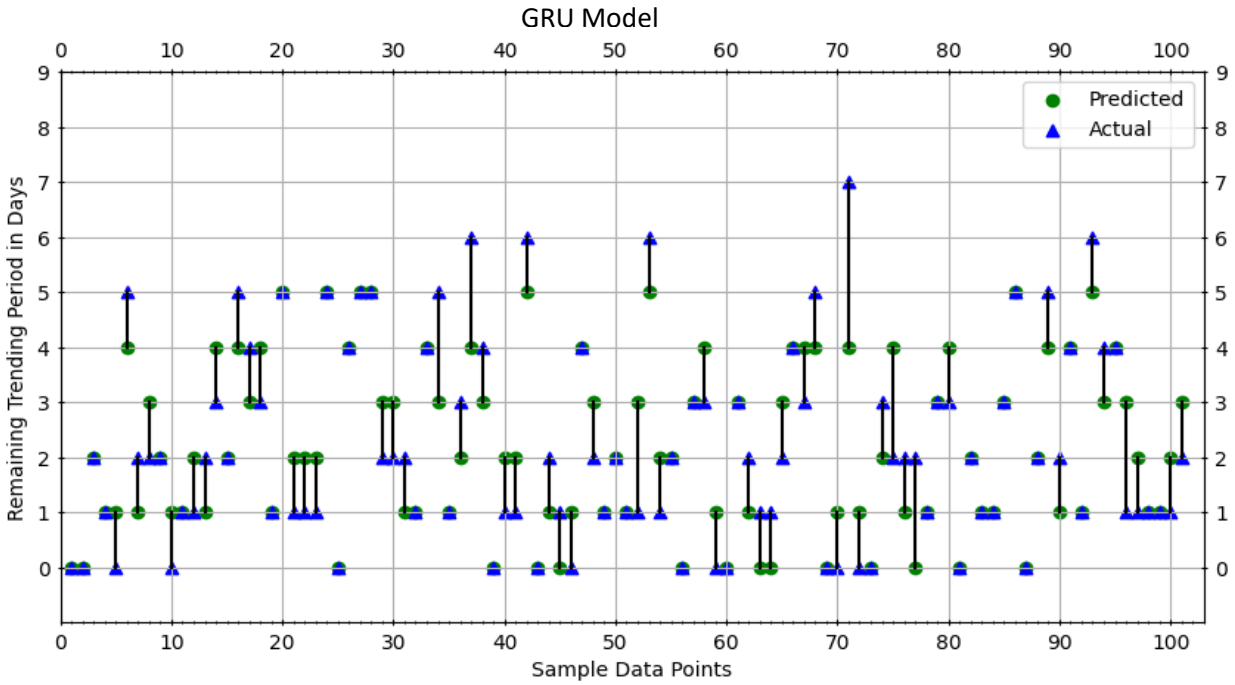

Figure 39: Trending Days Actual vs. Predicted Values using GRU Model.

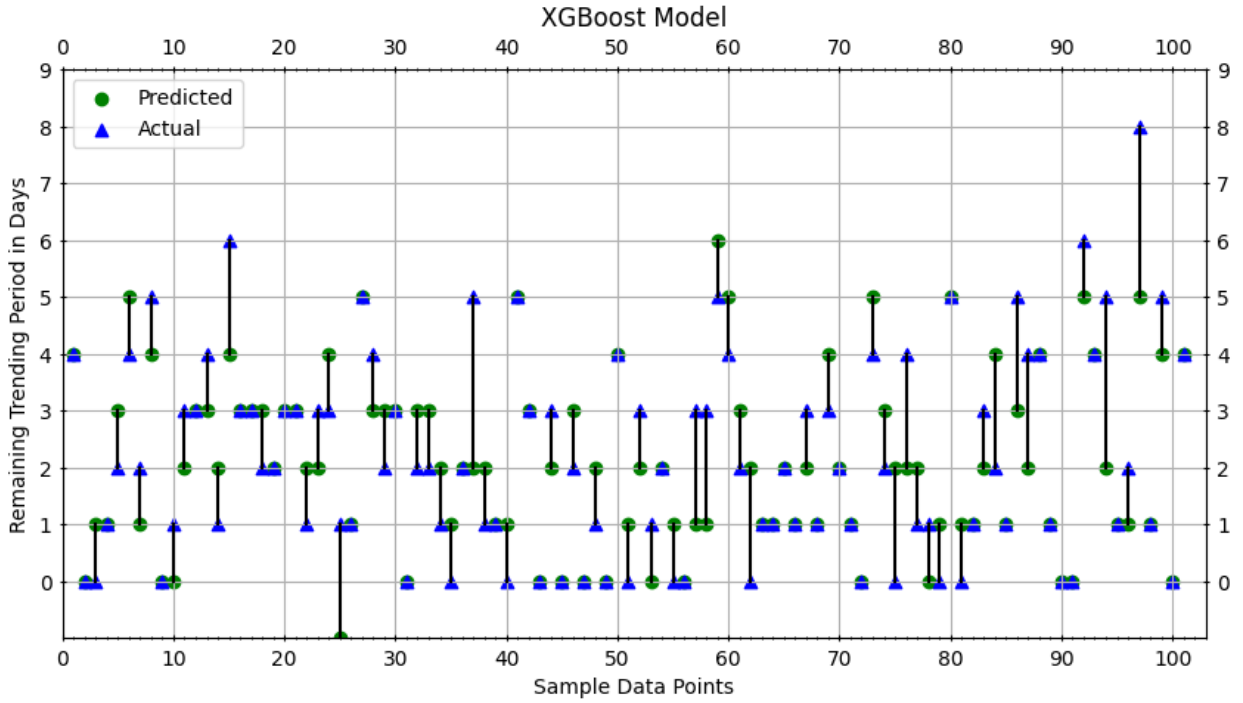

Figure 40: Trending Days Actual vs. Predicted Values using XGBoost Model.

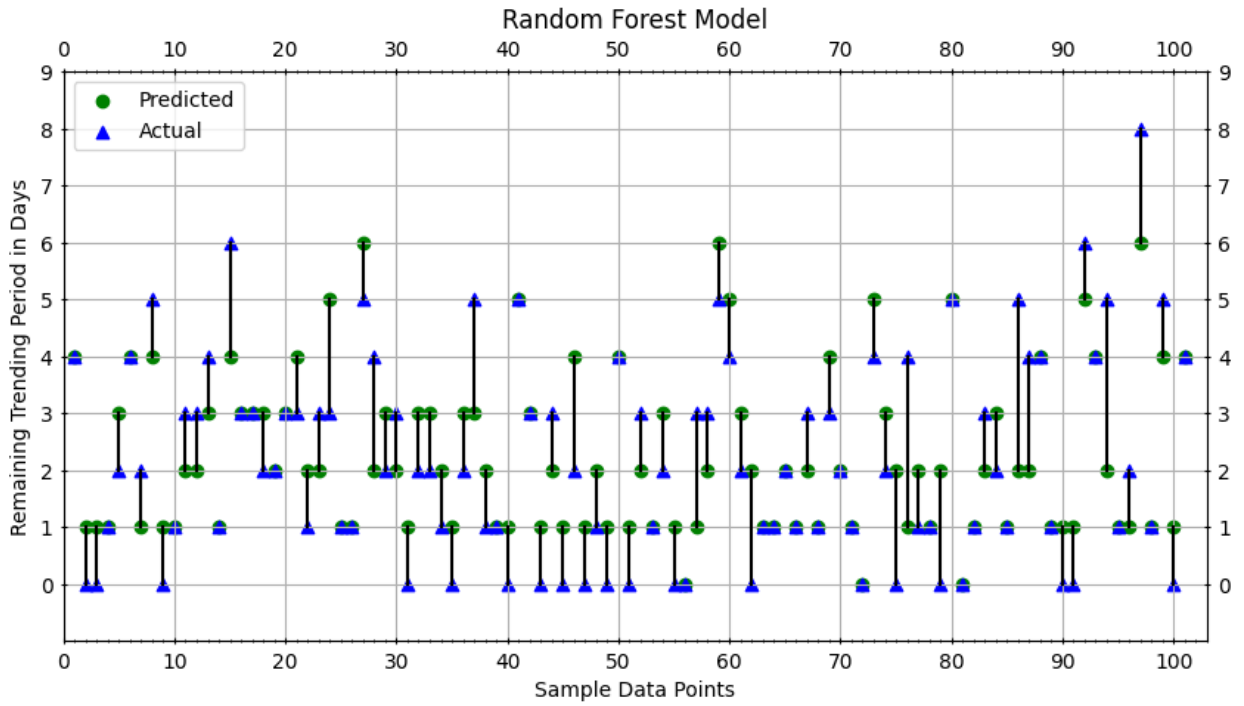

Figure 41: Trending Days Actual vs. Predicted Values using Random Forest Model.

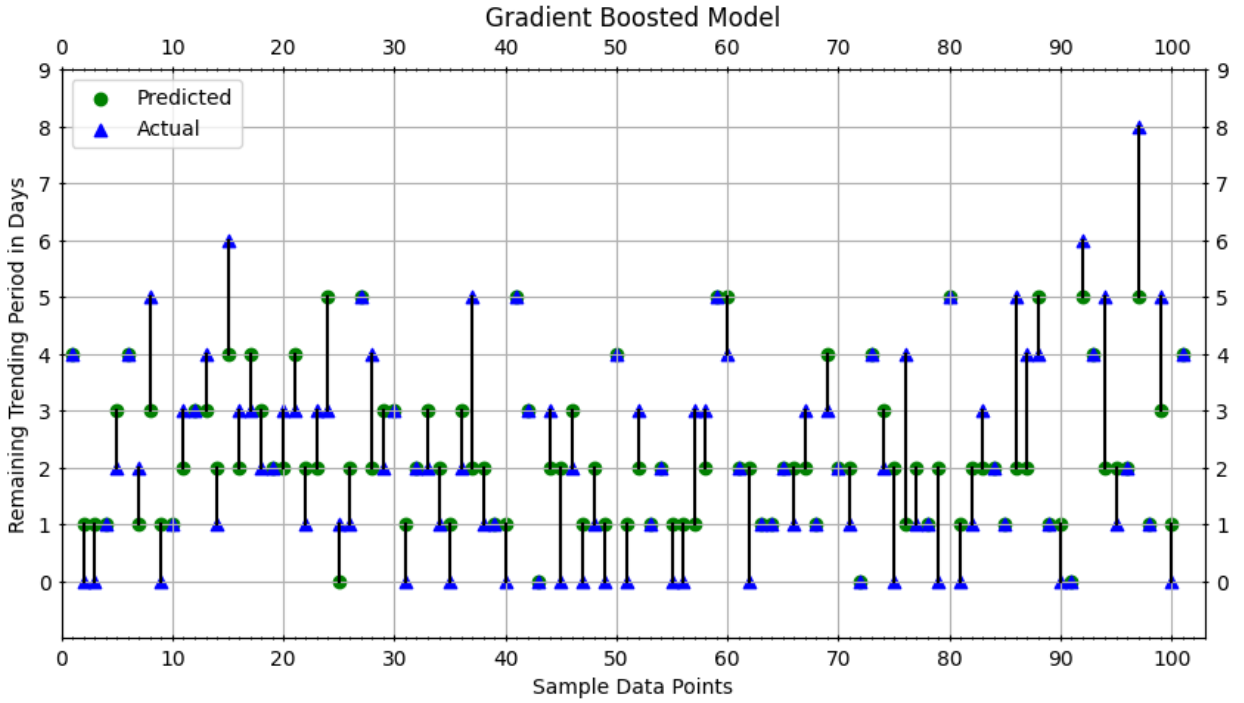

Figure 42: Trending Days Actual vs. Predicted Values using Gradient Boosted Decision Trees Model.

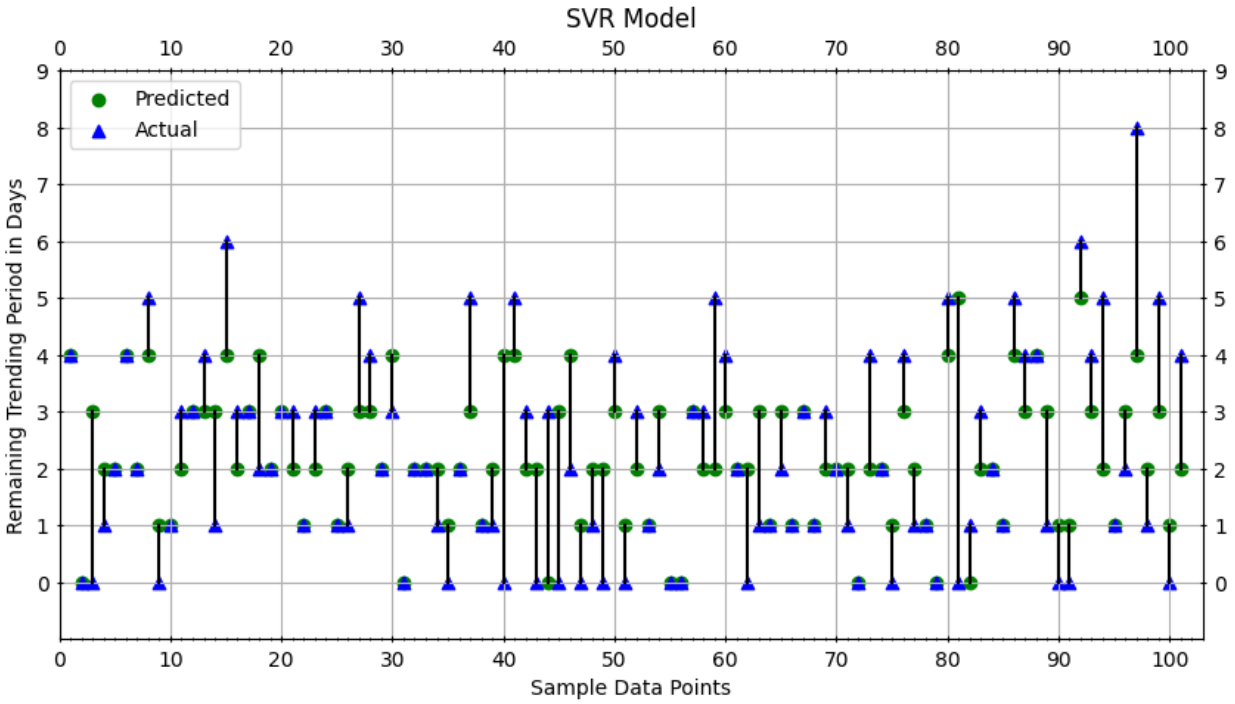

Figure 43: Trending Days Actual vs. Predicted Values using SVR Model.

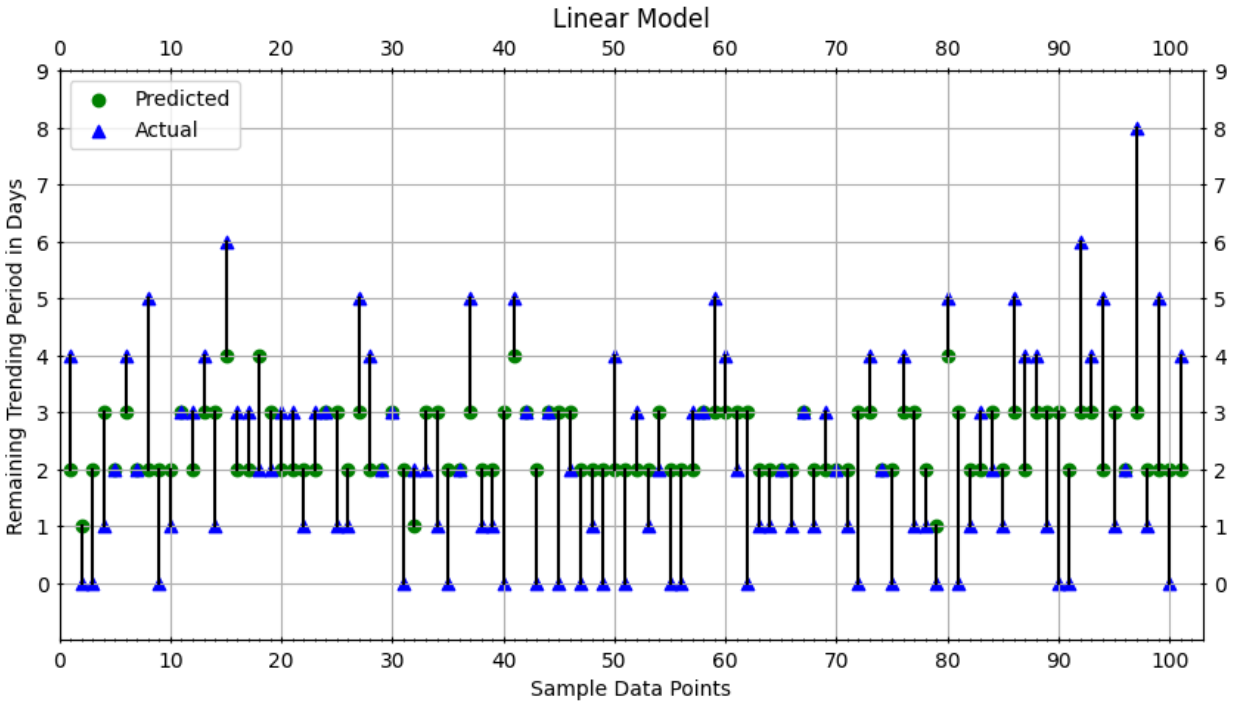

Figure 44: Trending Days Actual vs. Predicted Values using Linear Regression Model.

Chapter Five

Conclusion and Future Work

This thesis proposed an implementation of a GRU model to predict the number of days for a popular YouTube video to continuously remain trending given that the video has been trending for at least three consecutive days. The analysis revealed several interesting insights about the key factors influencing video trending days and the performance of the GRU model.

One key finding is that YouTube trending videos are highly influenced by the change in views, comments, like. These changes were expressed in the model in the form of gains and gain rates. From these gains, views were found to have the most influential effect on trending videos followed by comments and then likes, while dislike gains showed weaker influence. Moreover, thresholds. total engagement and likes ratio were also found to be influential on the video data making numeric features of highest importance for predicting popularity.

It is worth mentioning that likes ratio, is set to measure how many people like to the total number of likes and dislikes and gains express the change in views, comments, likes and dislikes based on the views, comments, likes and dislikes threshold while gain rates express the change in these features from time t to t+1.

As a result, views gain rates were found to be the most influential from gain rates, followed by comments, likes while dislikes gain rate showed weaker influence. Moreover, dislikes gain rate was found to be more influential than dislikes gains.

Moreover, thresholds represent the values that caused a specific video to enter the trend list. From these thresholds, comment threshold showed highest influence on trending data followed by views and likes while dislikes threshold had lower influence.

From date analysis of publish and trend time, videos published in evening, night, noon and late night showed higher influence on trending videos. In addition, publishing videos on Friday and Saturday seem to affect video trends more than any other weekday while for trending, all week days seem to be influential on the trending data.

Analysis results reinforced the importance of frequently used features such as title length, title keywords ratio, title subjectivity and comments thresholds. Lastly, it is worth mentioning that music and engagement categories were the most influential features from video category to affect the target variable.

As for the GRU model, the primary objective of this thesis was to design and implement a GRU model for popularity prediction and compare its' performance with other commonly used algorithms, namely XGBoost, gradient boosted decision trees, random forest, linear regression and SVR.

The findings demonstrated that the GRU model outperformed the other algorithms in terms of prediction performance metrics, as evidenced by the higher R-squared value of 0.767 and 0.755 MSE, meaning that approximately 76.7% of the variability in the remaining number of trending days can be explained by the GRU model.

The superior performance of the GRU model can be advocated for its ability to capture sequential dependencies and patterns in the input data. By utilizing the recurrent nature of GRU cells, the model can effectively process and analyze the temporal information present in the video trending data.

In addition, the performance of the comparative models show close runner up in R^2 and MSE from XGBoost with 0.728 and 0.880, random forest with 0.695 and 0.996, gradient boost decision tress with 0.665 and 1.08 and SVR with 0.325 and 2.187, respectively. It is worth mentioning that the linear regression showed poor predictive power of the regression task in hand.

Nonetheless, the outcomes of this thesis have important implications for various stakeholders, including content creators, marketers, and platform administrators. Accurately predicting the remaining number of trending days can assist in making informed decisions about resource allocation, content promotion, and overall content strategy. By leveraging the GRU model, these stakeholders can optimize their efforts and increase the visibility and impact of trending videos.

It is also important to acknowledge the limitations of this thesis. Larger GPU power were needed in order to test and compare the GRU model with other algorithms such as nonlinear regression. However, due to computation power limits, the non-linear models could not be trained.

Moreover, another important limitation is that the predictive performance of the GRU model might vary across different datasets as the GRU algorithm and its effectiveness can be influenced by various factors such as data quality, sample size, and specific characteristics and distribution of the video trends. Additionally, factors such as user engagement and external events, may also affect the duration of video trends. For example, sports category is considered unimportant feature, however if new data is included the importance score may change due to the World Cup that occurred many months ago.

For future work, it is recommended to explore the potential of decreasing the minimum number of sequencings needed to be able to predict the remaining trending period as well as gathering more data instances so to be fed into the network to create enhanced predictions. Additionally, conducting a comprehensive analysis of the model's interpretability and understanding the underlying factors driving its predictions could provide valuable insights for further refinement.

In conclusion, the implementation of the GRU model has shown promising results in predicting the remaining number of trending days for popular videos. Its improved performance compared to the other algorithms displays its potential for accurate trend duration estimation. Thus, as the field of video analytics continues to evolve, the GRU model offers a valuable tool for understanding the dynamics of trending videos and ultimately benefiting various stakeholders in the process.

References

Asur, S. and Huberman, B., 2010. Predicting the future with social media. In 2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, 492-499.

Bahdanau, D., Cho, K., and Bengio, Y.,2014. Neural machine translation by jointly learning to align and translate.

Baheti, P., 2023. Train Test Validation Split: How To & Best Practices [2023]. Retrieved from: https://www.v7labs.com/blog/train-validation-test-set

Bao, P., Shen, H. W., Huang, J., & Cheng, X. Q., 2013. Popularity prediction in microblogging network: a case study on sina weibo. In Proceedings of the 22nd international conference on world wide web, 177-178.

Bengio, Y., Simard, P., and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5, 157-166.

Cai, Y. and Zheng, Z., 2022. Prediction of News Popularity Based o Deep Neural Network.

Chung, J., Gulcehre, C. and Cho, K., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches.

Cisco, 2020. Cisco Annual Internet Report. Retrieved from:

https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html

Cleantext, 2023. Retrieved from: https://pypi.org/project/cleantext/

De Sa, S., Rocha, A. and Paes, A., 2021. Predicting Popularity of Video Streaming Services with Representation Learning: A Survey and a Real- World Case Study. 21.

En_core_web_lg, 2023. Retrieved from: https://spacy.io/models/en#en_core_web_lg

Figueiredo F., 2013. On the Prediction of Popularity of Trends and Hits for User Generated Videos. *WSDM'13*.

Gao, L., Liu, Y., Zhuang, H., Wang, H., Zhou, B., and Li, A., 2021. Public opinion early warning agent model: A deep learning cascade virality prediction model based on multi-feature

fusion. Frontiers in Neurorobotics, 15, 674322.

Gao, L., Wang, H., Zhang, Z., Zhuang, H., and Zhou, B., 2022. HetInf: social influence prediction with heterogeneous graph neural network. Frontiers in Physics, 9, 787185.

Goodfellow, I., Bengio, Y., & Courville, A., 2016. Deep Learning. MIT Press. Retrieved from: http://www.deeplearningbook.org

Google Support., (n.d.). Trending on YouTube. Retrieved from: https://support.google.com/youtube/answer/7239739?hl=en#:~:text=Trending%20helps%20viewers%20see%20what's,surprising%2C%20like%20a%20viral%20video.

Graves, A., 2013. Generating sequences with recurrent neural networks.

Haimovich D., Karamshuk D., Leeper T., Riabenko E., and Vojnovic M., 2022. Popularity Prediction for Social Media over Arbitrary Time Horizons. PVLDB, 15, 841-849.

Hastie, T., Tibshirani, R., and Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hayes, A., 2023, YouTube Stats: Everything You need to Know. Retrieved from: https://www.wyzowl.com/youtube-stats/#:~:text=You%20need%20to%20have%20at,monetizing%20your%20channels%20through%20ads.

Ho, T. K., 1995. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278–282).

Hochreiter, S., and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9, 1735-1780.

Iqbal M., 2022. YouTube Revenue and Usage Statistics (2022). *Business for Apps.* Retrieved from: https://www.businessofapps.com/data/youtube-statistics/

Itertools, 2023. Retrieved from: https://docs.python.org/3/library/itertools.html

Joblib, 2023. Retrieved from: https://pypi.org/project/joblib/

Lin, S., Kong, X., and Yu, P., 2013. Predicting trends in social networks via dynamic activeness model. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 1661-1666.

Ma, Z., Sun, A., and Cong, G., 2013. On predicting the popularity of newly emerging hashtags in

twitter. Journal of the American Society for Information Science and Technology, 64, 1399-1410.

Massimiliano V., Brunelli L.. and Susto G. A., 2021. Instagram Images & Videos Popularity Prediction: A Deep Learning-Based Approach. *Statwolf*.

Nisa, M., et al. 2021.Optimizing Prediction of YouTube Video Popularity Using XGBoost. *Electronics*,10, 2962.

NLTK, 2023 Retrieved from: https://www.nltk.org/

Orishko, 2020. YouTube Engagement Prediction. Retrieved from: https://github.com/orishko-py/YouTube-Engagement-Prediction/blob/master/exploratory-data-analysis.ipynb

Pytorch, 2023. Retrieved from: https://pytorch.org/

Rathord p., Jain A. and Agrawal C., 2019. A Comprehensive Review on Online News Popularity Prediction using Machine Learning Approach. 5.

Re, 2023. Retrieved from: https://docs.python.org/3/library/re.html

Saeed R., Abbas H., Asif S., Rubab S., Khan M., Iltaf N. and Mussiraliyeva S., 2022. A Framework to Predict Early News Popularity using Deep Temporal Propagation Patterns. 195.

Sarker, I.H, 2021. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021).

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks, 61, 85 117.

Sehl K. and Tien S., 2023. Engagement Rate Calculator + Guide for 2023. Retrieved from:https://blog.hootsuite.com/calculate-engagement-rate/

Sermush, 2023. Most Visited Websites by Traffic in the world for all categories, January 2023. Retrieved from: https://www.semrush.com/website/top/

Shang, Y., Zhou, B., Wang, Y., Li, A., Chen, K., Song, Y., and Lin, C., 2021. Popularity prediction of online contents via cascade graph and temporal information. Axioms, 10, 159.

Shang Y., Zhou B., Zeng X., Wang Y. Yu H. and Zhang Z., 2022. Predicting the Popularity of

Online Content by Modeling the Social Influence and Homophily Features. Frontiers in Physics, 10.

Shapiro, S.S. and Wilk, M.B., 1965. An Analysis of Variance Test for Normality (Complete Samples). Biometrika, 52, 591-611.

Sharma R., 2022. YouTube Trending Video Dataset (updated daily). Retrieved on 26/09/2022 from: https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset?resource=download

Shen, H., Wang, D., Song, C., & Barabási, A. L. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In Proceedings of the AAAI Conference on Artificial Intelligence, 28.

Sibo Q., Shanchen P., Min W., Xue Z. and Feng D., 2021. Online Video Popularity Regression Prediction Model with MultiChannel Dynamic Scheduling Based on User Behavior. *Chinese Journal of Electronics*, 30(5).

Pedregosa, et al., 2011. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830. Retrieved from: https://scikit-learn.org/stable/

Spacy, 2023. Retrieved from: https://spacy.io/

Stats model, 2023. Retrieved from: https://www.statsmodels.org/stable/index.html

String, 2023. Retrieved from: https://docs.python.org/3/library/string.html

Tang L., Huang Q., Puntambekar A., Vigfusson Y., Lloyd W. and Li K., 2017. Popularity Prediction for Facebook Videos for Higher Quality Streaming. Princeton University.

Trzciński T., Andruzskiewicz P., Bocheński T. and Rokita P., 2017. Predicting Popularity of Online Videos Using Support Vector Regression. in IEEE Transactions on Multimedia, 19, 11, 2561-2570.

Tsur, O., and Rappoport, A., 2012. What's in a hashtag? Content based prediction of the spread of ideas in microblogging communities. In Proceedings of the fifth ACM international conference on Web search and data mining, 643-652.

Unidecode, 2023. Retrieved from: https://pypi.org/project/Unidecode/

Weng, L., Menczer, F., and Ahn, Y., 2014. Predicting successful memes using network and community structure. In Proceedings of the international AAAI conference on web and social media, 8, 535-544. Retrieved from:

XGBoost, 2023. Retrieved from: https://xgboost.readthedocs.io/en/stable/

Xin Y., Kong L., Liu Z., Chen Y., Li Y., Zhu H., Gao M., Hou H. and Wang C., 2018. Machine learning and deep learning methods for cybersecurity. IEEE Access, 6, 35365–81.

Zhao, Q., Erdogdu, M., He, H., Rajaraman, A., and Leskovec, J., 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 1513-1522.

Appendix