**IE 7275 Final Project: Youtube Video Likes Prediction**

**Milestone: Project Proposal**

Group x

Hans Huray

Joshua Ji

(206) 307-5017

(206) 291-0050

[huray.h@northeastern.edu](mailto:huray.h@northeastern.edu)

[ji.yix@northeastern.edu](mailto:ji.yix@northeastern.edu)

**Percentage of Effort Contributed By Student 1: 50%**

**Percentage of Effort Contributed By Student 1: 50%**

**Signature of Student 1: Hans Huray**

**Signature of Student 2: Yixuan Ji**

**Submission Date: 02/13/2022**

**Problem Settings:**

Covid 19 has impacted the world for more than two years and affected different areas of businesses. This resulted in huge decline in stock prices and downsizing in companies due to the social measures taken to slow the spread of the virus and inevitably affect the business's income. In hindsight, covid has negatively affected the firms, but this opened up new digital opportunities. Firms that focus on these aspects such as Zoom, Youtube, and Twitch had huge increases in their users. Although vaccinations have been administered to the population and work policy has been going back to normal, while working from home still remains as a feasible option to work indefinitely and digital platforms as an integrated part to daily lives, we believe that youtube has limitless potential to grow.

The goal of this case study is to predict the views to youtube videos through the information given such as the title, description, and other data. The analysis will focus on understanding the potential of youtube creators to the online community.

**Problem Definition:**

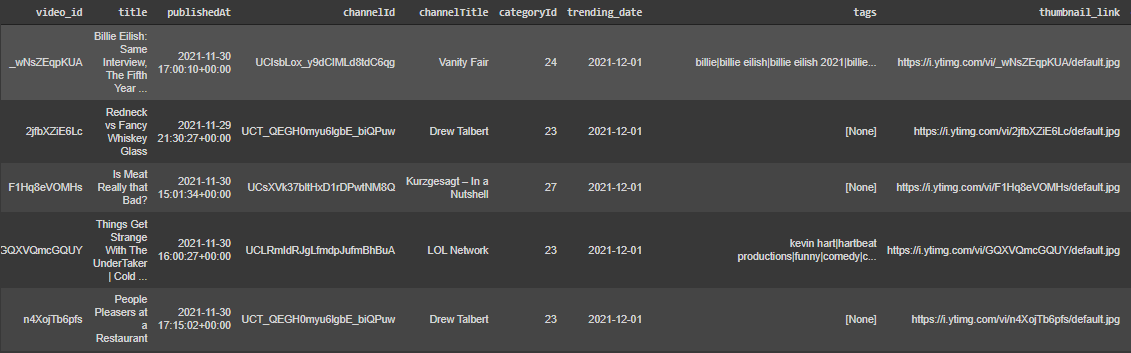
* How to transform “.parquet” data files and perform data manipulations on it.
* Understand the different features in the training dataset and test dataset, and how are they built (dates, number of rows, different in features between sets).
* Feature engineering: how to “mine out” more information from the features that are provided in the training set.
* Is it worth it to add more features to the dataset, ie. add more features based off of the thumbnails (image processing)?
* Have a quick mental run down, intuitively what would be some features that would affect the predictions? (celebrity name drops, channel followers, etc)
* Balancing between feature processing and model building, should the emphasis be more on training the model or transforming features and how to find a balance point between these two tasks. Trying to be less time-consuming and resource-consuming.
* Tuning hyperparameters: Is it worth it to spend so much time trying to optimize this task?
* Something we learned online, how to create target-based features that would not overfit the model but skillfully improve the final score.

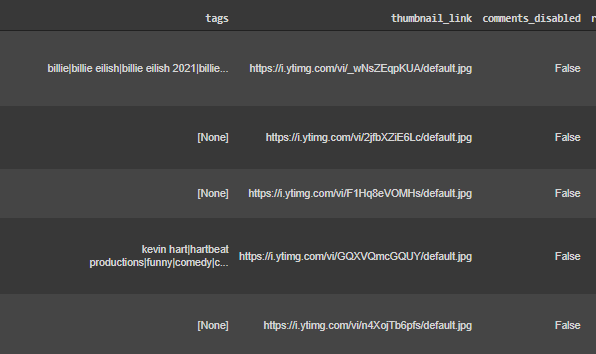
**Data Sources:**

Pog Champs, 2022, *Predict\_Youtube\_Video\_Likes* (v.1), Pog Champs, https://www.kaggle.com/c/kaggle-pog-series-s01e01/data

**Data Description:**

The dataset is divided into two parts, the training and test dataset. The test dataset comprises 5800 rows and 16 columns while the train dataset comprises 92275 rows and 20 columns. There are 15 variables for the predictors (15 columns). The other 5 columns are the response which is calculated from the other 4 columns (likes, dislikes, views, and comment count). The variables are mainly categorical variables with the duration being the only quantitative variable.

First 5 rows of the Test Dataset:





**Sample Dataset Dictionary:**

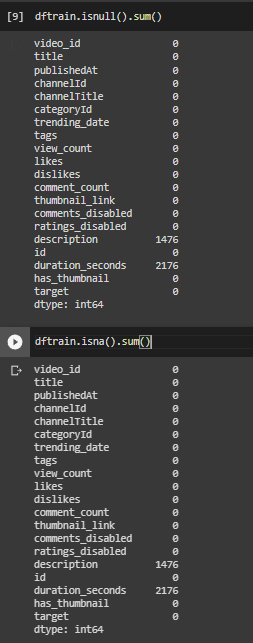
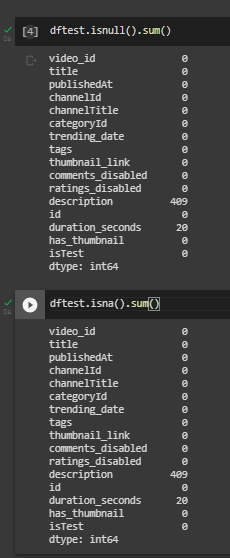
| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| title | string | Title of the video |
| publishedAt | datetime | Date that the video is published |
| channelTitle | string | Title of the channel (publisher) |
| categoryId | integer | Genre ID |
| comment\_count | integer | Number of comments in the video |
| ratings\_disabled | boolean | If true, enables the youtube algorithm to rate the video (for example, mature content or drug reference content) |
| duration\_seconds | integer | Duration of the video |
| target | float | Response of the analysis which is count of likes divided by view count |

**Preprocessing Data Summary:**

* Dropped features from train data set that are not in the test data set “dislikes”, “comment count”
* Removed rows of ‘Ratings\_disabled’ = True (Ratings\_disabled=True will cause the target variable to be 0)
* Created a new column for age of the video, this describes how long the video has been posted age=‘trending\_date’ - ‘published\_at’
* Transformed data types: - ‘trending\_time’ to type datetime
* Changed ‘description’ from (string) to length of description (int)
* Transformed ‘tags’ (string) into number of tags
* Created new feature to classify whether a video is a ‘Youtube\_shorts’

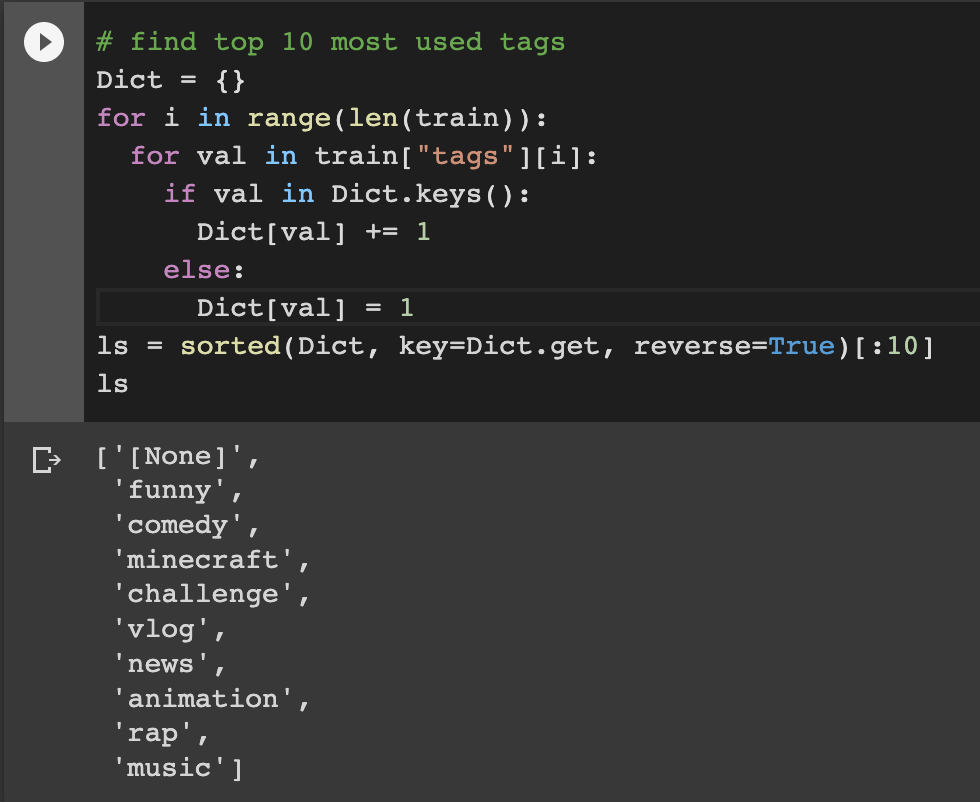
**Exploratory Data Analysis & Data Visualization:**

**NaN Values in columns:**

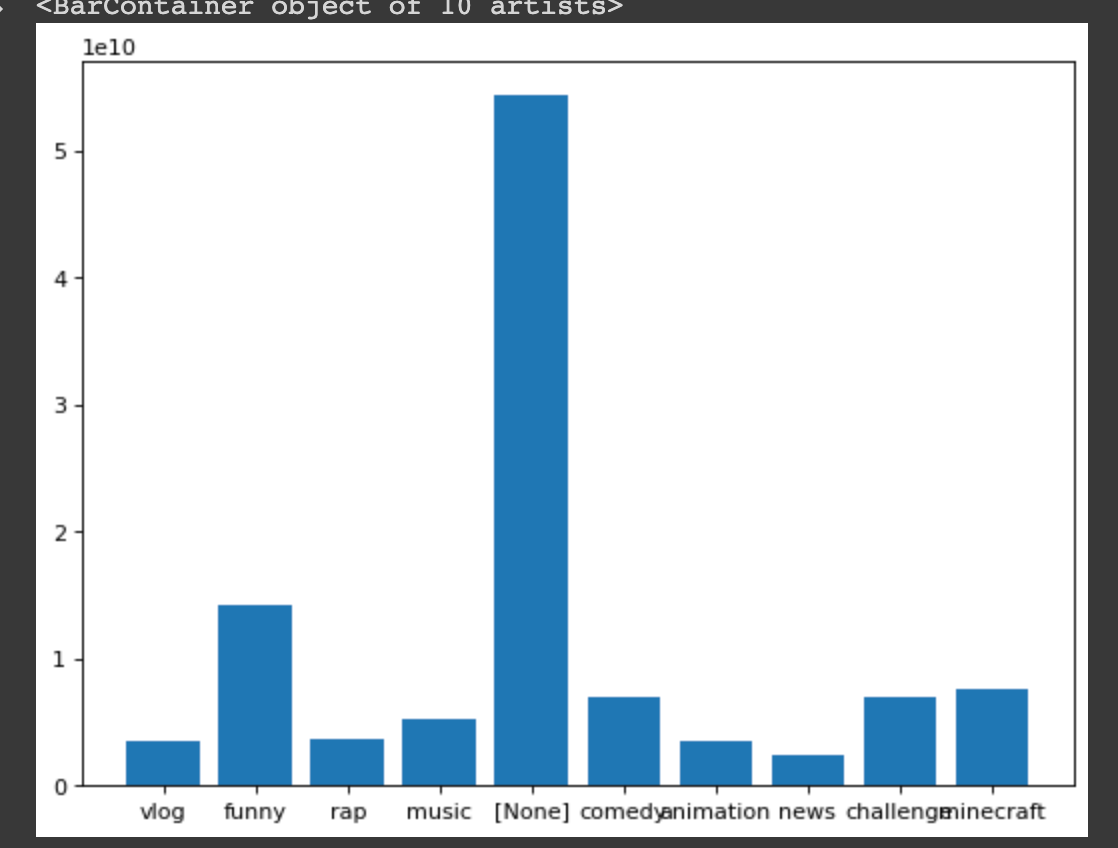


There are some NaN values in seconds and description. NaN description refers to blank description since description is optional and NaN duration\_seconds refers to posts instead of videos.

**Find top 10 most used tags:**

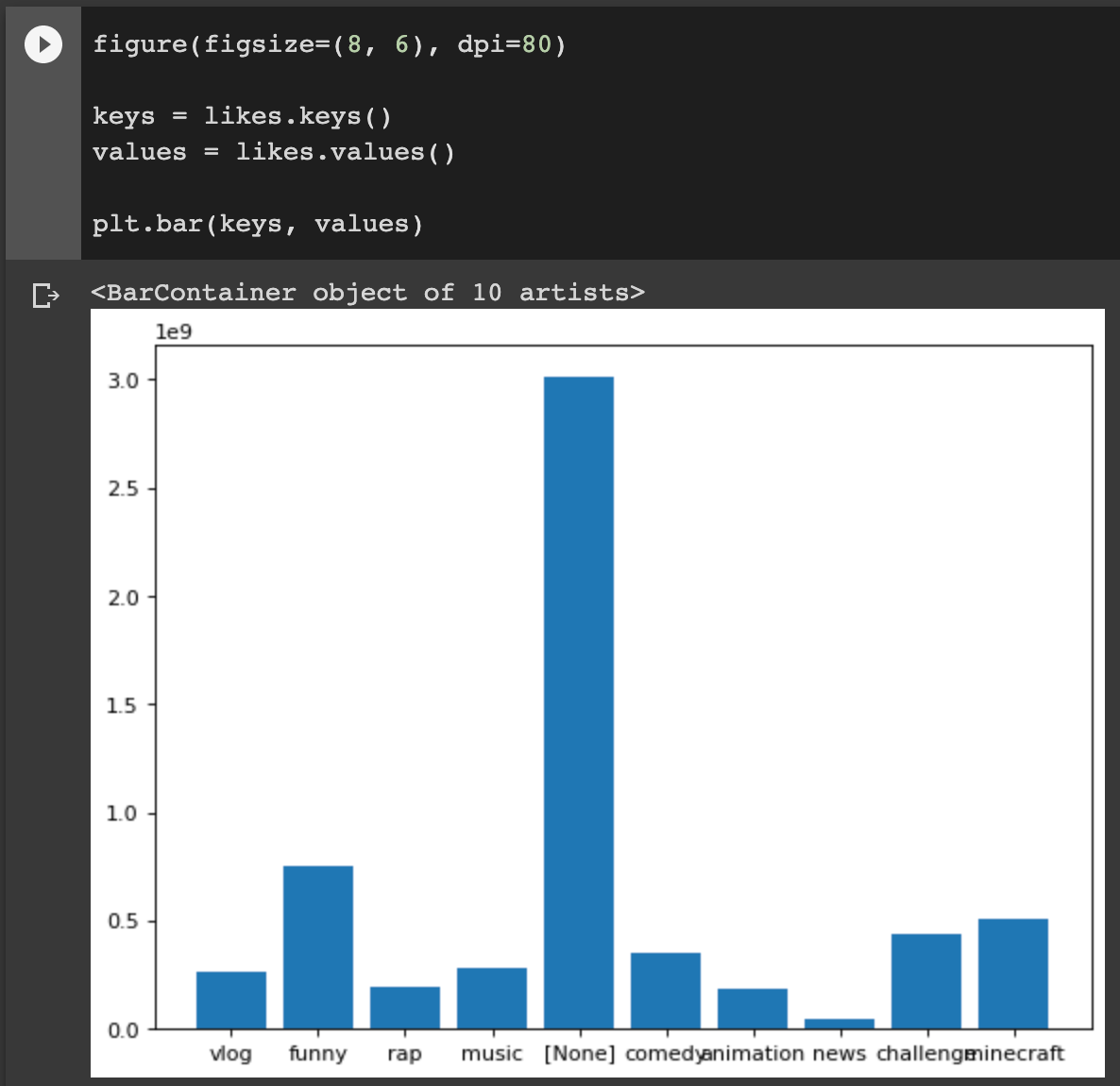


**Top 10 most used tags view count:**

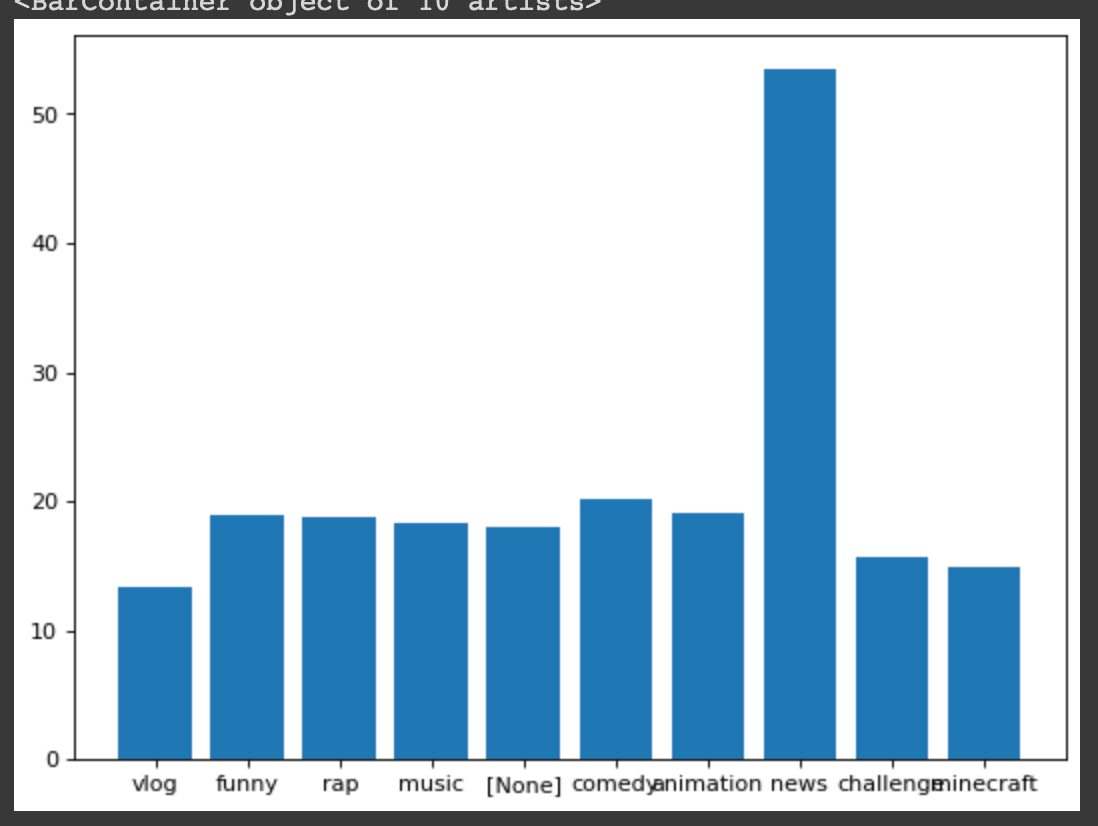


In the graph, it can be seen that tags do not necessarily determine the view count, “None” tag proves to have the most view compared to other tags

**Top 10 most used tags like count:**

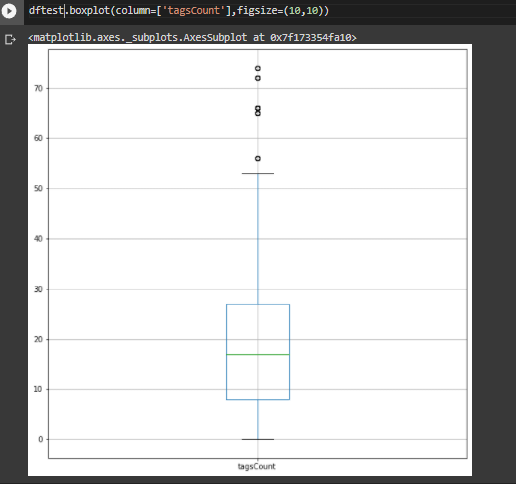
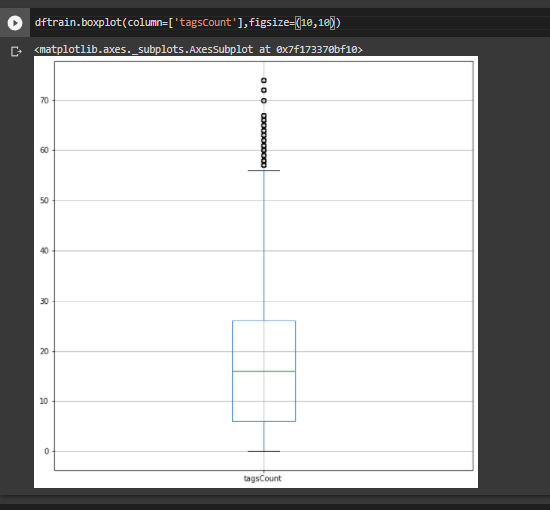


**Top 10 most used tags view/like ratio:**



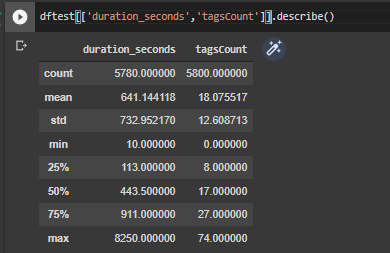
“None” tag has the most like count. This might be caused by the number of videos if [None] tag, but news tag has the most like to view ratio.

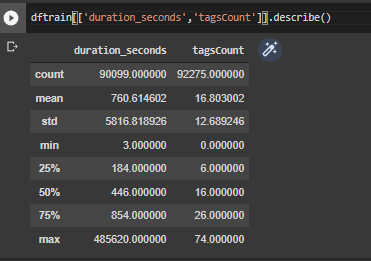
**Tags Count Boxplot:**



From the boxplot, it can be seen that the median of the tags count is roughly the same for both train and test dataset. However the train dataset has a lot more higher outliers compared to the test dataset, transforming the tags count of the train dataset might be necessary.

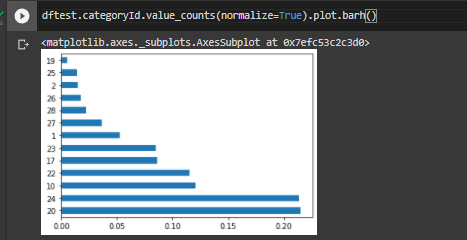
**Numerical data description:**

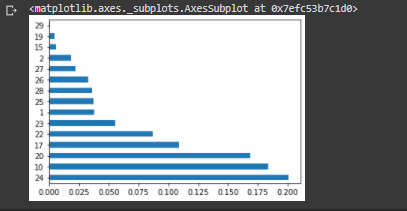




From the data description, the train data is more spread out compared to the test data as it has significantly higher standard deviation.

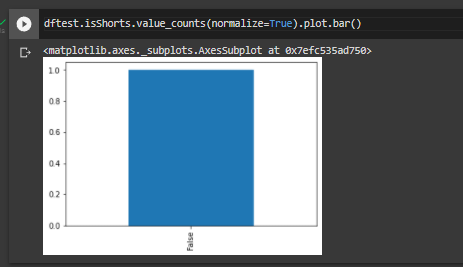
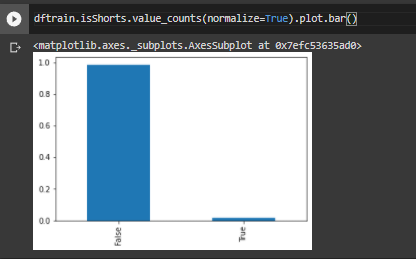
**CategoryID percentage:**





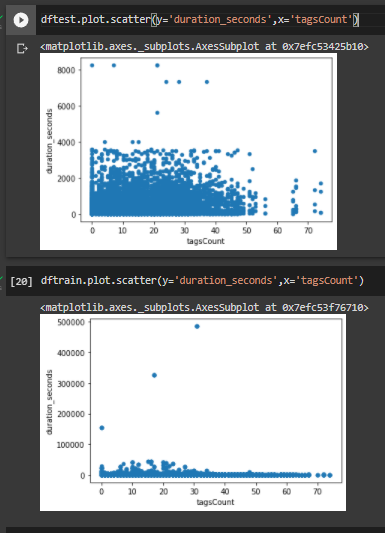
In the test dataset, it can be seen that categoryID 20 and 24 are the most stated in the videos while in the train data set, category 24 is the most used in the videos

**IsShort Barchart:**



From the barchart, some of the train data has youtube short videos (portrait video that has less than 60 seconds duration) but in the test data all the data is either normal videos or posts.

**Scatter plot between duration and tagsCount**

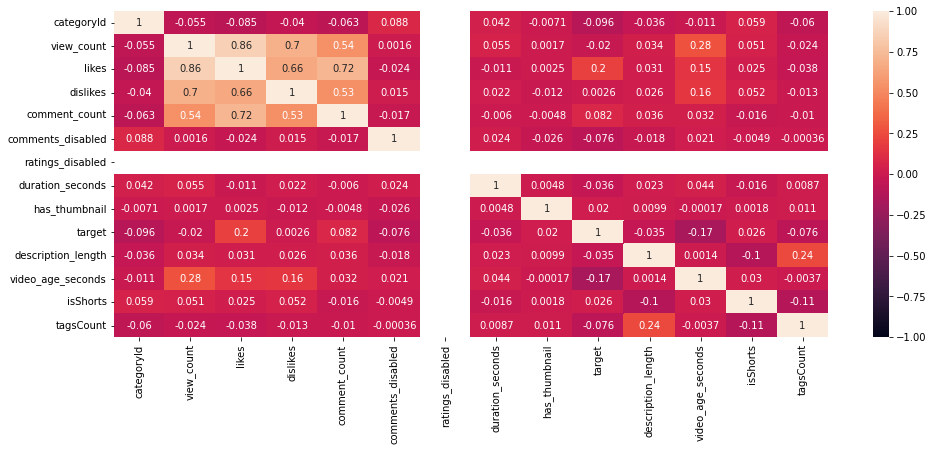


From the chart, it can be seen that the tagsCount does not correlate with the video as it does not represent any relationship between those two variables.

### **Feature Selections**

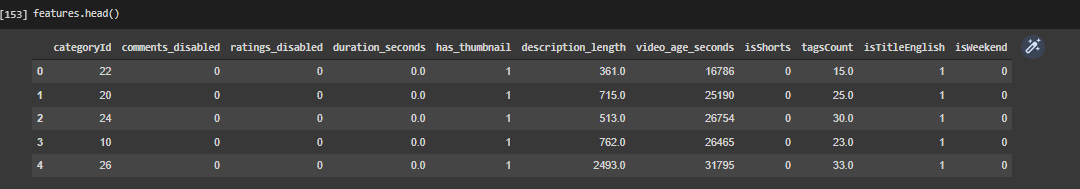
* Improve the accuracy with which the model is able to predict new data.
* Reduce computational cost.
* Produce a more interpretable model.

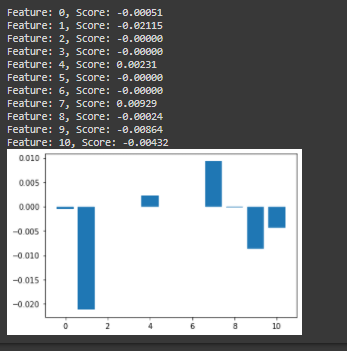
Correlation heatmap



From the correlation heatmap, lighter color indicates higher correlation between variables. Choosing one variable from the other with high correlation should be considered since the other variable can be considered redundant. The heatmap shows that the train variables in general are not correlated with each other. We can see that the tagsCount and description length might have a minor correlation but other variables are not correlated.

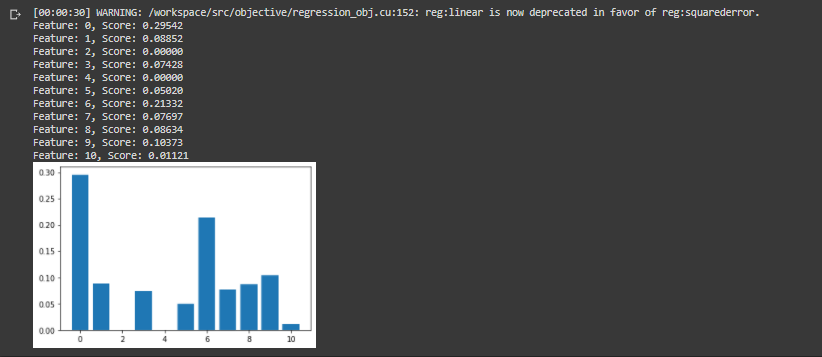
**Linear regression feature selection**





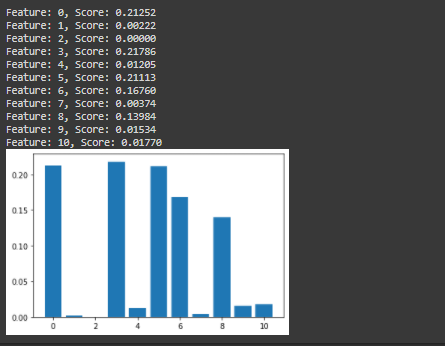
Based on the linear feature selection, best feature are comments disabled followed by, video\_age\_seconds, isTitleEnglish, isWeekend, duration\_seconds, categoryId, and tagsCount

**Xgboost feature selection**



Based on xgboost, best features are categoryID, followed by video\_age\_seconds, isTitleEnglish, comments\_disabled, tagsCount, isShorts, duration\_seconds, description length, isWeekend

**Random forest feature selection**



Based on random forest, best features are duration\_seconds, followed by categoryId, description length, video\_age\_seconds, tagsCount

**Model Selection**

For model selection, we are leaning towards forest tree regression as the variables that we have are mainly categorical. The results that we get are purely from the data without splitting, to avoid overfitting, we will split the data and refit the model



### **Implementation of the Selected Model**

For the implementation, we split the data into 75% training data and 25% validation data. Then, we applied the Forest Tree regression as most of the data are categorical data.

Forest Tree Regression score on train data



Regression score on validation data set

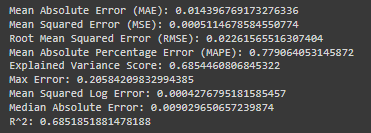


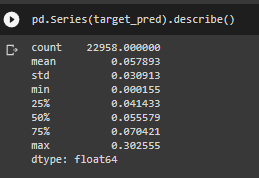
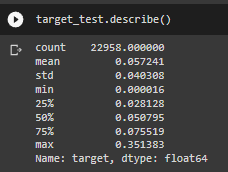
The model outputs rsquared of 0.698 on the validation dataset which is good but needs more improvement



Some improvements that can be done is by putting more parameters in the forest tree regression to create a better model

### **Performance Evaluation and Interpretation**





From the performance evaluation, it can be concluded that the average residuals is 0.0144, the variance of the residuals is 0.0005, the standard deviation of the residuals is 0.0226. The mean absolute percentage error is considered high (77%). This might be caused by the small values of the numbers which resulted in high percentage if there is an inconsistency to small values. The explained variance score is considered good as it explains more than 60% of the variance and therefore this model has significance in forecasting. The maximum error is 0.2 which is high compared to the data statistics description (low numbers with max of around 0.3). The mean squared log error is considered good (0.04%). Median absolute error is 0.009, this can be improved as this is 18% value of the median from the target data. R^2 can be improved as we believe that R^2 should be at least 0.7 to be mathematically significant.