**Trending YouTube Video Statistics**

**Final Project Report**


Chia-Han Chiang
Khushboo Harjani

857-930-5593 (Tel of Chia-Han C.)

848-252-1951 (Tel of Khushboo H.)

chiang.ch@husky.neu.edu

harjani.k@husky.neu.edu


**Signature of Member 1:** ChiaHanChiang

**Signature of Member 2:** KHarjani

**Submission Date: December 2nd , 2019**

## I. Problem Setting

The YouTube platform has two kinds of users - the ones that post videos and others that view them. Videos are always posted under a parent channel. YouTube channels, aim to acquire a large enough fan base so that they can make their own channel a profit-making marketing business. In order to determine if a YouTube channel should be paid for marketing a product there are metrics that can support when a video is said to have successfully acquired an audience.

## II. Problem definition

The goal of this project is to explore data mining techniques to create a reliable predictive model that can **foresee YouTube video trendiness** given that a video has already shown up on the platform's top trending list, once. Trendiness across all videos is measured by the frequency in which a video is displayed on the trending list. The videos that are captured on the "trending list" do not necessarily consist of the most **viewed, liked or commented** on videos. For a video to be selected on the top trending list, a combination of factors are taken into consideration.

## III. Data Sources

The dataset is obtained from the [www.kaggle.com](http://www.kaggle.com) website, where information is continuously updated. Everyday a list of 200 trending videos are added to the dataset. Each record on the dataset represents an instance when a video is displayed on the trending list.

## IV. Data Description

The attributes that are in the original dataset are as follows:

| video_id | trending_date | title | channel_title | category_id | publish_time | tags | views | likes | dislikes | comment_count | thumbnail_link | comments_disabled | ratings_disabled | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

The chosen portion of the dataset is a collection of the top 200 trending videos daily in the span of 6 months, arranged by ascending trending date. The data for training, validation, and testing only cover a time frame of 2 months (60 days) each. In order for the measure of degree of trendiness to be consistent in each of the training, validation and test data [Min:1, Max: 60], we partition the data into three equal parts. Since there are different number of days in a month (28, 30 and 31), the average number of days in a month is approximately 30. 36,000 records encompasses almost 6 months. The data partition is shown below:

**Table 1 - Data Partition into Training, Validation and Testing Datasets**

| Training Data (60 days/bins) | Validation Data (60 days/bins) | Testing Data (60 days/bins) |
|---|---|---|
| Nov 14, 2017 – Jan 14 , 2018 | Jan 15, 2018 – Mar 16, 2018 | Mar 16, 2018-May 16 , 2018 |
| Records 1-12,000 | Records 12,001-24,000 | Records 24,001 – 36,000 |

After partitioning the data into training, validation and test sets, the records are aggregated by video_id (unique video identifier) and the attributes that are associated with each video are:

| video_id | largest_views | largest_likes | largest_dislikes | largest_commentcount | comments_disabled | ratings_disabled | frequency |
|---|---|---|---|---|---|---|---|

1. Largest_views - Most up-to-date number of views
2. Largest_likes - Most up-to-date number of likes
3. Largest_dislikes - Most up-to-date number of dislikes
4. Comment Count - Most up-to-date number of comments
5. Comments_enabled - Comments are enabled in video.
6. Ratings_enabled - Likes and dislikes are enabled in video.
7. Frequency (Response)- Total number of times a video identifier is on the trending list

This new dataset along with its attributes is the basis on which a predictive model is built to predict video "trendiness".

## V. Data Exploration

There are 4 numerical attributes: **views**, **likes**, **dislikes** , and **comment_count**.

Numerical Variables:

```
> summary(df $views)
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
     549    242329    681861   2360785   1823157 225211923
> summary(df $likes)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0    5424   18091   74267   55417 5613827
> summary(df$dislikes)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0     202     631    3711    1938 1674420
> summary(df$comment_count)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0     614    1856    8447    5755 1361580
```

**Figure 1 - Summary of numerical attributes**

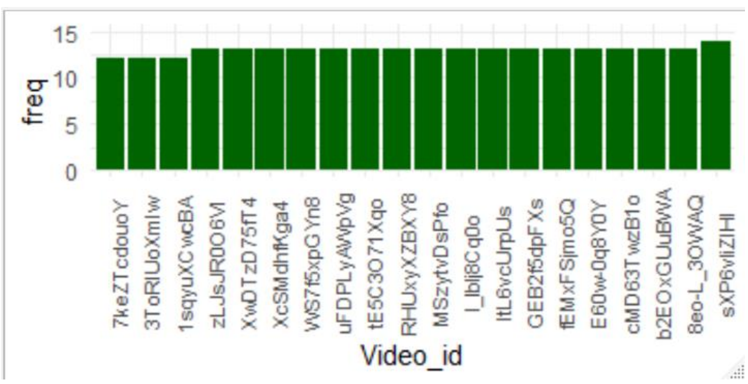


**Figure 2- Number of times a video appears on trend list (top 20 video_id's)**

The top twenty most frequent trendy videos have a similar frequency between 12-14 times (from a total of 60 possible times of being displayed on the trending list over the course of 60 days). Frequency in the remaining videos decrease gradually.

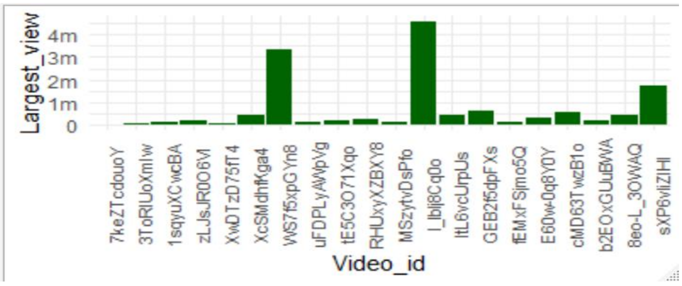

**Figure 3 - Top twenty trendy videos, by number of views**

The above figure shows that although the top trendy videos have the same degree of trendiness, the number of views differ greatly. Also as shown in the scatterplot matrix in the next section, the number of views is a relatively weak predictor of trendiness.

The next step to create a more reliable predictor is to transform the number of views numerical variable into a categorical variable. The frequency is also defined in a new categorical response variable made up of two classes (>5 and <5 frequency of trendiness)

Video id, sXP6vliZIHI (on Figure 3), has the highest frequency of trendiness, however the number of views are not the highest. This video has a relatively high number of likes, dislikes and comment count. It also belongs to category 10 which is a popular category. The video(sXP6vliZIHI) belongs to the channel named “Cardi B”. It is the only trendy video on this channel and the number of views are relatively high (17,540,613).

A portion of YouTube posts have **disabled comments and/or ratings.**

**Table 2 - Video types based on combinations of enabled and disabled ratings/ comments.**

| | Videos Categorized by Enabled Ratings and Comments | Ratings | Comments |
|---|---|---|---|
| 1(RC) | videotype_RC | Enabled | Enabled |
| 2 (R) | videotype_R | Enabled | Disabled |
| 3(C) | videotype_C | Disabled | Enabled |
| 4 (NA) | videotype_NA | Disabled | Disabled |

In order to check if there is a significant difference in the average number of views for each category on Table 2 above, a bar chart is used which displays the average frequency from each category (RC, R, C, NA).

There is an under-representation of classes C (Comments enabled only) and NA (none enabled) therefore the samples within these classes are small and can be misleading. However the average frequency in R (Ratings enabled only) is higher than RC (Ratings and Comments enabled).

Videos with only ratings have the highest average number of times on the trend list. The chart below indicates a reasonable difference and therefore the enabling of ratings alone may have a significant impact on the frequency on the trendlist.

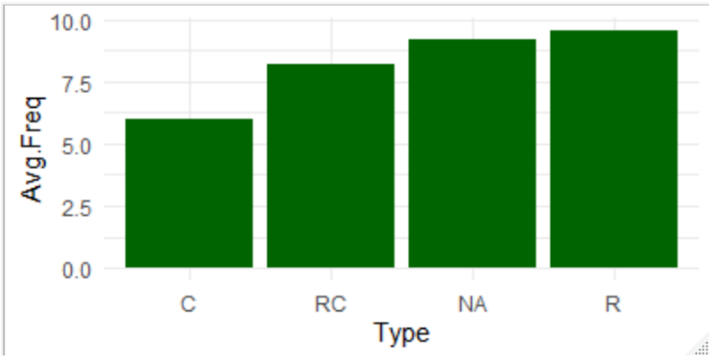


**Figure 4 - Average frequency by variations of enabled and disabled ratings and comments**

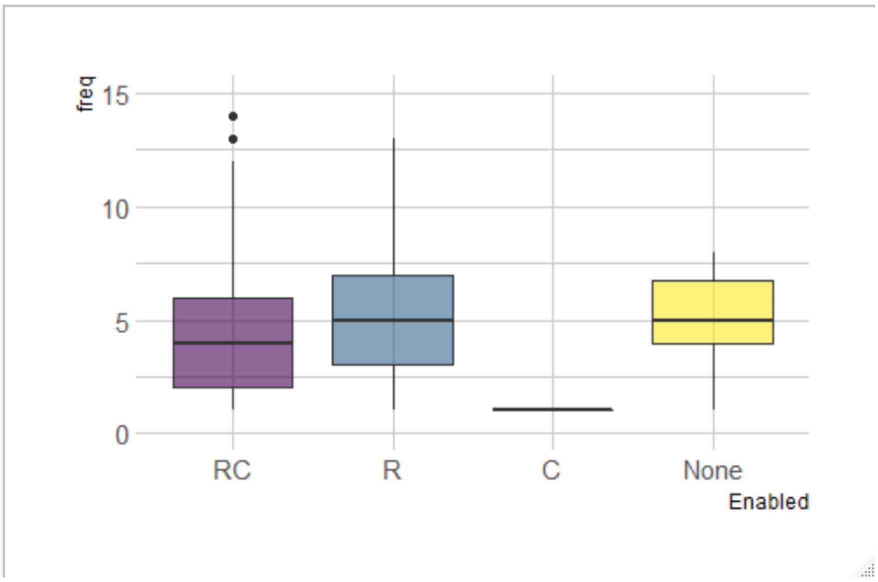


**Figure 5 - Box Plot of videos types RC, R, C, None**

After narrowing down the videos (including only enabled ratings and enabled comments) *(videotype_RC)*. A scatterplot matrix is created as shown below. The scatterplot matrix includes only the data points in the training data that have enabled rating and comments. Variables observed for correlation include likes, dislikes, comment count, and views. The attribute likes has a strong correlation of 0.866 with views. Dislikes and Comments have weaker correlations of,  0.599 and 0.659 respectively.

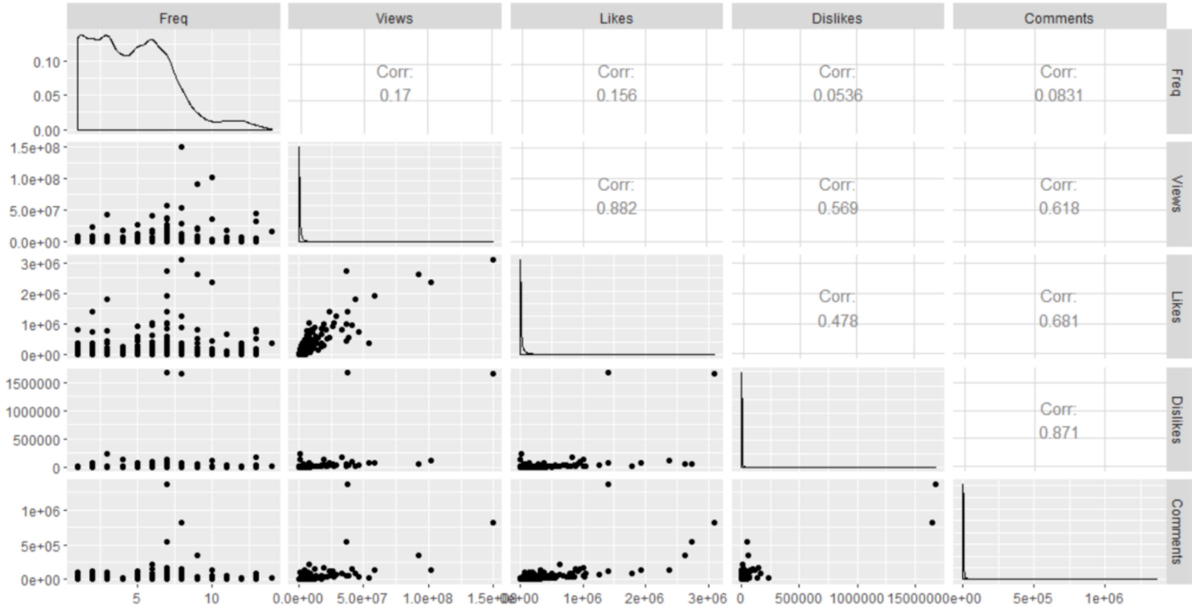


**Figure 6 - Scatterplot Matrix of all training data of frequency, views, likes, dislikes and comments (each point represents a video_id)**

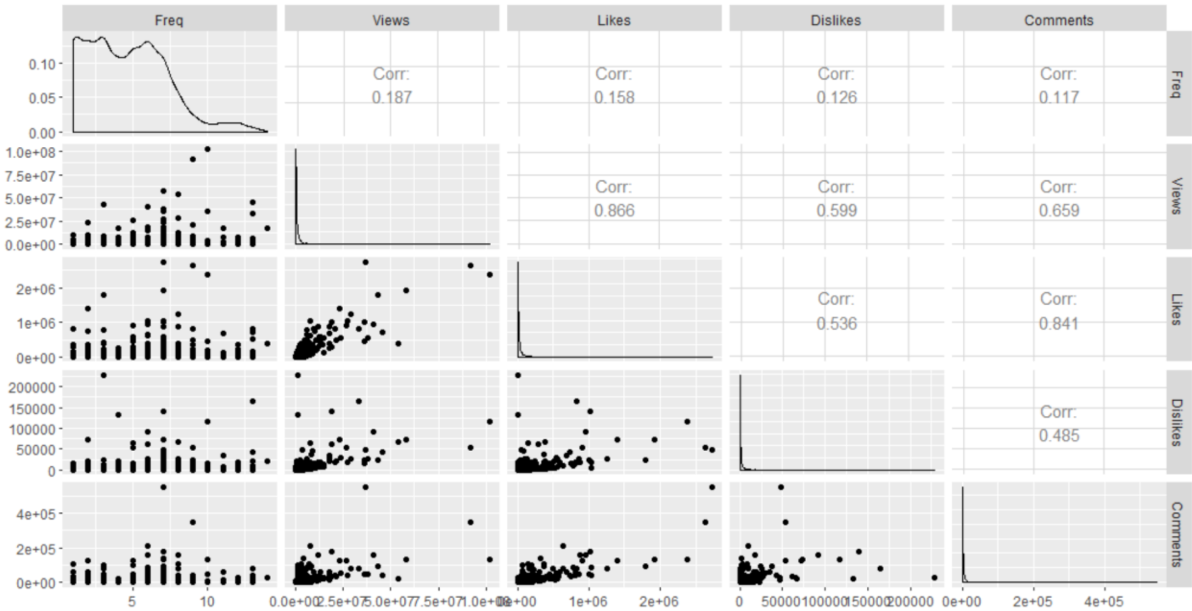


**Figure 7 - Updated Scatterplot Matrix after outliers are removed**

Significant outliers were removed from the data, and the correlation levels increased by a noticeable amount.

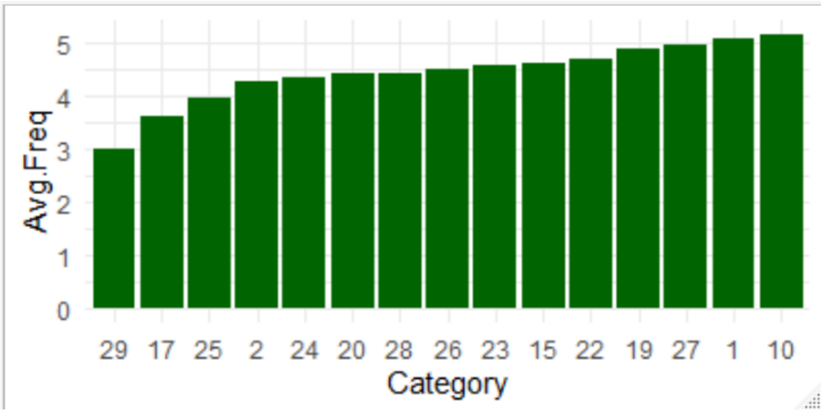


**Figure 8 - Average number of frequency by category**

The degree of frequency of videos is somewhat evenly distributed across categories (Ranges from 3 to 5).  As shown above. The category 29 has the least frequent trendiness.

**Most Popular Channels**

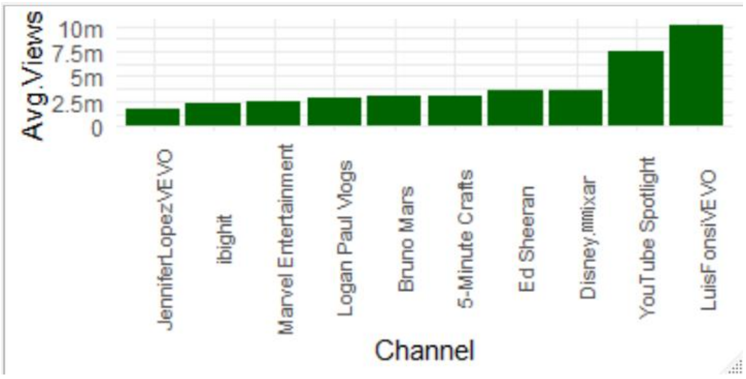


**Figure 9 - Highest average number of views by channel**

Popularity amongst channels in terms of average views, is highest in the top two, and relatively flat in the following videos. As shown in Figure 10, the average frequency of a video on the trend list is the same, whether the videos were published on a weekday or weekend.

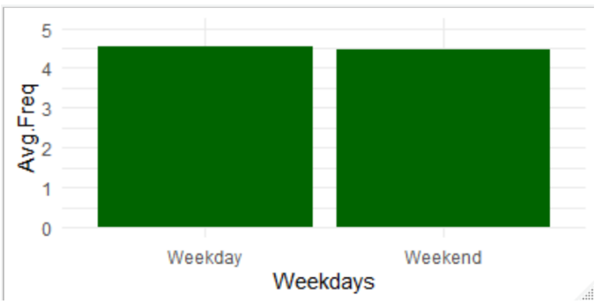


**Figure 10 - Average frequency of videos published on a weekday vs. weekend**

Since the average frequencies of videos published on weekday versus weekend are the same, we will not consider this variable in the prediction of video trendiness.

From the original dataset, attributes are used to conduct analysis, however due to lack of the ability to differentiate between them, they have been removed.

**Table 3 - Attributes that are eliminated post-data analysis**

| **Attribute** | **Reason for Elimination** |
|---|---|
| **description** | Tags are a better way to obtain key words |
| **tags** | After identifying the 30 most common words (tags), based on the selected key words, there was no significant difference in the frequency of trendiness in videos that contained the keywords versus the videos that did not contain the keywords. So tags were not useful. |
| **thumbnail_link** | The thumbnail link redirects to a picture of each YouTube video cover. Since we are not conducting any in depth pictorial recognition analysis methods in this project, this attribute was removed. |
| **comments_disabled**<br>**ratings_disabled** | Although there was a significant difference in trendiness frequency between videos that enabled ratings only than the videos that had disabled ratings or comments or both, were less than 2%, and therefore make up an insignificant proportion of the population.<br><br>The attributes (comments_disabled, and ratings_disabled) were used to narrow down the dataset to videos that had both enabled only (97% of the records). |

Other attributes that are used as a reference only, however not used in the process of developing data mining methods: **video_id, trending_date, category, channel_title.**

## VI. Data Mining Tasks

In order to predict the *trendiness level* of a youtube video that has previously appeared on the trending list, three models are explored: **Multiple Linear Regression (prediction)**, **K-NN Neighbors (classification), and Classification trees.**

The data is initially normalized. The transformed data ranges from approximately -3 to 3. Normalizing the data, allows to understand the position of a point relative to the rest of the population, therefore if in the future if it becomes usual to see a point greater than the current dataset (higher social media involvement), the point should be considered acceptable. However, a big challenge is that the mean and standard deviation of the set of future records is unknown as well.

## VII. Data Mining Models

### Multiple Linear Regression

The numerical predictors are views, likes, dislikes and comments. The scatterplot matrix in Figure 11 shows that the correlations between predictors and the **response variable (frequency)** is significantly low:

- Views (18.7%)
- Likes (15.8%)
- Dislikes (12.6%)
- Comments (11.7%)

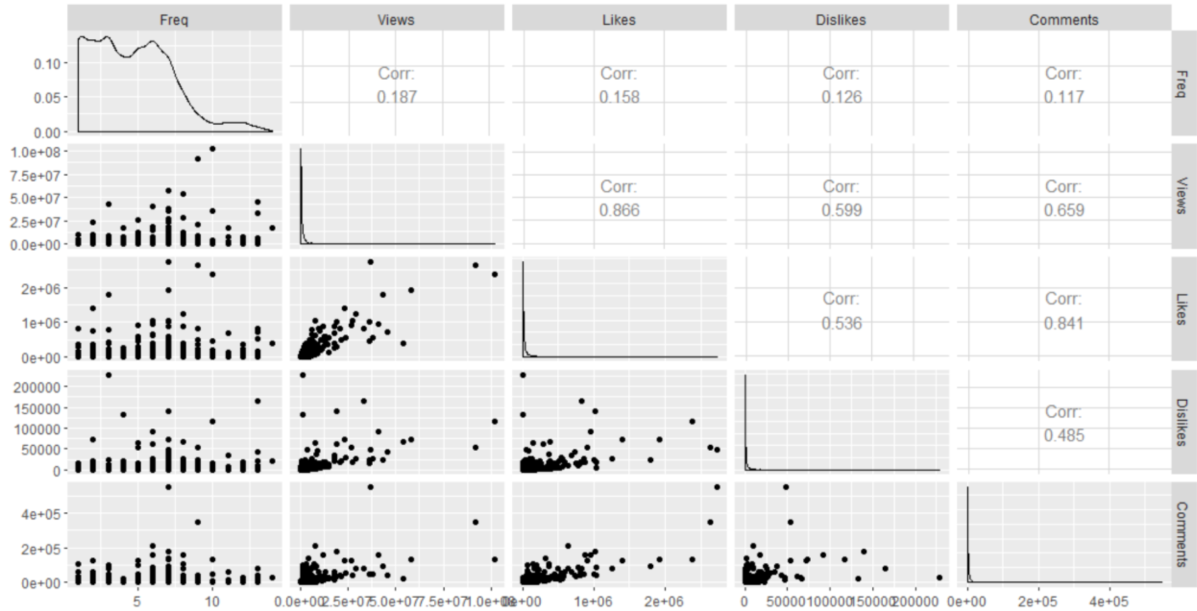


**Figure 11 - Scatterplot Matrix of Training data after outliers are removed**

Since the data is cluttered to  the bottom, the log of each variable is taken to create the following new scatterplot shown in Figure 12. Even after variable transformation, there is no apparent linear, polynomial or exponential trend between any predictor and the response variable (frequency). In addition, predictors have a clear linear correlation. It is expected that the multiple linear regression model will not perform well, therefore **the simple MLR method** is dropped, and **Principal Component Analysis (PCA)** is explored to observe if fewer principal component variables can be used as better predictors, and predictors for which common correlation is removed.

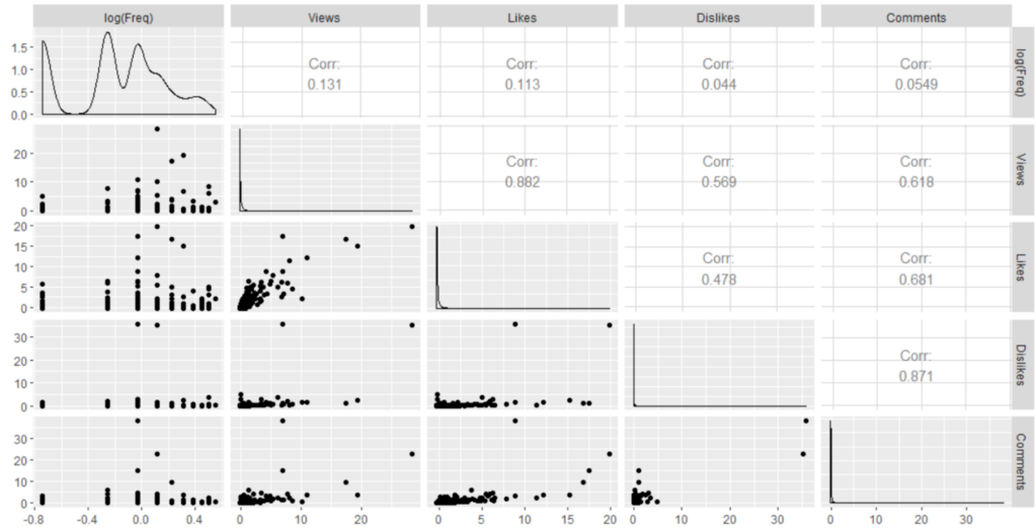


**Figure 12 - Scatterplot Matrix of Training data (log of each variable)**

## <u>Principal Component Analysis (PCA)</u>

Figure 11 showed that there was a relationship between predictors. All numerical predictors have significant correlation:

- Likes and Views (86.6%)
- Comments and Likes (84.1%)
- Comments and Views (65.9%)
- Dislikes and Views (59.9%)
- Dislikes and Likes (53.6%)
- Comments and Dislikes (48.5%)

The correlations listed above, demonstrate covariation between predictors - how much variation in one variable is duplicated by variation in the second variable.

After transforming the original variables (x1, x2, x3, x4) into Principal Components, the Principal components become the new predictors. The principal components are estimated in such a way that most of the variation in the data can be captured by the least amount of components - reducing the number of predictors and improving model performance.

```
> summary(pca)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7472 0.8470 0.43611 0.19924
Proportion of Variance 0.7632 0.1793 0.04755 0.00992
Cumulative Proportion  0.7632 0.9425 0.99008 1.00000
> pca$rot[,1:3]
                    PC1        PC2        PC3
Largest_view -0.5032611  0.4643518 -0.5824818
Likes        -0.4996801  0.5152948  0.4318374
Dislikes     -0.4765362 -0.6057547 -0.4268611
Comment      -0.5195781 -0.3897561  0.5403897
```

**Figure 13 - Variance Captured by new Product Components**

Since the first three product components capture 99% of the variation in the data - we chose to eliminate PC4. This would leave Product Components 1, 2 and 3 as the new predictors.

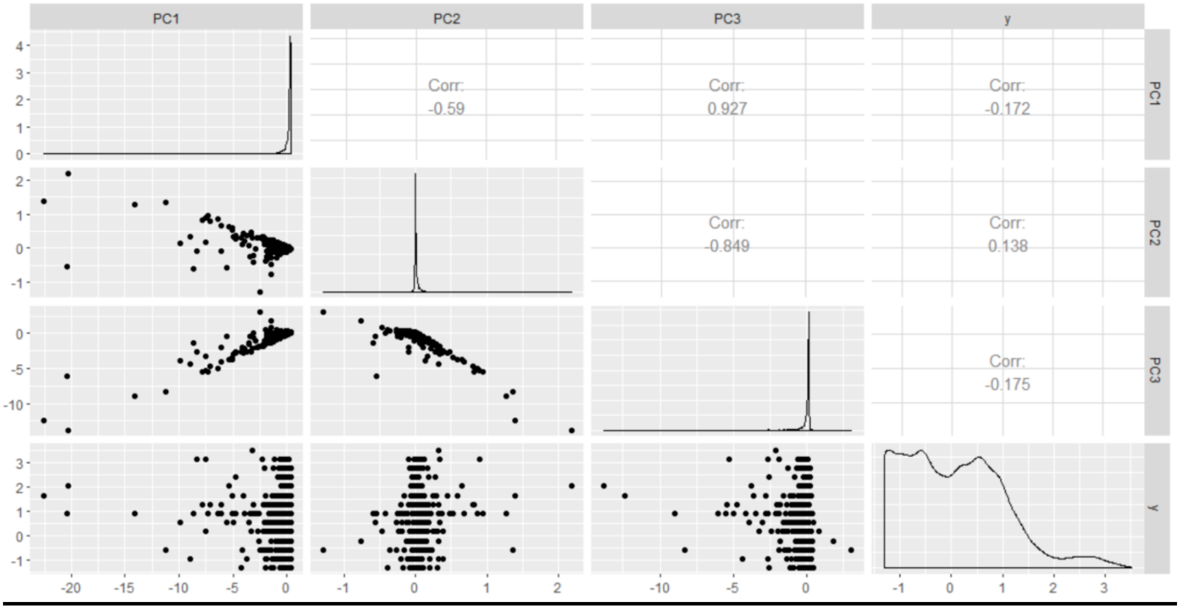


**Figure 14 - Scatterplot Matrix of Product Component Variables**

The correlations between the individual product components and the response variable are significantly low. Therefore we can conclude that there is no evident linear relationship between product components and the response variable (frequency).

```
lm(formula = y ~ PC1 + PC2 + PC3, data = outpca)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4624 -0.8946 -0.1486  0.5960  3.1835

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.242e-14  1.926e-02   0.000   1.0000
PC1         -7.528e-01  3.860e-01  -1.950   0.0512 .
PC2          5.767e+00  3.270e+00   1.764   0.0779 .
PC3          1.752e+00  1.072e+00   1.634   0.1024
```

```
> stepAIC(fit, direction ="backward")
Start:  AIC=-76.32
y ~ PC1 + PC2 + PC3

       Df Sum of Sq    RSS     AIC
<none>              2531.0 -76.317
- PC3   1    2.5889 2533.6 -75.644
- PC2   1    3.0159 2534.0 -75.204
- PC1   1    3.6892 2534.7 -74.509

Call:
lm(formula = y ~ PC1 + PC2 + PC3, data = outpca)

Coefficients:
(Intercept)          PC1          PC2          PC3
 -1.242e-14   -7.528e-01    5.767e+00    1.752e+00
```

**Figure 15 - MLR Model of Product Components**

Although an MLR model is expected to fail, we choose to create the MLR model with the product component variables.

The model is:

```
y = −1.242e−14−7.528e−01*PC1+5.767e+00*PC2+1.752e+00*PC3
```

```
> RMSE_valid
[1] 0.9825646
> RMSE_test
[1] 0.9781543
```

**Figure 16 - MLR Root Mean Squared Error**

The Mean Squared Error rate of this model is 98.2%, which means it’s accuracy 1.8% which is not any better than the Naive Benchmark. A numerical prediction of frequency (“trendiness level”) is not a good model for this particular dataset.

**<u>K-NN Neighbors</u>**

The k-nn is expected to be a better model because the predictors are highly correlated and there is a high likelihood for clusters to present in the data. The k-nn requires that variables be normalized so that the distance between the k-neighbors is comparable across the predictor variables. Normalization is done separately for each: the training, validation, and test data, because our goal is to see the performance of a video in comparison to its current competitors (goal is for a video to be higher than the 50th percentile)

This model incorporates “category” (the genre that a video belongs to). The reason that we do not include the variable “channel” is because we observed that there are a significantly large number of channels in the validation data which are not present in the training dataset (whereas “category” type is consistent).

To convert “category” into a numerical predictor,  we calculate the average frequency of a video in each category. We normalize the average frequencies of each category. The normalized values are used in the determination of k-nearest neighbors in the training dataset. The response variable is classified into two outcomes, success (> Median) or failure ( < Median).

## VIII. Performance Evaluation

**K-NN Performance Evaluation**

```
> accuracy.df
    k  accuracy
1   1 0.5625276
2   2 0.5563411
3   3 0.5735749
4   4 0.5775519
5   5 0.5912506
6   6 0.5974370
7   7 0.6098100
8   8 0.6067167
9   9 0.6177640
10 10 0.6120194
11 11 0.6164384
12 12 0.6261600
13 13 0.6212992
14 14 0.6177640
15 15 0.6235086
16 16 0.6252762
17 17 0.6204154
18 18 0.6235086
19 19 0.6292532
20 20 0.6261600
```

**Figure 17 -Accuracy levels of models that have k number of neighbors**

As shown above, the number of neighbors that gives the highest accuracy on the validation data is k= 19. Given a high number of neighbors, we can see that the data is smoothed to the trend rather than being fitted to local positions.

We test k=19 on the "test data" and calculate the accuracy of the model. The Accuracy level of the final model is 64.5%. We can see that K-nn is a relatively good model to use for this dataset.

```
          Reference
Prediction   0   1
         0 144 257
         1 241 762

Accuracy : 0.6453
```

**Figure 18 - Accuracy of models when tested on test data**

## Classification Trees

The KNN model could handle clusters of data. However classification trees can identify single variable (bin) associations with higher frequency (trendiness). It is expected that the classification tree model would perform much better in comparison.

We use  a complexity parameter (cp), which is the minimum improvement in the model needed at each node (in order to have a split). Since leaf nodes are classified using majority rule, the model is re-tested on the same training data to observe accuracy level.

Similarly, the model is tested on the test data to ensure that it is not overfit to the training data, and accuracy level is calculated.

## Classification Tree Performance Evaluation

| | CP | nsplit | rel error | xerror | xstd |
|---|---|---|---|---|---|
| 1 | 0.2221337580 | 0 | 1.0000000 | 1.0000000 | 0.02033772 |
| 2 | 0.0127388535 | 1 | 0.7778662 | 0.8025478 | 0.01981347 |
| 3 | 0.0087579618 | 4 | 0.7396497 | 0.7874204 | 0.01974160 |
| 4 | 0.0047770701 | 5 | 0.7308917 | 0.7898089 | 0.01975326 |
| 5 | 0.0039808917 | 8 | 0.7165605 | 0.7921975 | 0.01976480 |
| 6 | 0.0037154989 | 9 | 0.7125796 | 0.7921975 | 0.01976480 |
| 7 | 0.0035828025 | 17 | 0.6823248 | 0.7921975 | 0.01976480 |
| 8 | 0.0031847134 | 19 | 0.6751592 | 0.7937898 | 0.01977244 |
| 9 | 0.0027070064 | 23 | 0.6624204 | 0.8033439 | 0.01981713 |
| 10 | 0.0023885350 | 28 | 0.6488854 | 0.8152866 | 0.01987038 |

**Figure 19 - Table of complexity parameter values and associated tree errors**

The smallest xerror is #3, which has 0.7874, and has 4 splits for a best pruned tree.

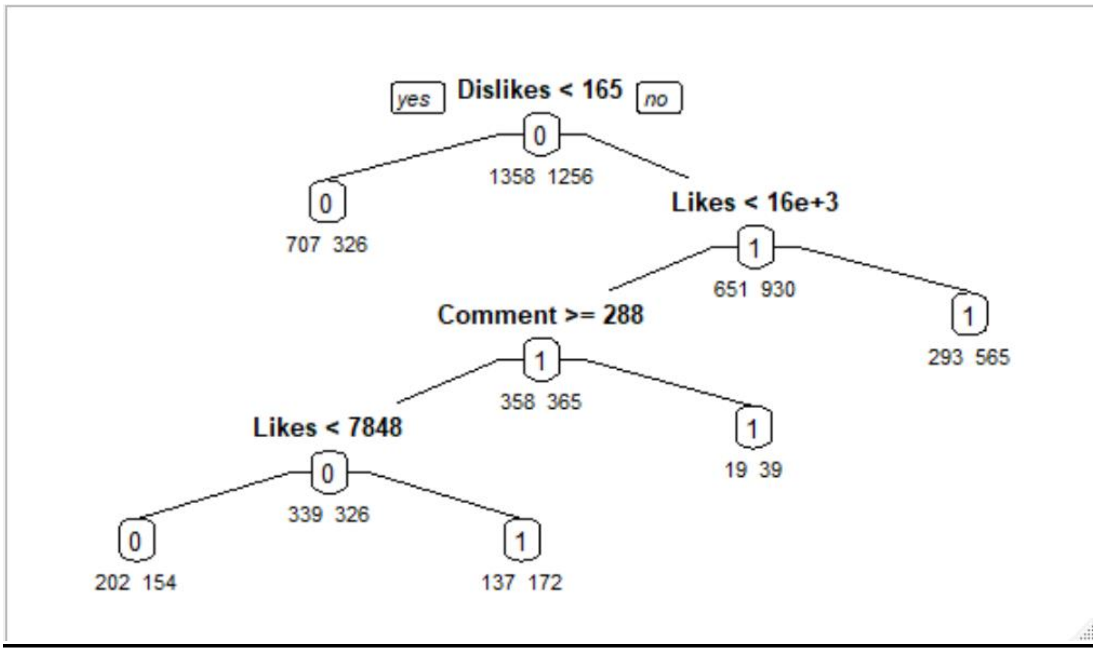


**Figure 20 - Classification tree for data**

```
> confusionMatrix(factor(ct.pred.train), factor(trainfreq$freq))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 909 480
         1 449 776

               Accuracy : 0.6446
```

**Figure 21 - Confusion Matrix for training data (Default tree)**

```
> confusionMatrix(factor(ct.pred.test), factor(testfreq$freq))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 107 163
         1 278 856

               Accuracy : 0.6859
```

**Figure 22 - Confusion Matrix for test data (Default tree)**

```
> confusionMatrix(factor(dct.pred.train), factor(trainfreq$freq))
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1358    0
         1    0 1256

               Accuracy : 1
```

**Figure 23 - Confusion Matrix for training data (Deeper tree)**

```
> confusionMatrix(factor(dct.pred.test), factor(testfreq$freq))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 171 405
         1 214 614

               Accuracy : 0.5591
```

**Figure 24 - Confusion Matrix for test data (Deeper tree)**

The chosen tree is the default tree because it captures the trend rather than the noise. Classifications of the validation data were correct 68.6% of the time, therefore this is a good model to use in order to ensure that the frequency of a video is at least 5.

## IX. Project Results

From the three data mining models chosen, MLR, K-NN, and Classification trees the predictive performances are as follows:

1. Multiple Linear Regression (PCA): 1.8%
2. K-NN neighbors: 64.5%
3. Classification trees: 68.6%

## X. Impact of the Project Outcomes

The highest predictive performance from the three models are the classification trees with a performance of 68.6%, This is higher than the naive benchmark (50%) and would therefore create a lift given predictor information.

## XI. Appendix

| Category ID | |
|---|---|
| 0 | Film & Animation |
| 1 | Autos & Vehicles |
| 2 | Music |
| 3 | Pets & Animals |
| 4 | Sports |
| 5 | Short Movies |
| 6 | Travel & Events |
| 7 | Gaming |
| 8 | Video Blogging |
| 9 | People & Blogs |
| 10 | Comedy |
| 11 | Entertainment |
| 12 | News & Politics |
| 13 | How to & Style |

| | |
|---|---|
| 14 | Education |
| 15 | Science & Technology |
| 16 | Nonprofits & Activism |
| 17 | Movies |
| 18 | Anime/Animation |
| 19 | Action/Adventure |
| 20 | Classics |
| 21 | Comedy |
| 22 | Documentary |
| 23 | Drama |
| 24 | Family |
| 25 | Foreign |
| 26 | Horror |
| 27 | Sci-Fi/Fantasy |
| 28 | Thriller |
| 29 | Shorts |
| 30 | Shows |
| 31 | Trailers |