Machine Learning
Capstone Project # 1 - Predicting Popularity of YouTube Videos

Code:

https://github.com/fariha23/YouTube_Data_Analysis_Video_Categories

Scenario:

YouTube (the Client) would like to identify which group of videos is popular i.e. which group gets the most views and engagement from the viewers. This could help with pushing those videos to new markets,identifying traits that make them popular and incentify the viewers to make videos with such traits. YouTube is looking to minimize Type II errors of the prediction model i.e. labeling videos that are popular as not popular should be kept at the minimum. If there is an increased number of incidences where popular class video is labeled as not popular class video, YouTube will not advertise those videos to the new markets and lose possible new audiences.

Approach:

1. Using feature engineering, create, split or modify the features of the data frame to make the data more robust for machine learning algorithms
2. Identify the groups (clusters) in the data using clustering algorithm KMeans
3. Once the groups have been identified, tags each video with the appropriate label for that group
4. Using KNN, Random Forest, Gradient Boost and AdaBoost classification methods train and test the data and calculate metrics of their performance

Exploratory Data Analysis:

The criteria on which a video is deemed popular are it's likes, dislikes, views and comments it receives. The YouTube video data that was used in this project came from Kaggle and it contains all those features (and more) for thousands of videos for different countries. The countries selected for this project are the USA and Mexico because they represent two very different kinds of viewers from a socioeconomic perspective.

Combining the two data from two countries created a data frame with over 76,000 rows. The data was collected over 2017 and 2018 on several different days. A lot of videos therefore are present in the data frame multiple times with the statistics for each of the collection day. In order to avoid duplication of the statistics only the most recent data was kept for the analysis. This exercise brought down the total number of rows from over 76,000 to around 36,000.

The likes, dislikes,comment_count and views data was spread on different scales and not distributed normally. In order to see the distributions of these statistics a log transformation was applied and a histogram was plotted. Figure 1 shows histogram plots for a non-transformed and a log-transformed data.

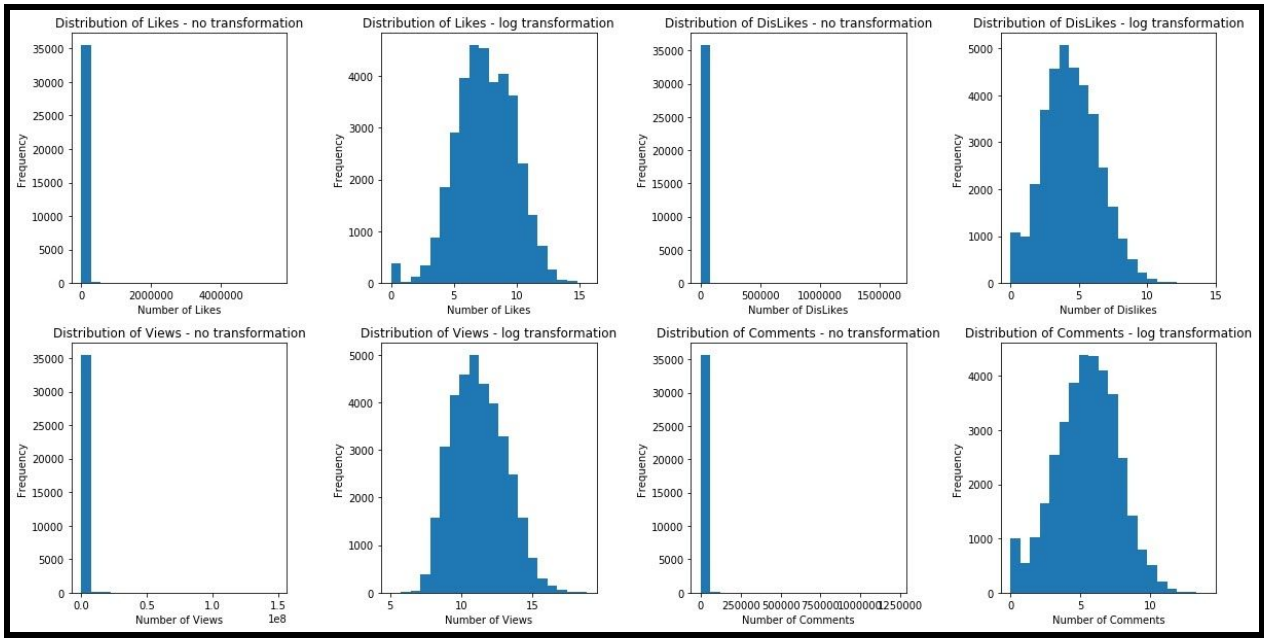

Figure 1: Distributions of Likes/Dislikes/Comments/Views

Figure 1 shows that before transformation the data is left skewed for all the parameters. After log transformation the distribution is normal. This shows that for machine learning analysis the transformation might be necessary.

Feature Engineering and Data Wrangling:

1. A new statistic was created using likes, dislikes and comment_count to show the engagement level of the users for the YouTube videos. The formula is given below. The comment_count feature is given twice the weight of likes and dislikes because if a user comments on the video he/she is technically putting in more effort compared to someone who hits likes/dislikes buttons

$$Engagement\ Score\ (En) = Number\ of\ likes + Number\ of\ dislikes + 2 * Number\ of\ comment\ count$$

2. Deleted rows that showed comments_disabled and ratings_disbled are true since they do not reflect a user's engagement level for those videos as the creator has disabled the ability to like/dislike or comment on the video
3. Added additional features for title length (title_len)and channel title length (channel_title_len). These could be informative features for the classification models
4. The log of views (views_log) and log of En (En_log) were added as transformed features. Having these on a same scale helps with clustering of the data

5. Text Extraction - 10 common words from the “title’:
    a. The title of the videos have a lot of different words and some are commonly used by the content creators repeatedly
    b. Using sklearn’s CountVectorizer converted a collection of text documents (titles of the videos) to a sparse matrix with their respective word counts
    c. The vocabulary formed after fit_transform method is extracted by using .get_feature_names() and saved in a list.
    d. The vocabulary is converted to an array and then to a data frame with feature names as columns
    e. Stopwords in english and spanish and also some of the expressions used in the titles that are not ‘words’ are dropped from the vocabulary
    f. Choose top used common words from the vocabulary array
    g. Concatenate the data frame with top words and the original dataframe

Clustering:

Using KMeans in sklearn, perform clustering using En_log and Views_log features. Plot knee-elbow graph to find the optimal number of clusters. Following is the plot showing the optimal number of clusters is either 3, 4 or 5. The knee point is loosely defined as the point of maximum curvature in a system i.e. where the inertia is least changed from one point to the other. Inertia is an attribute to identify the sum of squared distances of the samples to the nearest cluster

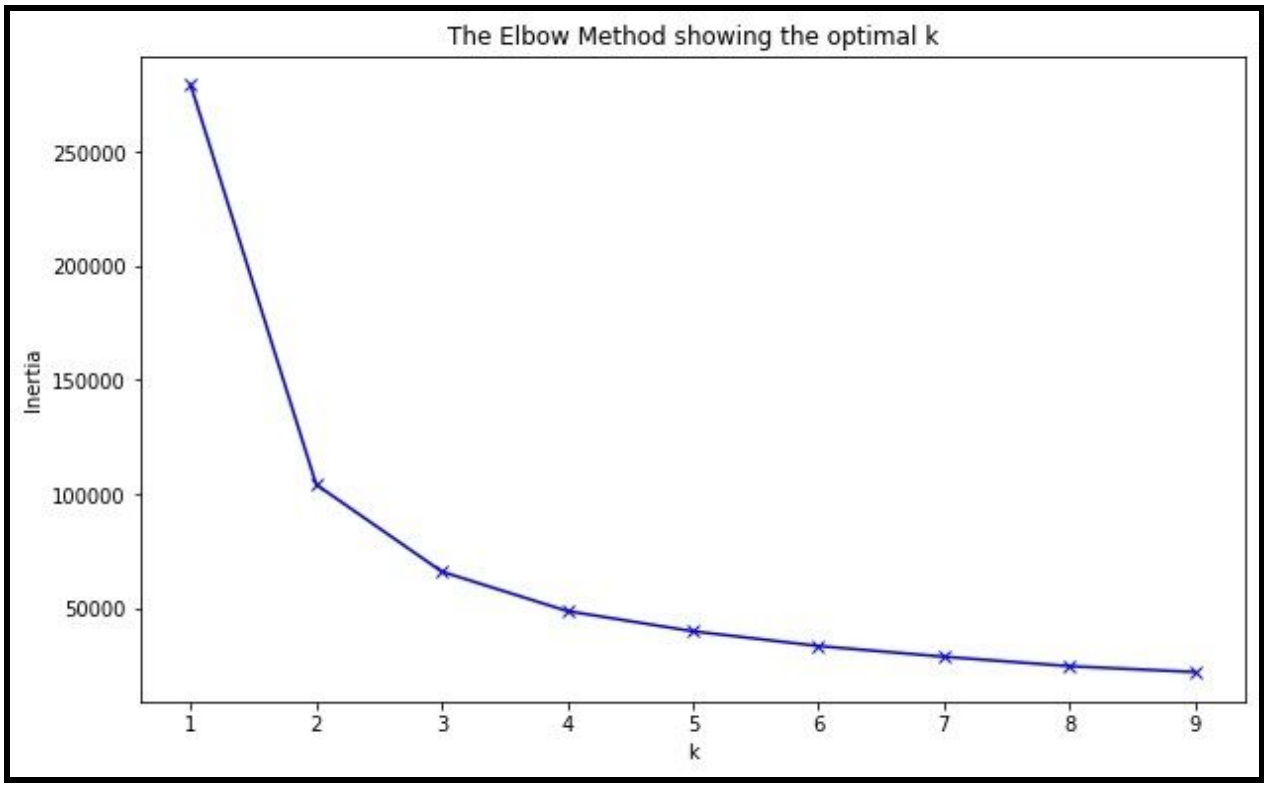

Figure 2: Knee-Elbow Plot

Classes were created using 3 clusters. Figure 3 shows scatter plot of the distribution of classes over En_log and Views_log data.

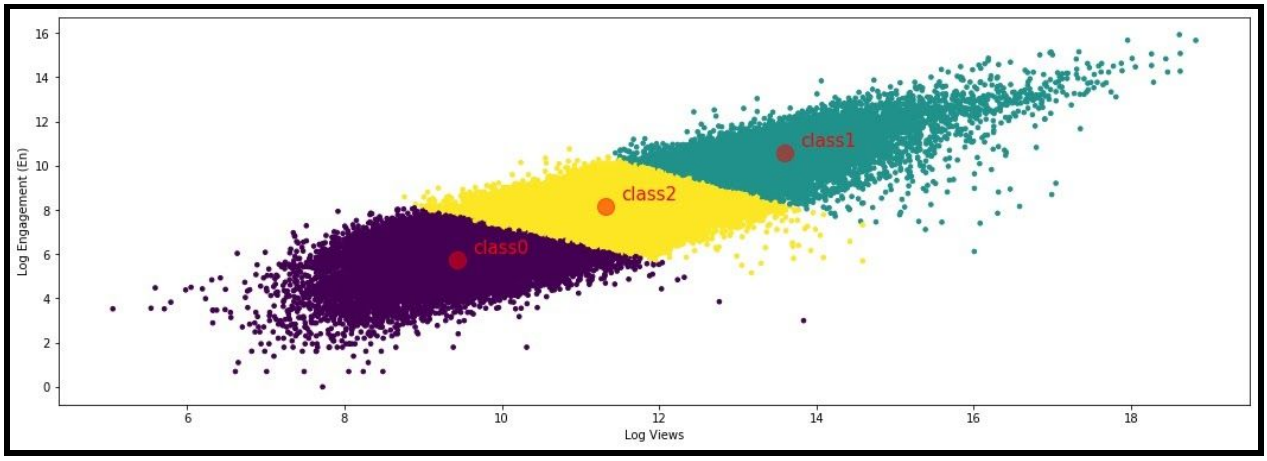

Figure 3: Class distribution with 3 clusters

Classes Description:

class 1 - views from 1.2M (e^{14} = 1202604.2841648) and onwards

class 2 - views from 0.2M (e^{12} = 162754.791419) to under 1.2M

class 0 - views under 0.2M

class 1 - engagement Score (En)is around 60K and onwards {likes/dislikes/2*cc (e^{11} = 59874.141715198)}

class 2 - engagement score (En) is within 2900 to under 60K (e^{8} = 2980.9579870417 to e^{11})

class 0 - engagement score (En) is within 0 to less then 2900

Majority of the videos are in class 2 as can be seen by the above description and Figure 3 i.e. most videos from 2017/2018 got views between (0.2M and 1.2M) and Engagement score between 2900 and 60K. The classes are not balanced as can be observed from the following table:

pop_class3	
0	12150
1	9152
2	13851

Table 1: Popularity classes - distribution of videos

Classification with imbalanced class data:

The imbalanced datasets when used with machine learning techniques have poor performance towards minority class and usually minority class is the most important class. SMOTE (Synthetic Minority Oversampling Technique) is a technique that oversamples the minority class. After splitting the data into train and test sets, the train set was oversampled for the minority class using SMOTE. The new distribution yielded 10,379 samples for each class.

Using KNN, Random Forest, Gradient Boost and AdaBoost prediction models, classification was performed. GridSearchCV was used to loop over multiple parameters of the models and find the best possible score for each model.

The summary of the performances is shown below

Accuracy Score:

	Best Score	**Best Params**
KNN	0.583	{'n_neighbors': 1}
Random Forest	0.639	{'max_depth': 20, 'max_features': 6, 'n_estimators': 100}
Gradient Boost	0.548	{'n_estimators': 90}
AdaBoost	0.522	{'n_estimators': 140}

Table 2: Accuracy Scores and Best Parameters

The accuracy of all the models is not very high. AdaBoost has the lowest accuracy and Random Forest has the highest. However, choosing a model for the prediction depends upon the goal at hand. The scenario to consider here is as follows and to evaluate the model's performance based on that criteria, confusion matrix and precision/recall scores are calculated.

Scenario: From Figure 3 the most relevant class to YouTube in class 1 since the videos in that class bring the highest views and engagement from the audiences. YouTube will advertise those specific videos to new markets to gain traction. If a class 1 is labeled as class 0 or 2, YouTube will not advertise that video and potentially not gain a new audience.

Therefore YouTube is looking to minimize Type II errors of the prediction model. A high incidence of False Negatives will be the worst outcome compared to having high incidence of False Positives. Keeping this in mind the precision, recall and f-scores below for the four models are calculated for class 1 only (considered as positive class).

Confusion Matrix:

Confusion Matrix is used to describe the performance of a classification model. The results show the predicted values of a test set vs. the true values.

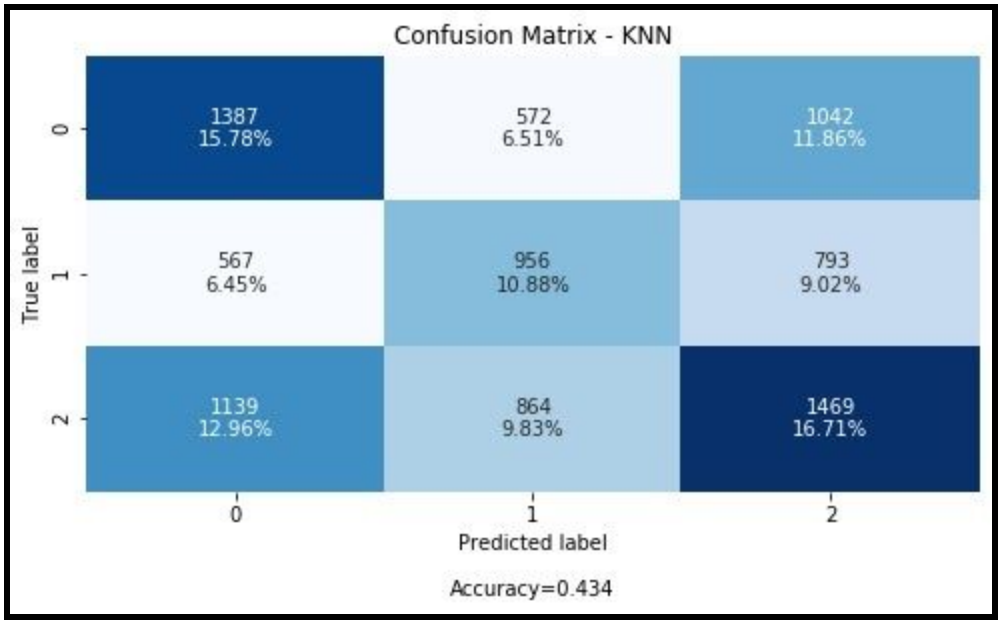

Figure 4: Confusion Matrix - KNN

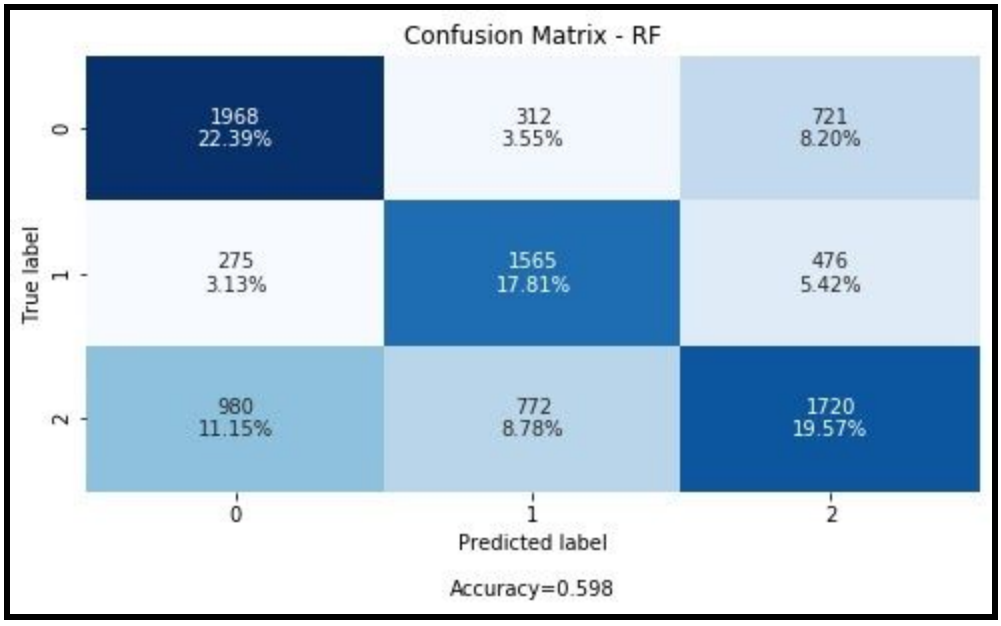

Figure 5: Confusion Matrix - Random Forest

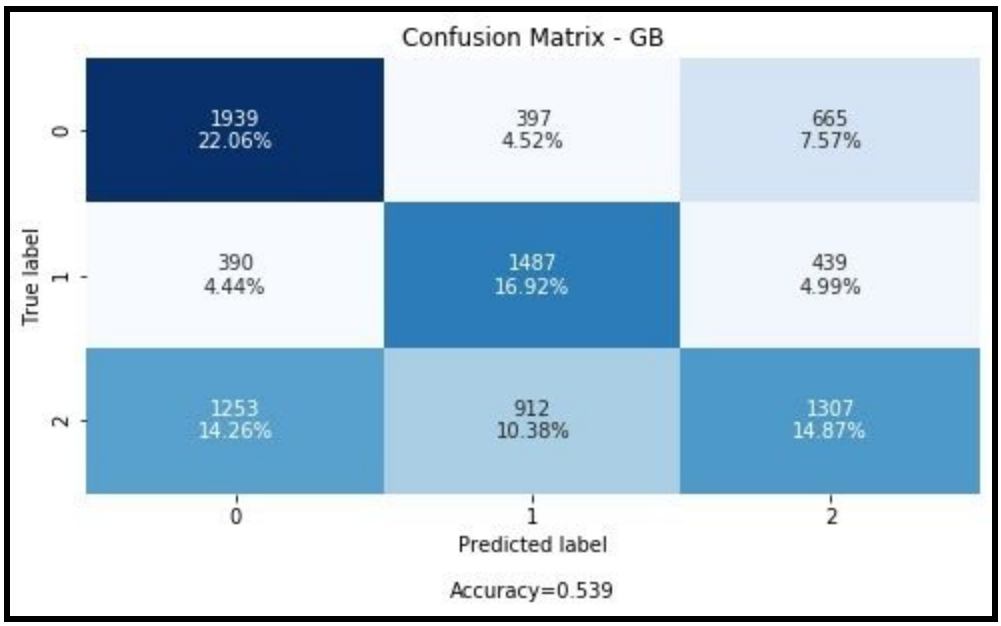

Figure 6: Confusion Matrix - Gradient Boost

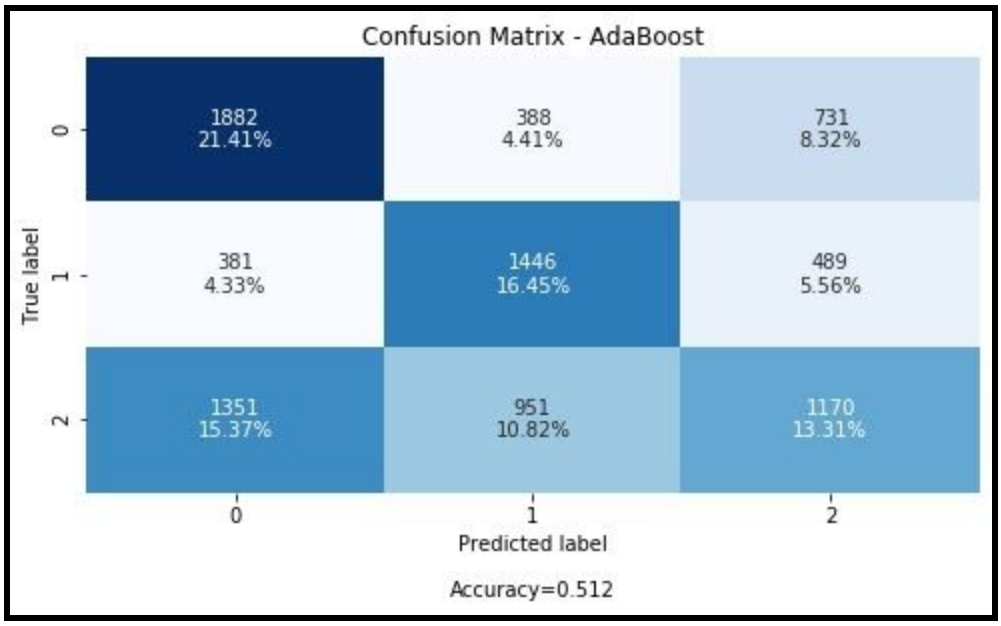

Figure 7: Confusion Matrix - AdaBoost

Precision/Recall/F-Scores - Class 1:

	Precision	**Recall**	**F-Score**
KNN	0.4	0.41	0.41
Random Forest	0.59	0.68	0.63
Gradient Boost	0.53	0.64	0.58
AdaBoost	0.62	0.52	0.57

Table 3: Precision, Recall and F-Score - class 1

Definitions:

Precision: Percentage of true positive from all the predicted positives = $\frac{TP}{TP+FP}$

Recall: Percentage of true positives from all the actual positives = $\frac{TP}{TP+FN}$

F-Score = Harmonic mean of Precision and Recall: $\frac{2\,(Precision)(Recall)}{Precision + Recall}$

From Table 3, since we are concerned with having low Recall (goal is to have least number of False Negatives), Random Forest and Gradient Boost could be eliminated from consideration. Both Random Forest and Gradient Boost predicted more false negatives(high recall score) i.e. class 1 being labeled as class 0 or class 2 making YouTube drop some of the popular videos from being advertised and possibly losing new audience.

AdaBoost seems to have a lower Recall and adequate Precision score. F-Score is more than KNN's F-score. Therefore it could be chosen as the model to achieve the required goal.

Conclusion:

The data provided is not adequate enough in determining the correct clusters. The classes that resulted from this activity were imbalanced which puts a challenge for predicting popularity of the videos correctly. SMOTE helped with making the data artificially balanced and along with the addition of several new features in the data helped with analysis and making accuracy of the models better than using the raw data itself. Since the goal at hand was to find popular YouTube videos and advertise them to new markets to gain more audiences, it was important to have least amount of Type II errors.Having False Negatives i.e. class 1 being labeled as class 0 or 2 would make YouTube drop important content to advertise and therefore hinder the progress on achieving the goal of engaging new audiences. After applying four different classification methods, AdaBoost model produced the recall score acceptable in comparison to others. F-Score of AdaBoost is also high which makes it better model for the purpose of achieving the goal at hand.