Analyzing Youtube Trends

Capstone Project by Liyan Cen

1 INTRODUCTION

As YouTuber becomes one of the most popular freelancer jobs nowadays, it's getting more challenging and competitive to achieve fame on the platform. In other words, to make money by uploading youtube videos is just becoming more difficult.

So, how does a YouTuber make money? When a YouTuber links Google AdSense to his or her channel, youtubers earn 68% of the as revenue. At the same time, YouTube charges advertisers when a viewer watches 30 seconds or more of the ad. On average, it chagres around $.18 per view. Furthermore, the youtuber earns only when viewers view the ad. The longer time viewers spend on watching ads, the more profits youtubers could earn. However, only about 15% of viewers will be counted as a “paid view” since many of them skip. In sum, average youtubers earn around $18 per 1,000 views.

Accordingly, view count is the direct and critical factor that affects a YouTuber’s income. There are many factors such as category, comment counts, likes, dislikes, publish time, trending month etc.. which can possibly affect the view counts. We use datasets from sources containing these factors to build supervised machine learning models to determine which factor has the most significant influence to the view counts.

2 Data Acquisition and Cleaning

The dataset we acquire includes several months (and counting) of data on daily trending YouTube videos for US with up to 200 listed trending videos per day.

Given the fact that the original dataset only has category ID, we add another column to have the ID translated into specific category names. The names we add in are 'Film and Animation', 'Cars and Vehicles', 'Smartphone', 'Music', 'Pets and Animals', 'Sports', 'Travel', 'Game', 'People and Blogs', 'Comedy', 'Entertainment', 'News and Politics', 'How to and Style', 'Education', 'Science and Technology', 'Nonprofits and Activism', 'News & Politics'. These names are based on the associated JSON file, “US_category_id.json”.

For our machine learning models, we have “comment counts”, “likes”, “dislikes”, “publish time”, “trending month” as feature variables, and “view counts” as our target variable. Hence, in order to make all feature variables uniform in type, we convert all the values to numeric. In specific, for “trending_date” and “public_time”, we only extract the month and hour respectively, and convert

them into integers. More details on the data cleaning process can be found from the IPython notebook. The cleaned dataset is then ready for explorations.

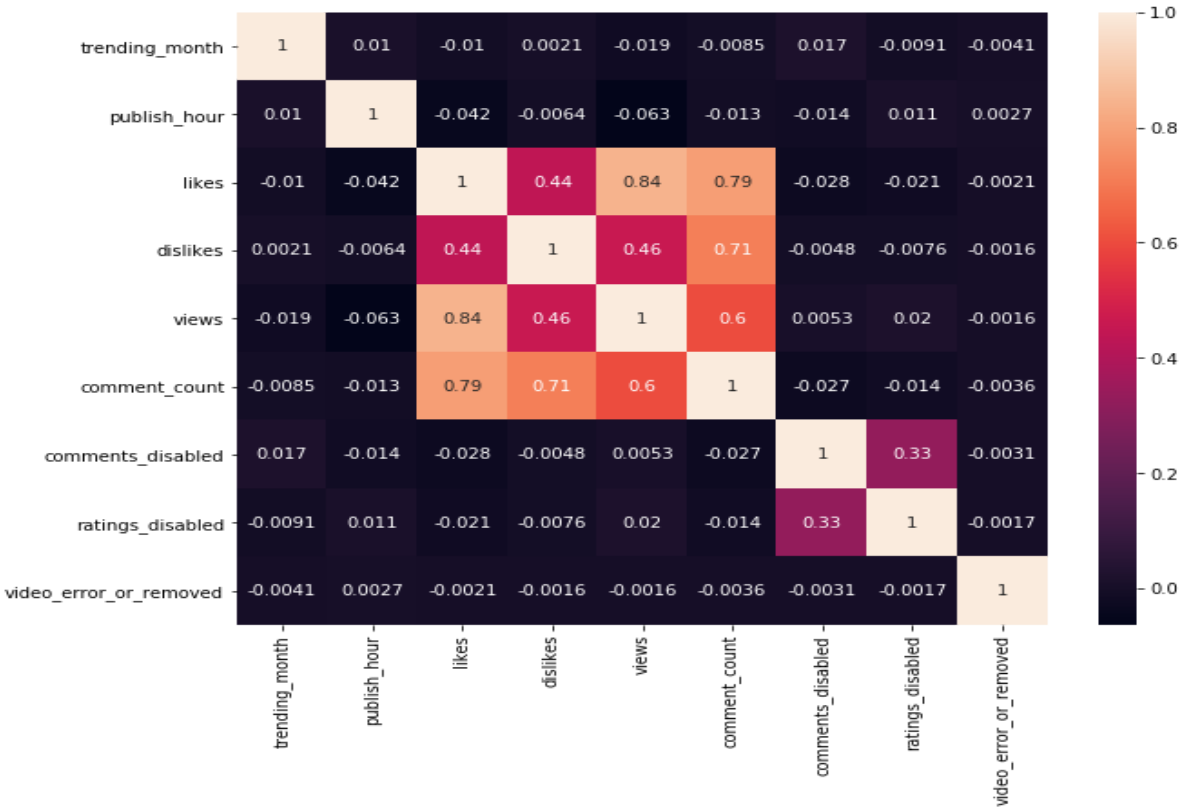
3 Data Exploration

3.1 Introduction to the cleaned data

There are 37549 records on daily trending videos in US for several months.We will go through most of the fields (or columns) in the dataset to explore their relationships with view counts. The target column for this project is called “views”.

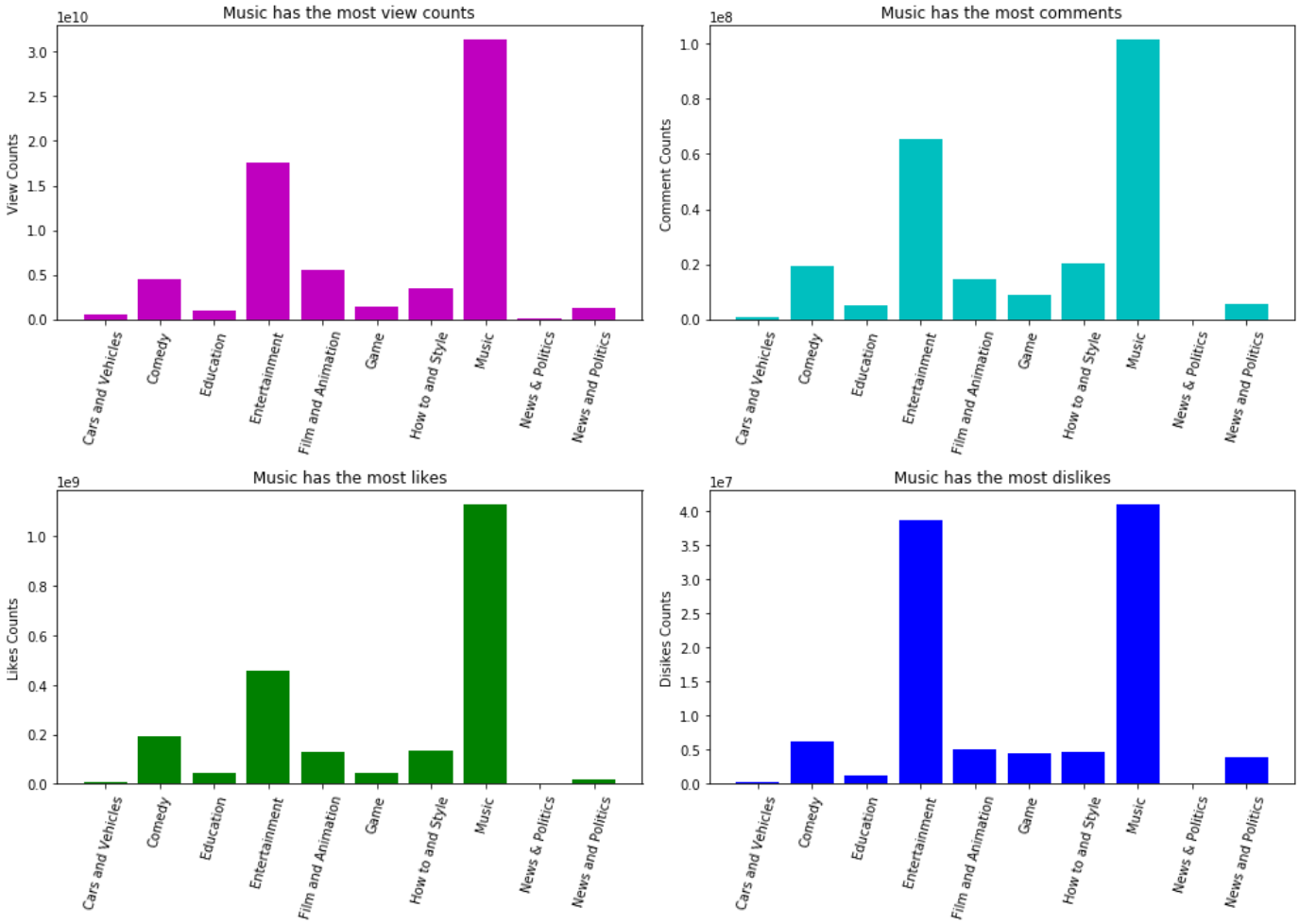
3.2 Correlation between variables

Based on the correlation matrix figure shown below, “likes”, “dislikes”, and “comment_counts” are more likely to have a correlation with the ‘views”.



3.3 Most popular video category on youtube is “Music”

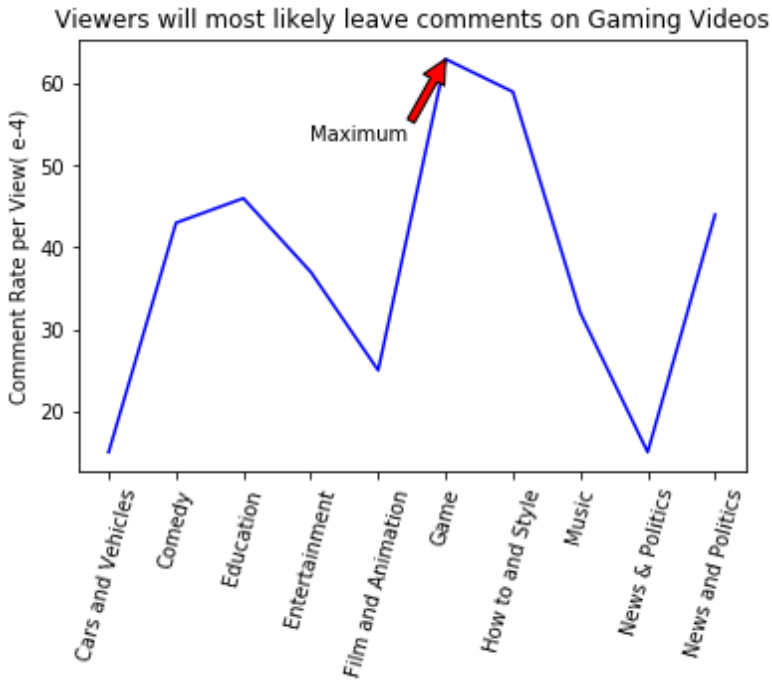
By referring to 4 graphs shown below, and by comparing the total views counts, comment counts, likes, and dislikes, we can conclude that music has the most of all the features.



However, the results turn out differently if we compare each video category by total feature counts dividing the total view counts for each category, which are explained in details in the next 5 sections below.

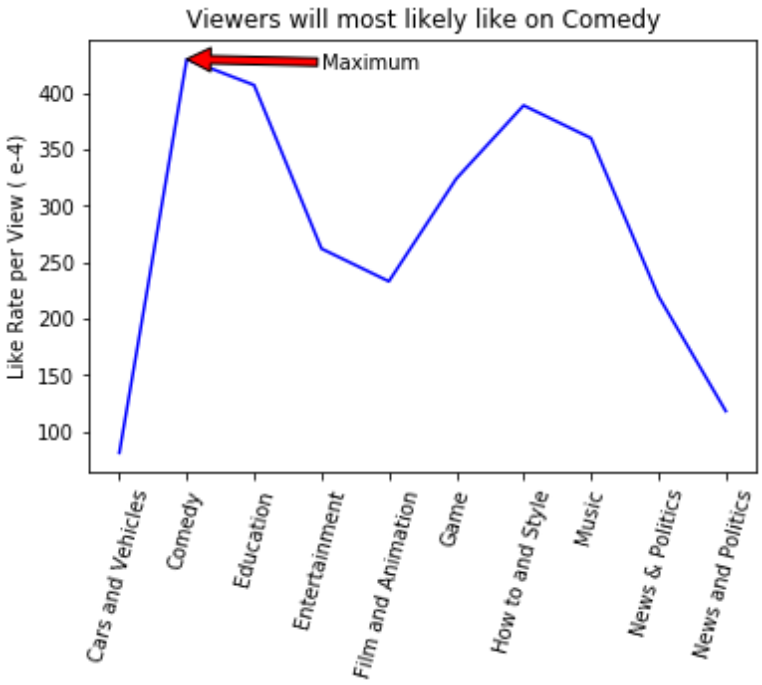
3.4 Viewers will most likely leave comments on Gaming Videos

By comparing comment counts for each video category and total view counts for each category, we get the conclusion that viewers mostly like leave comments on gaming videos.



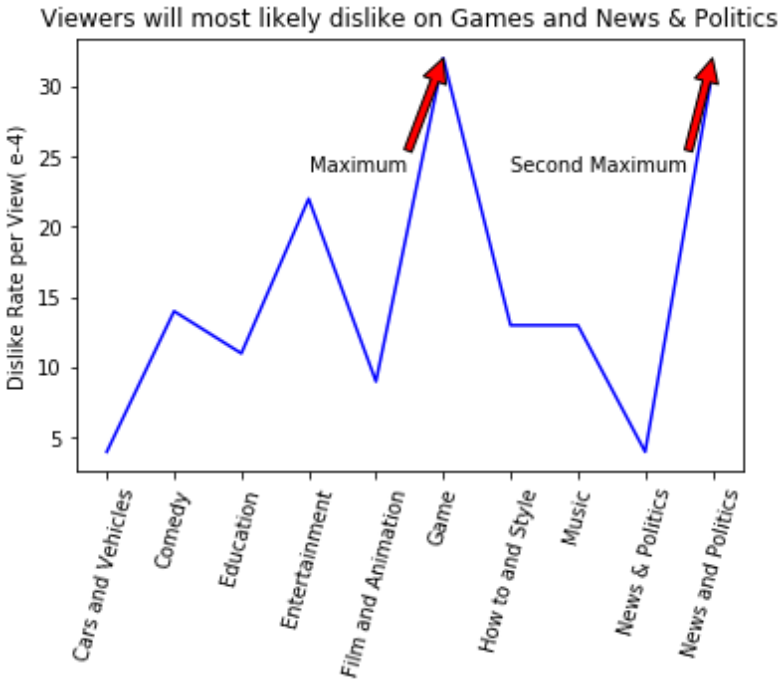
3.5 Viewers will most likely click on “like” on Comedy

By comparing like clicks for each video category and total view counts for each category, we get the conclusion that viewers mostly likely click on “like” on comedy.



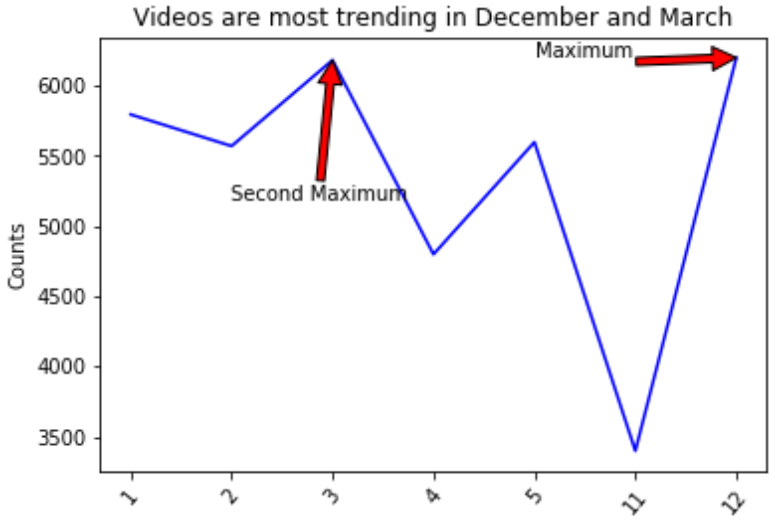
3.6 Viewers will most likely dislike on Games and News & Politics

By comparing dislike click counts for each video category and total view counts for each category, we get the conclusions that viewers mostly likely dislike on gaming videos, News and Politics.



3.7 Videos are most trending in December and March

By comparing the total video counts for each month by the total trending videos overall, we get the conclusions that videos are most likely to be trending in December and March.



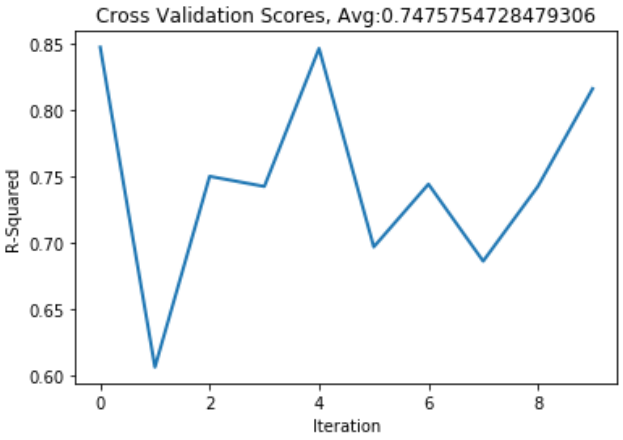
3.8 4pm is the time most of Youtubers publish their videos

By comparing the total video counts for each hour in day by the total trending videos overall, we get the conclusions that Youtubers most likely publish their videos at 4 pm.

4 Modeling

We use supervised learning algorithms to build predictive models, and find out the best model based on their performance. As we mentioned earlier, we will have “comment counts”, “likes”, “dislikes”, “publish time”, “trending month” as our feature variables, and the “ views” as the predictor. We perform Linear Regression, LinearSVR, Gradient Boosting, DecisionTreeRegressor, and RandomForestRegressor, and compare their “R^2” and “root mean square error” to figure out the best model, and best parameter.

4.1 Linear Regression



R^2: 0.7475754728479306
Negative Root Mean Squared Error: -11158621864039.723

First of all, based on our assumption that the feature variables and the target variable are roughly linear, we perform the Linear Regression because it can help us turn most non-linear features into linear pretty easily. By performing cross validation training, and linear regression model, we get R^2 of 0.75, which indicates that the model explains 75% the variability of the response data around its mean. Hence, this performance is not bad.

4.2 Support Vector Regression

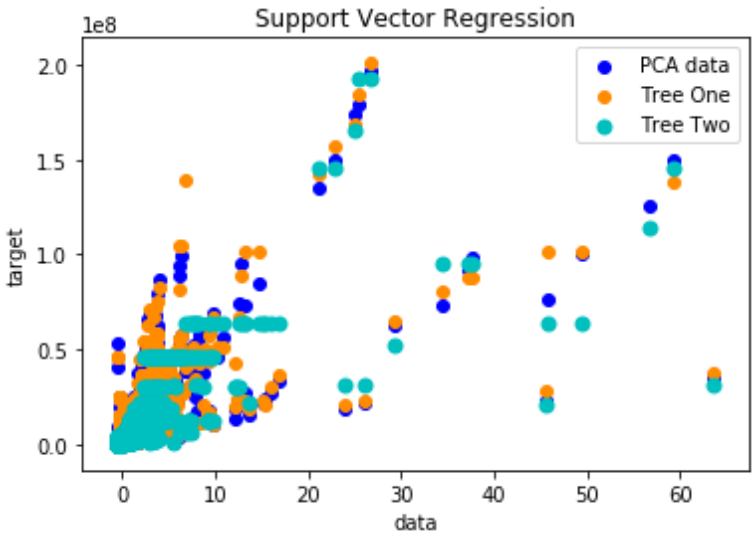
R^2: 0.6223845633914988
Root Mean Squared Error: 4277014.810241759

Given the fact that this project is to predict on a feature set with the large dimensional space, and the possibility that target variables may not be linearly related, we choose to perform SVC models, one of models in SVM, with a non-linear kernel, RBF.

By performing the SVR linear regression model, we get R^2 of 0.62, which indicates that the model explains 62% the variability of the response data around its mean.

This is the model that gives us the lowest R^2 of all the models we perform, it might be due to the limitation of the parameter tuning.

4.3 Decision Tree



Tree one R^2: 0.9253810156876101 # max_depth = None (no limit)
Tree two R^2: 0.843513452809086 # max_depth = 5
Tree one Root Mean Squared Error: 1901255.7957531882
Tree two Root Mean Squared Error: 2753305.4116950906

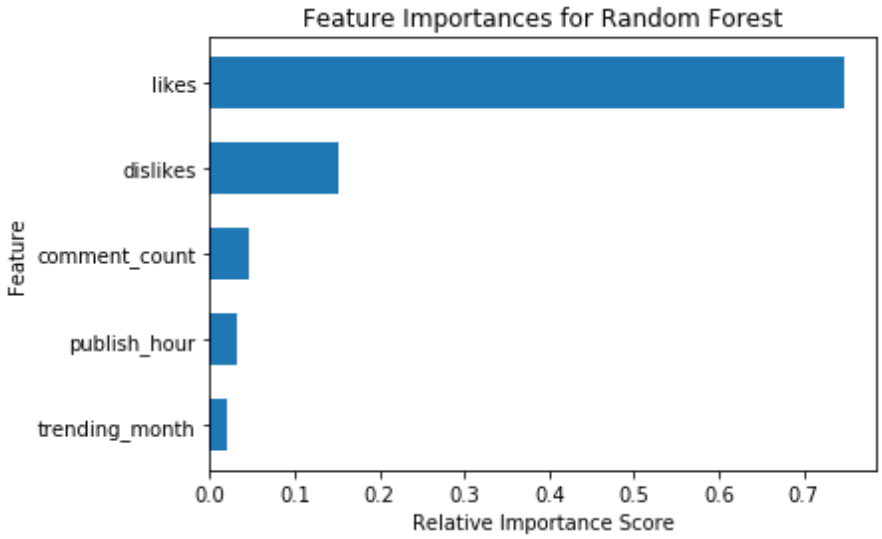
Besides examining the dataset as a whole, we also perform decision tree models to make the most optimal decision at each step. This model helps us to examine how each feature in specific has an effect on the target variables.

Based on its nature, this model tends to be overfitting. Hence, we set the max_df to be “None”, which helps to limit the risk of overfitting.

In order to compare, we set max_depth to be “None” and “5” for “Tree one” and “Tree two” respectively. Max_depth equalling to “None” means nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. Hence, this parameter will definitely be larger than “5”. As a result, the R^2 value and the Root mean squared error are better when parameter equals to “none” than “5” .

By comparing its results with other models, we can tell that the Decision Tree model does not perform as well as the other ones. This might be due to the nature of decision tree model, which it tends to make the most optimal decision at each step, instead of taking into account the global optimum. The other reason may also be that decision tree trains data based on the results of the last training data.

4.4 Random Forests

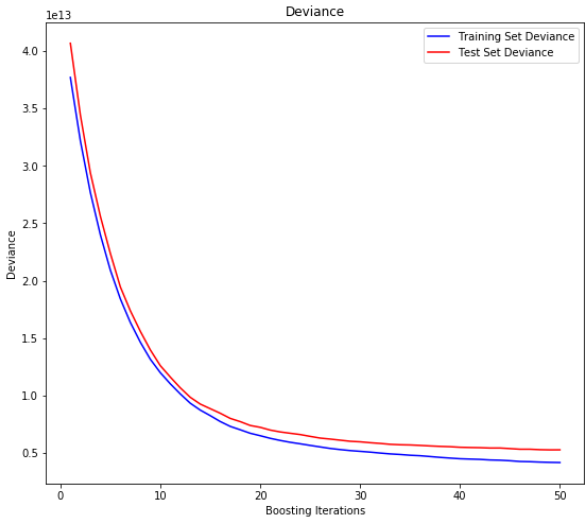


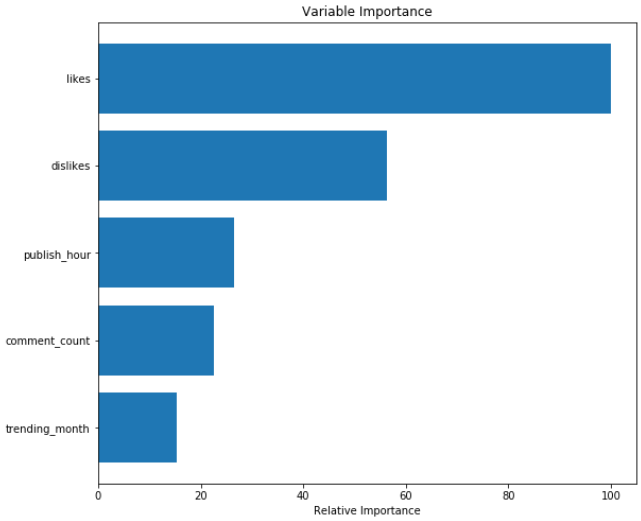
R^2: 0.9563351726789749
Root Mean Squared Error: 1454392.9404071828

Instead of running all features in a dataset, randomly selecting a subset of features for each step and running each one independently might ease the overfitting issue mentioned in single decision tree model. Accordingly, we try out Random Forests model to fix the problem. As a result, the output shows that Random Forests performs stronger than a single decision tree does.

Based on the feature names ranked above, we get a sense that the variable “like” has the most effect in our models, which means “likes” is the feature variable that has the most significant influence to “views” in this dataset. Furthermore, by performing the random forests model, we get a best R^2 value and best root mean squared error among all model results we get.

4.5 Gradient Boosting





R^2: 0.9339981072293979
Root Mean Squared Error: 3197335125828.107

Even though Random Forest performs well on this dataset, it does have its limitations like the features were chosen randomly with replacement. Hence, we try out another popular decision tree model, Gradient Boosting, to fix this issue by training on data instances that had been modeled poorly in the overall system before.

We get a R^2 value of 93%, which indicates that the model fits pretty well. And it is actually the second best model among all the models we perform. The reason why Random Forests performs a little bit stronger might be due to the fact that Gradient Boosting' training is based on last training result whereas in Random Forests, data is trained independently from the rest.

Based on all the models listed above, it turns out that the Random Forest and Decision Tree model give us the best results of all. In order to re-confirm which model is the best for our dataset, for the next two subsection, we use grid search to figure out the best hyperparameter by performing "DecisionTreeRegressor" and "RandomForestRegressor".

4.6 Hyperparameter tuning for the best model Grid Search - DecisionTreeRegressor

param_grid = {"criterion": ["mse"], "min_samples_split": [2, 3], "min_samples_leaf": [10, 20, 30], "max_leaf_nodes": [20,40,60]}

R-Squared::0.8481297179350443
Best Hyperparameters::
{'criterion': 'mse', 'max_leaf_nodes': 60, 'min_samples_leaf': 10, 'min_samples_split': 2}

Tree one R^2: 0.8755169363559091
Tree one Root Mean Squared Error: 2455675.7321696724

After performing the cross validation for the training data, and the DecisionTreeRegressor, we get our R^2 value to be 84%. The result is not as good as the RandomForestRegressor, which may be due to the parameter limitation.

4.7 Hyperparameter tuning for the best model Grid Search - RandomForestRegressor

param_grid = {'n_estimators': [50, 100, 200], "criterion": ["mse"], "max_features": ['auto','log2', None]

R-Squared::0.9382609850453063
Best Hyperparameters::
{'criterion': 'mse', 'max_features': 'log2', 'n_estimators': 100}

Tree one R^2: 0.9656872061674027
Tree one Root Mean Squared Error: 1289270.1411325312

As a result, the RandomForestRegressor gives us the best R^2 value among all the models we have performed. Hence, it is our best hyperparameter and model for this test.

5 Limitations

This dataset includes several months (and counting) of data on daily trending YouTube videos for the US, so they're not the most-viewed videos overall for the calendar year.

Additionally, for this project, we use likes, dislikes, tending month, trending time, and comment counts as our feature variables to predict the view counts. However, there are many other factors which could also have significant influence on view counts like comments, share counts, subscription counts, notifications counts and etc.

6 Other data and future work

Other than the original dataset, we can also acquire or feature engineer more datasets which might enhance the predictive power of the machine learning morel. Following is a list of some possibilities:

1. Knowing the subscription counts of each video channel.
2. Knowing the share counts of each video
3. Knowing the notification counts of each video
4. Knowing the comments for each video, both negative and positive.

Each one of these ideas can be difficult if the data is not easily accessible. However, we believe that the predictive power of the model will be improved by incorporating these factors.

7 Conclusions

7.1 Data Exploration Conclusions

Even though “Music” has the most “likes”, “dislikes”, “comment counts” and “views” based on the total counts of each feature in the dataset, but the rank differs if we compare each video category by total feature counts dividing the total view counts for each category. As a result, if we compare each category in ratio, the top performers on the YouTube trending list are music videos, but viewers will most likely leave comments on Gaming Videos, click like on Comedy, and click dislike on Games, News, and Politics.

On the other hand, videos are most trending in December and March. Also, most of the YouTubers like to publish their videos around 4 pm.

7.2 Modeling Conclusions

The feature “likes” is the factor that gives us the most significant influence on “views”. The RandomForestRegressor gives us the best R^2 and parameter among all the models we have performed.