YouTube Video Popularity Predictor

Kanav Bhagat
IIIT-Delhi
Okhla Phase III, New Delhi
kanav16046@iiitd.ac.in

Mohit Juneja
IIIT-Delhi
Okhla Phase III, New Delhi
mohit17067@iiitd.ac.in

Aarish Chhabra
IIIT-Delhi
Okhla Phase III, New Delhi
aarish17212@iiitd.ac.in

Abstract

This document intends to give the reader a detailed idea about the course project based on predicting the view count of a youtube video using different data mining and machine learning techniques. The data set comprising around over 6 Million entries are used along with different libraries like sci-kit learn, pandas and numpy. We made 2 models for the analysis. The first model uses integral features and train simple models(linear regression, Random Forest etc) for the task. In the second model, thumbnail of the video is used to train a Convolutional Neural Network, the output of that is combined with other features to train a SVM and a Random Forest Model separately. [CNN + Random Forest] gave the best measure (score=0.95) of the views. This project can be extended to create applications predicting number of view for a user based on the input fed to the model. The users can analyse and modify their content accordingly to generate more revenue.

1. Introduction

Youtube is unarguably the largest video-sharing and streaming website on the internet where users upload, view, rate, share, add to favourites, report, and comment on videos. There is a large possibility for examining data present on YouTube and getting useful insights out of it. **Our motivation is to help a huge set of people to predict the popularity of their next video. Influencers can modify the content accordingly and generate greater revenue from the views.** We want to predict the view count on a video using Machine Learning algorithms. There are many things on which the views on the video depends such as subscriber count, title, thumbnail etc. We will be implementing various techniques like Data Collection, Feature engineering, Feature selection and modelling in our project to get to the final goal. Before fitting the data, we will be cleaning the data using data mining techniques so that we can eliminate as much as noise so that clean and relevant data can be used to run the models. Also, we will be implementing various machine learning algorithms and try to find out the one giving the best result. Some of the algorithms we have implemented in this project are - CNN, linear regression, SVM regressor etc. We also try to find out the best model which can accurately predict the views on youtube video using a given set of features. We evaluate all our models based on the R2 score on training as well as testing data.

2. Related Work

While doing our literature survey we came across the work which was related to the work we are planning to do. Below are the explanations to those works:

- **Youtube-View Predictor[1]** - This work predicts the popularity of youtube videos by using Title and Thumbnail of videos as features. Videos of a specific domain, "Fitness and Gym", were extracted from the Youtube-8M. We are currently focused on using the numerical features to predict the view instead of using NLP.
- **Youtube-Like Count Predictor[2]** - This work is just an analysis of youtube videos by predicting likes using other features of the same video. The features involve Views, Dislikes, Comments, favorites values of a particular video. In our work, we are using features that are already present with us i.e. the **statistics of the previous videos** along with other channel attributes.
- **Convolutional Neural Networks deep implementation[3]** - This is an online source which gives detailed functioning of Convolution networks. The deep implementation of CNN is being explained which was used in our model. The source also explains on how to evaluate a CNN model and what data-cleaning techniques to use.

3. Methodology

3.1. Dataset

Dataset was downloaded from the YOUTUBE - 8M dataset. The dataset of cookery videos was saved locally

into a CSV file with features:

Title, Description, CategoryId, PublishedAt, CurrentTime, Life Definition, Caption, Duration, Dimension, Latitude, Longitude, LikeCount, DislikeCount, ViewCount, FavoriteCount, CommentCount, Tags, Thumbnail, ChannelId, ChannelTitle, ChannelSubscribers, ChannelUploads, ChannelViews, ChannelComments, Country.
Considering features with numerical value only, the following attributes were decided:

Duration, LikeCount_PreviousVideo, DislikeCount_PreviousVideo, CommentCount_PreviousVideo, NumberOfTags, ChannelSubscribers, ChannelUploads, ChannelViews, VideoLength, DescriptionLength, SocialLinks

Here **Previous Video** refers to the recent video uploaded by the channel at the time of a new video.
After scraping the above data from youtube, we have in total **3,47,624** videos extracted. Currently, we are using 20% of the dataset for testing and the rest of the data is used for training.

3.2. Thumbnail of Videos

As mentioned earlier, the thumbnail is an important feature for the predictions and hence we extracted and used thumbnails of all videos using the video ids. Python 'PIL' library is used in this case to convert the image to grey-scale and the pixel values are stored in numpy arrays. The dimension of a thumbnail is 180X320 and images with other dimensions (very few in number) are not included in the dataset. These images are passed in the convolutional neural network. The results from thumbnails are used as a feature to train further models and hence can be added in the above list.

Features after thumbnail extraction:
Duration, LikeCount_PreviousVideo, DislikeCount_PreviousVideo, CommentCount_PreviousVideo, NumberOfTags, ChannelSubscribers, ChannelUploads, ChannelViews, VideoLength, DescriptionLength, SocialLinks, Thumbnail_Result

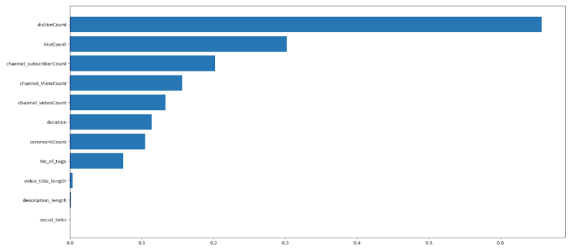
3.3. Evaluation

As the problem is based on regression, we are using the following evaluation metrics currently,

- R2 Score
- Mean Squared Error
- Mean Absolute Error

3.4. Feature Selection

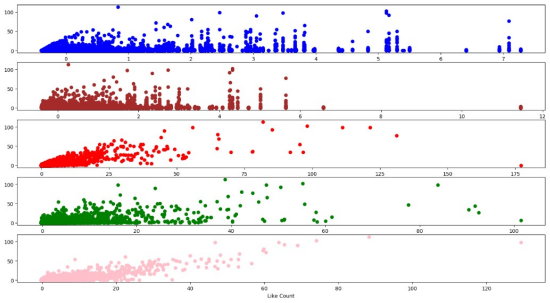
The analysis was made with respect to the importance of the feature (calculated using the Impurity techniques used in the Random Forest Classifiers). The following graph has been observed:



Studying the above graph, 3 features have been removed, namely, *Video Title Length, Description Length, Social Links.* It can be analyzed from the above graph that features such as Dislike Count, Like Count play a major role in predicting the results. The other features also influence the results to a nice extent.

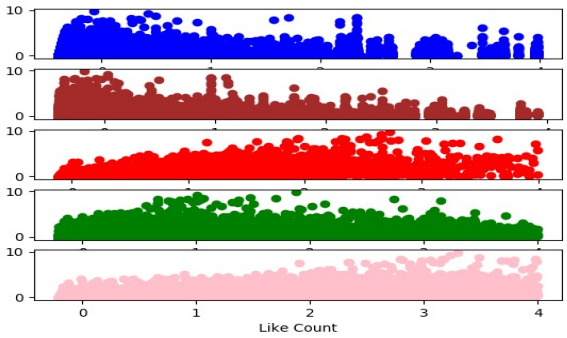
3.5. Outlier Removal

Picking the features with weights of the order of 10^{-1}, we have analyzed the data further by plotting the scatter plots of the data with respect to these features individually.



It is evident from the above scatter plot that data has significant number of outliers which have been further removed using the Z-Score Algorithm, keeping a threshold of 4 for the features and 10 for the View Count or the labels (as number of views are generally significantly more than the number of likes or dislikes)

After removing outliers:



3.6. Training the Models - I

Initially, the data have been trained on 4 models:

- Linear Regression
- Decision Tree Regression
- Random Forest Regression
- Support Vector Machine

Training the data with Linear Regression gives the weight parameters as shown below: [1.03654489e-03, -4.57557467e-04, 5.18023910e-01, -1.49883281e-01, 5.93377076e-01, 1.41848689e-01, -2.11171319e-01, 1.84621909e-02, 9.38064735e-03, -2.36220954e-03, -5.22645850e-03]

['duration','description_length','dislikeCount' ,'commentCount','likeCount','channel_ViewCount', 'channel_subscriberCount,'channel_videoCount', 'video_title_length','No_of_tags','social_links','viewCount']

These weights also confirm the feature selection process as the features removed have lesser weights than those selected.The results (accuracies) from these models were not up to the mark and thus thumbnail is introduced in II part to improve the output.

3.7. Training the Models - II

The training of the model has been done in 2 parts:
1) The first part focuses on extracting the feature, **"thumbnail_result"**. Convolutional Neural Network has been used for the same. Dataset for the CNN consists of the **Thumbnails** and the **labelled_views**. The actual views have been mapped into labelled_views by classifying the views into **10** classes. The data is then trained with the CNN and the resulting labels, giving an approximation of the actual views are stored as "thumbnail_result" to be used further.

Architecture of CNN:
Number of Layers: 6
Layer1: Convolution Layer with, kernel size = 5, padding = 2, stride = 1, dropout = 0.25, mapping one input channel into 16 channels.
After this, ReLu is applied on the outputs.

Layer2: Max Pooling Layer with stride = 2

Layer3: Convolution Layer with, kernel size = 5, padding = 2, stride = 1, dropout = 0.25, mapping 16 input channel into 32 channels.
After this, ReLu is applied on the outputs.

Layer4: Max Pooling Layer with stride = 2

Layer5: Fully Connected Layer mapping into 100 neurons, dropout = 0.25.

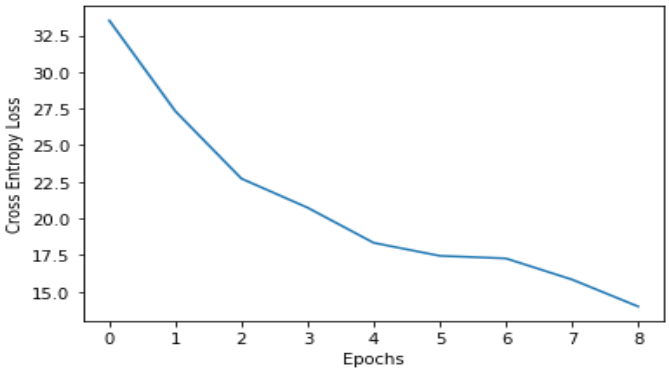
Layer6: Fully Connected Layer mapping 100 neurons into 10 class probabilities.

Intuition behind the Idea
Dropout of 0.25 is being used to generalize the results or to decrease the variance of the model.
The architecture includes just 2 FCC layers. More number of FCC layers would intuitively lead to overfitting. The overfitting is also evident from the R2 scores obtained when 3 FCC layers were used. The variance increased by 0.2-0.3. The pooling layers help to decrease the spatial dimensions of the image.
The batch size chosen for training is 128.
The loss on the training set Vs Epochs curve is provided below:

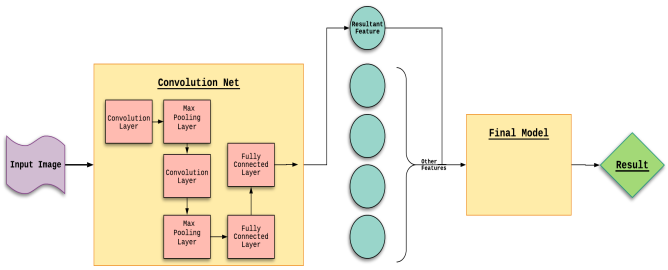
2) The **"thumbnail_result"** feature is now used with the above-mentioned features and has been trained with the following models, with the target variable being the actual number of views.

- Support Vector Regressor
- Random Forest Regressor

Generally, we are always more attracted to SVM as it works on the concept of maximizing the margin and thus



provides us with more general results. The RandomForest has been used as the dataset of the problem is still expected to have the outliers with variance in the number of views. The better capability of the Random Forest to handle the outliers provides an optimum reason for the choice.



Hyperparameter Tuning
Grid Search has been applied for the tuning of hyper-parameters. Grid Search has been applied with cross-validation=3.
Deciding the kernel: GridSearch chose 'RBF' as the most optimum kernel for the dataset.

Deciding the Max-depth of the tree in Random Forest: Grid Search chose 6 as the most optimum depth for the dataset.

Deciding the number of trees in Random Forest: Grid Search chose 10 as the most optimum number for the dataset which is infact the default value given by sklearn library.
The above results are completely in sync with the results we observed with our earlier scheme of training the data. (Training the Data - I)

4. Results

4.1. Training Model - I

After Running our data on Linear Regression, Decision Tree Regression and Random Forest Regressor, following are the results we gained.
Linear Regression:
R2 Score on the Training Data: 0.6364104183115034
R2 Score on the Test Data: 0.6282098478473409

Decision Tree Regressor:
R2 Score on the Training Data: 0.999999740665929
R2 Score on the Test Data: 0.5009736557269098

Random Forest Regressor:

- *Max-Depth 3:*
  R2 Score on the Training Data: 0.6185722789871666
  R2 Score on the Test Data: 0.6048359135782098

- *Max-Depth 4:*
  R2 Score on the Training Data: 0.6761023913595483
  R2 Score on the Test Data: 0.6428325354048319

- *Max-Depth 5:*
  R2 Score on the Training Data: 0.7278063118746587
  R2 Score on the Test Data: 0.6816166621953484

- *Max-Depth 6:*
  R2 Score on the Training Data: 0.775641123130922
  R2 Score on the Test Data: 0.6881128149014231

- *Max-Depth 7:*
  R2 Score on the Training Data: 0.8161590545191802
  R2 Score on the Test Data: 0.7178058576749329

- *Max-Depth 8:*
  R2 Score on the Training Data: 0.8469852309853958
  R2 Score on the Test Data: 0.7421699388091583

SVR:
R2 Score on the Training Data: 0.7536050917458403
R2 Score on the Test Data: 0.7255821853378172

4.2. Training Model - II

After using the **"thumbnail_result"**, models ran on Support Vector Regressor and Random Forest Regressor gave the following results:

SVR:
R2 Score on the Training Data: 0.955
R2 Score on the Test Data: 0.92

RandomForest::
R2 Score on the Training Data: 0.97
R2 Score on the Test Data: 0.95

More testing and trying with the increasing or decreasing the depth would surely lead to overfitting and underfitting respectively and the same even reflected in the results. So,we persisted with the max-depth of 6 only for the final model. Increasing the number of trees beyond 10 doesn't seem logical as it would surely lead to a downfall in terms of R2 Scores of the training set. Moreover, the results with this model are perfect, with decent R2 Scores on both the sets, having really less variance and bias.

5. Future Work

- In the future, a simple application could be made where a use can enter all the features of his/her upcoming video having details of thumbnail along with other attributes. The application would run our algorithm in the background and predict the number of view for the user. This would help to use to modify his content as required and earn more from the video.
- Though we are getting good results but again we haven't considered few more important features like Title of the video and Description of the video. Since these two features also play important role, we can use some sort of NLP techniques to fuse these two features with current feature set.

6. Conclusion

Thumbnail proved out an extremely important feature for sure as the results suggest. Without using thumbnail, many models were tried so as to increase the R2 Score of the model.

- Linear Regression, being the basic model for regression was initially tried but the result was not satisfying.
- Random Forest Regressor gave decent results for max-depth 5 and max-depth 6 whereas for lower and higher max-depths, underfitting and overfitting respectively were observed.
- The model trained by Decision Trees was highly overfittied as the R2 Scores suggest.
- Support Vector Regressor gave better R2 scores, comparable to Random Forest Regressor with optimal max-depths.

In quest of improving the R2 scores, thumbnail of the videos was chosen. The results improved significantly. Seeing the results without thumbnail, only 2 models were considered most optimal, the SVR and the Random Forest Regressor.

CNN + Random Forest turns out to be the best model for this dataset.
Through this project, many YouTubers/influencers can tune in their video before uploading to get the maximum view count and viewers will also get to watch the brewed content. As we described in Future work, we can take into account more and more features so that the model matures.

References

[1] https://towardsdatascience.com/youtube-views-predictor-9ec573090acb

[2] https://github.com/ayush1997/YouTube-Like-predictor

[3] https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-an