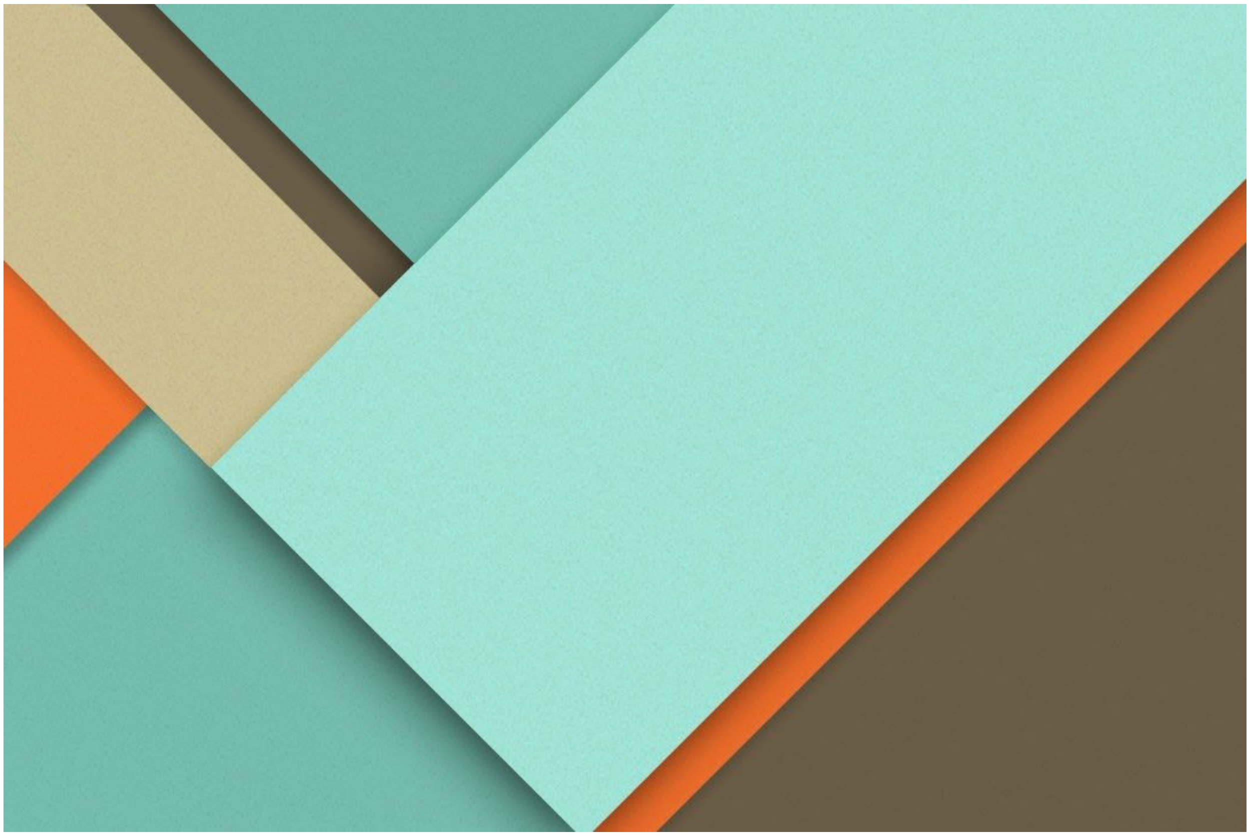


Analysis on Prediction of Trending YouTube Videos Applying Machine Learning Techniques

Final Project Report

Kahang Ngau
Qingyuan(Nicole) Xie

George Washington University

Data Science Introduction & Practicum

Abstract

The use of Machine Learning(ML) has become prevalent and almost inevitable nowadays, as a result of the nearly limitless amount of available resources and data, the rapid growth of cheaper but more powerful processing methods, and the improvement of storing data capacity. The practice of applying ML is a quickly growing field in computer science and it solves problems that are too difficult or time consuming for humans to solve. It is even more appealing to the other fields such as social media, where Big Data is analyzed through ML. In this project, we will demonstrate a detailed process of applying ML to understand one of the largest social media's data, YouTube.

Introduction

YouTube is an influential and popular online video-sharing tool that is rated as one of the largest search engines owned by Google. Because of its convenient feature of uploading and sharing videos, it has reached 1.9 billion users worldwide by the end of 2019. Each day, more than 1 million videos are being viewed in the U.S., and almost 5 million videos are being viewed globally. As such a well-known and wide-used social media platform, YouTube is impacting people’s life. Different types of businesses use YouTube to advertise and promote their new products; singers use it to drop their newest songs, other people use it to share various aspects of their daily lives.

On YouTube webpage, it has a list of the top trending videos of the day, which is generated through measuring users’ interaction, such as the number of views/likes/comments/shares. That is to say, the most trending videos are not the most viewed videos, but a combination of those factors. Of course, the more likes/views/comments you get, the more likely your video will become trending in the Youtube community. Therefore, understanding what is trending on Youtube could give us information about its users, because it indicates what most users are interested in at the moment. Plus, with some demographic information of the users, we can draw more precise conclusions in terms of a specific group of users on YouTube.

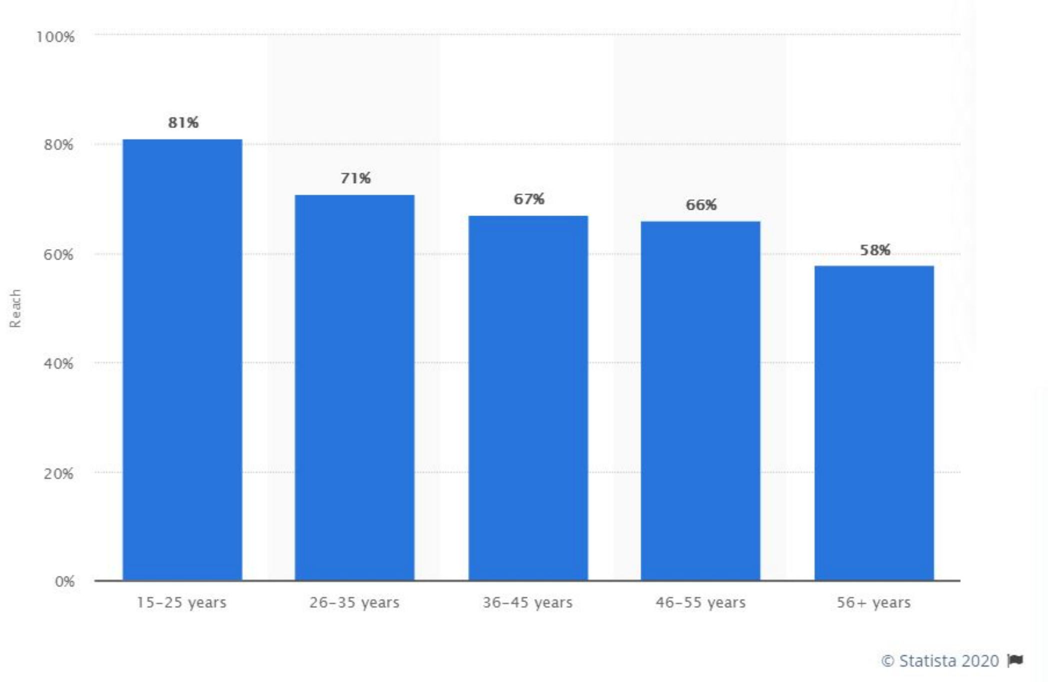


Fig.1 Youtube Users' demographic by age

Hence, our goals for this project are: To identify key features that predict trending videos are being liked the most in the U.S. Use Machine Learning to train model(s) on prediction and then evaluate and improve model performance. We targeted a broad range of audience, basically anyone who is interested in the topic. Ideally, we wanted to show the process of using big data and build the model to explain and predict the question of interest. The question that we are trying to solve are following:

I. What information can we extract from our dataset? How do we clean and transform the data so we can best do the analysis?

II. Which features influence the most on prediction on the number of likes on trending YouTube videos?

III. How do the models perform and how should we evaluate and compare different models. What is the Root Mean Square Error and R-square figure of each model?

Methodology

Data Identification

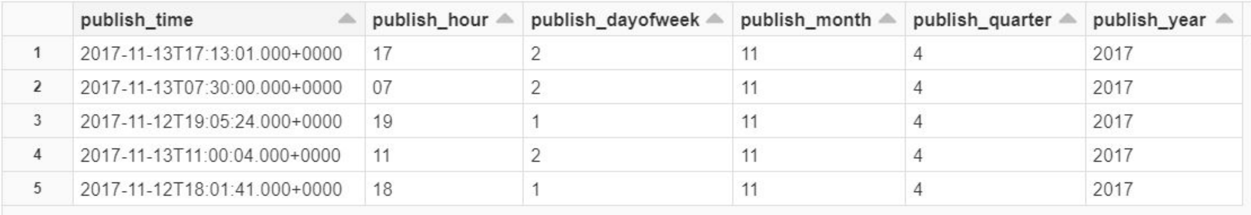
The data was pulled from Kaggle(link of the web data), and we used the method that was provided on Kaggle to retrieve the data set that we want to study. The dataset was built originally from kaggle with script scraping the most relevant information from videos that are currently trending on YouTube in a specific set of countries. The data information contains about six month of trending YouTube videos on it. And the country we focused on is the United States in this research. The reason why we chose to use the dataset from kaggle is that it has richful information and the contributor of this data set is an experienced author that has published a variety of work online.

Data Acquisition & Filtering

The data set was finalized with 40,949 rows and 16 columns. Although the YouTube API allows us to efficiently collect the dataset, it does not guarantee that every single feature variable will be suitable for model prediction. In order to better understand the list of feature variables, we checked how many null values each variable has to better understand how we should handle null values within our dataset. Besides the variable “Description” that has 570 empty rows, every other variable has zero empty rows. Since the nulls in this column are hard to correctly impute and may represent different meanings and dropping those empty rows creates bias among the whole dataset, therefore, we decided to drop “Description”. The second group of variables that we decided to drop are “videos_id” and “thumbnail_link”. The reason to drop them is because they are meaningless to the response as they are both unique to each video and may not provide any information to our response variable.

Data Extraction

As for checking the data type of all variables, we found that variable “publish_time” is a time data type that contains the detail of date and time of which the video was published. Therefore, We dissected the column “publish_time” and extracted several time features, including “publish_hour”, “publish_dayofweek”, “publish_month”, “publish_quarter” and “publish_year” into individual feature columns.

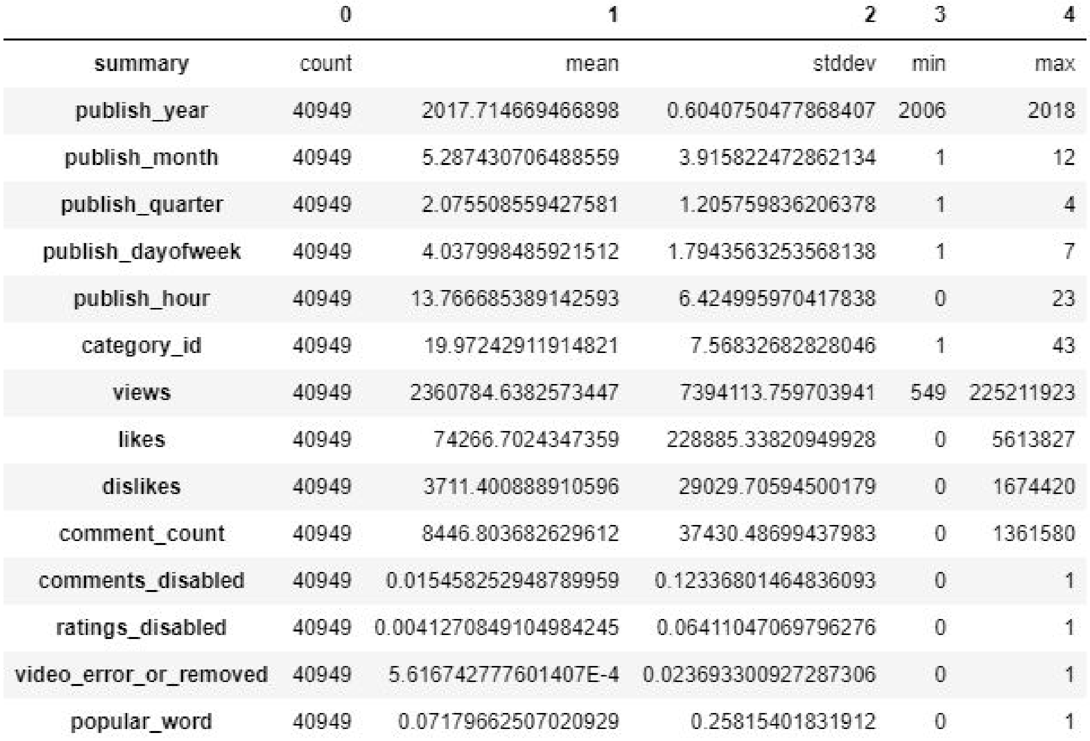


	publish_time	publish_hour	publish_dayofweek	publish_month	publish_quarter	publish_year
1	2017-11-13T17:13:01.000+0000	17	2	11	4	2017
2	2017-11-13T07:30:00.000+0000	07	2	11	4	2017
3	2017-11-12T19:05:24.000+0000	19	1	11	4	2017
4	2017-11-13T11:00:04.000+0000	11	2	11	4	2017
5	2017-11-12T18:01:41.000+0000	18	1	11	4	2017

Fig.2 Extracting hour, dayofweek, month, quarter, and year from 'publish_time'

Data Preparation

In this part, we conducted data checking by calculating the mean, standard deviation, max, and min figures of each variable column. By doing that, we can see if there are any outliers or bad inputs. We also wanted to avoid having string values be misplaced in the numerical columns.

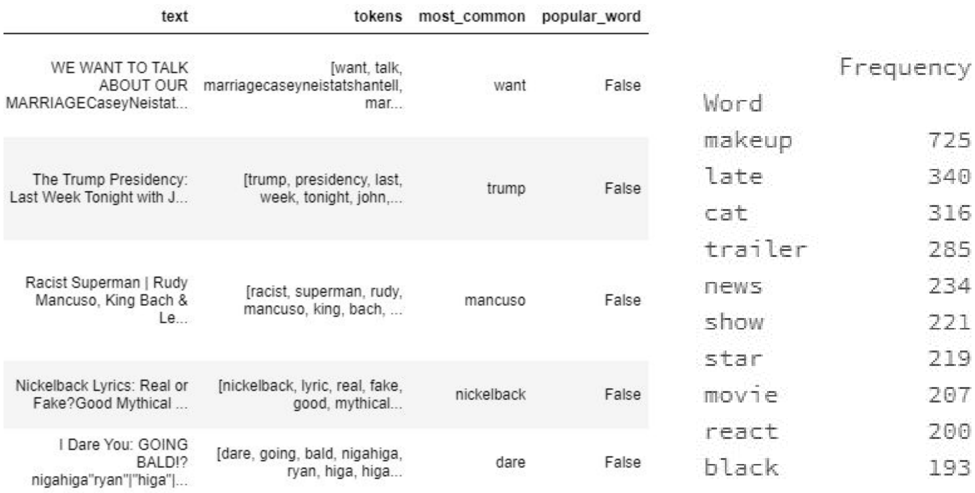


	0	1	2	3	4
summary	count	mean	stddev	min	max
publish_year	40949	2017.714669466898	0.6040750477868407	2006	2018
publish_month	40949	5.287430706488559	3.915822472862134	1	12
publish_quarter	40949	2.075508559427581	1.205759836206378	1	4
publish_dayofweek	40949	4.037998485921512	1.7943563253568138	1	7
publish_hour	40949	13.766685389142593	6.424995970417838	0	23
category_id	40949	19.97242911914821	7.56832682828046	1	43
views	40949	2360784.6382573447	7394113.759703941	549	225211923
likes	40949	74266.7024347359	228885.33820949928	0	5613827
dislikes	40949	3711.400888910596	29029.70594500179	0	1674420
comment_count	40949	8446.803682629612	37430.48699437983	0	1361580
comments_disabled	40949	0.015458252948789959	0.12336801464836093	0	1
ratings_disabled	40949	0.0041270849104984245	0.06411047069796276	0	1
video_error_or_removed	40949	5.616742777601407E-4	0.023693300927287306	0	1
popular_word	40949	0.07179662507020929	0.25815401831912	0	1

Fig.3 Count, Mean, Stddev, Min, and Max of all variables

Data Aggregation & Representation

Natural Language Processing (NLP) is one of the most important parts in exploratory data analysis of machine learning. We used NLP analysis to process three string type variables, “tag”, “title”, and “channel_title”. We first combined these three variable variables to column “text”. Then we cleaned the words that had no meaning in the text and removed redundant words that might not add much value to the meaning of the response from this column. This is to only keep the words that could be informative in the text to better use it in the later modeling step. We then separated each single word and stored them in a new column ‘tokens’. Next, we found the most frequent words of each row in the ‘tokens’ column and stored them in the “most_common” column. We then formalized a list of top 10 most frequent words from column “most_common” and finally, returned True or False in the newly created column “popular_word” associated with the word in “most_common” column if it is also in the list of top 10 most frequent words.



text	tokens	most_common	popular_word
WE WANT TO TALK ABOUT OUR MARRIAGECaseyNeistat...	[want, talk, marriagecaseyneistatshantell, mar...	want	False
The Trump Presidency: Last Week Tonight with J...	[trump, presidency, last, week, tonight, john,...	trump	False
Racist Superman | Rudy Mancuso, King Bach & Le...	[racist, superman, rudy, mancuso, king, bach, ...	mancuso	False
Nickelback Lyrics: Real or Fake?Good Mythical ...	[nickelback, lyric, real, fake, good, mythical...	nickelback	False
I Dare You: GOING BALD!? nigahiga"ryan"|"higa"|...	[dare, going, bald, nigahiga, ryan, higa, higa...	dare	False

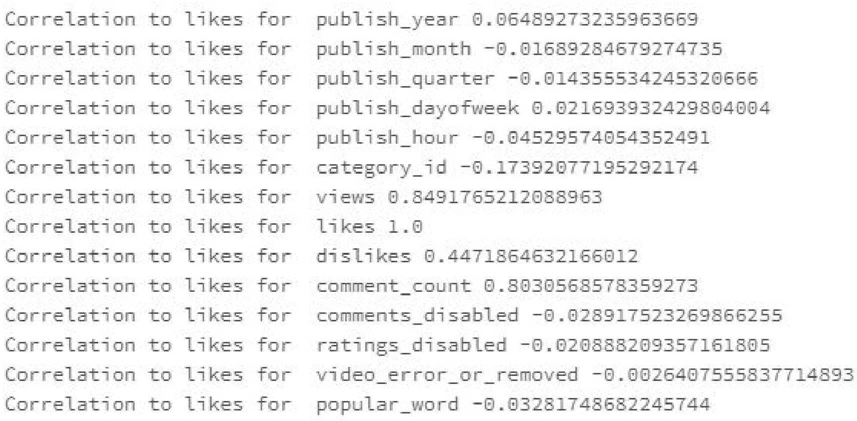
	Frequency
Word	
makeup	725
late	340
cat	316
trailer	285
news	234
show	221
star	219
movie	207
react	200
black	193

Fig.4 Columns created for NLP analysis and the list of the top 10 frequent words

Analysis

Exploratory Data Analysis

It is quite important and useful to understand the relationship between variables. And by computing the correlation between independent variables and target variable “likes”, we can know that as one independent variable changes, how much will the target variable tend to change in specific direction.



```
Correlation to likes for  publish_year 0.06489273235963669
Correlation to likes for  publish_month -0.01689284679274735
Correlation to likes for  publish_quarter -0.014355534245320666
Correlation to likes for  publish_dayofweek 0.021693932429804004
Correlation to likes for  publish_hour -0.04529574054352491
Correlation to likes for  category_id -0.17392077195292174
Correlation to likes for  views 0.849176521208896З
Correlation to likes for  likes 1.0
Correlation to likes for  dislikes 0.4471864632166012
Correlation to likes for  comment_count 0.8030568578359273
Correlation to likes for  comments_disabled -0.028917523269866255
Correlation to likes for  ratings_disabled -0.020888209357161805
Correlation to likes for  video_error_or_removed -0.0026407555837714893
Correlation to likes for  popular_word -0.03281748682245744
```

Fig.5 Correlation from all variables to target variable ‘likes’

The correlation coefficient ranges from –1 to 1. When it is close to 1, it means that there is a strong positive correlation; for example, the number of likes tends to go up when the views and common counts go up. When the coefficient is close to –1, it means that there is a strong negative correlation; the number of likes tends to go down when the videos' category id goes up. Finally, coefficients close to zero mean that there is no linear correlation. The bar chart below represents the distribution of likes corresponding to videos' category id group by publish year. YouTube videos are categorized into different groups, and from the plot is shows that group number 10 and 29 tend to have more likes than the other groups.

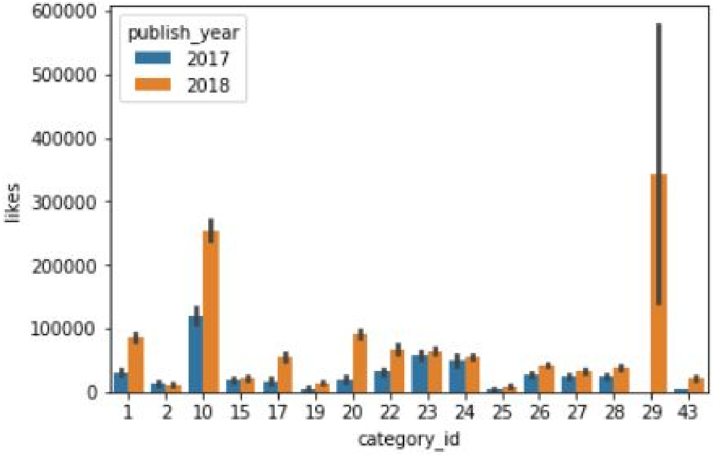


Fig.6 Distribution of category id and likes group by years

We then plotted the distribution of publish_hour, publish_dayofweek, publish_month, publish_quarter, publish_year in bar charts.

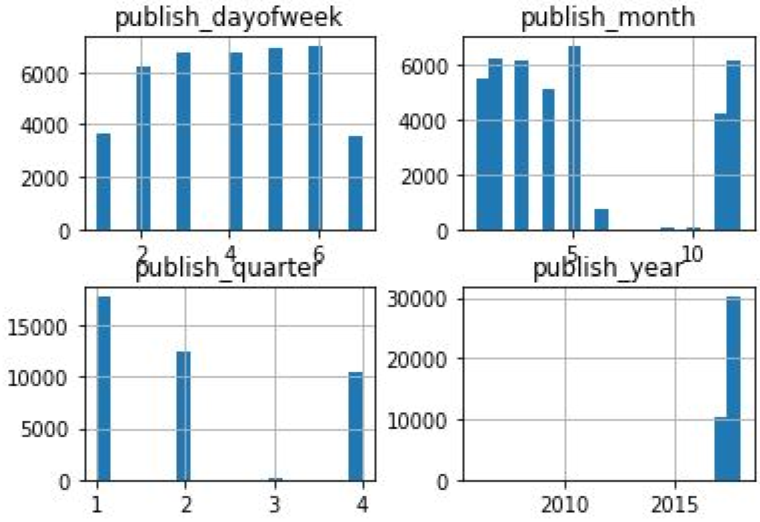


Fig.7 Distribution of dayofweek, month, quarter and year

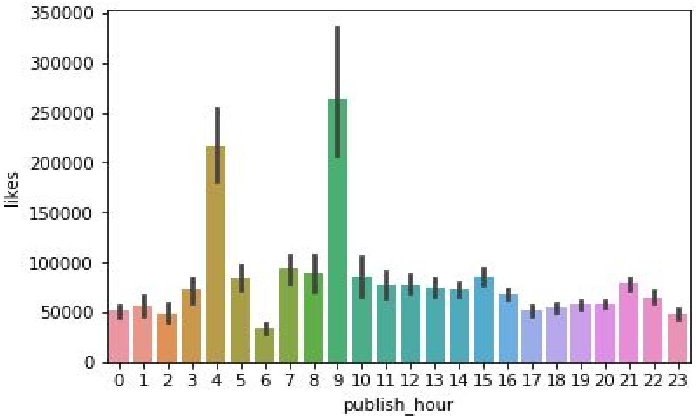


Fig.8 Distribution of publish hours

The published year of the YouTube Videos in 2017 and 2018 will be the primary focus in this study. And it is important to keep in mind that the dataset only captured a very small proportion of video data that are published at quarter 3. And many of the videos are published during Tuesday to Friday of the week and at 4 a.m and 9 a.m of the day.

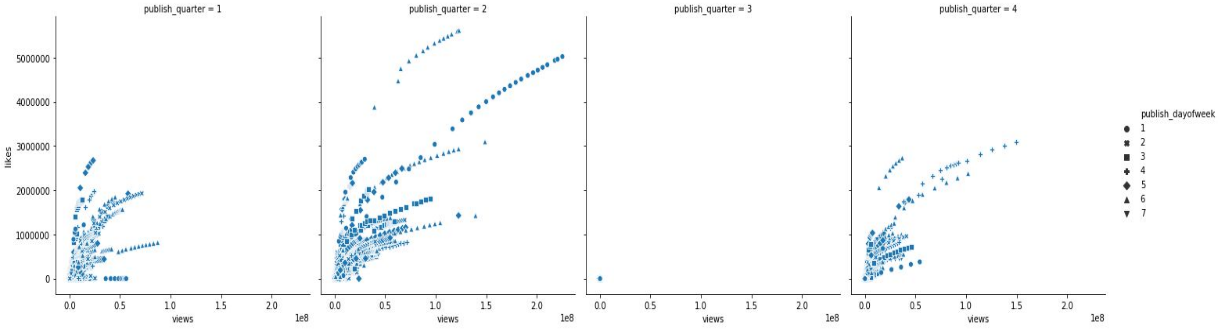
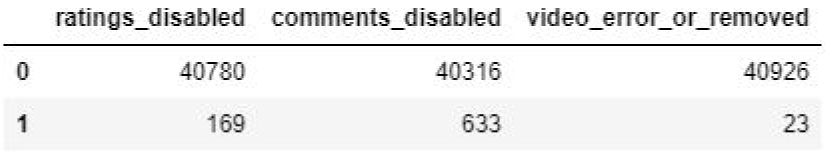


Fig.9 Views vs Likes among dayofweek distributed on quarters

The graph above shows the pattern between “likes” and “views” of dayofweek in different quarters. We can see that quarters, except quarter 3, have a similar pattern. And the graph below shows the numbers of True and False for the column ‘ratings_disabled’, ‘comments_disabled’, and ‘video_error_or_removed’.



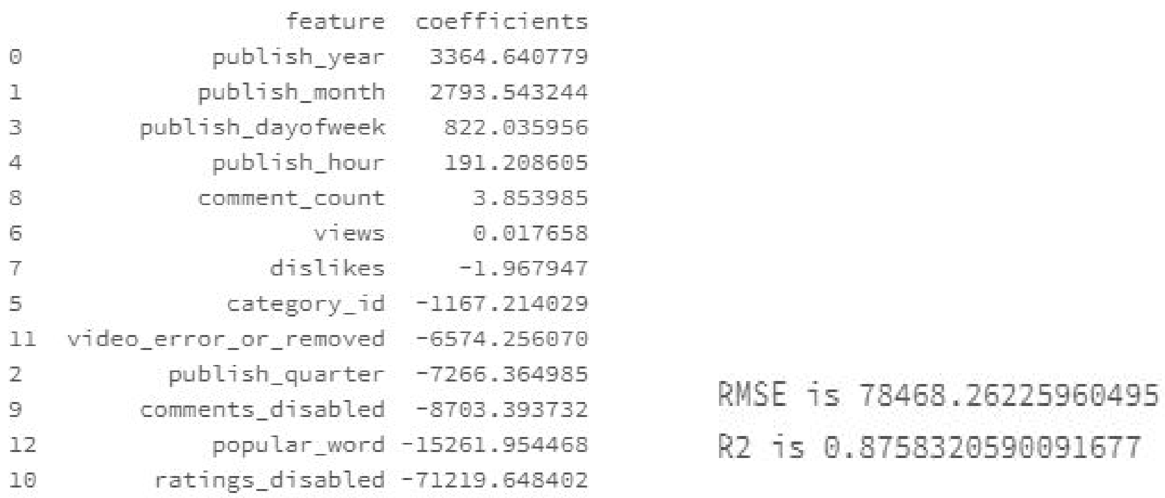
	ratings_disabled	comments_disabled	video_error_or_removed
0	40780	40316	40926
1	169	633	23

Fig.10 Numbers of True and False on these 3 columns

Results

Linear Regression

Linear regression model is one of popular and useful machine learning techniques in projecting results and identifying the strength of effect that the independent variables have on a dependent variable. We first built up the linear regression model and splited the dataset randomly into two datasets, train and test. The train data set contains 80% of the original dataset and the test dataset contains 20%. We set the features column with all the list of all feature variables, label column will be the target variable 'likes'. The parameter of the model will be maxIter=10, regParam=0.3, and elasticNetParam=0.8. We then fitted the model with the train dataset.



```
                      feature   coefficients
0                publish_year    3364.640779
1               publish_month    2793.543244
3           publish_dayofweek     822.035956
4                publish_hour     191.208605
8               comment_count       3.853985
6                       views       0.017658
7                    dislikes      -1.967947
5                 category_id   -1167.214029
11    video_error_or_removed   -6574.256070
2             publish_quarter   -7266.364985
9           comments_disabled   -8703.393732
12               popular_word  -15261.954468
10           ratings_disabled  -71219.648402
```

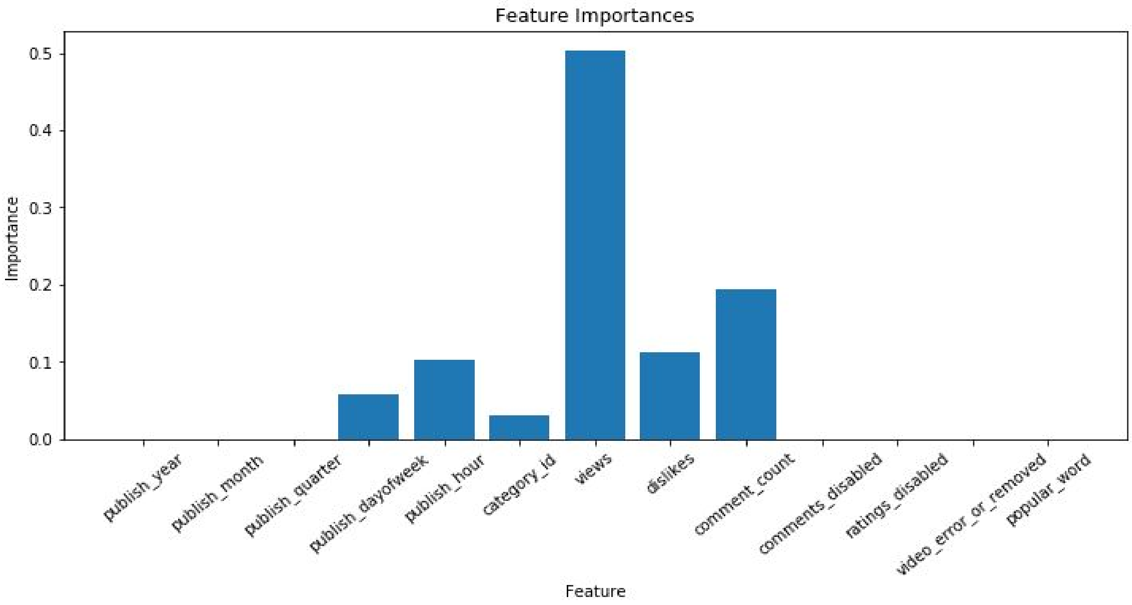
```
RMSE is 78468.26225960495
R2 is 0.8758320590091677
```

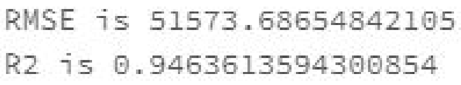
Fig.11 Coefficients, RMSE, and R-squared of linear regression model

The coefficients chart above indicates the top predictors that positively influence the projecting response are publish_year, publish_month, publish_dayofweek, and publish_hour. These predictors have great contributions to the model. The predictor rating_disabled is the one that has the greatest negative influence on the model's response. RMSE measures the differences between predicted values by the model and the actual values. In this case, we used the model to transform the test data set and to make comparison. R squared at 0.88 indicates that in our model, approximate 88% of the variability in "likes" can be explained using the model. This is in alignment with the result from Scikit-Learn.

Decision Tree

The Decision Tree regression model is another powerful machine learning technique that uses a set of binary rules to calculate target values and to make predictions. Each individual tree is a fairly simple model that has branches, nodes and leaves and it generally uses less amount of time to process. Before using the decision tree model to fit with the training data set, we used Spark's ParamGridBuilder to find the optimal hyperparameters in a more systematic approach. We defined the grid of hyperparameters to be maxDepth=[2, 5, 10] and maxBin=[10, 20, 40, 80, 100]. Then we also conducted a 3-fold cross validation to identify the optimal maxDepth and maxBin. Cross Validation is a technique which involves reserving a particular sample of a data set on which we do not train the model, and this is used when the dataset is relatively small. Later, we test our model on this sample before finalizing it.





```
RMSE is 51573.68654842105
R2 is 0.9463613594300854
```

Fig.12 Bar chart of feature importances, RMSE, and R-square of decision tree regression

The model's result indicates that it out-performed the linear regression model by having a smaller Root Mean Square Error and a much higher R-squared. It also means that the decision tree regression model, after hyperparameter tuning, better fits with our dataset and more percentage of the variability in “likes” can be explained. The feature importance bar chart above explains which variables take an influential role in helping the decision tree model to make predictions. Variable views contributed to the model's response the most. Comment count, publish hour, and dislikes are some of the other great influential predictors to the model. The best combination that generated the optimal solution is with maxDepth=10 and maxBin=80.

Conclusion

As we explained above, the main goal of machine learning is to train and generate a productive and efficient computation model. This report aims to showcase the different performances from the two different machine learning models, the linear regression and the decision tree regression. Both of these two techniques are tied to supervised learning, which uses a training dataset to train and fit the model. The linear regression model performed with a worse RMSE of 78468 and lower R-squared of 0.88. And it intercepted variable publish_year is the greatest positive influence on the model and variable ratings_disable is the greatest negative influential predictor. However, it can potentially imply that the model is underfitting; it cannot detect the interaction between variables, and also cannot capture the non-linear relationship. It ultimately leads to a high RMSE and it ends up with lack of capacity to make accurate predictions with both training data set and new data.

On the other hand, the decision tree regression model is a better predictive model that outperforms the linear regression model. Decision tree regressor has the advantage of performing a comprehensive analysis on the consequences of each possible decision and it also allows for partitioning the train data set in different levels. By combining with the use of paraGrid and cross validation, it offers us the option to set the range of the parameters and to choose the optional combination from it. The fact that the decision tree model makes explicit all possible alternatives and traces each alternative to its conclusion in a single view leads to a better performance of RMSE and R-squared.