Members:
Haomin Hu
Zhenmin Hua

DSCI 551 Project: Evaluation and Managing ML data and models for TikTok Trending Videos Estimation

1. INTRODUCTION

We are probably all familiar with TikTok. People tend to spend hours each day scrolling through the millions of videos that are uploaded every single day. Not to mention the uploaders who are giving anything to get as many likes and followers as possible. But what makes one TikTok video become trending or dismissed from the recommendation pool? Is it generated when a Tiktok video is uploaded, or the number of followers, thumbs up, and hashtags may cause a huge impact?

Our project is to discover the characteristics of the first 1000 trending videos scraped from Tiktok and random 8000 videos from Tiktok that aren't in trending videos, clustering the similarities of them and analyzing which attributes of the videos may upgrade the video to be popular. After that, we try to analyze the algorithm and set up rules to distinguish whether a video can be on the trending board(with the percentage of probability and time) and predict the thumb up rates by all other attributes.

Besides, users may use this application to analyze the e-commerce advertising spreading rate based on the figures of short videos, users would just need to offer their video url with necessary attributes. In addition, the media and KOL companies can use this tool to track their accounts on Tiktok, understanding how to produce trending videos.

2. DATA DESCRIPTION

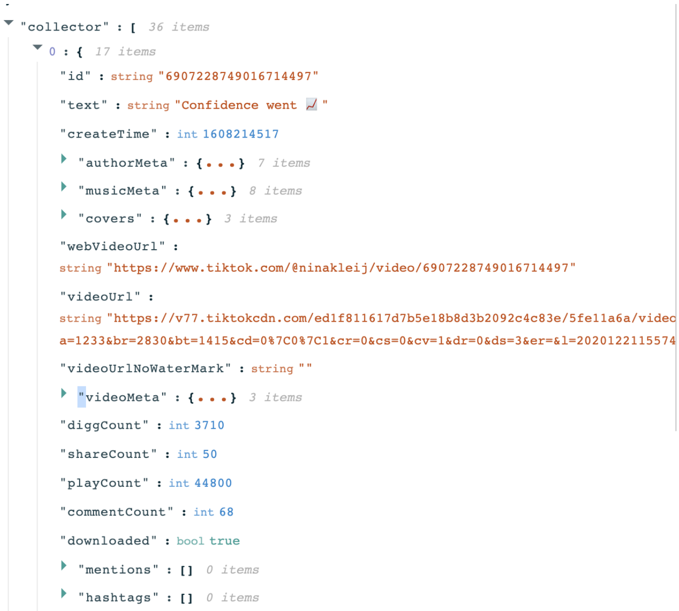
2.1 Data Collection

The dataset of TikTok Videos(trending videos scraped from TikTok) was scraped down from an unofficial Tiktok web-scraper. Notable, the current trending videos are generated based on the

account information that we provided. However, the fact is a certain type of people may receive the same type of videos recommendation and preference.
It is available to be downloaded on kaggle website in case needed:
https://www.kaggle.com/erikvdven/tiktok-trending-december-2020?select=trending.json.

2.2 Data Overview

The dataset contains 9693 rows of data entries, with 37 attributes(dimensions) for all entries. Attributes in this dataset contain basic information such as id, text, createTime, authorMeta, shareCount, videoUrl, and etc. As well as non-text data such as videos (mp4 format) and background music(transformed into binary files already). Besides, there is unfiltered music genre information, such as genre names, release date, etc.

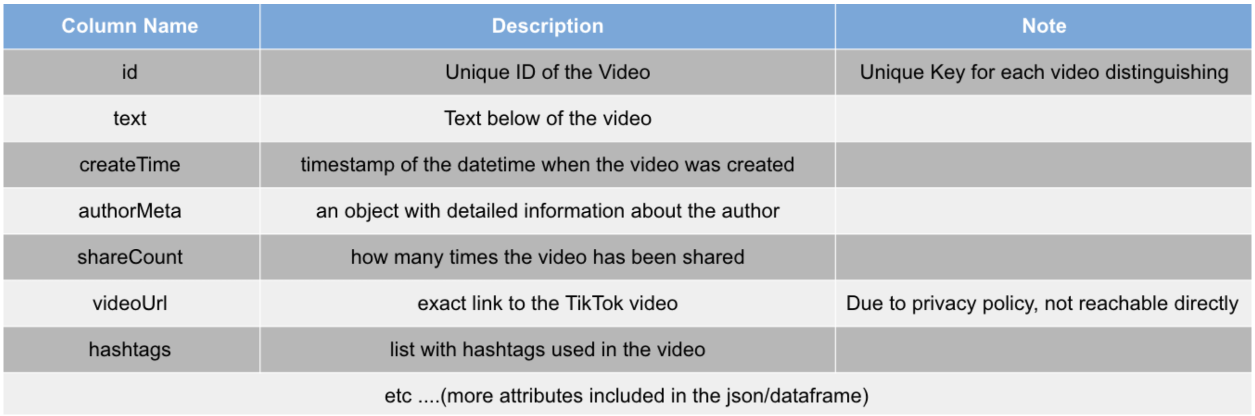


```
"collector" : [  36 items
  0 : {  17 items
    "id" : string "6907228749016714497"
    "text" : string "Confidence went 📈 "
    "createTime" : int 1608214517
    "authorMeta" : {...}  7 items
    "musicMeta" : {...}  8 items
    "covers" : {...}  3 items
    "webVideoUrl" :
    string "https://www.tiktok.com/@ninakleij/video/6907228749016714497"
    "videoUrl" :
    string "https://v77.tiktokcdn.com/ed1f811617d7b5e18b8d3b2092c4c83e/5fe11a6a/videc
    a=1233&br=2830&bt=1415&cd=0%7C0%7C1&cr=0&cs=0&cv=1&dr=0&ds=3&er=&l=2020122115574
    "videoUrlNoWaterMark" : string ""
    "videoMeta" : {...}  3 items
    "diggCount" : int 3710
    "shareCount" : int 50
    "playCount" : int 44800
    "commentCount" : int 68
    "downloaded" : bool true
    "mentions" : []  0 items
    "hashtags" : []  0 items
```

Figure 2.1 Json file Overview[1]

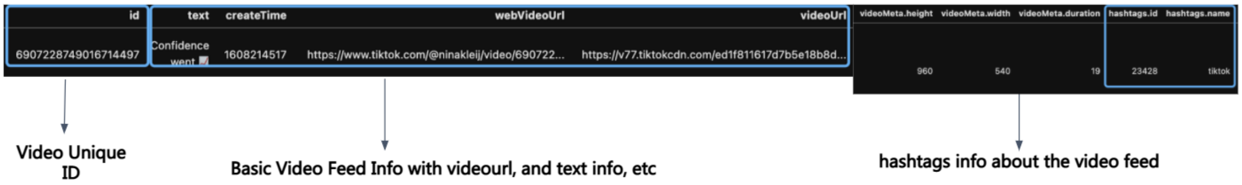
The data in the attributes of choices/selections will be the foundation of the analysis, as they will be the features of comparison. Due to the purpose of this dataset we set is to analyze the final thumbs up number and the relationship of becoming trending, so the attributes of playCount, shareCount and CommentCount will be the feature to be used to estimate watch count, which connected tightly to determine videos in recommendation pool or not. Regarding the open-ended information and non-binary inputs, the sentence formatted answers and videos need to be preprocessed into an analyzable format using Word2Vec and feature extraction, which will be explained in 2.4.

[1] Refers to the dataset of TikTok Videos(trending videos scraped from TikTok)



Column Name	Description	Note
id	Unique ID of the Video	Unique Key for each video distinguishing
text	Text below of the video	
createTime	timestamp of the datetime when the video was created	
authorMeta	an object with detailed information about the author	
shareCount	how many times the video has been shared	
videoUrl	exact link to the TikTok video	Due to privacy policy, not reachable directly
hashtags	list with hashtags used in the video	
etc ....(more attributes included in the json/dataframe)		

Figure 2.2 Description of attribute from JSON files



id	text	createTime	webVideoUrl	videoUrl	videoMeta.height	videoMeta.width	videoMeta.duration	hashtags.id	hashtags.name
6907228749016714497	Confidence went 📈	1608214517	https://www.tiktok.com/@ninakleij/video/690722...	https://v77.tiktokcdn.com/ed1f811617d7b5e18b8d...	960	540	19	23428	tiktok

Video Unique ID
Basic Video Feed Info with videourl, and text info, etc
hashtags info about the video feed

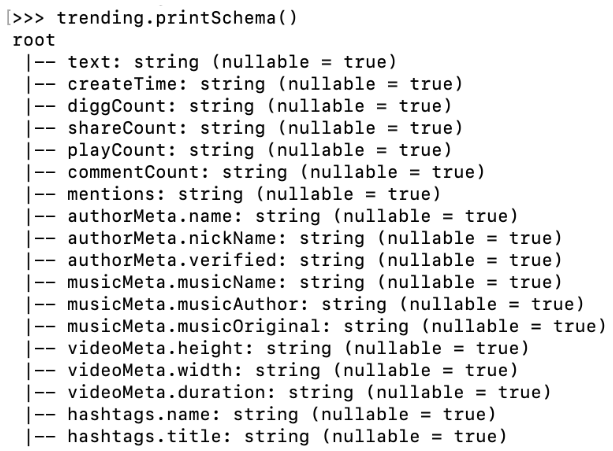
Figure 2.3 Description of attribute from JSON files[2]

2.3 Data Cleaning

2.3.1 Missing values:

Many entries of video metadata are left blank, which would be considered missing values. However, by digging into the attributes configuration, we may have a clear picture that the dataset can be null and id doesn't affect the accuracy in describing the dataset. As a result, we decide to use all the attributes directly and eliminate the input where lack of 3 pivotal attributes, after checking the pivotal attributes previously mentioned in Chapter 2.2.

[2] At this time, we have transformed the JSON file into dataframe using Python pandas



```
[>>> trending.printSchema()
root
 |-- text: string (nullable = true)
 |-- createTime: string (nullable = true)
 |-- diggCount: string (nullable = true)
 |-- shareCount: string (nullable = true)
 |-- playCount: string (nullable = true)
 |-- commentCount: string (nullable = true)
 |-- mentions: string (nullable = true)
 |-- authorMeta.name: string (nullable = true)
 |-- authorMeta.nickName: string (nullable = true)
 |-- authorMeta.verified: string (nullable = true)
 |-- musicMeta.musicName: string (nullable = true)
 |-- musicMeta.musicAuthor: string (nullable = true)
 |-- musicMeta.musicOriginal: string (nullable = true)
 |-- videoMeta.height: string (nullable = true)
 |-- videoMeta.width: string (nullable = true)
 |-- videoMeta.duration: string (nullable = true)
 |-- hashtags.name: string (nullable = true)
 |-- hashtags.title: string (nullable = true)
```

Figure 2.4 Checking about the attributes characteristics using Spark

2.3.2 Unusual values or Outliers:

No noise or outliers for the sentence attribute.
In description, there are no outliers or noise, as all description and video configuration (non-binary info) are taken in consideration of the feature/label extraction process.
The noises and outliers may be generated after the Word2Vec step, as the words being mentioned such as "the" or "a" will become noise. However, the data in short videos hardly demonstrates these kinds of words.

2.3.3 Duplicates: None of the entries(especially unique ID of videos)are duplicated, since they are reflecting the videos uniquely and solely.

2.4 Data Preprocessing

2.4.1 Word2Vec

The Word2vec is an algorithm that uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.[3]

[3] Reference 1."Gensim: Topic Modelling for Humans." *Models.word2vec – Word2vec Embeddings - Gensim*, 30 Aug. 2021, https://radimrehurek.com/gensim/models/word2vec.html.

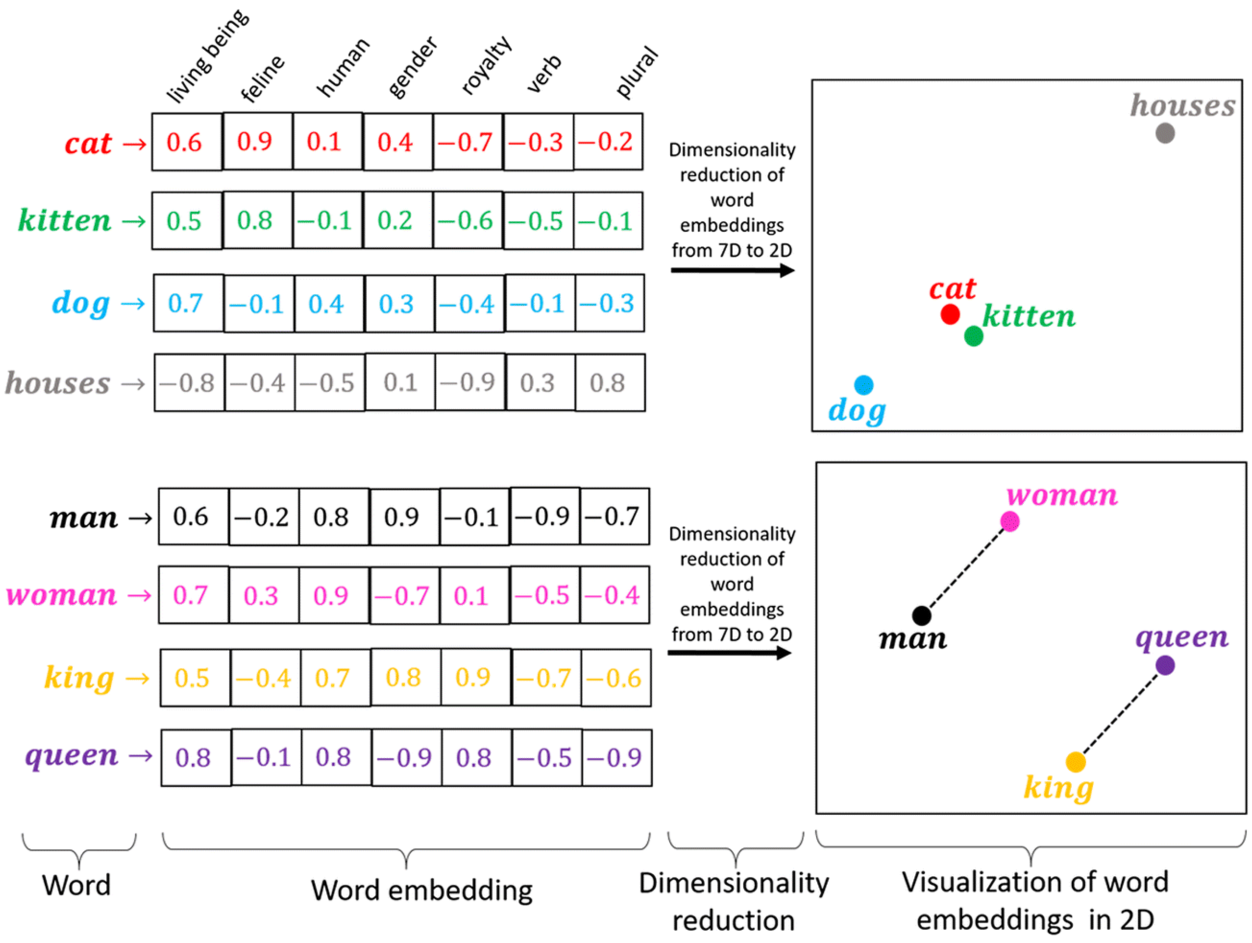
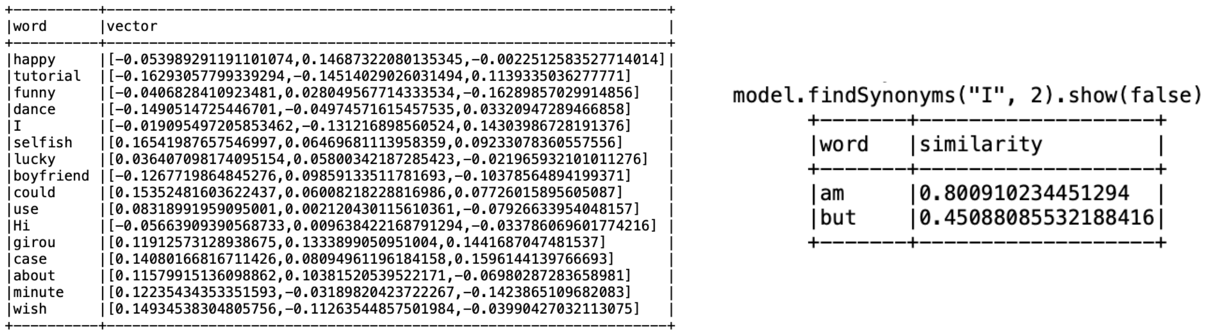

living being
feline
human
gender
royalty
verb
plural
cat → 0.6 0.9 0.1 0.4 −0.7 −0.3 −0.2
kitten → 0.5 0.8 −0.1 0.2 −0.6 −0.5 −0.1
dog → 0.7 −0.1 0.4 0.3 −0.4 −0.1 −0.3
houses → −0.8 −0.4 −0.5 0.1 −0.9 0.3 0.8
Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
man → 0.6 −0.2 0.8 0.9 −0.1 −0.9 −0.7
woman → 0.7 0.3 0.9 −0.7 0.1 −0.5 −0.4
king → 0.5 −0.4 0.7 0.8 0.9 −0.7 −0.6
queen → 0.8 −0.1 0.8 −0.9 0.8 −0.5 −0.9
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Figure 2.4 Visualization of how Word2vec works[4]



```
+----------+------------------------------------------------------------------+
|word      |vector                                                            |
+----------+------------------------------------------------------------------+
|happy     |[-0.053989291191101074,0.14687322080135345,-0.0022512583527714014]|
|tutorial  |[-0.16293057799339294,-0.14514029026031494,0.1139335036277771]    |
|funny     |[-0.0406828410923481,0.028049567714333534,-0.16289857029914856]   |
|dance     |[-0.1490514725446701,-0.04974571615457535,0.03320947289466858]    |
|I         |[-0.019095497205853462,-0.131216898560524,0.14303986728191376]    |
|selfish   |[0.16541987657546997,0.06469681113958359,0.09233078360557556]     |
|lucky     |[0.036407098174095154,0.05800342187285423,-0.021965932101011276]  |
|boyfriend |[-0.1267719864845276,0.09859133511781693,-0.10378564894199371]    |
|could     |[0.15352481603622437,0.06008218228816986,0.07726015895605087]     |
|use       |[0.08318991959095001,0.002120430115610361,-0.07926633954048157]   |
|Hi        |[-0.05663909390568733,0.009638422168791294,-0.033786069601774216] |
|girou     |[0.11912573128938675,0.1333899050951004,0.1441687047481537]       |
|case      |[0.14080166816711426,0.08094961196184158,0.1596144139766693]      |
|about     |[0.11579915136098862,0.10381520539522171,-0.06980287283658981]    |
|minute    |[0.12235434353351593,-0.03189820423722267,-0.1423865109682083]    |
|wish      |[0.14934538304805756,-0.11263544857501984,-0.03990427032113075]   |
+----------+------------------------------------------------------------------+
```

```
model.findSynonyms("I", 2).show(false)
+--------+-------------------+
|word    |similarity         |
+--------+-------------------+
|am      |0.800910234451294  |
|but     |0.45088085532188416|
+--------+-------------------+
```

Figure 2.5 Using Spark to extract words pop up most, giving them vector statistic attributes; Showing the similarity rate with "I" by descending order

In this part we used Spark to finish data preprocessing(attribute extraction) on the "description" part. The reason that we are using Spark is based on the handy operation and quick output speed. And through that we have a clear big picture of how the trending videos are and facilitate the next step: adapting XGboost towards our final goal on result prediction.

[4] Reference 2. Gautam, Hariom. "Word Embedding." *Medium*, Medium, 1 Mar. 2020, https://medium.com/@hari4om/word-embedding-d816f643140.

3. EXPLORATORY ANALYSIS

3.1 Data Visualization

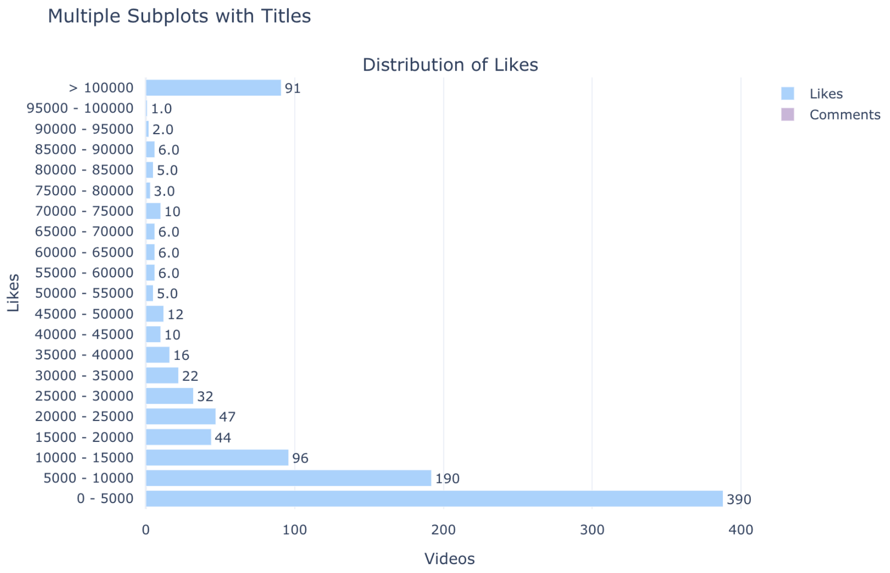


Figure 3.1: Bar Chart of the relationship between videos in count and likes(thumbs up)

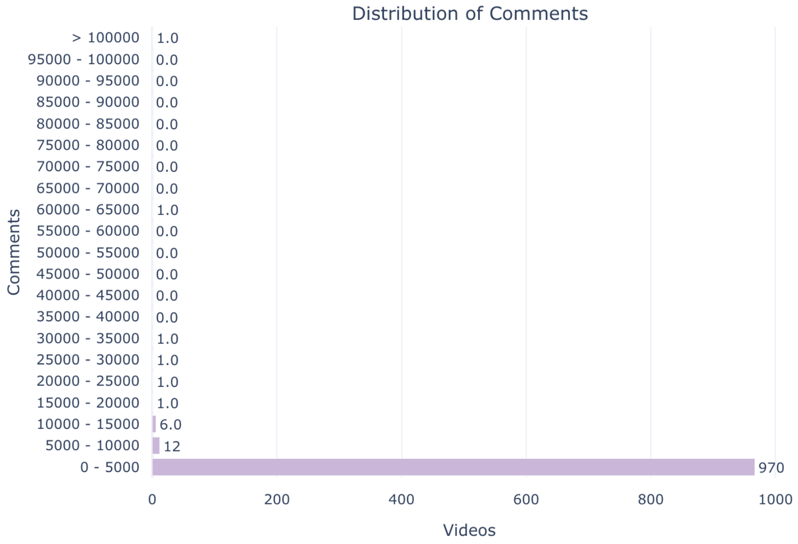


Figure 3.2: Bar Chart of the relationship between videos in count and comments

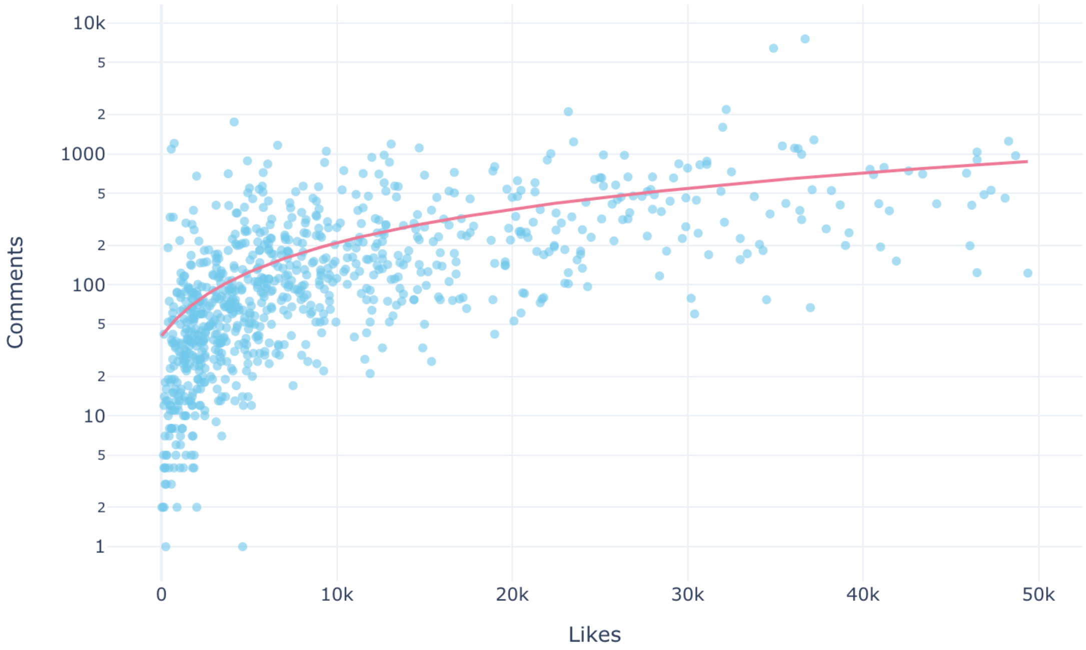


Figure 3.3: Pie Chart for Race distribution of all tiktok videos in the dataset

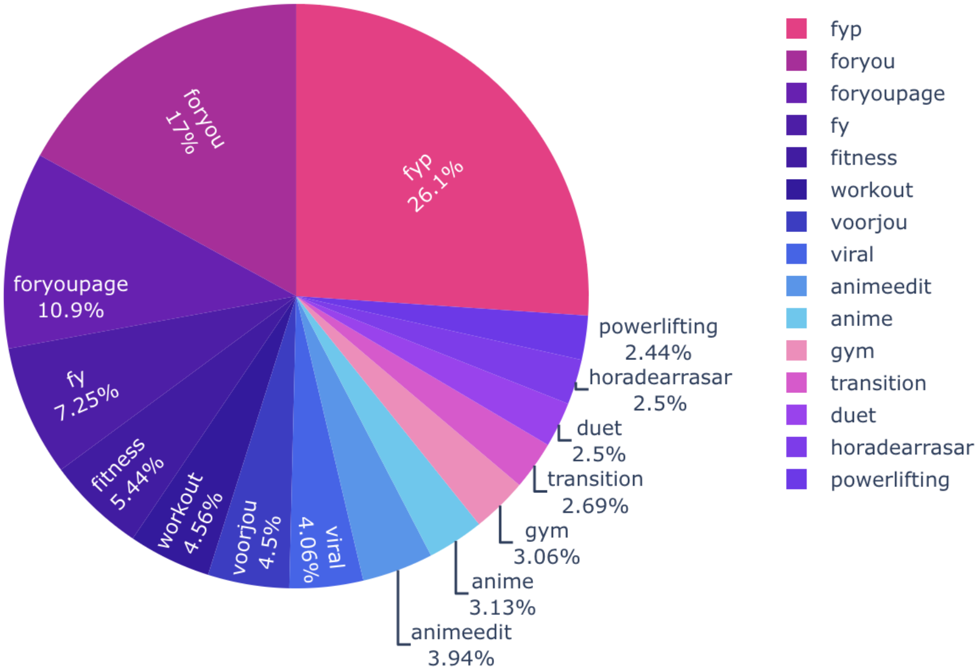


Figure 3.4: Pie Chart for tag distribution of all tik tok videos

Figure 3.1 & 3.2 plots the bar charts of all tiktok videos in our dataset based on likes and comments. From the pie chart, it is clear that most of the videos are on the road of becoming trending. Nearly 60% of all videos are below or only slightly above the 15,000 likes. At the same time, only about 1% of all these videos are receiving 100,000 likes. From here, we acknowledge that the composition of trending video pools comes from two parts: top rated video across all platforms and new emerging videos that are starting to gain likes. At the same time, almost all the comments are gathered between the 0 - 5000 category, showing the characteristics that both trending and non-trending videos share the same characteristics.

In the Figure 3.3 plot, we use log-scale for the y-axis, to allow a large range to be displayed without the small values being compressed down into the bottom of the graph. The R^2 value indicates that the independent variable (likes) is not explaining much in the variation of the dependent variable (comments). And we receive interesting results based following:

- Most videos gathering under 20,000 likes
- Videos with a low amount of likes can still be high in comments (interactive rate is high)
- There are two interesting outliers around 35,000 likes which receive lots of comments.

Figure 3.4 shows a significant imbalance distribution of tags among all the videos. More than 50% of the videos are with the tags “for you page”. It is still not clear why this is split into four different topics. As far as we are concerned for now, the For You page is the front page of TikTok, the landing place when users open the app. The videos on this page are populated by a recommendation engine that's a black box to anyone outside the company. This provides a guideline how often a trending video would pop up during the daily usage, and based on the estimation rate from above can we start with the estimation.

4. MODEL DEVELOPMENT

4.1 What is XGBoost

XGBoost is an open-source software library which provides a regularizing gradient boosting framework for C++, Java, Python, R, Julia, Perl, and Scala. It works on Linux, Windows, and macOS. From the project description, it aims to provide a "Scalable, Portable and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library". It runs on a single machine, as well as the distributed processing frameworks Apache Hadoop, Apache Spark, Apache Flink, and Dask. It has gained much popularity and attention recently as the algorithm of choice for many winning teams of machine learning competitions.

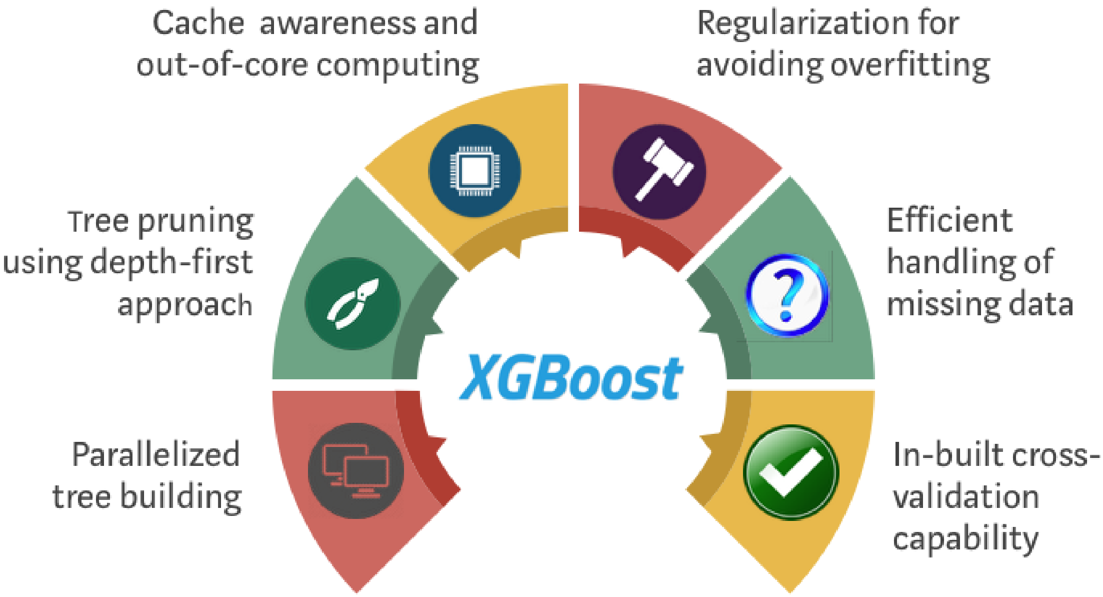

Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Figure 4.1: Abstract process of how XGBoost works

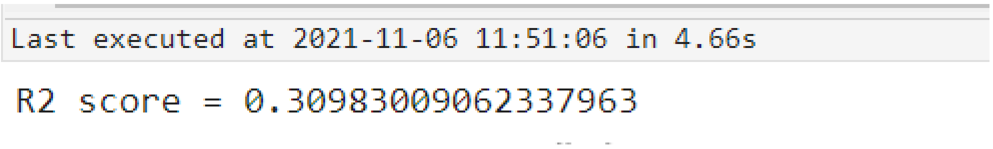
4.2 The goals of this project

- **Primary:** To predict the digCount(number of people who like the video) of a video, which could help to determine whether a future video can become trending or not. The media channels and KOL companies can use this tool to track their accounts on Tiktok and estimate newly published videos.
- **Secondary:** Finding out the connections of each trending videos' attributes.
  The algorithm will find out whether a video would be tagged as trending from the moment when a video is uploaded or perhaps the amount of followers and other attributes are more important factors
- **In future:** Understanding how to produce trending videos and using them for advertising & e-commence. This tool may provide a guide book in third party analysis and people are interested in the recommendation system pool.

5. PERFORMANCE AND RESULTS

5.1 Performance

We use the XGBoost machine learning algorithm to predict the digCount of a certain video. According to the feature extraction results, five features are extracted and put into the model. After fitting the model to our dataset, the prediction R2 score is about 0.31 as the diagram shows below.



```
Last executed at 2021-11-06 11:51:06 in 4.66s

R2 score = 0.30983009062337963
```

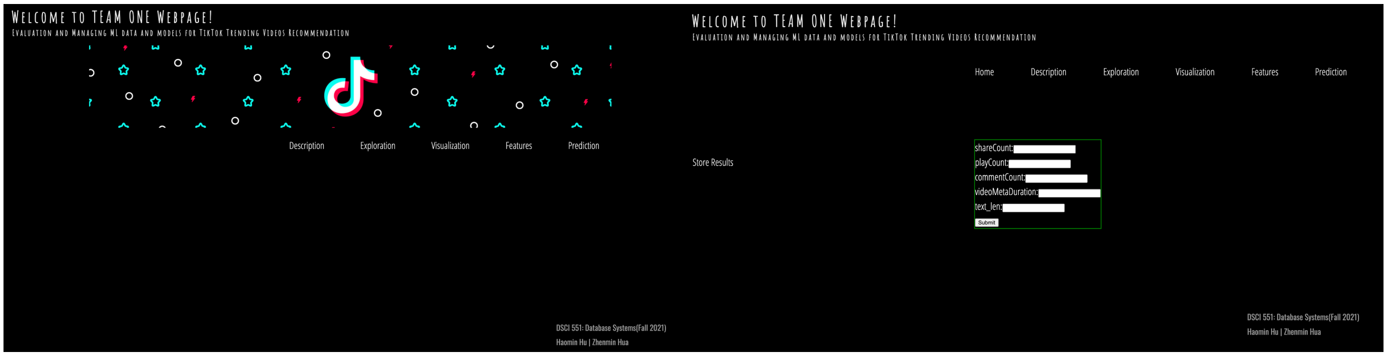
5.2 Result Analysis

Since the scale of our dataset is not quite large and there is no model comparison analysis about the dataset, the result of our model prediction still needs optimization. In the future, we could focus more on implementing better machine learning models on our dataset. Model comparison analysis and model parameters optimization are required for it as the following plan in the future.

6. CONCLUSION, FUTURE WORK AND WEBSITE DESIGN

This paper and analysis employ the data from the dataset of TikTok Videos(all trending and non-videos scraped from TikTok) , with the two main goals. The first goal is to predict the digCount(number of people who like the video) of a video, which could help to determine whether a future video can become trending or not. The media channels and KOL companies can use this tool to track their accounts on Tiktok and estimate newly published videos. The second goal is to find out the connections of each trending videos' attributes.

As shown in figure 6.1 & 6.2, we have developed a website that facilitates the user in using our project with UI/UX design. Not only can the users explore the metadata with visualized ways but also they can fill in the necessary information of the video and make predictions on their video likes amount. Also, we understand that the current prediction result is in low precision. This mainly resulted from the lack of inputdata. With more data in the future system, combining with the current data, the accuracy will be exponentially improved and provide useful trending prediction.



WELCOME TO TEAM ONE WEBPAGE!

EVALUATION AND MANAGING ML DATA AND MODELS FOR TIKTOK TRENDING VIDEOS RECOMMENDATION

DSCI 551: Database Systems(Fall 2021)

Haomin Hu | Zhenmin Hua

WELCOME TO TEAM ONE WEBPAGE!

EVALUATION AND MANAGING ML DATA AND MODELS FOR TIKTOK TRENDING VIDEOS RECOMMENDATION

DSCI 551: Database Systems(Fall 2021)

Haomin Hu | Zhenmin Hua

Figure 6.1 & 6.2 Parts of the screenshots on our Webpage

Acknowledgements

For this project, we greatly appreciate Professor Wensheng Wu, Teaching Assistant Dexuan Luo & Yichao Zhu, and all classmates in DSCI-551 class, Tuesday session for helpful suggestions and feedback. Thanks again for the wonderful course experience in this fall semester.