

# Rakuten – Viki Challenge

Le Nguyen The Dat

# About me

- 2010: MSc. Computer Science – Oxford University
- 2011: Research Engineer – A\*STAR DSI
- 2013: Data – ZALORA Group
- 2015: Data – Commercialize TV



<https://github.com/lenguyenthedat>




<https://sg.linkedin.com/in/lenguyenthedat>



# Challenge descriptions

<https://www.viki.com/>

1,000,000,000



GLOBAL TV TV GUIDE EXPLORE


COMMUNITY MORE SIGN IN

CHINA

ON AIR

Love Me if You Dare

A brilliant criminal psychologist solves the most mysterious and violent crimes with the help of his observant young assistant.


 Channel Manager

8


Languages


Play Now ▶


Check out our Viki mobile and TV apps!


 Like

5.2m











Popular Shows

See All <>




The Glamorous Imperial Con...

EN 0% • China, Romance




Mischievous Kiss: Love in T...

EN 100% • Japan, Romance




I Order You

EN 100% • Korea, Idol Drama



Legend of Lu Zhen

EN 100% • China, Historical



Down With Love

EN 100% • Taiwan, Idol Drama

# Challenge descriptions

<http://www.dextra.sg/challenges/rakuten-viki-video-challenge/>

RAKUTEN-VIKI GLOBAL TV CHALLENGE

Overview

About The Host

Data & Resources

FAQ

GO TO CHALLENGE

## Rakuten Viki - global TV recommender challenge



challenge host:



organizer:



supported by:



**BUILD A MODEL TO RECOMMEND TV DRAMA EPISODES TO VIEWERS.**

A Data Challenge hosted by Rakuten Institute of Technology and Rakuten-Viki based on the online TV viewing data

GO TO CHALLENGE

DOWNLOAD CASE STUDY

# Challenge descriptions

## Data:

- (880,000) User Attributes (*country – gender*)
- (600) Video attributes (*country – language – genre – owner – casts*)
- (4,880,000) User viewing behavior (*video – user – score*)

## Task:

- Recommendation engine - prediction for each user  
(*user – top 3 videos*)
- Insights

## Case study:

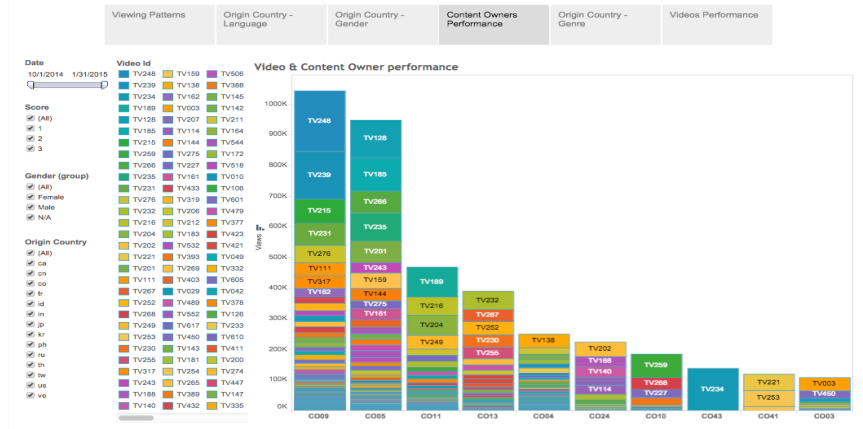
- [http://www.dextra.sg/wp-content/uploads/2015/09/CaseStudy\\_Viki.pdf](http://www.dextra.sg/wp-content/uploads/2015/09/CaseStudy_Viki.pdf)

# Useful Links

## Tableau Public Visualization:

<http://tiny.cc/viki-viz>

Dextra Rakuten-Viki Data Science Challenge 2015



## Source Code:

<http://tiny.cc/viki-src>



Unwatch 1

My solution for Dextra Data Science Challenge #43 (Rakuten/Viki) <https://challenges.dextra.sg/challenge/43> — Edit

51 commits


1 branch


0 releases

 **1 contributor**

Branch: master dextra-viki-2015 / +

Updated README file.

 lenguyenthedat authored a day ago

latest commit 6e0f4a7b92 


 failed\_attempts Finalize. 7 days ago

 [.gitignore](#) [Attempted traditional scikit-learn ML...](#) a month ago

 README.md Updated README file. a day ago

 requirements.txt Add hotness as a similar scoring method. a month ago

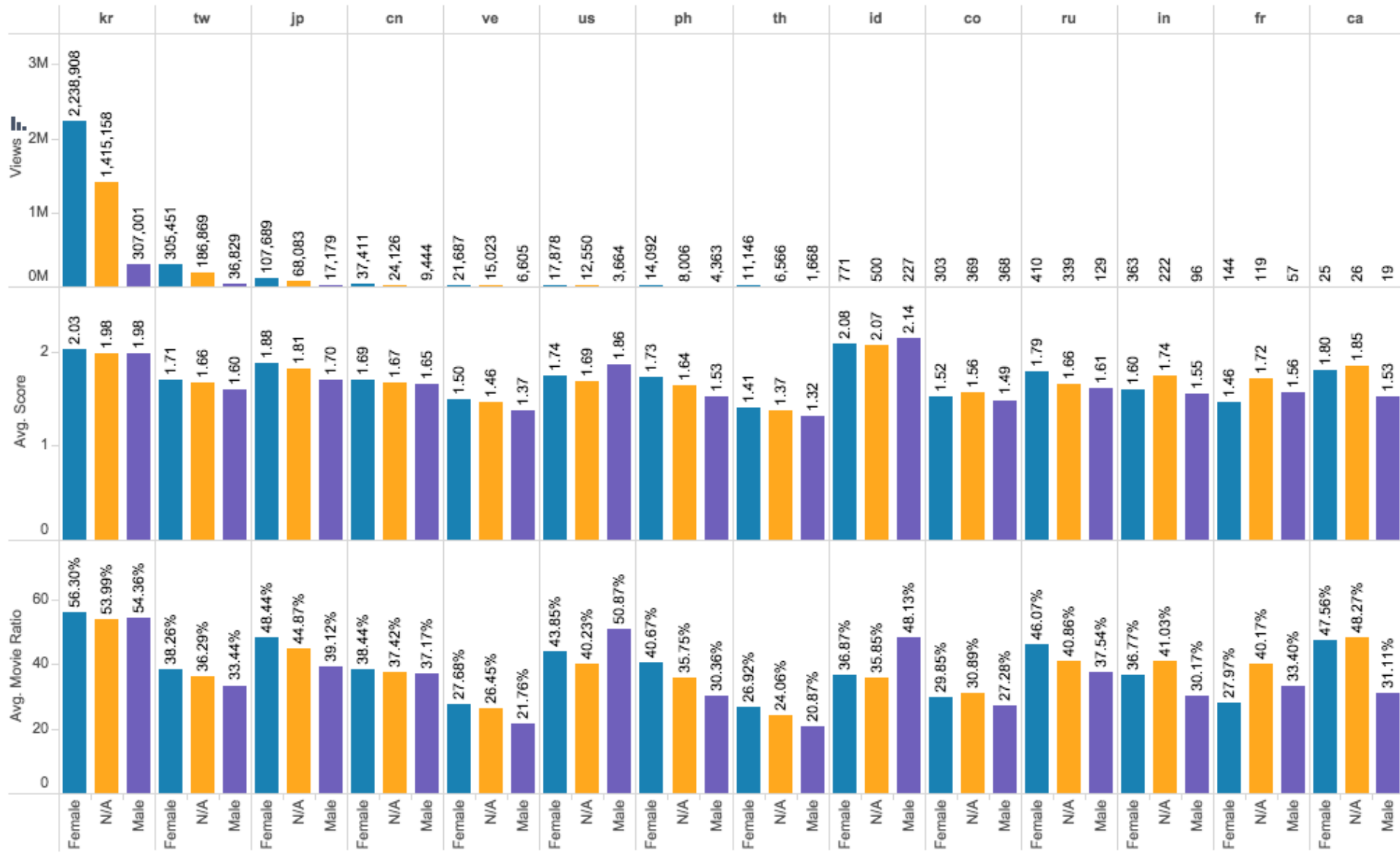
 [viki-users-recommender.py](#) Added cosine similarity - 144 0.21493 11 days ago

 [viki-videos-similarity.py](#) Added cosine similarity - 144 0.21493 11 days ago

# Preliminary Analysis

# Analysis – Gender

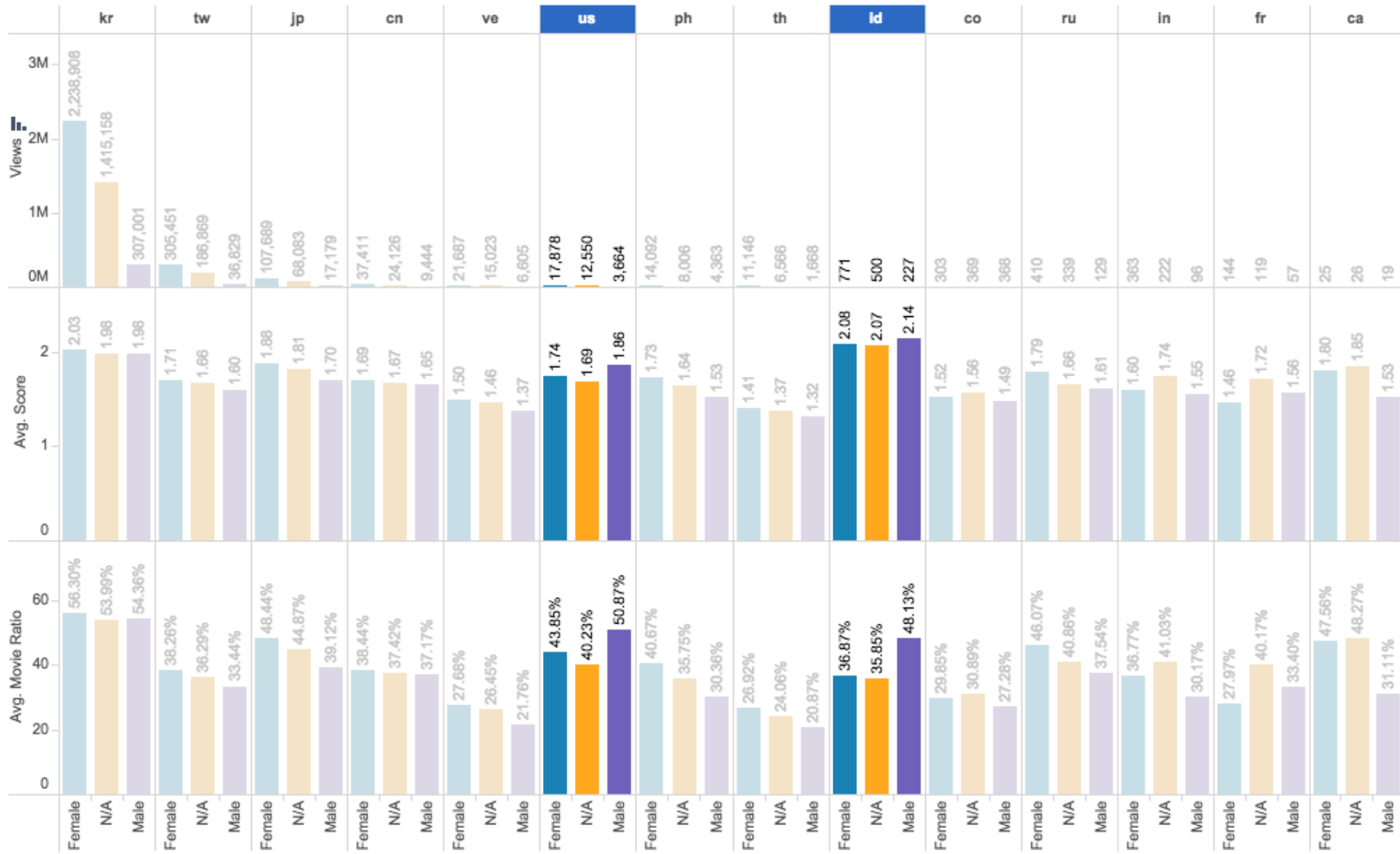
Video's Origin Country and Viewer's Gender





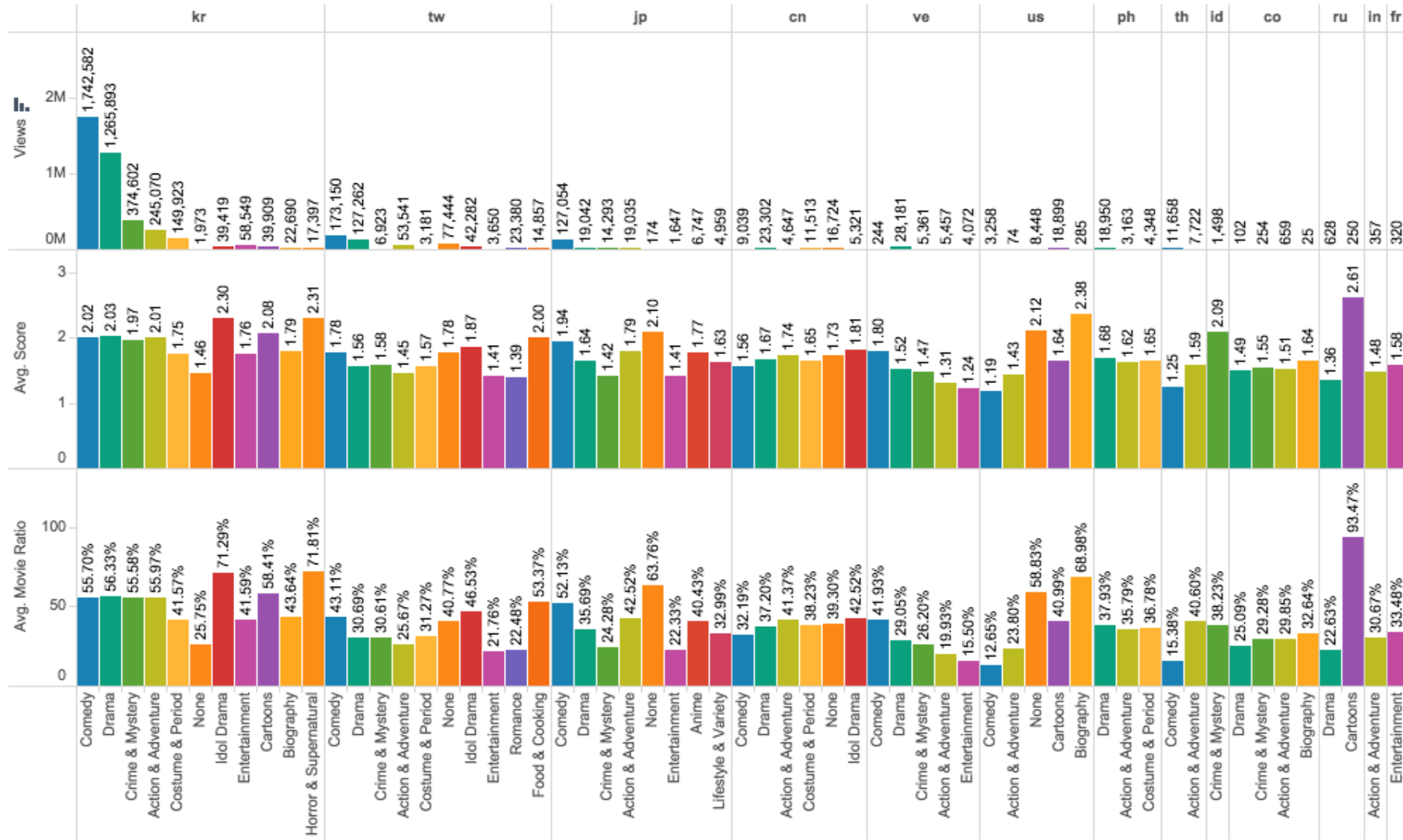
# Analysis – Gender

### Video's Origin Country and Viewer's Gender



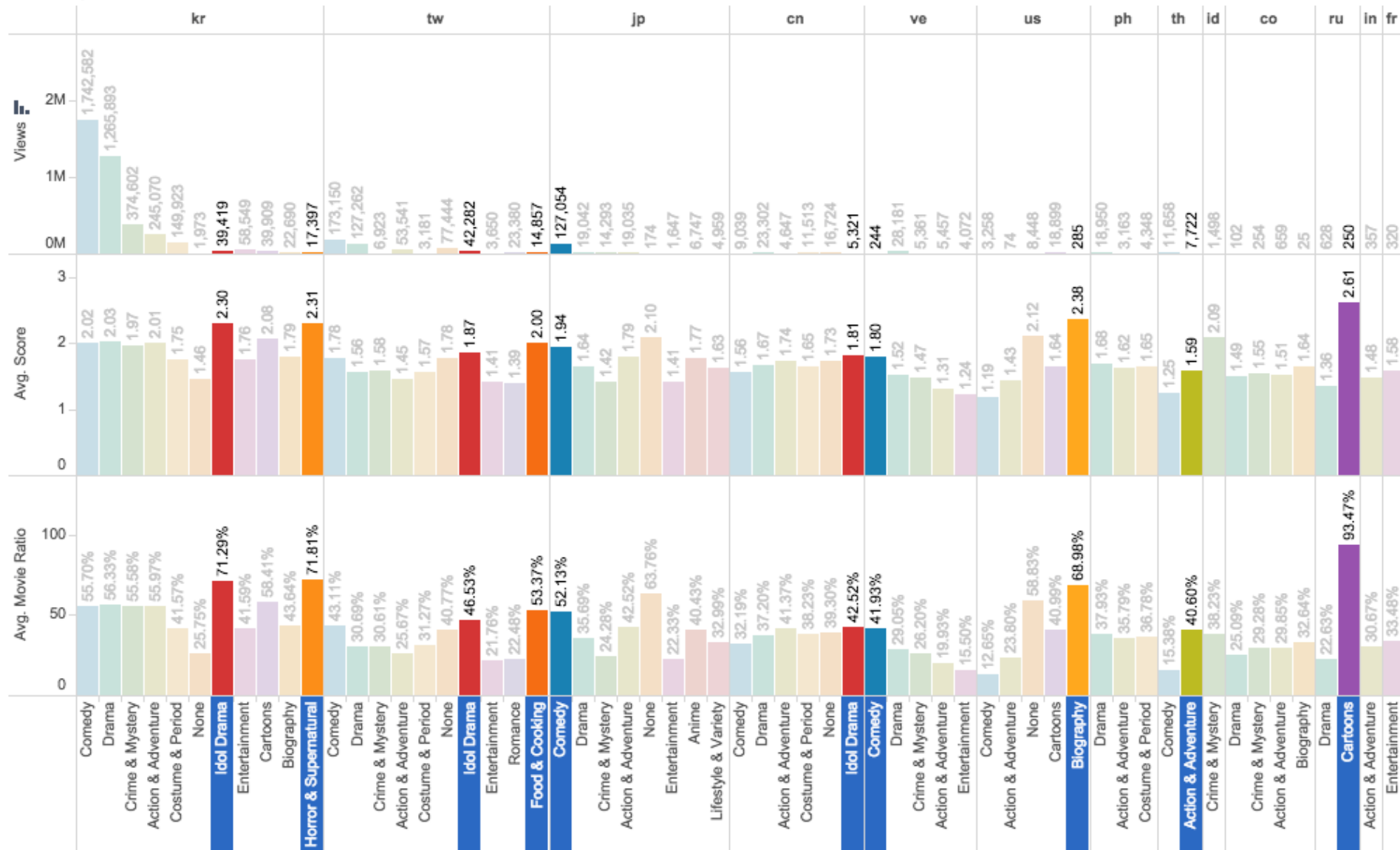
# Analysis – Genre

Country - Genre



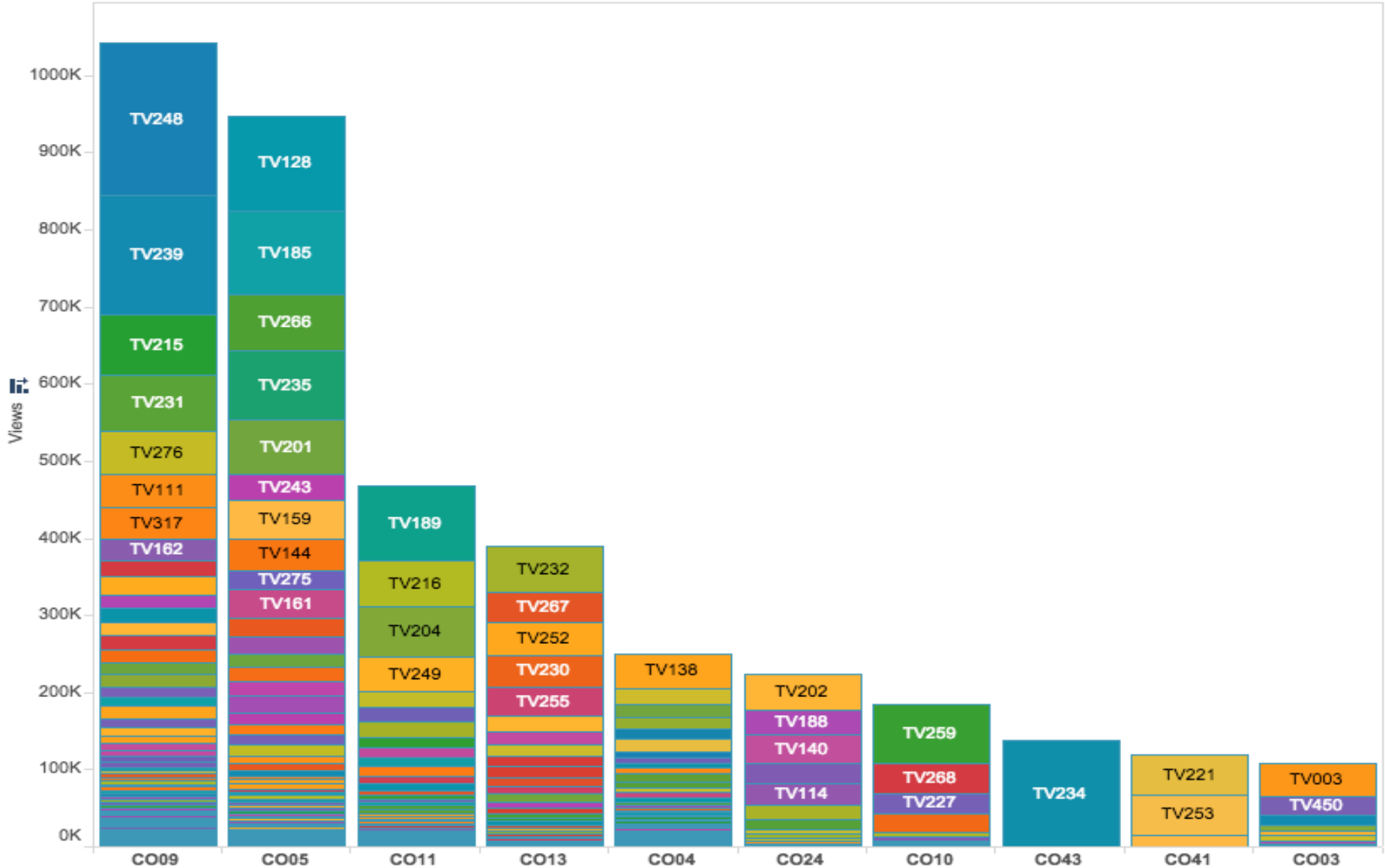
# Analysis – Genre

Country - Genre



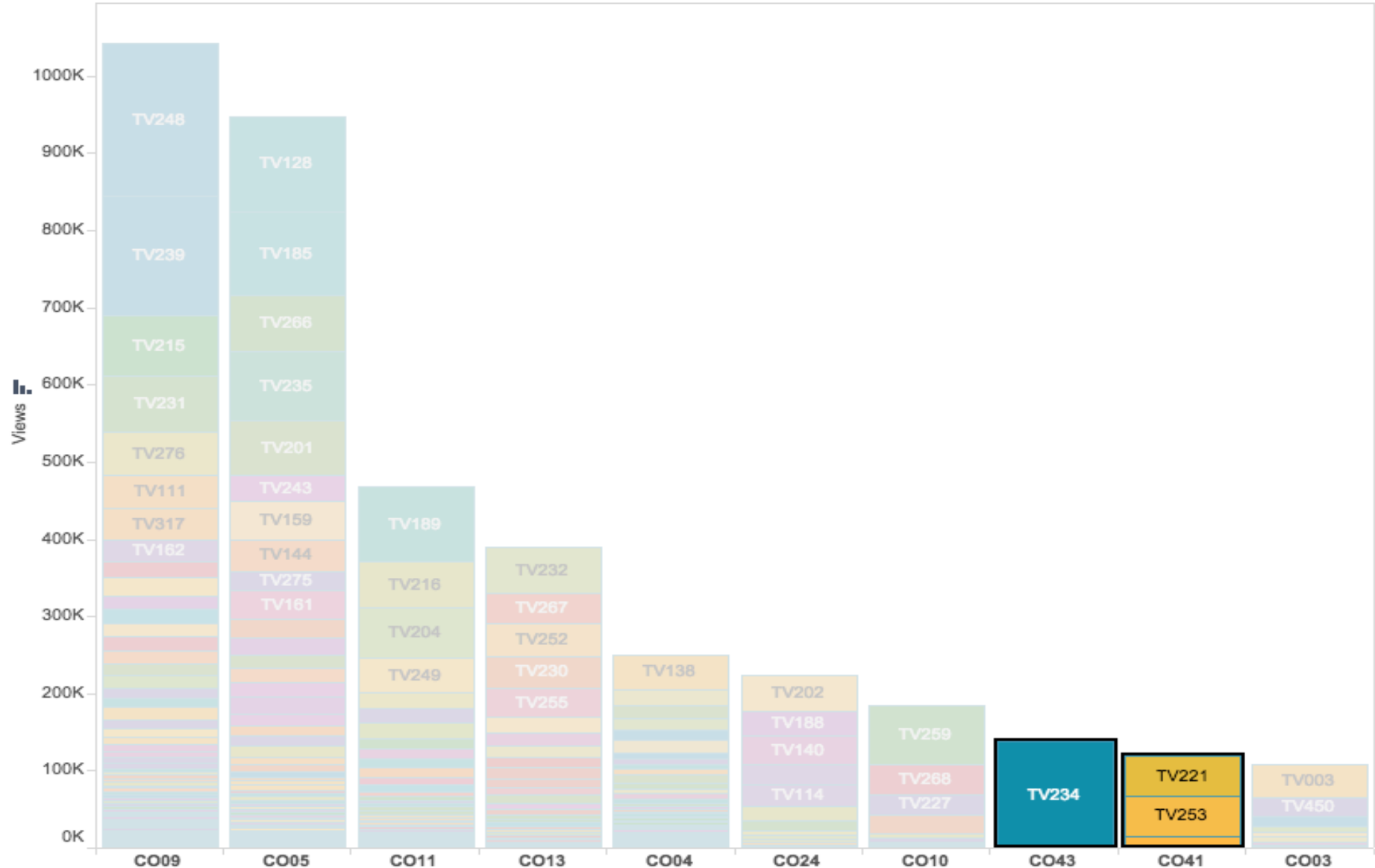
# Analysis – Content Owner

## Video & Content Owner performance



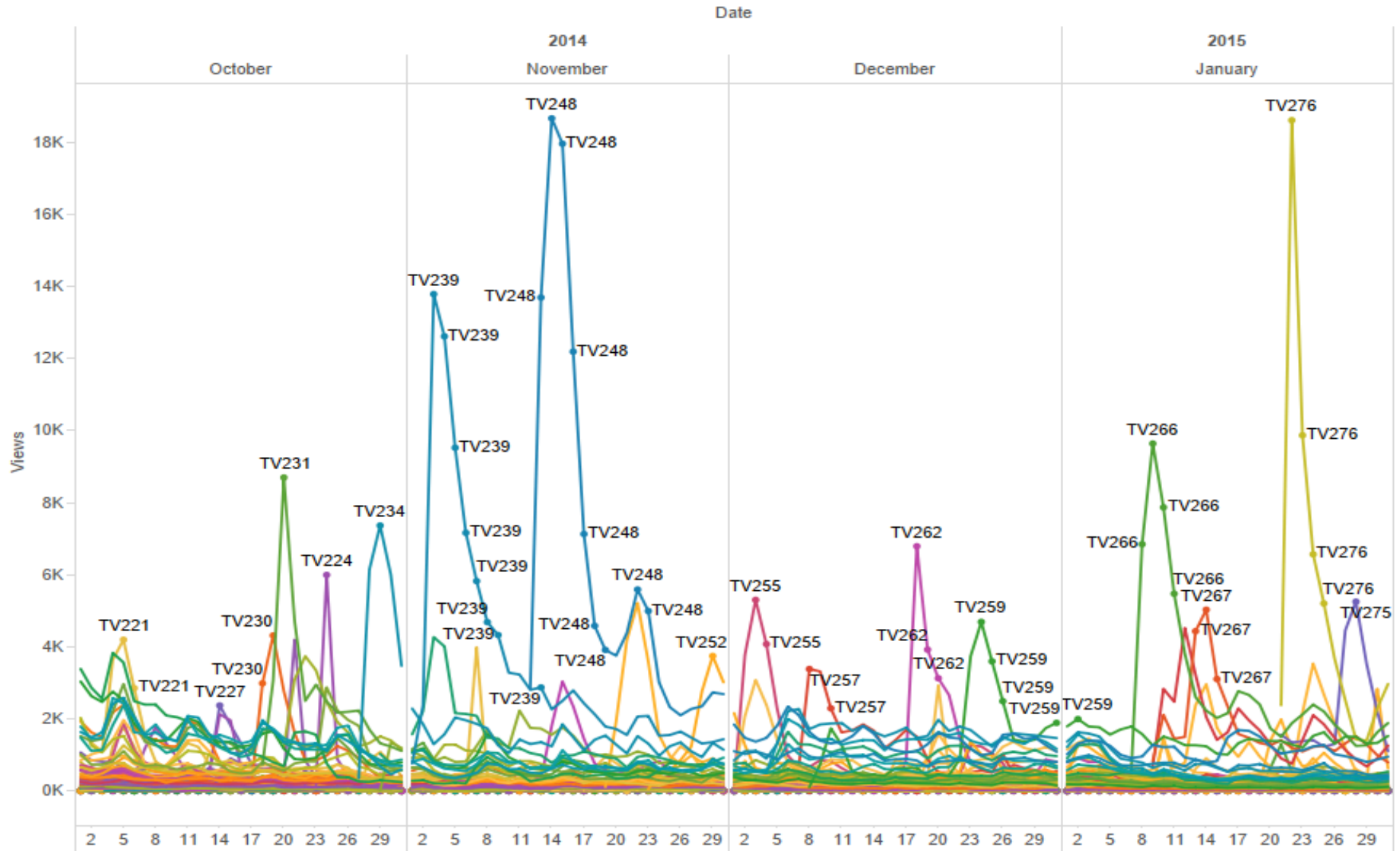
# Analysis – Content Owner

Video & Content Owner performance



# Analysis – Videos Traffic

Videos Performance Overtime



# Algorithm Overview

# Training phase

## Videos Overall Performances

- Hotness
- Freshness

## Videos Similarity Matrix

- Content Similarity
- Collaborative Filtering



```
graph TD; A[Videos Overall Performances] --> C((Recommendation Engine)); B[Videos Similarity Matrix] --> C;
```

Recommendation Engine



# Training phase

## Videos Overall Performances

- Hotness
- Freshness

## Videos Similarity Matrix

- Content Similarity
- Collaborative Filtering



```
graph TD; A[Videos Overall Performances] --> C((Recommendation Engine)); B[Videos Similarity Matrix] --> C;
```

Recommendation Engine

# Training phase

Videos overall performances:

$$Hotness* = \frac{\sum usersWatched}{firstDate - lastDate}$$

\*With gender filter applied

$$Freshness = \frac{1}{(broadcastDate - currentDate)^2}$$

# Training phase

## Videos Overall Performances

- Hotness
- Freshness

## Videos Similarity Matrix

- Content Similarity
- Collaborative Filtering



```
graph TD; A[Videos Overall Performances] --> C((Recommendation Engine)); B[Videos Similarity Matrix] --> C;
```

Recommendation Engine

# Training phase

## Videos similarity Matrix – Content Similarity

- Original Country:

$$V_1.country == V_2.country$$

- Original Language:

$$V_1.language == V_2.language$$

- Adult Content:

$$(V_1.adult == 1) \& (V_2.adult == 1)$$

- Content Owner:

$$V_1.contentOwner == V_2.contentOwner$$

# Training phase

## Videos similarity Matrix – **Content Similarity**

- Episode Count:

$$\frac{\min(V_1.\text{episodeCount}, V_2.\text{episodeCount})}{\max(V_1.\text{episodeCount}, V_2.\text{episodeCount})}$$

- Genre:

$$J(v_1, v_2) = \frac{G_1 \cap G_2}{G_1 \cup G_2}$$

- Cast:

$$J(v_1, v_2) = \frac{C_1 \cap C_2}{C_1 \cup C_2}$$

# Training phase

## Videos Overall Performances

- Hotness
- Freshness

## Videos Similarity Matrix

- Content Similarity
- Collaborative Filtering



```
graph TD; A[Videos Overall Performances] --> C((Recommendation Engine)); B[Videos Similarity Matrix] --> C;
```

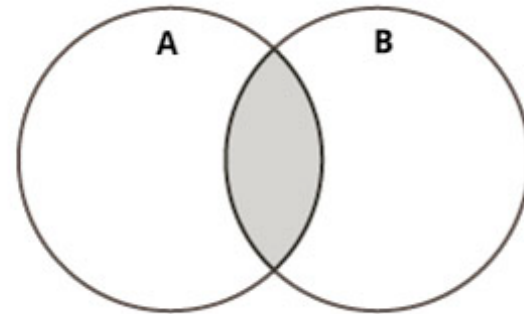
Recommendation Engine

# Training phase

## Videos similarity Matrix – Collaborative Filtering

- Jaccard Index - [https://en.wikipedia.org/wiki/Jaccard\\_index](https://en.wikipedia.org/wiki/Jaccard_index):

$$J(v_1, v_2) = \left| \frac{U_1 \cap U_2}{U_1 \cup U_2} \right|$$



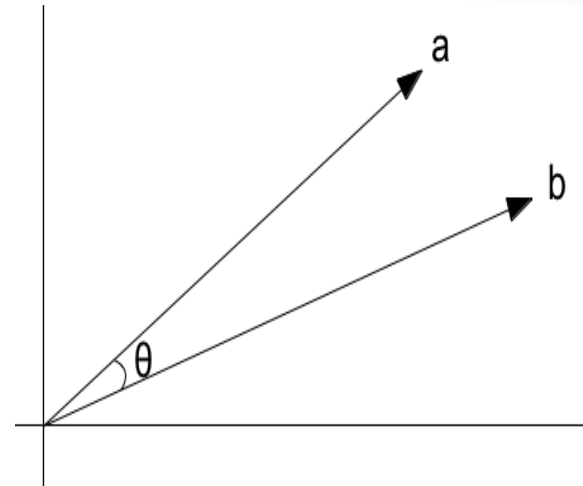
- Set theory
- Ratio of intersection gives similarity score
- Sensitive to sparse input – limit to only top 25% videos

# Training phase

## Videos similarity Matrix – Collaborative Filtering

- Cosine Similarity - [https://en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity):

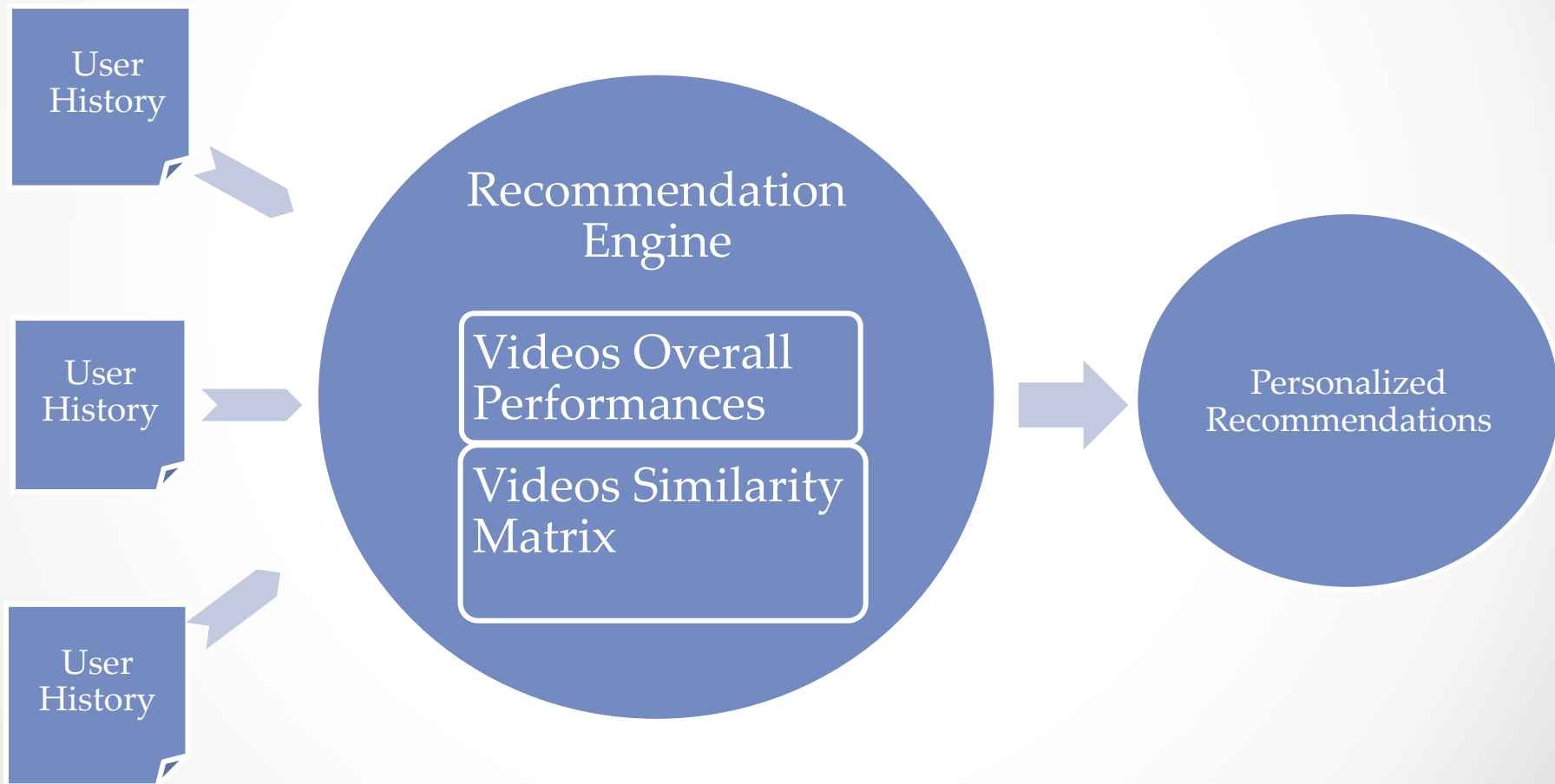
$$\cos(v_1, v_2) = \frac{\vec{U}_1 \cdot \vec{U}_2}{\|\vec{U}_1\| \|\vec{U}_2\|} = \frac{\sum_{i=1}^n U_{1,i} \cdot U_{2,i}}{\sqrt{\sum_{i=1}^n U_{1,i}^2} \times \sqrt{\sum_{i=1}^n U_{2,i}^2}}$$



- Vector space model
- Angle between 2 vectors gives similarity score.
- Good for sparse input – can apply gender filter.



# Personalization phase



# Performance

Overall time & space complexity:

$$O(uv^2)$$

- $u$ : number of **users** (880,000)
- $v$ : number of **videos** (600)

## Advantages:

- Lightweight – fits in 8GB Macbook Air!
- Scalable (fully distributed with [SparklingPandas](#))

# Applications

## Flexibility:

- Custom **weightages** for:
  - Features
  - Collaborative filtering similarity scores
  - Video performances (hotness or freshness)
  - Individual User - Video scores

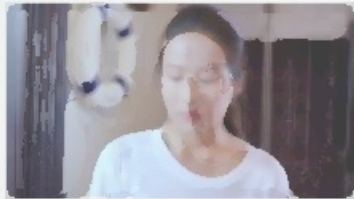
## Not just an engine but a framework:

- To create **different recommendation engines**.

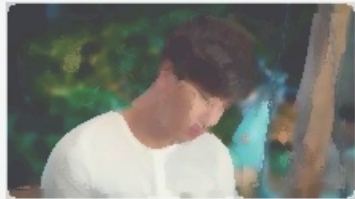
# Applications

Personally picked for you:

See All <>



Jiang Yan and Xia Bing beco...  
EN 100% • China



Xia Bing encourages Jiang Y...  
EN 100% • China



Xia Bing mistakes Jiang Yan...  
EN 100% • China



Promo: Best Get Going  
EN 100% • China



Jiang Yan and Xia Bing in Cl...  
EN 100% • China

Discovery Recommendations:

See All



A Wok Through Time  
EN 100% • Taiwan, Food & Cooking



Innocent Lilies 2  
EN 100% • Japan, Idol Drama



Love Frequency 37.2  
EN 100% • Korea, Melodrama



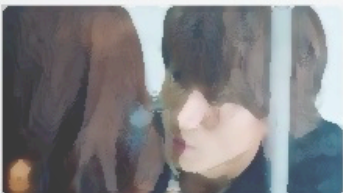
Be Arrogant  
EN 100% • Korea, Idol Drama



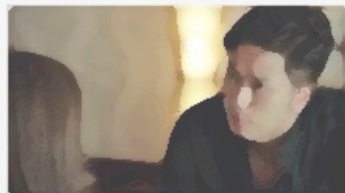
Leiji Matsumoto's OZMA  
EN 100% • Japan, SciFi & Fantasy

Shows with similar Genres & Actors, Actresses:

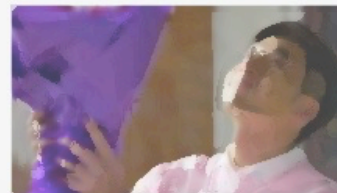
See All <>



Jerry Yan Pushes Maggie Ji...  
EN 100% • China



Ron Ng and Viola Mi Kiss an...  
EN 100% • China



Trailer 2: My Best Ex-Boyfrie...  
EN 0% • China



Trailer 1: My Best Ex-Boyfrie...  
EN 100% • China



INFINITE's Shoutout to Viki F...  
EN 100% • Korea

# Suggestions

- **Additional useful data sets:**
  - Explicit **user rating** is also very important.
  - User's contributions data (**subtitles**).
  - User's and video's interactions data (**live comments**).
- **Training & Testing data:**
  - Should **exclude top videos**.  
(Promoted on front-page or banners.)
- **Evaluation method:**
  - Equal test set splits will give an overall better result.  
(Models that work well with **Feb 2015** data might not work very well with **March 2015** data)

# Technology stack

- **Tableau Public**

- Free to [download](#)
- Publicly shared [workbooks](#)
- Interactive visualizations and insights

- **Python**

- [Pandas](#): data analysis library
- [Scikit-Learn](#): machine learning library
- [iPython Notebook](#): IDE for data analysis
- Other libraries:
  - Spotify's [annoy](#): approx. nearest neighbors calculation
  - PySpark's [Mllib](#): spark's machine learning
  - [panns](#): approx. nearest neighbors search
  - [python-recsys](#): recommendation system

Thank you!