ʃigmoid MasterChefs

Revolutionizing ad moderation with ML and Gurobi: Optimally matching ads to moderators, enhancing efficiency with violation flagging and text categorization, and ensuring timely ad deployment.

Inspiration

When reviewing advertisements, it is important to assist moderators which can help make the moderation process more efficient. As such, our system does not just take into account the characteristics of each advertisement and each moderator, but also seeks to provide accurate signals to better prioritize and assist moderators, namely through a CLIP video violation flagging and a SentenceTransformers ad categorization model.

What we did

Our pipeline contains 3 main components.

Advertisement Scoring System

Engineering

ıplementing the Ad Scoring System, two new features are created to be used as features in the optimization model.

: created a `days_diff` feature which is the difference between `start_time` and `pdate`. It is the number of days between the advertisement being uploaded on Ads and the date the advertiser wants the ad to start. If the difference is negative, we impute the mean number of days among the positive values. The lower the `:f` the more urgent the ad is and hence the higher priority.

iered advertisers which is calculated by the below formula:

hresholds are calculated based on quantiles and then each advertiser is assigned a `tier` from 1-10.

)ptimizer

: the ad priority score to be a product of `avg_ad_revenue`, `baseline_st` (which we can derive from `delivery_country`, `product_line` and `task_type_en` ıııpuıs), `ays_diff` (negative correlation) and `tier`, the optimization model will aim to maximize priority score

The objective function can be seen below:

: coefficients β and λ are derived from an iterative training and optimization process of the Gurobi optimizer.

tic term is combined to model a normal distribution from 0-1. A constraint where the mean of the priority scores = 0.5 is also implemented.

above, we ran the first 20,000 rows of the provided dataset and assigned a priority score to each ad. The results can be found in the appended "ad_dataset.xlsx" file. Each ad is also assigned a confidence score, and the assigned `moderator_id` is in the moderator column.

ı Flagging and Categorizing Ads

ı Flagging

for potential violation of the video advertisement, we sample K number of frames evenly across the video and utilise [CLIP](#) as a zero-shot classifier. We provide a :t of violation labels to check for but it can be customised for any violation that needs to be flagged.

To normalise the logits to a confidence score that can be interpretable, we compare the logits of our each label against a baseline (the empty string "") - a technique inspired from the unconditional sampling of diffusion models. This provides us a confidence score between 0-1 that is independent of the other labels being tested for.

Categorizing Ads

ecognise that advertisements cover a wide range of categories and each moderator has a few specific expertise within each domain. Therefore, a system that is f determining which category an advertisement belongs in and then matching it to the moderator with the most appropriate expertise can ensure timely and noderation.

ıge a biencoder, specifically the [BGE Embedding models](#), embedding each category and their corresponding description. We then compare the embedding of the nent description which each of the category embeddings. The most similar category embedding would be the category that our system has identified. Similarly, ɔries can be changed dynamically.

, since the current ads dataset does not include the `category` feature, we randomly generated a category for each ad.

ıce Metric

ıce would be a new feature added to each advertisement and it represents our model's certainty in its prediction regarding the presence or absence of a violation in `ıfidence` is calculated separately from `ad_score` as they require different types of moderators.

onfidence is high, the ad is "easier" to moderate as it is easy to spot the presence or absence of a violation. Hence the ad will be allocated to moderators that have h productivity so the ad can be cleared quickly, and low accuracy as a tradeoff
: ads with low confidence, it will be allocated to moderators that have high accuracy so that violations can be correctly identified, with the tradeoff being low ductivity

consist of two components:

- ıne model confidence determined from CLIP. The formula for determining the confidence from the score array of possible violations can be found below.

: punish_num shows how confident we can be in whether the advertiser is likely to have violations or not. High and low punish_num indicates higher confidence ile the mean punish_num would show low confidence. Below is how we derive confidence from the normalized punish_num:

icients, one can

...g the above two components, the overall confidence metric will be defined as such:

lose to 1
_score as they
an be cleared

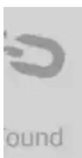
ich ad in the Excel sheet provided contains ad_score, category and confidence, all 3 are used in the moderator matching algorithm.

or Scoring and Matching System

System for Moderators

ise the Productivity and accuracy columns to determine how "good" the moderators are. After normalizing the 2 metrics, the formula for the score is:

ish_num would

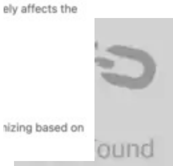


Max Tasks per Day

When the queue optimization system allocates ads to moderators, it should ensure that the ads are evenly allocated among all suitable moderators. As the TikTok Data eam expects an increase of utilization % by 10%, we set that as the threshold (i.e. each moderator's utilization can only increase by a maximum of 10%).

above, we can calculate the maximum number of tasks each moderator can take on per day with the below formula, it assumes that TikTok moderators work 8 s per day and that handling time is in ms (as shown from EDA).

ely affects the
izing based on



Generating Expertise

As mentioned above, the categories of the ads would be matched with the moderators' expertise to ensure timely and accurate moderation.

However, since the current dataset does not include this expertise feature, we will assume that all moderators do not have any expertise as of now. The optimization matching model will take note of this and still ensure that each moderator will only be given a maximum of 3 categories per day, to ensure that their work is more focused and efficient.

Moderator Allocation Gurobi Optimizer

The optimization model contains two objective functions, the first considers ad_score and the second considers ad_confidence.

The first objective function to minimize the difference between ad's score and moderator's score, this ensures that the top ads get matched with the best moderators

The second objective function is with regards to the confidence of the ad, which is defined as how confident our model is in identifying whether there is a possible violation in the ad. How it affects the moderator matching can be found in the "Confidence Metric" section above.

Combining these two objective functions, the overall objective function is as follows:

onstraints are also included in this optimization model.

e ad should only be allocated 1 moderator
_productivity e total tasks allocated to each moderator cannot exceed their max_tasks_per_day
nultiple iterations e delivery_country of the ad must match the market of the moderator
β_2, both noderator's category expertise is not null, expertise must match category of ad
ull, each moderator can only be assigned a maximum of 3 ad categories a day
- This is to ensure that the moderator's work is more focused to improve productivity

ıt would be each ad being assigned to a specific moderator. The number of tasks assigned to each moderator can also be seen on the submitted moderator_dataset and the % increase in utilization.

Full-Stack Web App of Ads Submission Form

In addition to scoring the provided ads and moderators, and also assigning each ad to a moderator, we also created a web app using a React front-end and Flask back-end to demonstrate how our optimization models, violation flagging and categorization of ads models work. You may refer to our Github README on how to run the web app on localhost.

On the webapp, advertisers would submit an ad to be sent for moderation. They are to key in fields such as Name, Description, Delivery Country and also upload the .mp4 of the ad video. After submitting, the results page has two main sections.

- The ads portion shows ad_score and the components from which it was derived, potential violations and the category of the ad is also shown
- The moderator portion shows the moderator the ad was assigned to, and the remaining tasks he/she can take in a day as well as the % increase in utilization

How we built it

For our Exploratory Data Analysis, we used Python packages such as pandas, numpy, Matplotlib and seaborn to analyze and plot visualizations (e.g. correlation matrix) about the data.

For our Ad Scoring and Moderator Allocation Optimization models, we used the optimizer to find an optimal solution. Our violation flagging model uses CLIP and the ad categorization model uses a biencoder.

For our full-stack web app, the front-end was built using React and the back-end was done using Flask.

b Guide

e the folders/files in our Github repository that our various components are located:

ploratory Data Analysis Notebooks and files can be found in folder "EDA"
and Moderator Scoring code and result datasets are found in folder "Scoring"
IP and SentenceTransformers models are found in `violation_checker.py` in folder "violations"
derator Queuing System and Optimization model found in `moderator-queue.ipynb`
act front-end found in folder "ad-input"
sk backend in `main.py`
tasets after running the provided ads and moderator data through our allocation models can be found in the folder "Results"

's next

r current code uses pre-defined weights for the different optimization equations, ideally the β values should be determined through an iterative training and ion process and also through business knowledge. Hence, prolonged usage of our Ads Prioritization and Moderator Matching algorithms will optimize the ıence leading to increased revenue and moderator utilization.

m can also be extended to handle ads in multiple languages, using language detection and translation tools to assign ads to moderators proficient in the respective s.

Biography

Poon Zhe Xuan is a highly motivated student at the National University of Singapore, where he is pursuing a Double Major in Business Analytics and Statistics. Fueled by his deep passion for Data Science and Machine Learning, he not only emerged as a Semi-Finalist in the prestigious Google Cloud Hackathon but also developed a full-stack web application that leverages semantic search and content-based filtering to provide tailored movie recommendations. Zhe Xuan's internship experiences at industry giants like Rakuten Viki and Shopee have honed his skills in utilizing tools such as Pyspark and SQL for data analysis, A/B testing, and the creation of dynamic dashboards.

Wei Hern Lim's remarkable journey in the fields of Computer Science and Statistics at the National University of Singapore is distinguished by his exceptional talent and unwavering dedication. His innovative capabilities were highlighted when he and his team secured the 1st runner-up position in the NUS Fintech Month Hackathon for the Best Machine Learning Project, introducing a groundbreaking Machine Learning-driven API powered by the T-5 Text-To-Text Transfer Transformer Model, poised to revolutionize chatbot efficiency in businesses. In the professional arena, Wei Hern has left an indelible mark, contributing significantly during his internship at DBS Bank and KPMG. His versatile skills, encompassing data analytics, financial instrument valuations, and corporate finance, backed by proficiency in Python, Tableau, and Excel, have consistently yielded insightful results

enia Purnama is a dedicated Business Analytics student at the National University of Singapore, pursuing a second major in Statistics. She demonstrates her ɔr software development as a Technology Associate at Google Developer Student Clubs NUS and honed her skills in web application development from her full-ject that aims to recommend movies using semantic search and content-based filtering. Kay's impressive analytical skills and critical thinking shine through her :tory at the ASEAN Data Science Explorers 2023 Competition. Kay's academic excellence is complemented by accolades such as making it to the Dean's List and ognized as the Top Student for BT2102 (Data Management and Visualisation) at NUS.

i Jia, Computer Science undergraduate at the National University of Singapore with a second major in Data Analytics, is undeniably a driven individual. His nal journey has seen him make a significant impact during an internship at Whatnot Startup Studio in Bangkok, where he led the design and implementation of nterface for Aira, an advanced LINE AI Chatbot and gained valuable experience in the field of entrepreneurship. Rui Jia boasts a diverse skill set spanning both ınd frontend development. He has undertaken many noteworthy projects, such as creating a tool to transfer profiles across Netflix and developing a YouTube der. Proficient in web development tools like React, CSS, and Flask as well as having a strong command of Python, Chan Rui Jia is poised for an exceptionally ş future in the dynamic world of technology and innovation.

Kian, Justin is a passionate Computer Science undergrad at the National University of Singapore (NUS). Transitioning from Quantitative Finance, he d his passion in computing through an introductory programming module. Ever since, Justin boasts a commendable academic record and experience in software and is proficient in Python, Java, Flutter, and Dart. Justin was offered a Teaching Assistant role in NUS's School of Computing, specialising in Programming logy and Discrete Mathematics. Beyond academia, Justin honed his data analytics skills during an internship at the Monetary Authority of Singapore and ated his digital prowess in web development and Google Ads management at Chatter Solutions. An active community leader, Justin contributes to various NUS ;, such as volunteering in Centre for Computing and Social Good & Philanthropy events, and seeks holistic growth through his involvement in multidisciplinary at Tembusu College.

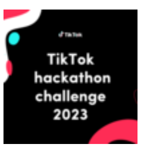
With

- clip
- flask
- gurobi
- chine-learning
- ıdas
- hon
- ct
- tencetransformers

out

Hub Repo

:d to

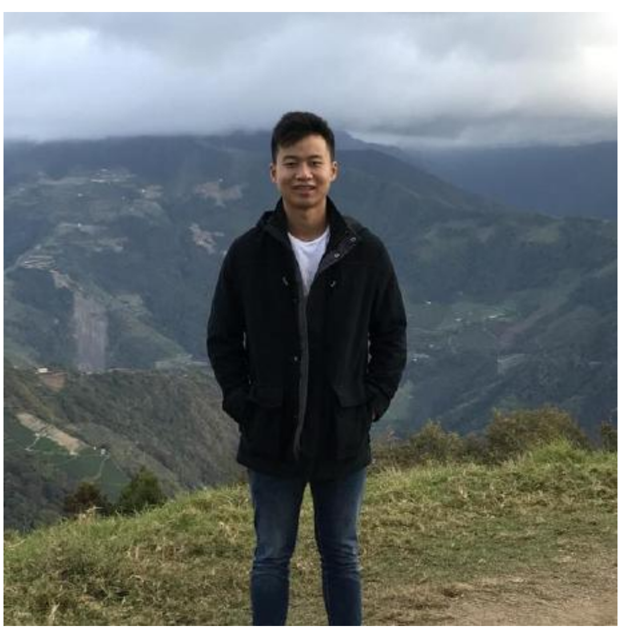


Tok Hackathon Challenge 2023

- Winner Top 5 Overall Submissions (Judge's Choice)

Created by

icients, one can

- 

Justin Tan

[an Rui Jia](#)

- 

  [Poon Zhe Xuan](#)

- 

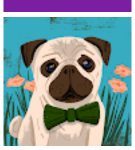
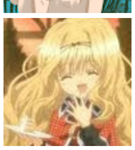
  [Wei Hern Lim](#)

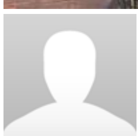
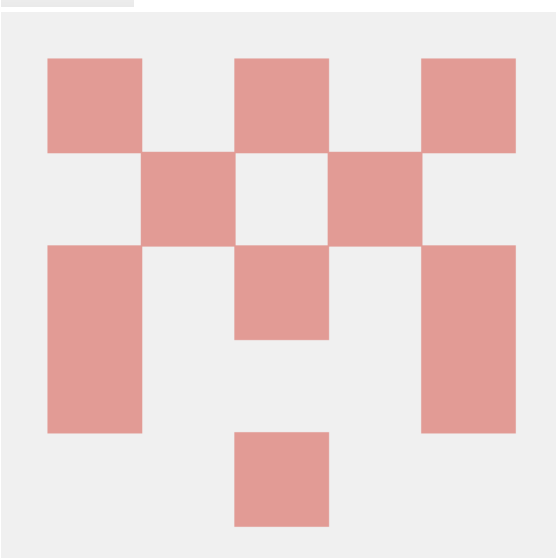
- 

  [Kay Eugenia Purnama]

[Like 56]
56 people like this:

- 
- 

- 
- 

- 
- 
- 
- 

- 
- 

[+ 46 more]
Share this project:

-
-
-

Updates

Poon Zhe Xuan started this project — 4 months ago

Leave feedback in the comments!

View previous comments

- 

  nigel eng · 3 months ago

  love this!

- 

  jackieanto · 3 months ago

Cool

- 

  Thái Lê · 51 days ago

  Oke love

- 

  Thái Lê · 51 days ago

  Thái Lê arenaofvlor

- 

  Abdul Hakim Uzzal · 38 days ago

  Hi

Log in or **sign up for Devpost** to join the conversation.

Devpost

- About
- Careers
- Contact
- Help

Hackathons

- Browse hackathons
- Explore projects
- Host a hackathon
- Hackathon guides

Portfolio

- Your projects
- Your hackathons
- Settings

Connect

- Twitter
- Discord
- Facebook
- LinkedIn

- Community guidelines
- Security
- CA notice
- Privacy policy
- Terms of service