



Trend of Twitch API Data

Tianyi Fang

Project Statement

Data Analysis of Twitch API endpoints datasets to better understand:

- **Which application developers are important to the developer ecosystem**
- **Which API endpoints are more important to continue to maintain**

There are two datasets: one is daily log data from Nov. 2017 to Feb. 2018, including client id that give application calling the Twitch API, API endpoint, number of requests and average latency. The other is application mapping data, including application names and application authors.

Initial Analysis

Key Metrics:

- **Number of Requests:** shows how frequently one application is called(**popularity**). API with larger number of requests or has increasing trend are more important for us. Some applications might have large count number but smaller number of request, means they are not that popular.
- **Average Latency:** shows how fast clients get response, it measures how well the application preforms(**efficiency**). Larger latency will increase user frustration and negative feedbacks. Popular applications with smaller latency are preferred by both clients and Twitch.
- **Trend:** shows how different endpoints and applications of different author changed overtime.

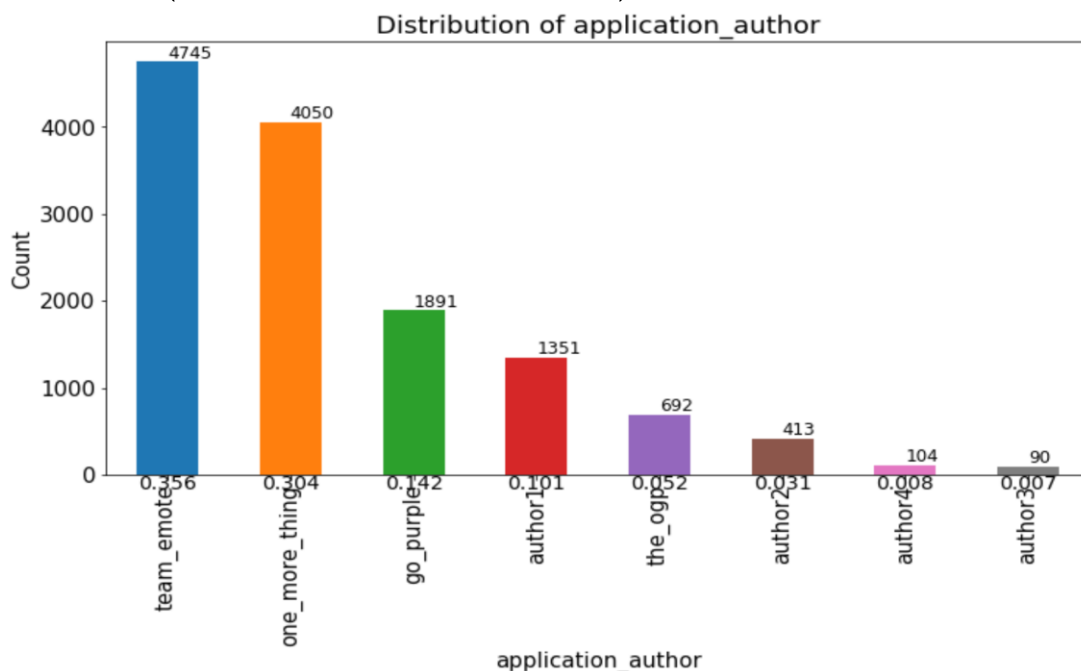
Technical Goal:

We want to find important developers, who **owns applications with most frequent calls and less latency time**; and **continuously-used popular endpoints, along with relatively longer latency time**.

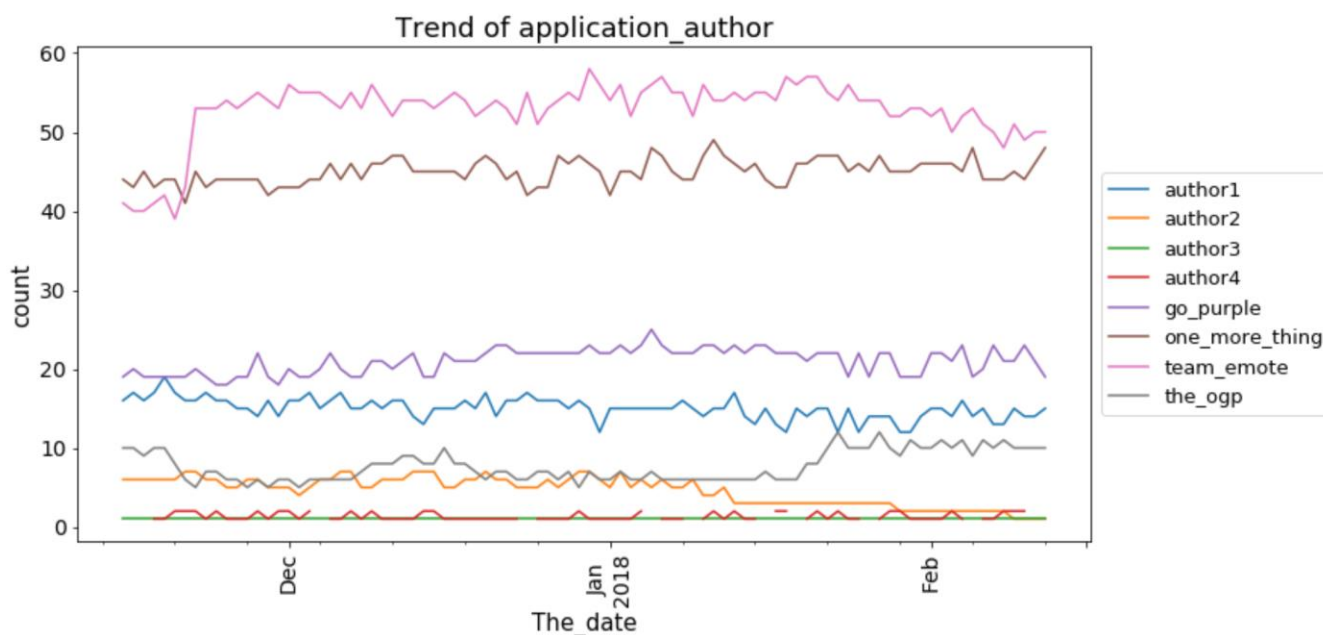
Notice: API version of /v3, /v5 will be removed at the end of 2018; old versions and new versions(helix) share the same API endpoint method name.

Analysis Visualization and Result

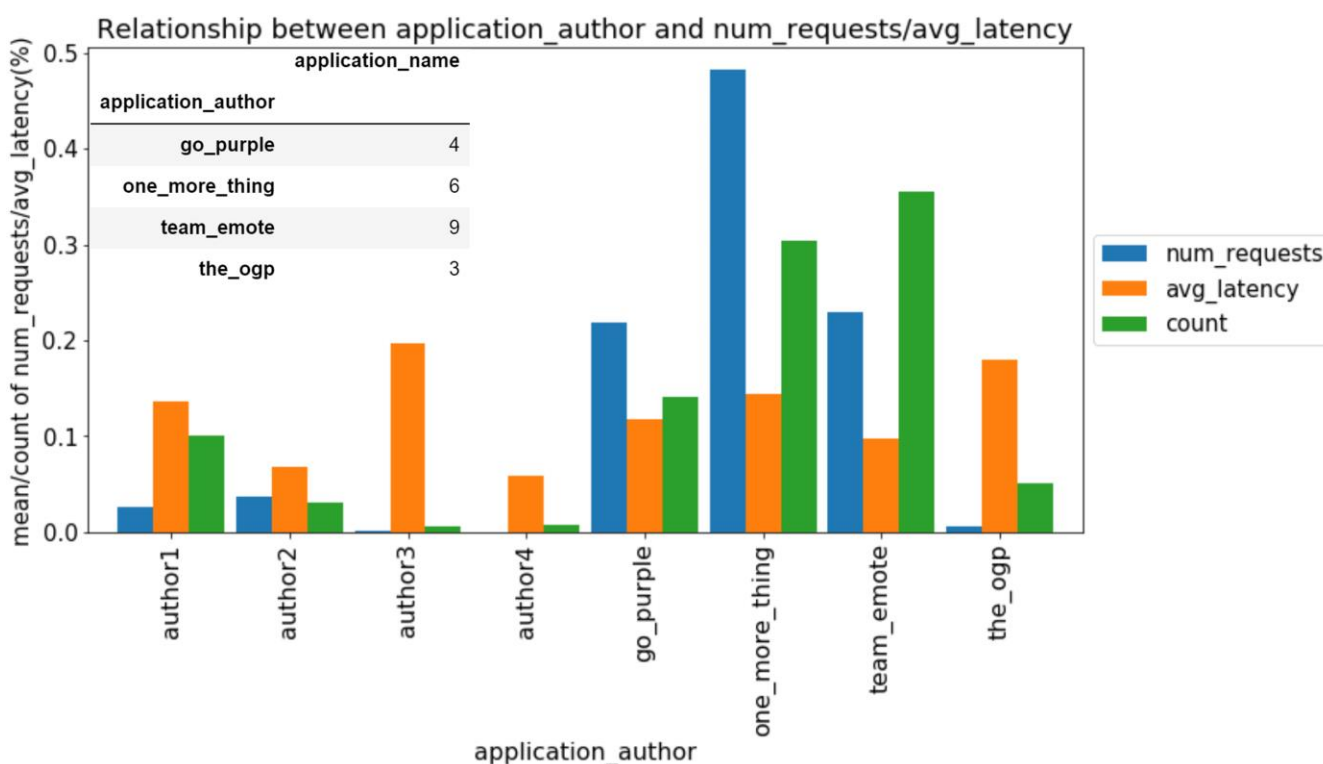
- Important Author (Distribution + Trend + Performance)



Team_Emote, One_More_Thing and Go_Purple are top three authors, whose applications are called most frequently, indicates that those author's applications might be the most popular among clients.



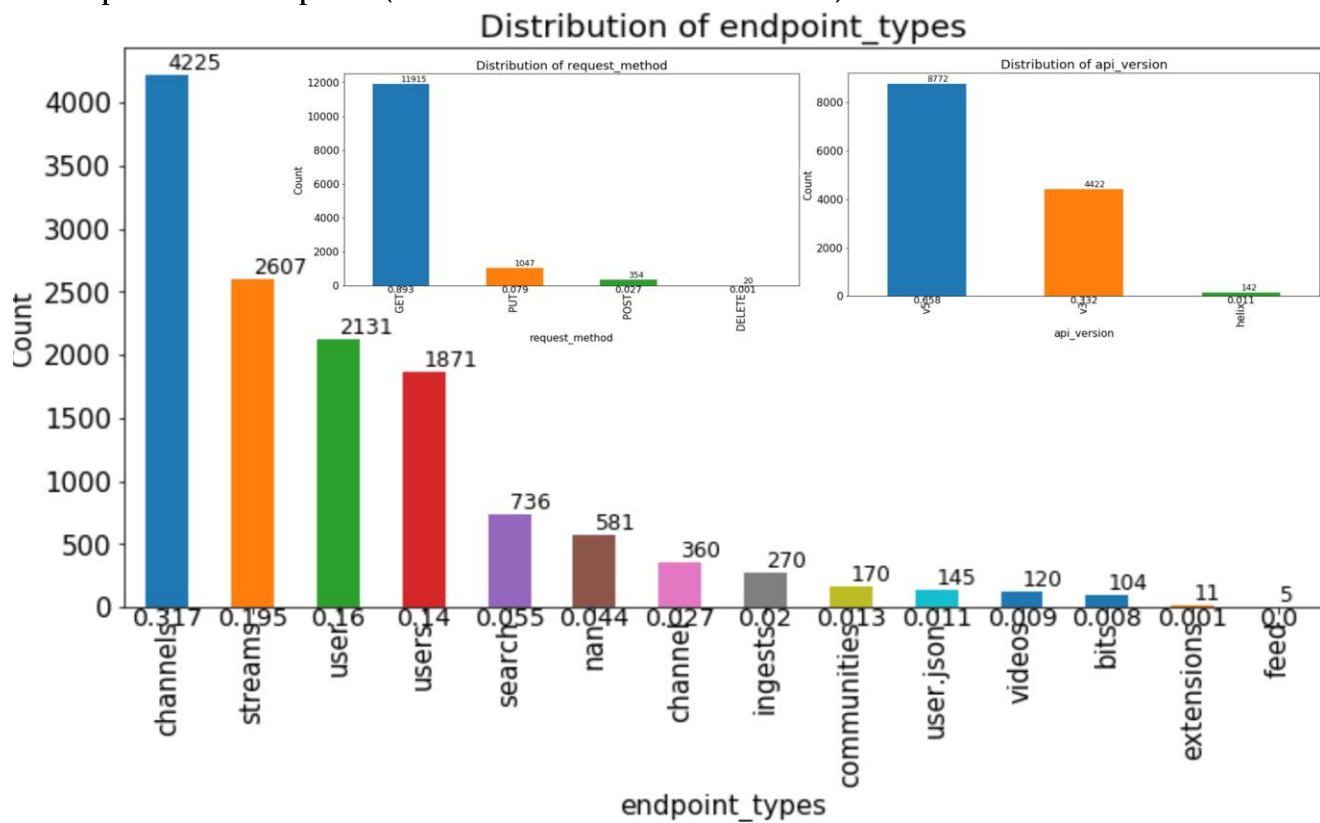
Team_Emote, One_More_Thing and The_Ogp show increasing trend. While applications of some unknown authors show slightly decline and being rarely called.



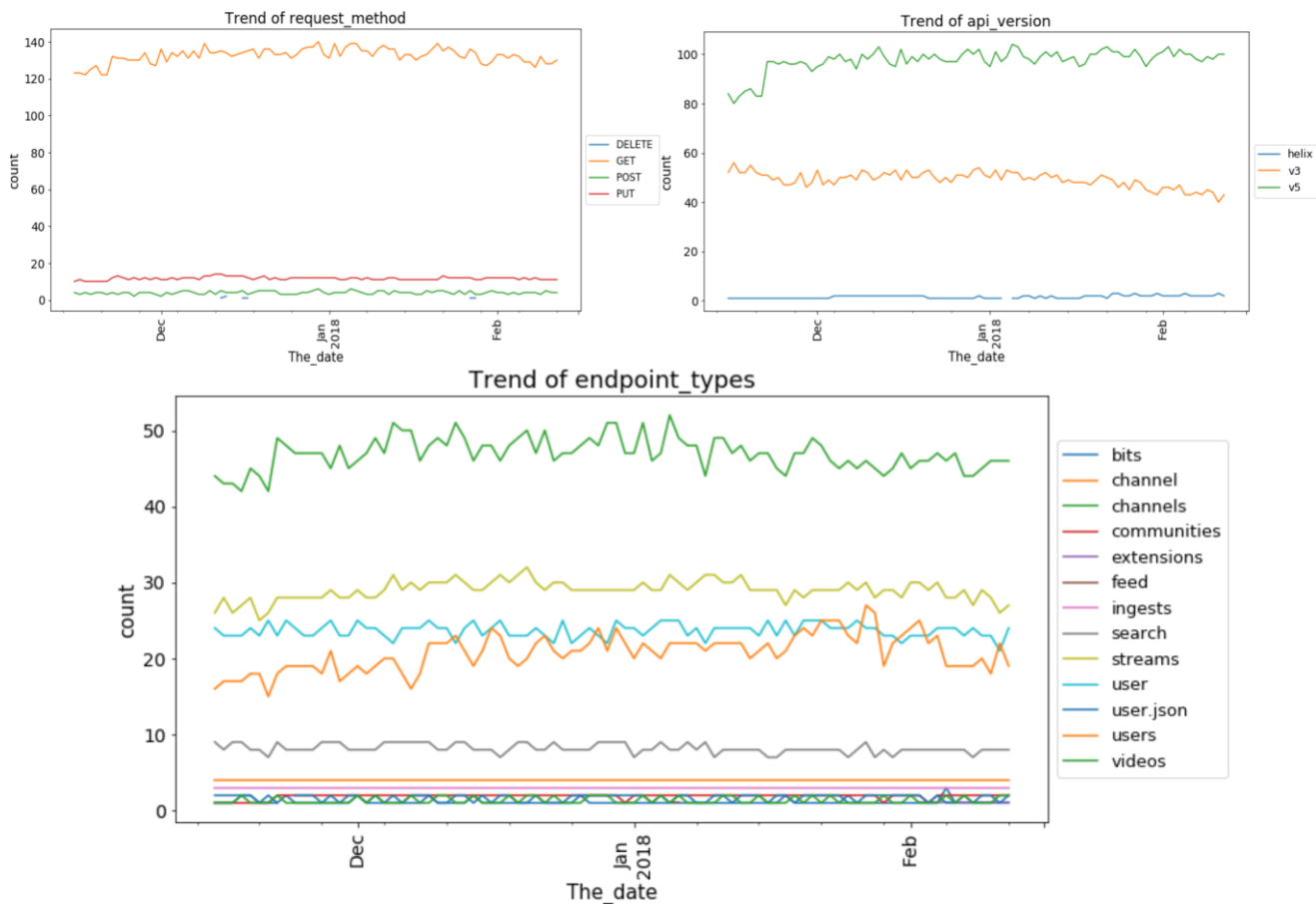
From this figure, we can find One_More_Thing has smaller total counts (smaller number of application owns) than Team_Emote, but he has much larger average number of requests, which indicates although his application quantity is less than Team_Emote, but his applications are more popular among clients and more important for the ecosystem. Taking average latency into consideration, although applications of anonymous authors has smallest latency (highest efficiency), those applications are barely requested by clients. Popular applications of One_More_Thing with relatively higher average latency than other popular applications of Team_Emote and Go_Purple indicates endpoints of those API need more effort to maintain and improve.

Important Developer: One_More_Thing, Team_Emote and Go_Purple are important developers to developer ecosystem, due to their well preformed, popular applications.

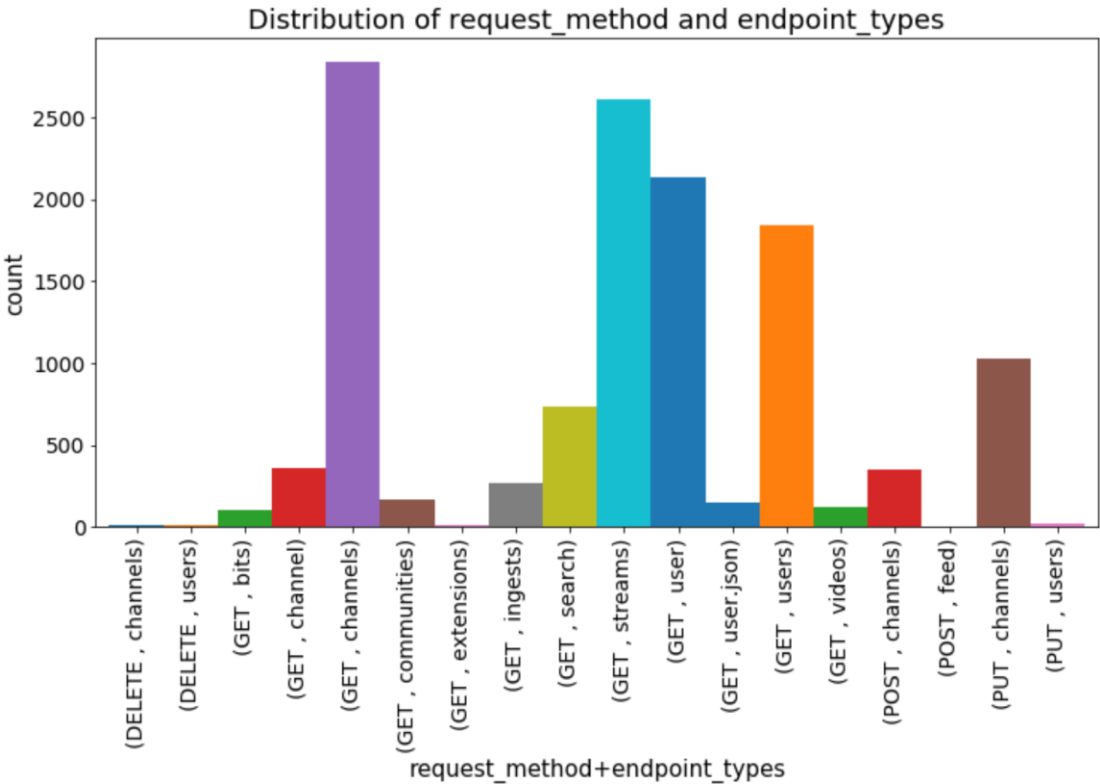
- Important API endpoints (Distribution + Trend + Performance)



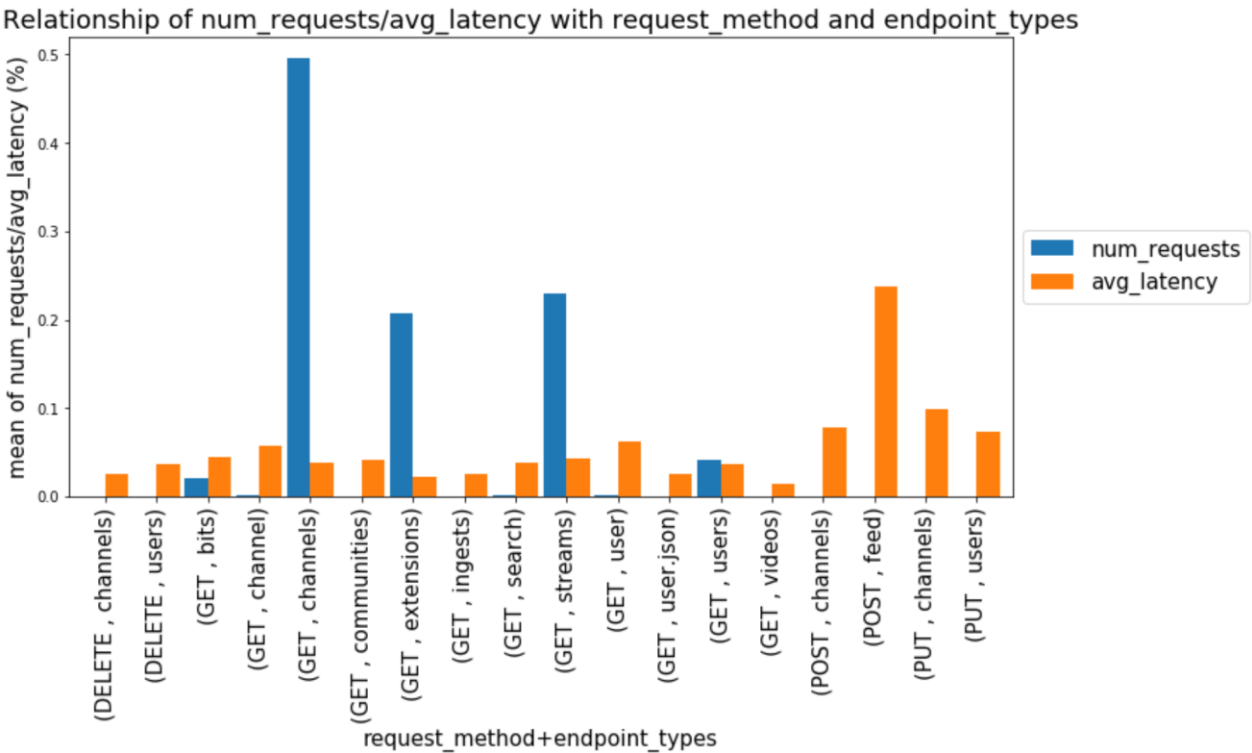
Separating API_endpoints into Request_method, API_version and endpoint_method helps better illustrate how they perform overtime. From the distribution, **GET** and **v5** are most frequently called methods and version, with 89% and 65% respectively. For endpoint_types, **channels**, **streams**, **user**, **users** and **search** are most popular.



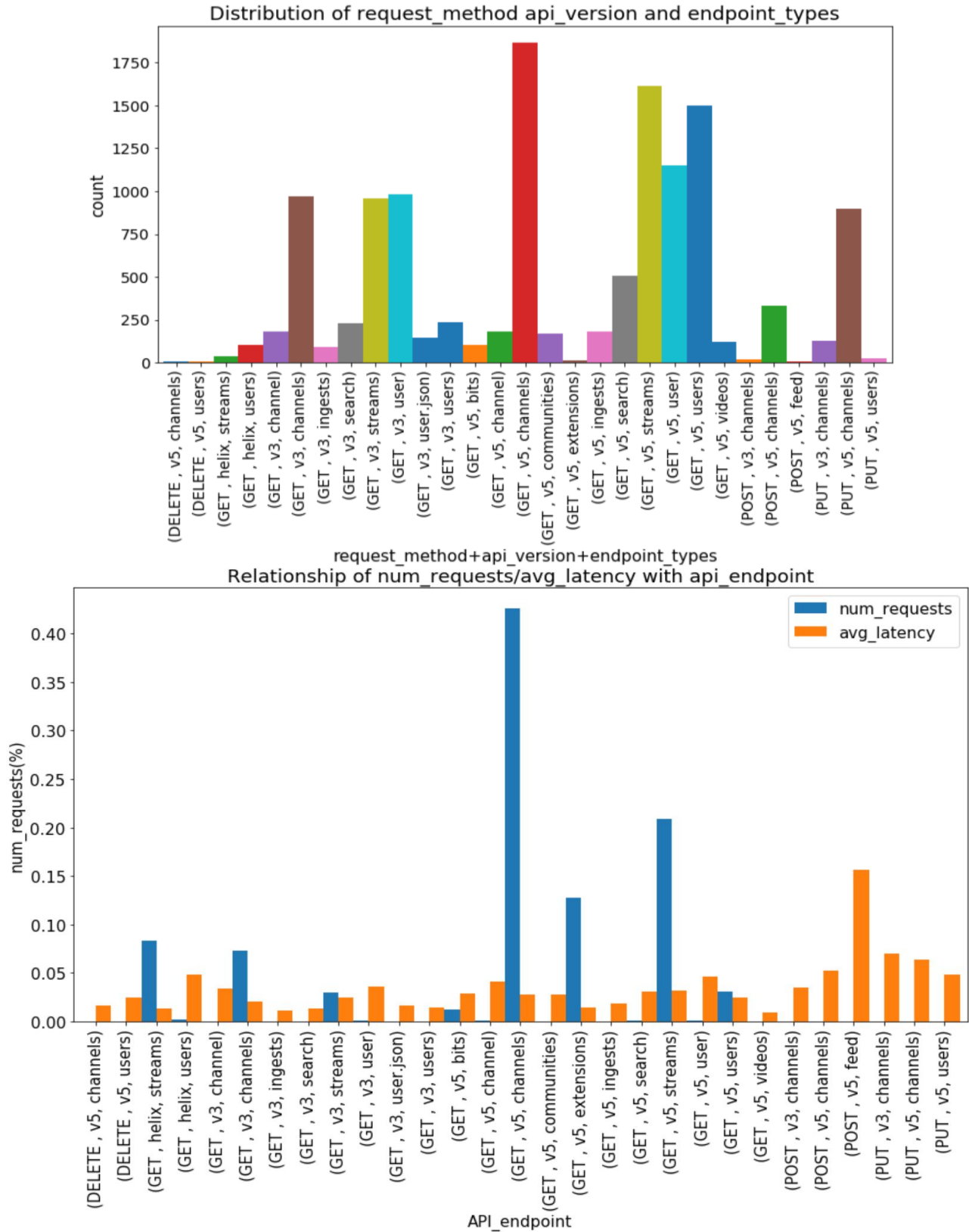
The trend of request_method and endpoint_types almost remain the same, even though with some fluctuation. Trend of API_version shows slow decrease in v3, small increase in v5, indicates **v5 are still the most popular version** in this period. New version helix is called in small percentage, that might because it is still under test.



As illustrated by Twitch, the API_version is neither related to request_method nor endpoint_types, **combine request_method and endpoint_types** as high-level illustration of what developers do such as Get channels, Post Channels help us find out the performance of API endpoints. The top five popular combination of request_method and endpoint_types are GET channels, GET streams, GET user, GET users and PUT channels.



Check how each combination preformed: GET channels, GET extensions, GET streams are top three most frequently used API endpoints. The average latencies of these endpoints are relatively small, indicates they are under well maintenance. Although GET extension shows less count, but each count contains larger number of requests, which also shows its popularity and importance. GET users and PUT channels shows many times, but they have very low number of requests, which means they are not that important.



If combine request_method, API_version and endpoint_types, we can have more detailed understanding of how each API endpoints perform. The top five popular of API request method_version_endpoints are all top five of API request method_endpoints in v5 version, the same as in v3 version. The performance in number of requests and average latency still tell us the important API endpoints are **GET channels**, **GET extension**, **GET streams in v5 version**; **GET channels in v3 version** and **GET streams in helix version**.

Notice that new version helix only shown up in GET streams and GET users.

Summary

Important developer:

One_More_Thing, Team_Emote and Go_Purple

Important API endpoints:

GET channels, GET extension, GET streams in v5 version;

GET channels in v3 version

GET streams in helix version

Notice

1. Anonymous author: there are only 22 client_id with 4 authors in app_metadata dataset, but 26 client_id in daily logs, which means the extra 4 applications are developed by unknown author. I assume each of them are created by different developer other than existed authors, but it is possible several applications are created by one author. Since application of author1 also being used in some degree, while applications of author3 is rarely used, it's better to set different author for each unknown application.
2. Version v3 and v5 will be removed 9 months later, but they are still largely in use recently, especially v5. Since most clients still use v5, API endpoints in v5 version are important and deserve to keep maintaining until the last several weeks. Moreover, Although in last 3 months, Helix version is only used in two API endpoints (just like under test), it will definitely be in increasing use in the following months and years. Thus, API endpoints in helix version are also important and need efforts to maintain, because its performance guides the following new-introduced applications and API endpoints.
3. Detailed and technical illustration can be found in supplement material.