PRO Feature Attribution

Jiaxuan Wang

Agenda

- Motivating the importance of feature attribution for PRO
- Computing feature attribution
- Using the feature attribution dashboard (bunnylol “go pro_attribution”) to
  - Learn about PRO’s logic
  - Monitor distribution shift

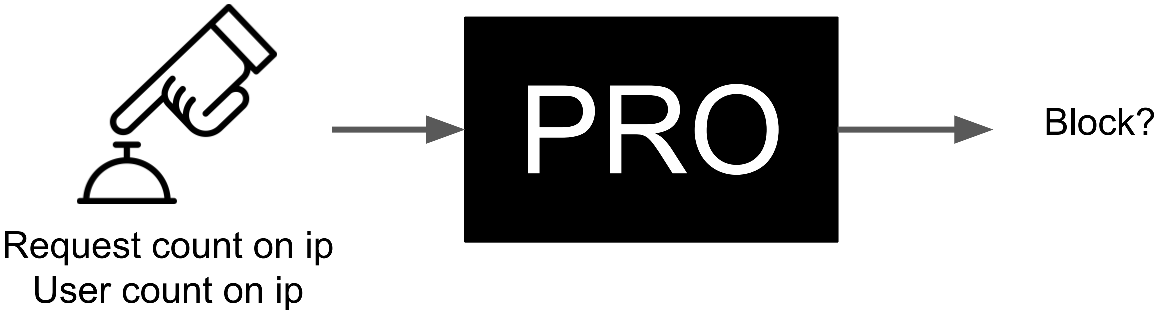
What is PRO?

PRO

Predictive response optimization (PRO) is a ML system that select actions to minimze

- **unauthorized scraping** (e.g., scraping data returned to scrappers)
- **friction to non scrappers** (e.g., Impact to Growth/DAU, Time spent, Revenue, Bug reports)

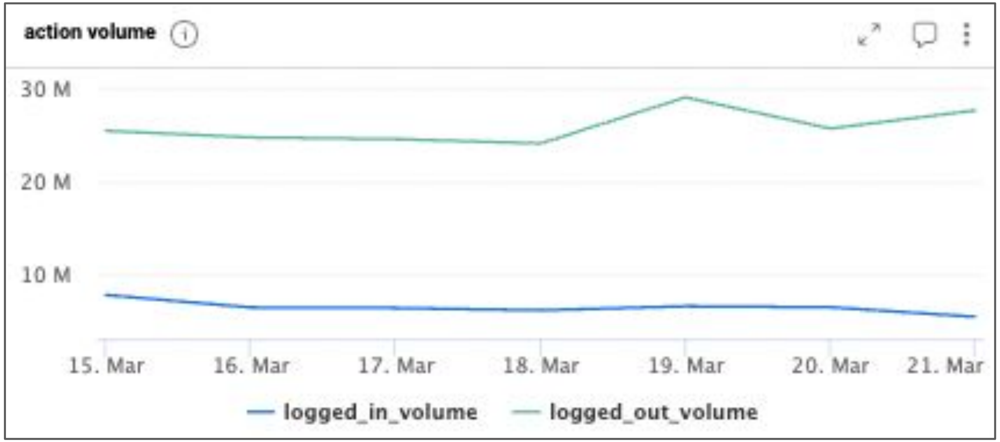
What is PRO?



Predictive response optimization (PRO) is a ML system that select actions to minimze

- **unauthorized scraping** (e.g., scraping data returned to scrappers)
- **friction to non scrappers** (e.g., Impact to Growth/DAU, Time spent, Revenue, Bug reports)

Need for Explanations



Despite acting on *millions of users* on IG logged in and logged out, we *cannot explain its logic* to defend and debug its decision

Problem: Black Box Models

Why is an IP blocked?

Why is there a spike in the amount of IPs blocked?

Problem: Black Box Models

Why is an IP blocked? → **Understand PRO's logic**

Why is there a spike in the amount of IPs blocked?

Problem: Black Box Models

Why is an IP blocked? → Understand PRO’s logic

Why is there a spike in the amount of IPs blocked? → **Monitor shifts in action/response selection**

Solution: Feature Attribution

Why is an IP blocked? → Understand PRO’s logic

Why is there a spike in the amount of IPs blocked? → Monitor shifts in action/response selection

Feature attribution is a principled way to answer those questions in terms of features

What feature is important for the decision?

Solution: Feature Attribution

Why is an IP blocked? → <u>The IP is blocked because request count is too high</u>

Why is there a spike in the amount of IPs blocked? → Monitor shifts in action/response selection

Feature attribution is a principled way to answer those questions in terms of features

What feature is important for the decision?

Solution: Feature Attribution

Why is an IP blocked? → <u>The IP is blocked because request count is too high</u>

Why is there a spike in the amount of IPs blocked? → <u>PRO blocks more IPs today because more IPs have high request count</u>

Feature attribution is a principled way to answer those questions in terms of features

What feature is important for the decision?

Solution: Feature Attribution

Why is an IP blocked? → <u>The IP is blocked because request count is too high</u>

Why is there a spike in the amount of IPs blocked? → <u>PRO blocks more IPs today because more IPs have high request count</u>

Answering those questions allows us to

- Understand, debug, and defend the model’s decision
- Monitor for irregularities in model’s behavior

Computing Feature Attributions

How do we know a feature is important?

- Short answer: we use SHAP [lundberg et al., 2017]

Computing Feature Attributions

How do we know a feature is important?

- Short answer: we use SHAP [lundberg et al., 2017]
- Long answer: we calculate the difference in output when we take out a feature.

Computing Feature Attributions: Example

How do we know a feature is important?

We have 2 features A and B

Attribution for A = PRO(A, B)

Computing Feature Attributions: Example

How do we know a feature is important?

We have 2 features A and B

Attribution for A = PRO(A, B) - PRO(B)

Computing Feature Attributions: Example

How do we know a feature is important?

We have 2 features A and B

Attribution for A = PRO(A) - PRO()

Computing Feature Attributions: Example

How do we know a feature is important?

We have 2 features A and B

Attribution for A = avg[PRO(A, B) - PRO(B), PRO(A) - PRO()]

Computing Feature Attributions: Example

How do we know a feature is important?

We have 2 features A and B

Attribution for A = avg[PRO(A, B) - PRO(B), PRO(A) - PRO()]

To ensure explanation uses the same setting as PRO’s decision, we compute feature attribution <u>*online*</u> when the decision was made.

What does it mean to take out a feature?

PRO outputs “block” when number of requests = 5 and number of users count = 1,
Consider B(,) as the function that outputs 1 when PRO outputs “block”, 0 otherwise

B(num requests, user count)

Attribution for num request = avg[B(5, 1) - B(0, 1), B(5, 0) - B(0, 0)]

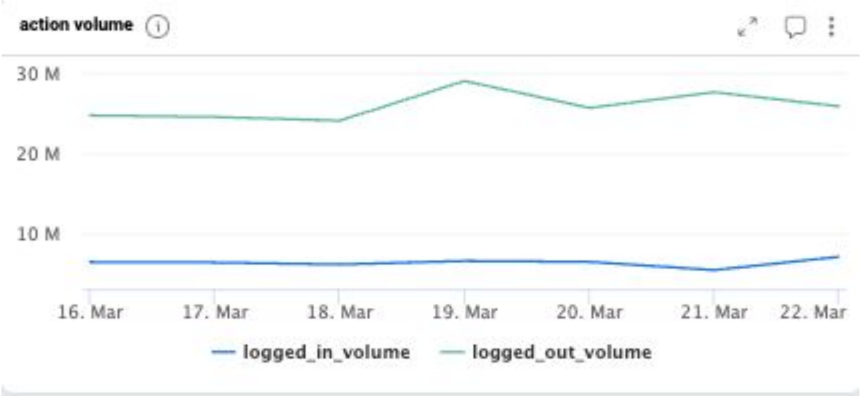
The red value is called foreground value (value of the sample we want to explain) and blue value is called background value (value of a sample for reference).

In PRO, background value is set to

- 0 for continuous features
- “” for context/categorical features

Logging Feature Attributions

Feature attribution is expensive to compute so we use randomly sampled inputs



Feature attributions are logged under the “**feature_attribution**” column of the tables “**pro_logged_in_ig**” / “**pro_logged_out_ig**”.

Why is an IP blocked?

(Understanding PRO’s Logic)

Why is *this* IP blocked?

Block

recommended_action	continuous_features	feature_attribution

1.0	{
	"bgp_subnet_score": 0.060089707,
	"bgp_subnet_score_account_endpoints":
	0.24628517230477,
	"bgp_subnet_score_not_enough_logging_fb_data": 1,
	"sdr_1_forecast_value": 0,
	"log_request_count_1d": 3.7972559397922,
	"log_user_count_on_ip": 0,
	"bgp_subnet_score_account_endpoints_unfiltered":
	0.068373071528752,
	"bgp_subnet_score_young_devices": 23807173.754865,
	"signup_events_1_forecast_value": 0,
	"cloud_hosting_score": 0.01,
	"sdr_labeled_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id_fb_data": 0,
	"bgp_subnet_score_rule_based_labels": 0.56907998735378,
	"time_spent_1d_1_forecast_value": 0,
	"bgp_subnet_score_netacuity_proxy_fb_data": 0,
	"request_count_1d": 43.57868989483,
	"endpoint_ratio": 0.27726344704115,
	"bgp_subnet_score_young_devices_fb_data":
	1569915645.6024,
	"num_users_1_forecast_value": 0,
	"num_requests_1_forecast_value": 0,
	"user_count_on_ip": 0,
	"num_labeled_requests_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id": 0.40450725744843,
	"estimated_turnstile_count": 5.5,
	"bgp_subnet_score_old_asbd_header_version":
	0.16701030927835
	}
		{"context_features":{},"continuous_features":
		{"num_labeled_requests_1_forecast_value":0,"num_requests_1_forecast_value":0,"num_users_1_forecast_value":0,"sdr_1_forecast_value":0,"sdr_labeled_1_forecast_value":0,"signup_events_1_forecast_value":0,"time_spent_1d_1_forecast_value":0,"log_request_count_1d":1,"log_user_count_on_ip":0}}

Only log_request_count has non-zero attribution

```
SELECT recommended_action, continuous_features, feature_attribution FROM pro_logged_out_ig WHERE feature_attribution != 'null' AND recommended_action = '1.0'
```

Why is *this* IP blocked?

The IP is blocked because log request count 1d has a value of 3.79

Block

recommended_action	continuous_features	feature

1.0	{
	"bgp_subnet_score": 0.060089707,
	"bgp_subnet_score_account_endpoints": 0.24628517230477,
	"bgp_subnet_score_not_enough_logging_fb_data": 1,
	"sdr_1_forecast_value": 0,
	"log_request_count_1d": 3.7972559397922,
	"log_user_count_on_ip": 0,
	"bgp_subnet_score_account_endpoints_unfiltered": 0.068373071528752,
	"bgp_subnet_score_young_devices": 23807173.754865,
	"signup_events_1_forecast_value": 0,
	"cloud_hosting_score": 0.01,
	"sdr_labeled_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id_fb_data": 0,
	"bgp_subnet_score_rule_based_labels": 0.56907998735378,
	"time_spent_1d_1_forecast_value": 0,
	"bgp_subnet_score_netacuity_proxy_fb_data": 0,
	"request_count_1d": 43.57868989483,
	"endpoint_ratio": 0.27726344704115,
	"bgp_subnet_score_young_devices_fb_data": 1569915645.6024,
	"num_users_1_forecast_value": 0,
	"num_requests_1_forecast_value": 0,
	"user_count_on_ip": 0,
	"num_labeled_requests_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id": 0.40450725744843,
	"estimated_turnstile_count": 5.5,
	"bgp_subnet_score_old_asbd_header_version": 0.16701030927835
	}
		{"context_features":{},"continuous_features":{"num_labeled_requests_1_forecast_value":0,"num_requests_1_forecast_value":0,"num_users_1_forecast_value":0,"sdr_1_forecast_value":0,"sdr_labeled_1_forecast_value":0,"signup_events_1_forecast_value":0,"time_spent_1d_1_forecast_value":0,"log_request_count_1d":1,"log_user_count_on_ip":0}}

Only log_request_count has non-zero attribution

Why is *this* IP blocked?

Block

recommended_action	continuous_features	feature_attribution

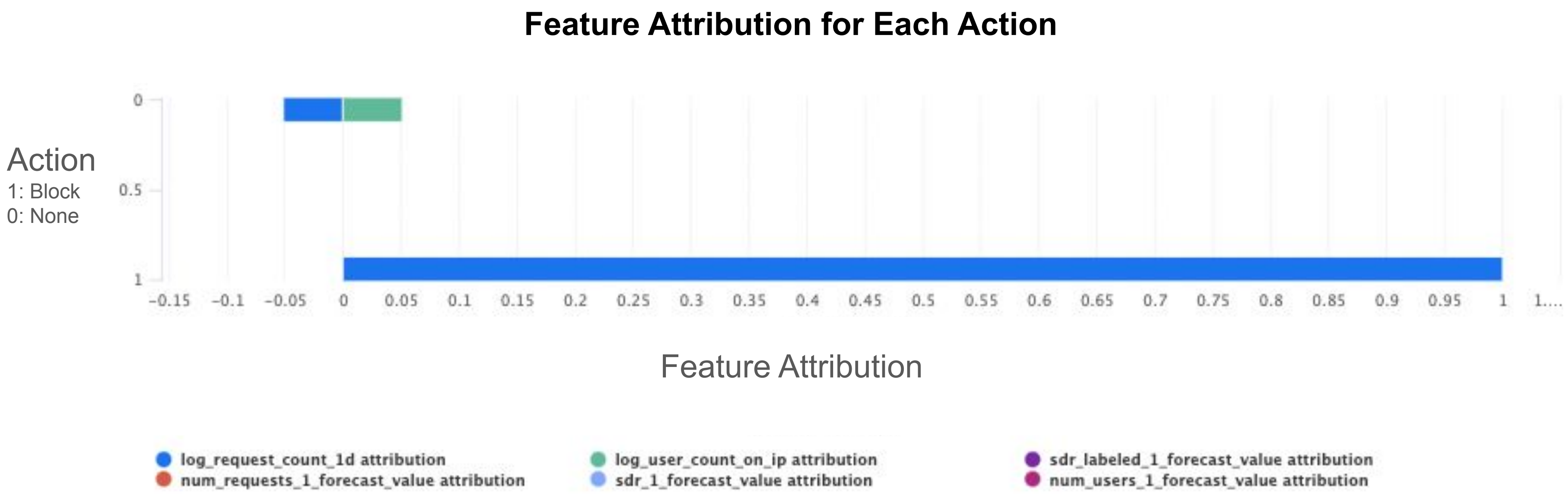
1.0	{
	"bgp_subnet_score": 0.060089707,
	"bgp_subnet_score_account_endpoints": 0.24628517230477,
	"bgp_subnet_score_not_enough_logging_fb_data": 1,
	"sdr_1_forecast_value": 0,
	"log_request_count_1d": 3.7972559397922,
	"log_user_count_on_ip": 0,
	"bgp_subnet_score_account_endpoints_unfiltered": 0.068373071528752,
	"bgp_subnet_score_young_devices": 23807173.754865,
	"signup_events_1_forecast_value": 0,
	"cloud_hosting_score": 0.01,
	"sdr_labeled_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id_fb_data": 0,
	"bgp_subnet_score_rule_based_labels": 0.56907998735378,
	"time_spent_1d_1_forecast_value": 0,
	"bgp_subnet_score_netacuity_proxy_fb_data": 0,
	"request_count_1d": 43.57868989483,
	"endpoint_ratio": 0.27726344704115,
	"bgp_subnet_score_young_devices_fb_data": 1569915645.6024,
	"num_users_1_forecast_value": 0,
	"num_requests_1_forecast_value": 0,
	"user_count_on_ip": 0,
	"num_labeled_requests_1_forecast_value": 0,
	"bgp_subnet_score_missing_device_id": 0.40450725744843,
	"estimated_turnstile_count": 5.5,
	"bgp_subnet_score_old_asbd_header_version": 0.16701030927835
	}
		{"context_features":{},"continuous_features":{"num_labeled_requests_1_forecast_value":0,"num_requests_1_forecast_value":0,"num_users_1_forecast_value":0,"sdr_1_forecast_value":0,"sdr_labeled_1_forecast_value":0,"signup_events_1_forecast_value":0,"time_spent_1d_1_forecast_value":0,"log_request_count_1d":1,"log_user_count_on_ip":0}}

Only log_request_count has non-zero attribution

Is this insight **generalizable** to other samples?

Why is *any* IP blocked?

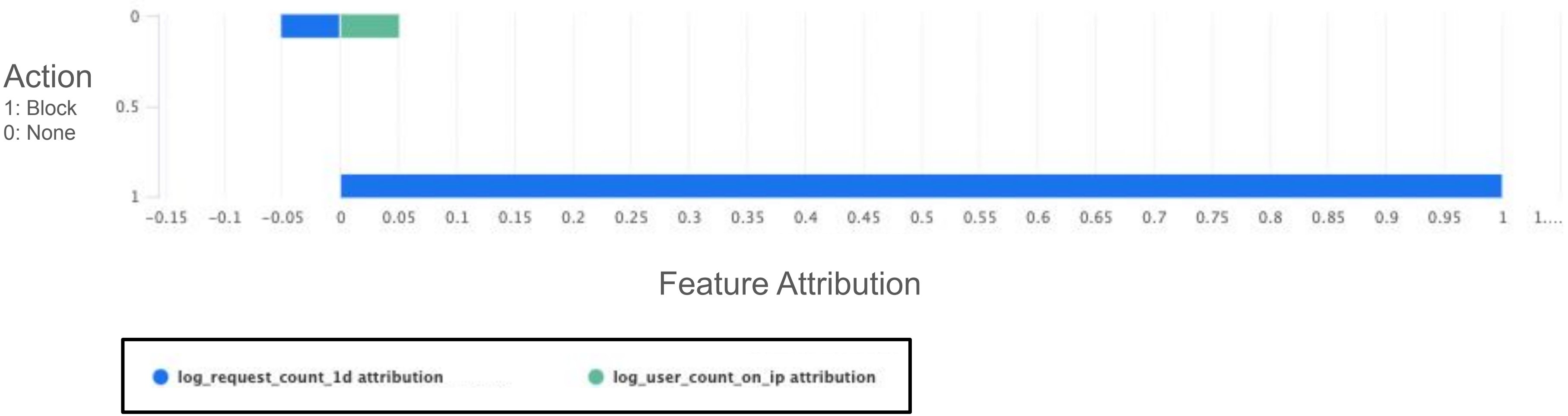
- Aggregate the feature attribution over all samples



Why is *any* IP blocked?

- Aggregate the feature attribution over all samples

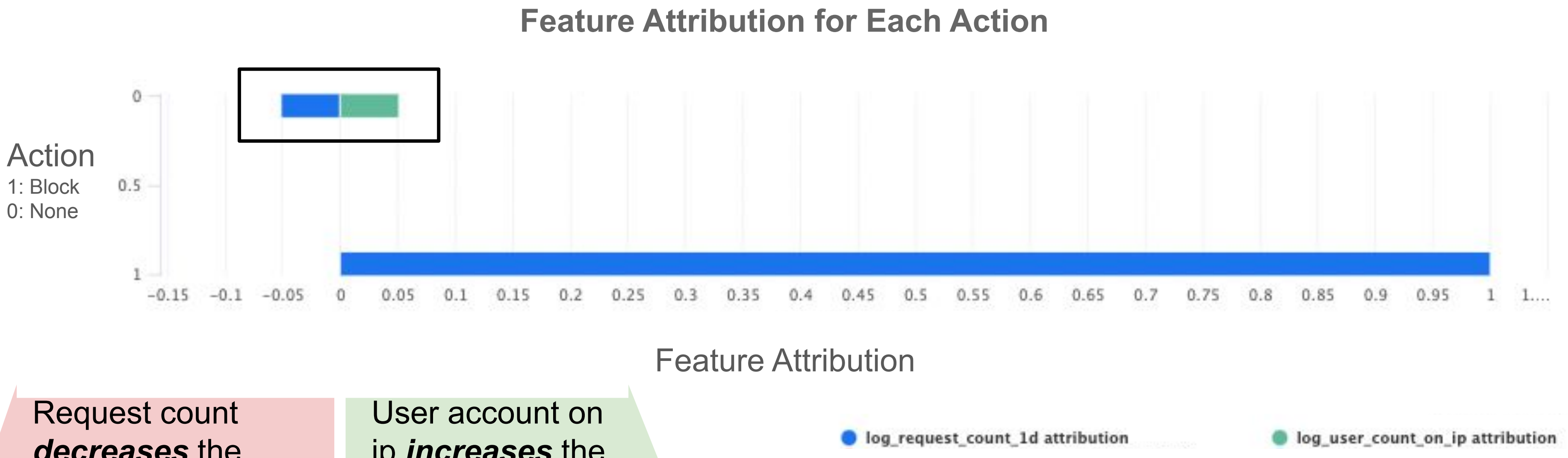
Feature Attribution for Each Action



Only “request count” and “user count on ip” affect PRO’s decision on IG logged out

Why is *any* IP blocked?

- Aggregate the feature attribution over all samples

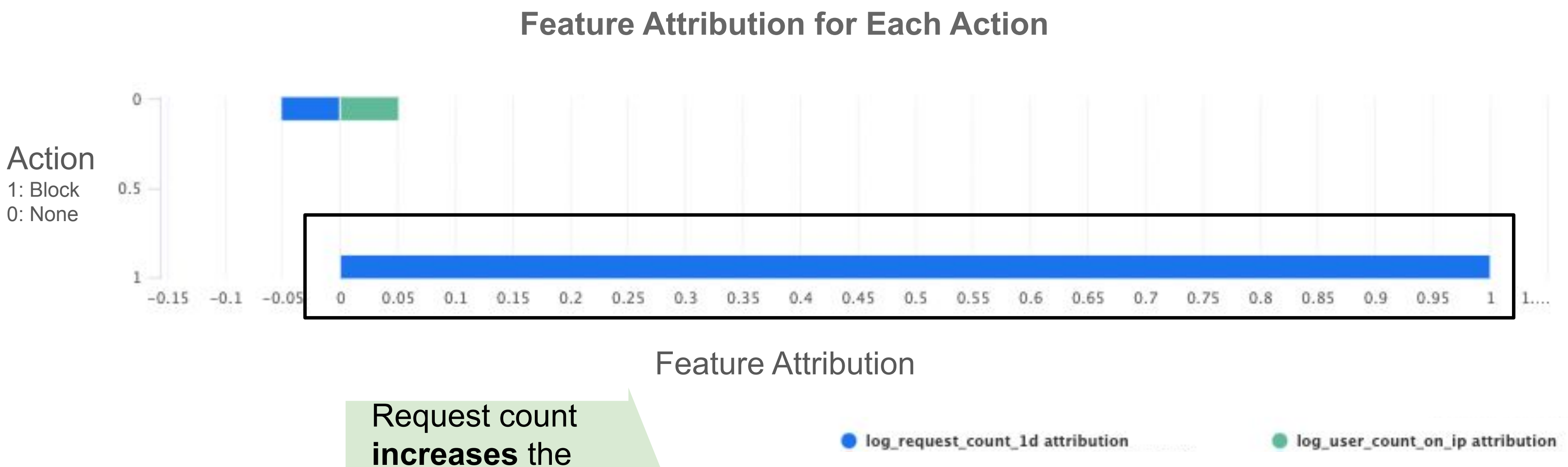


Request count ***decreases*** the chance of no response

User account on ip ***increases*** the chance of no response

Why is *any* IP blocked?

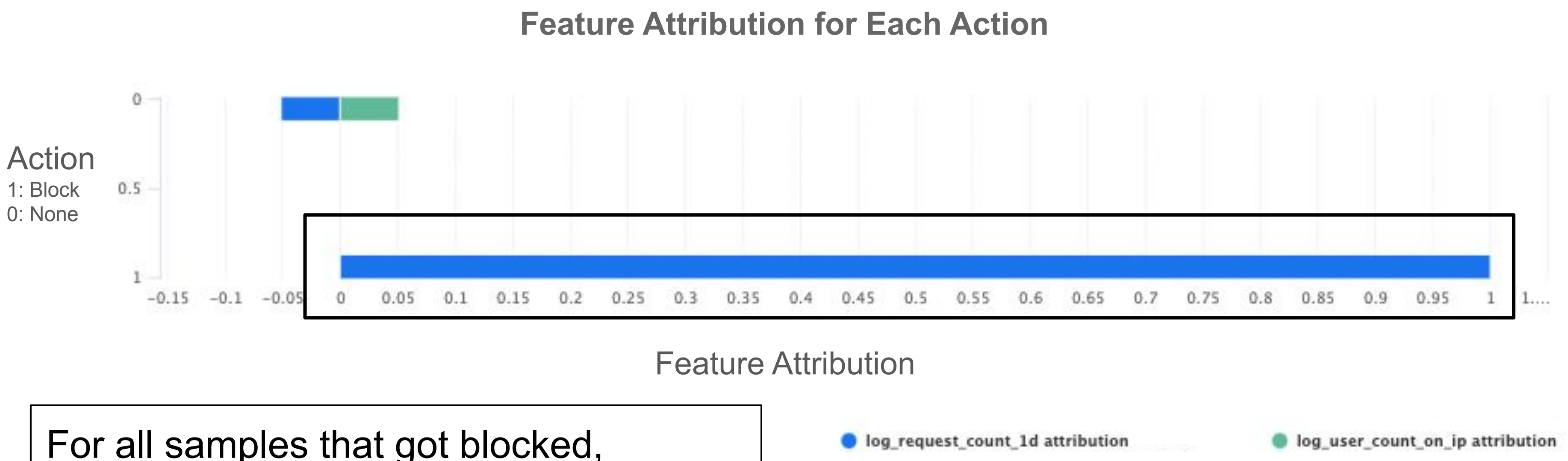
- Aggregate the feature attribution over all samples



Request count **increases** the chance of getting blocked

Why is *any* IP blocked?

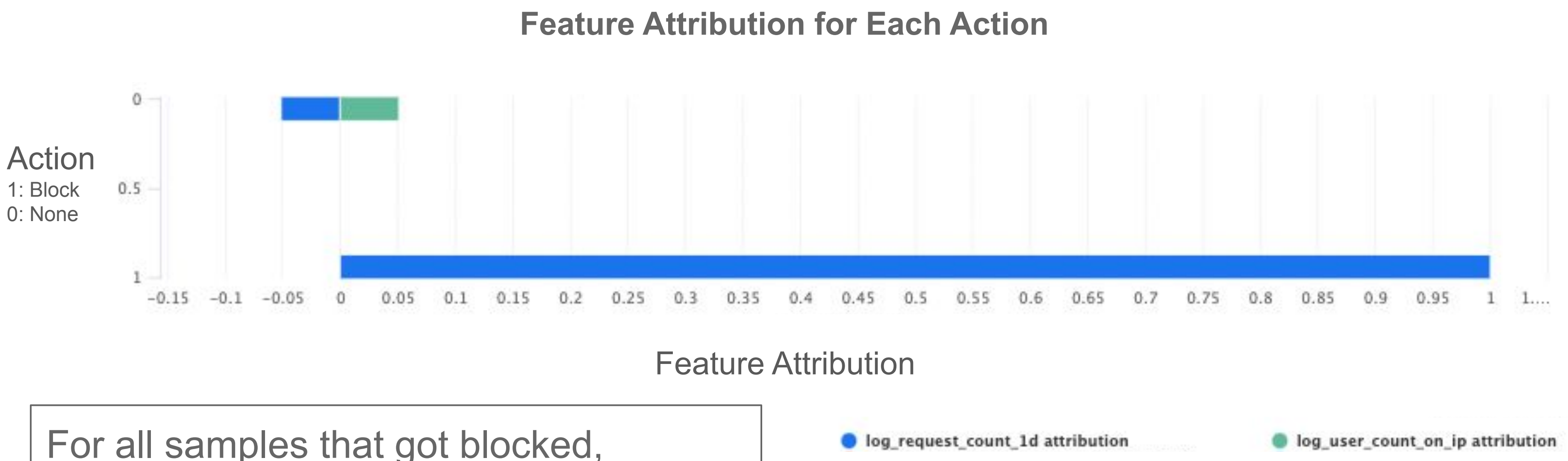
- Aggregate the feature attribution over all samples



For all samples that got blocked, log_request_count_1d is the culprit

Why is *any* IP blocked?

- Aggregate the feature attribution over all samples

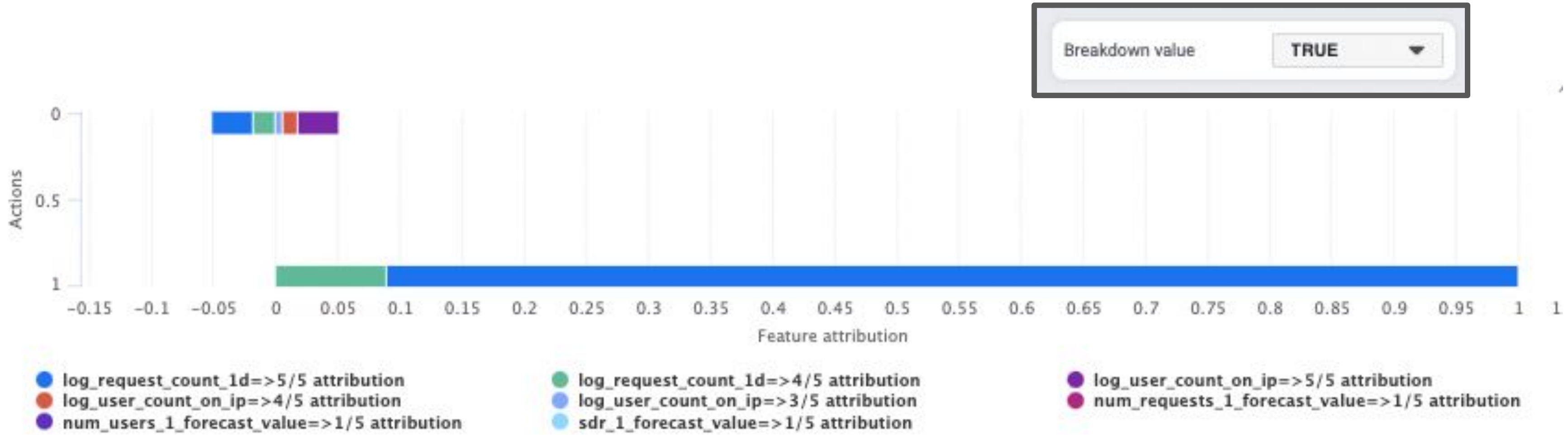


For all samples that got blocked, log_request_count_1d is the culprit

..How does request count leads to my blocking?

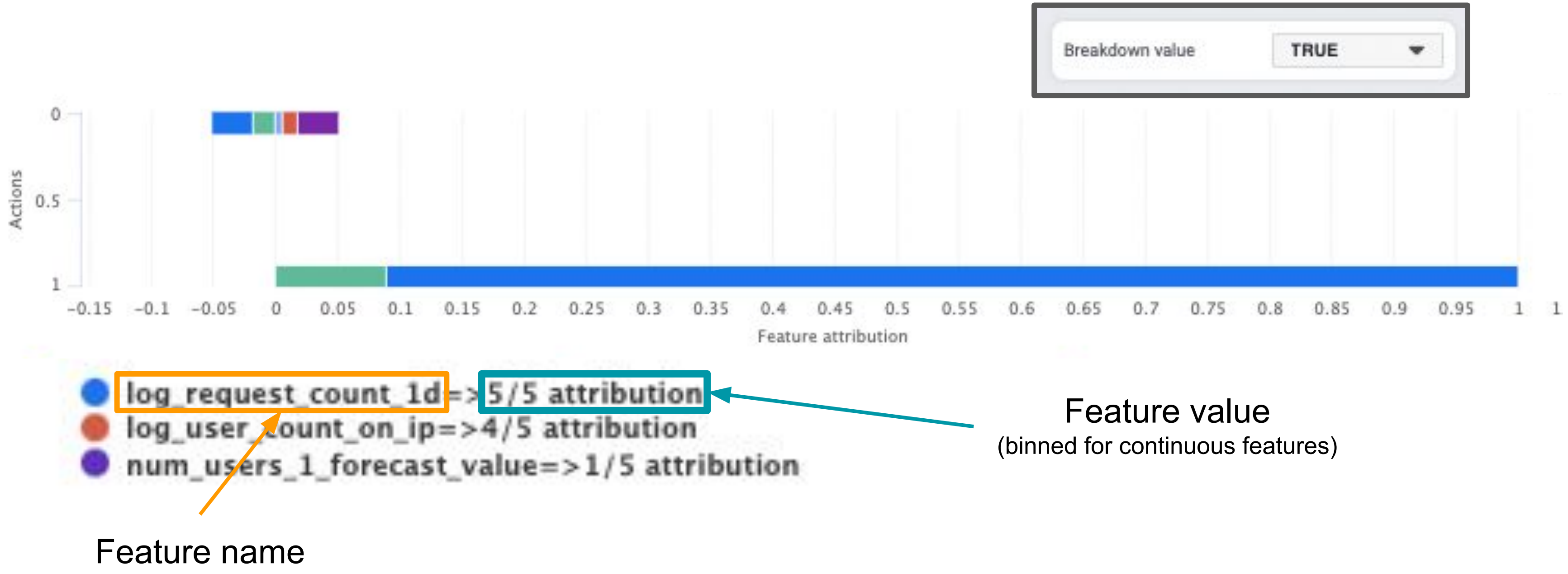
Breaking down feature attribution by values

- Taking a closer look at “**log_request_count**” and “**log_user_count**”



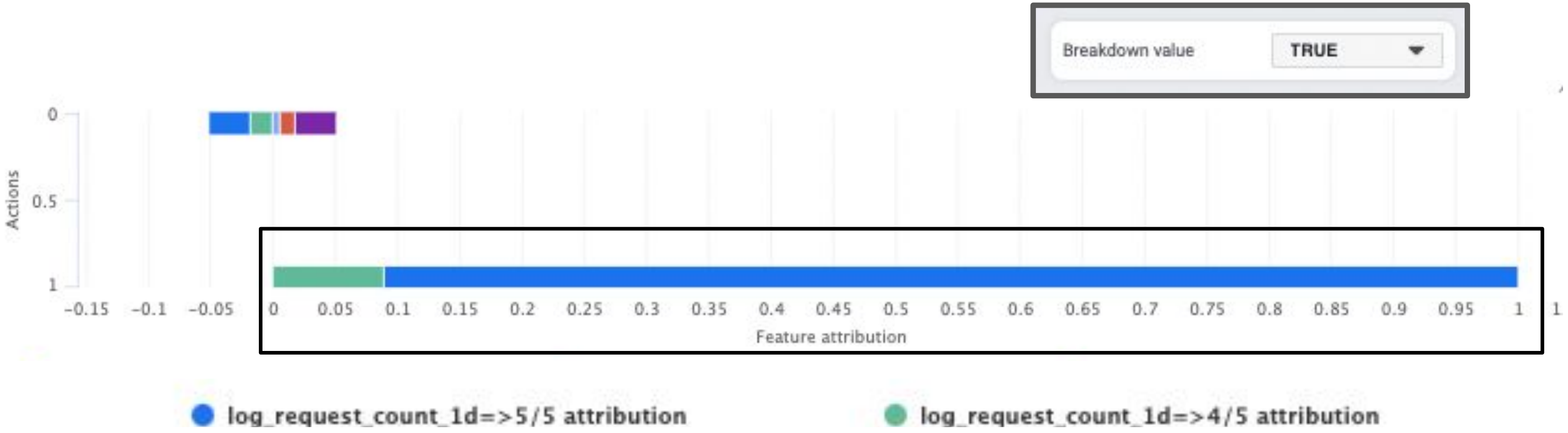
Breaking down feature attribution by values

- Taking a closer look at “log_request_count” and “log_user_count”



Breaking down feature attribution by values

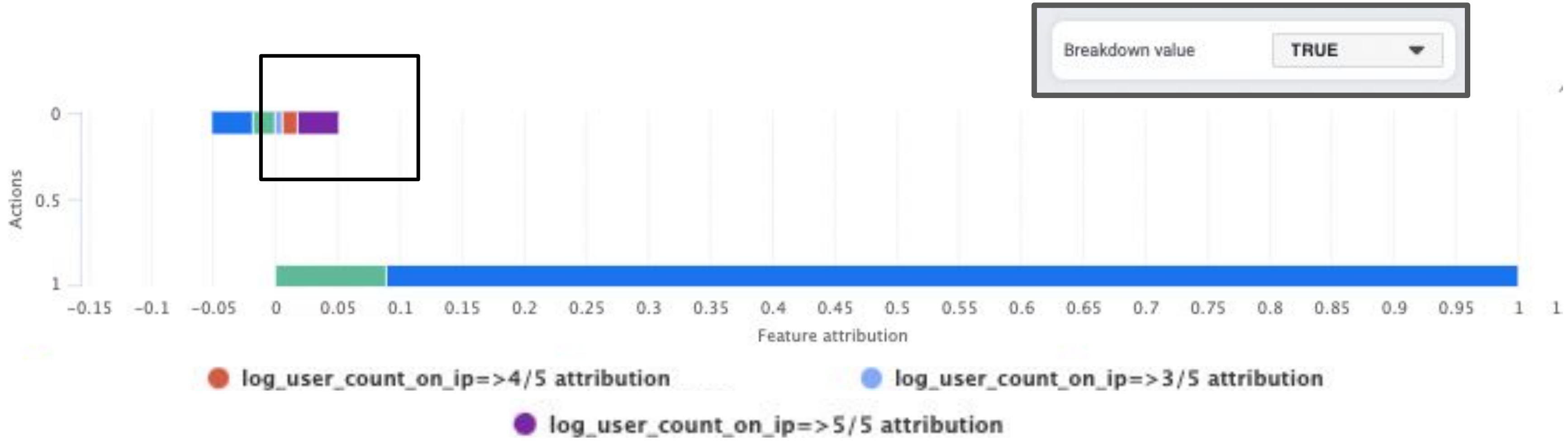
- Taking a closer look at “log_request_count” and “log_user_count”



High request counts lead to blocking!

Breaking down feature attribution by values

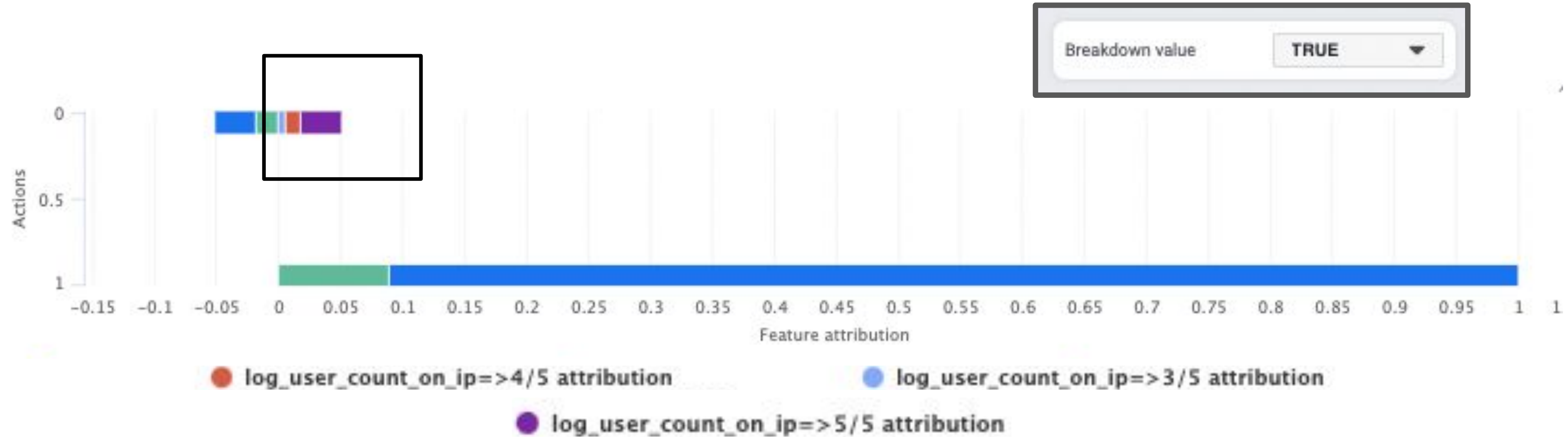
- Taking a closer look at “log_request_count” and “log_user_count”



High user count on ip get you unblocked!
→ *IP addresses with high request count but low user count on IP are blocked*

Breaking down feature attribution by values

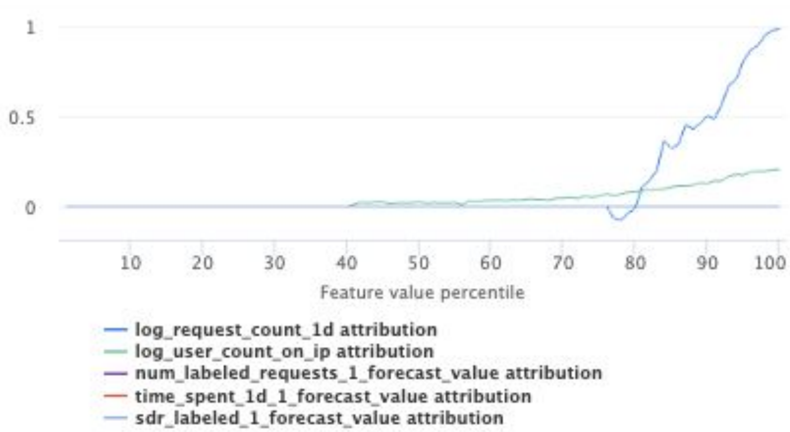
- Taking a closer look at “log_request_count” and “log_user_count”



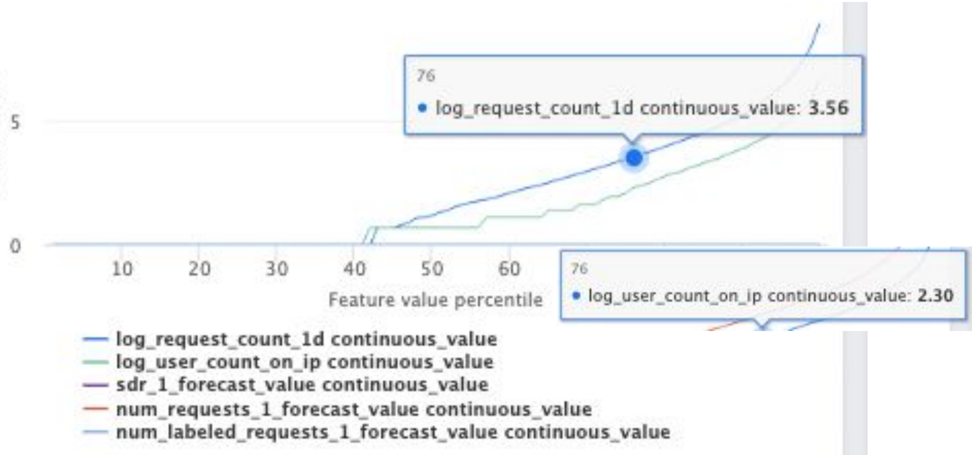
.. Can we be more precise?

Continuous break down of features by percentile

Feature Attribution

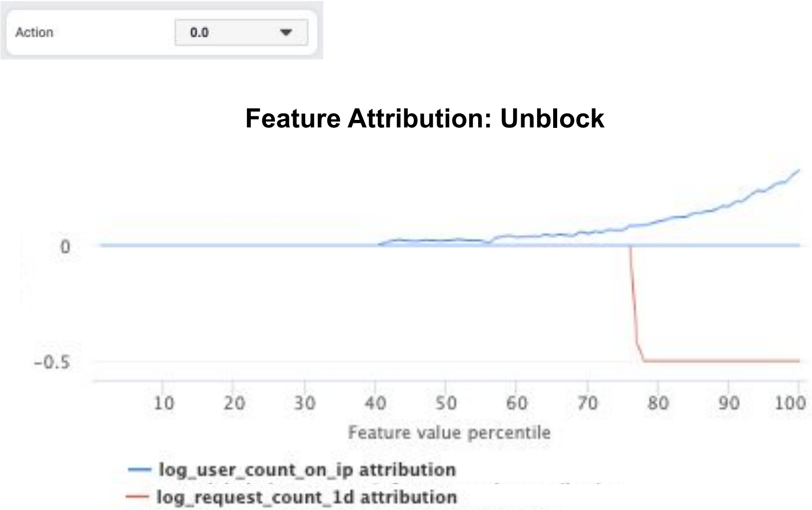


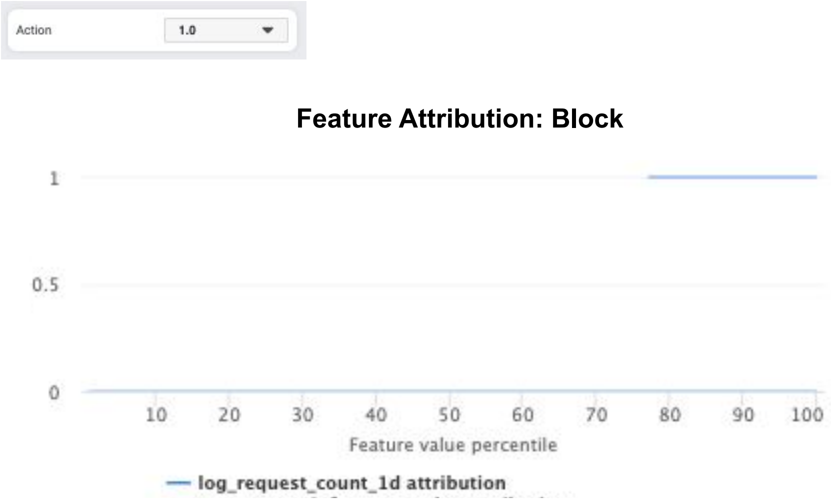
Feature Value



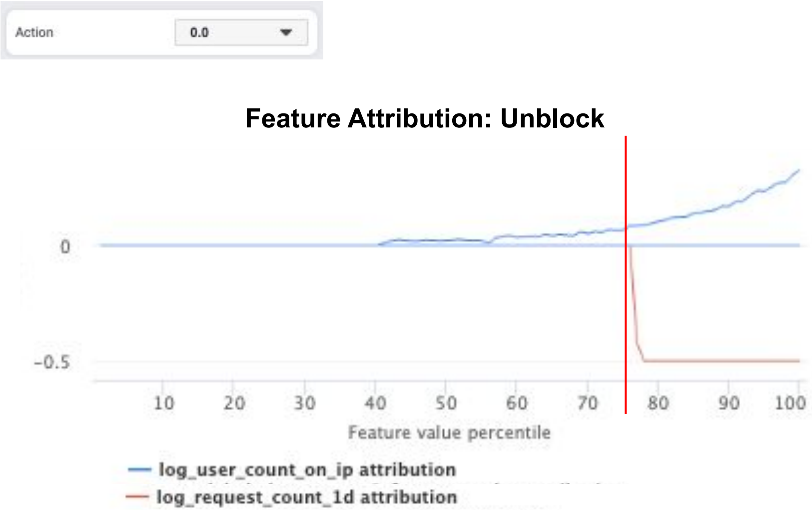
~75 Percentile
log_request_count_1d: 3.56
log_user_count_on_ip: 2.30

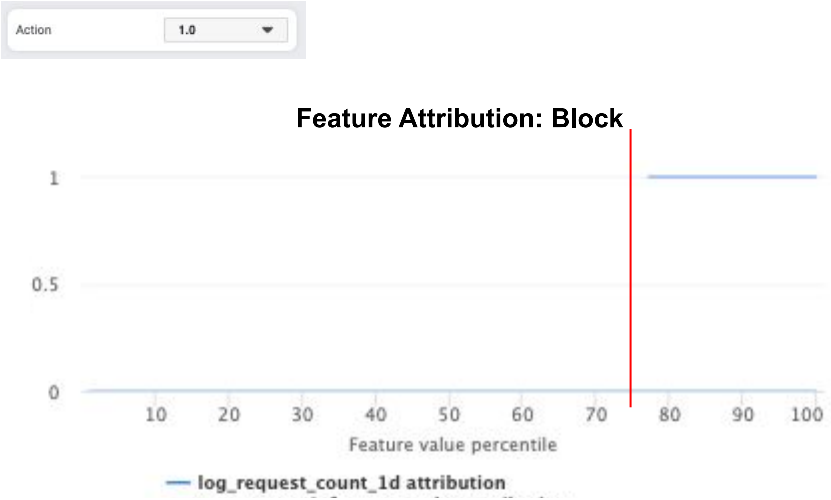
Continuous break down of features by percentile





Continuous break down of features by percentile

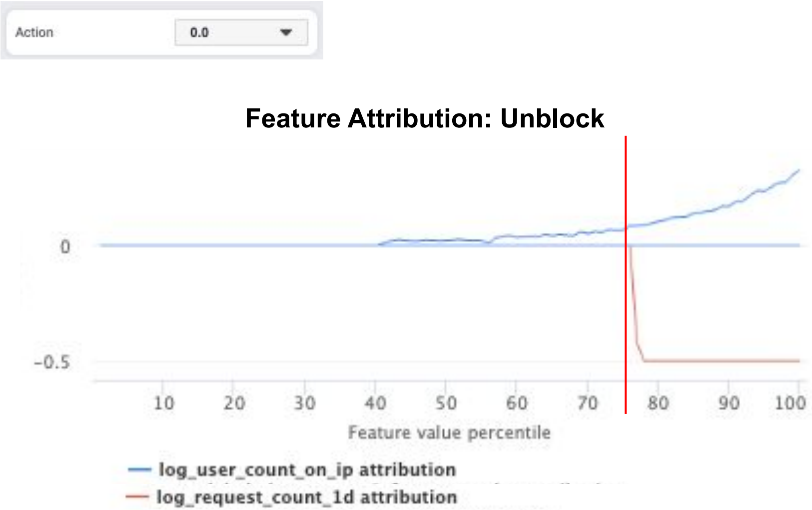


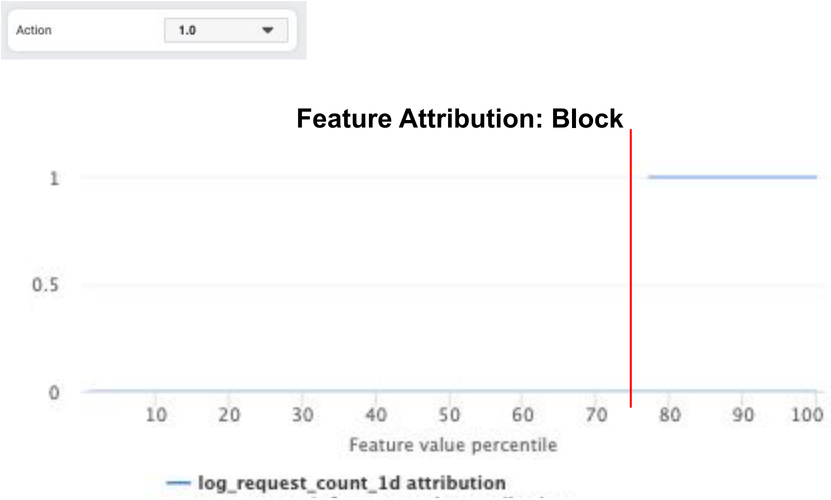


log user count > 75% → unblocking

log request count > 75% → blocking

Continuous break down of features by percentile





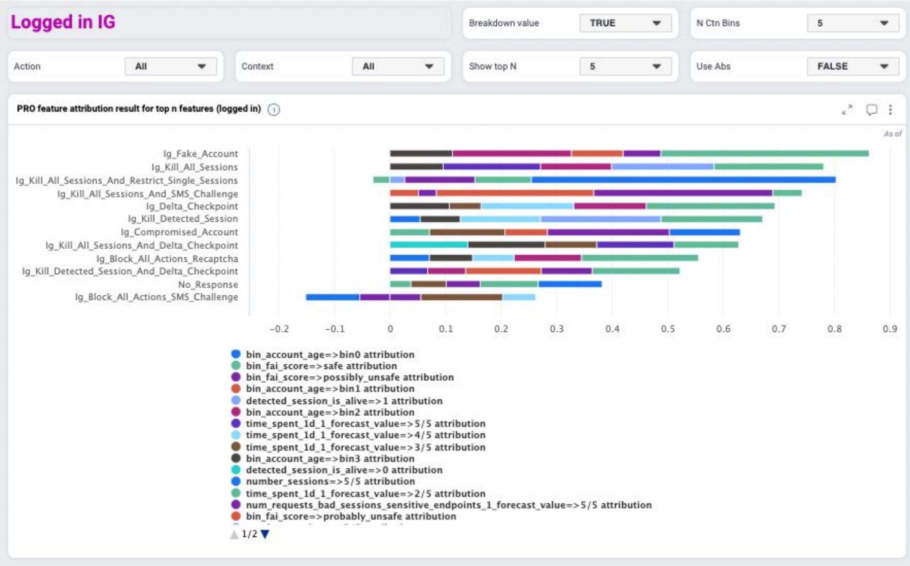
log user count > 75% → unblocking

log request count > 75% → blocking

IP addresses with log request count > 3.60 and log user count on IP < 2.3 are likely blocked

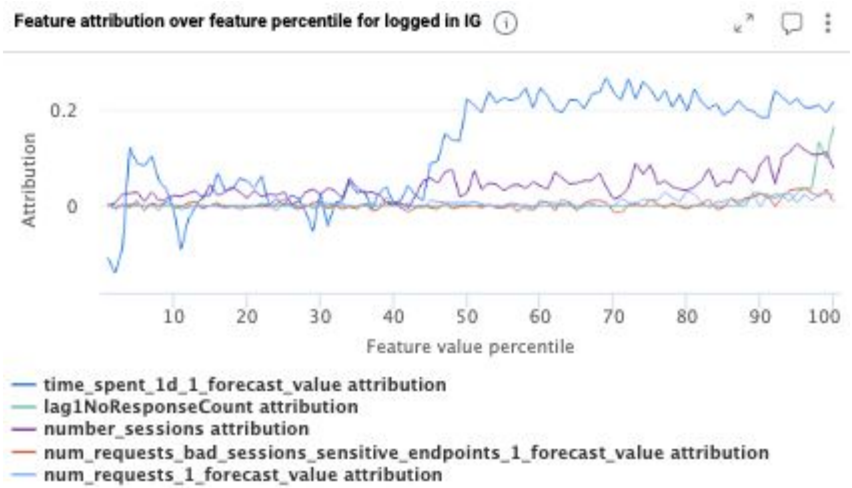
~75 Percentile
log_request_count_1d: 3.56
log_user_count_on_ip: 2.30

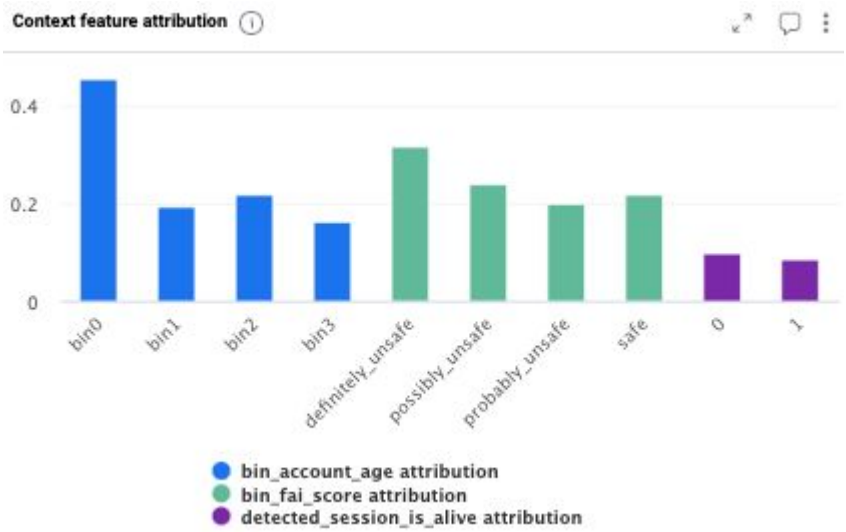
Similar idea applies to IG logged in ..



- Showing top N features for each action
- Custom filters for:
  - Actions
  - Context features
- Customize:
  - Number of features to show
  - Whether to use absolute transformation on attribution
  - Whether to break down attribution by feature value
  - Number of bins used for continuous feature breakdowns

Similar idea applies to IG logged in ..





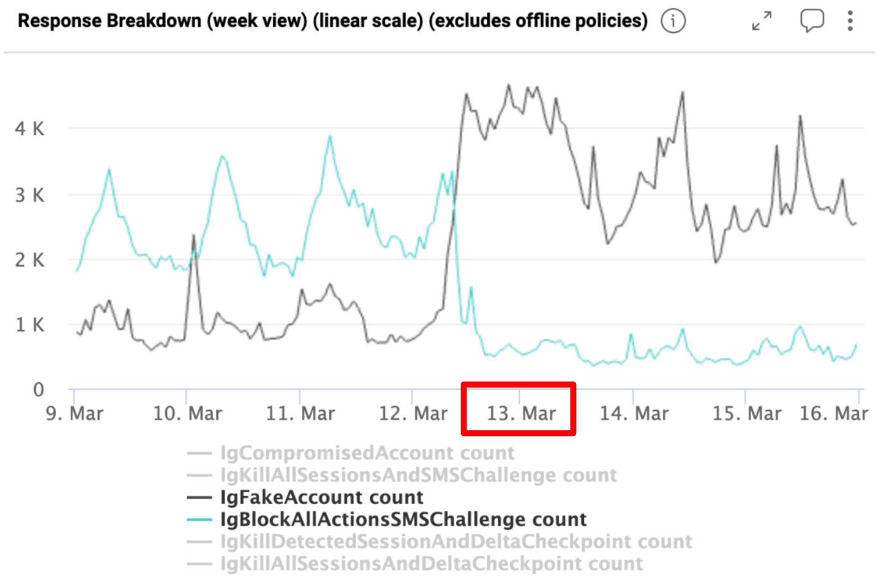
We can track both important **continuous** and **context/categorical** features

Why did a response spike?

(Monitoring Distribution Shifts)

Identifying a response spike

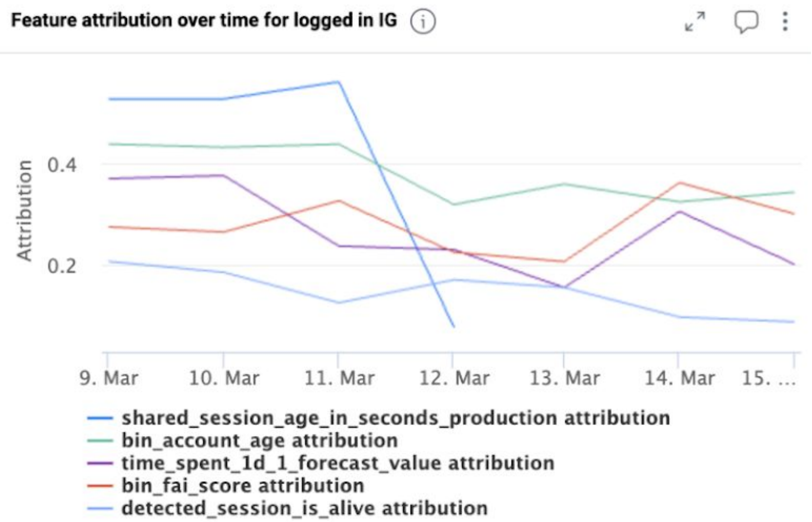
We observe a shift in the “**IgFakeAccount**” and “**IgBlockAllActionsSMSChallenge**” response



Response spikes can be <u>caused</u> by shifts in the <u>Covariate and/or Conditional Distribution</u>

Monitoring shifts in the **Conditional** distribution

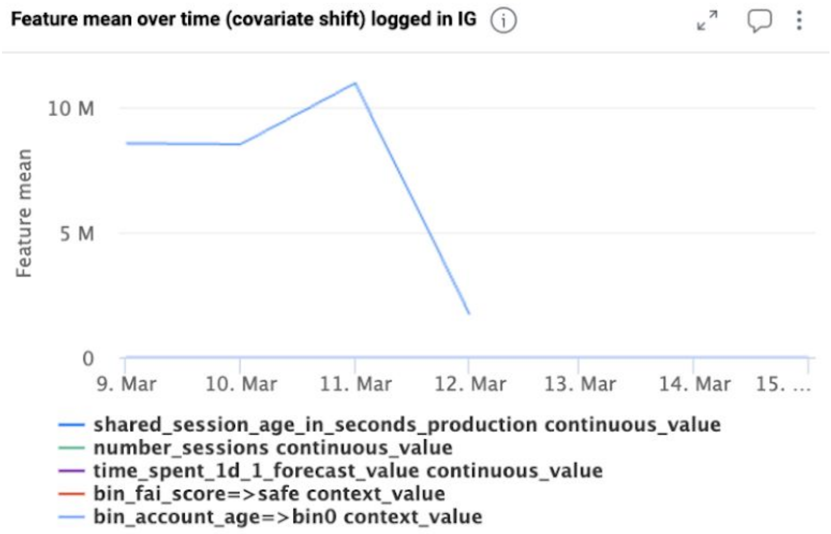
Conditional Distribution
(Y|X, “the learned relationship”)



- Changes to PRO’s learned model can be monitored using <u>*Feature Attribution*</u>.
- Tracks top N features along time, plotting feature attribution

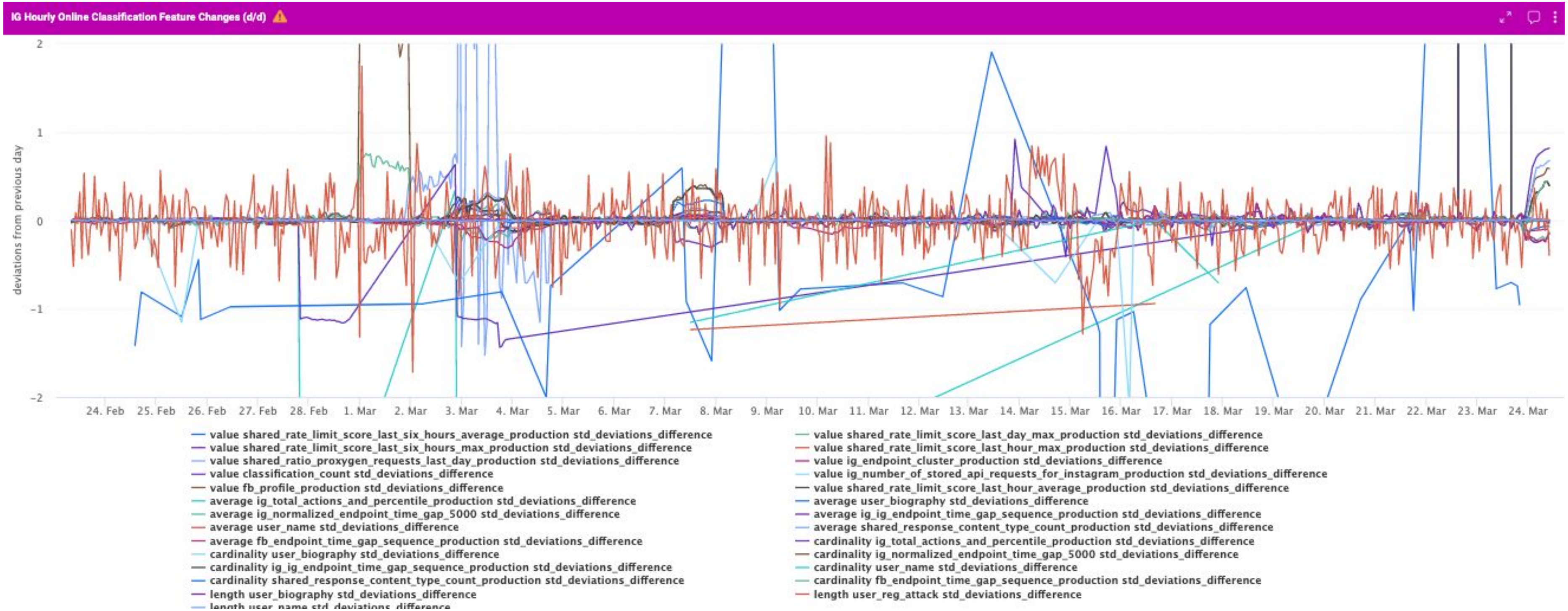
Monitoring shifts in the **Covariate** distribution

Covariate Distribution
(X, the inputs)

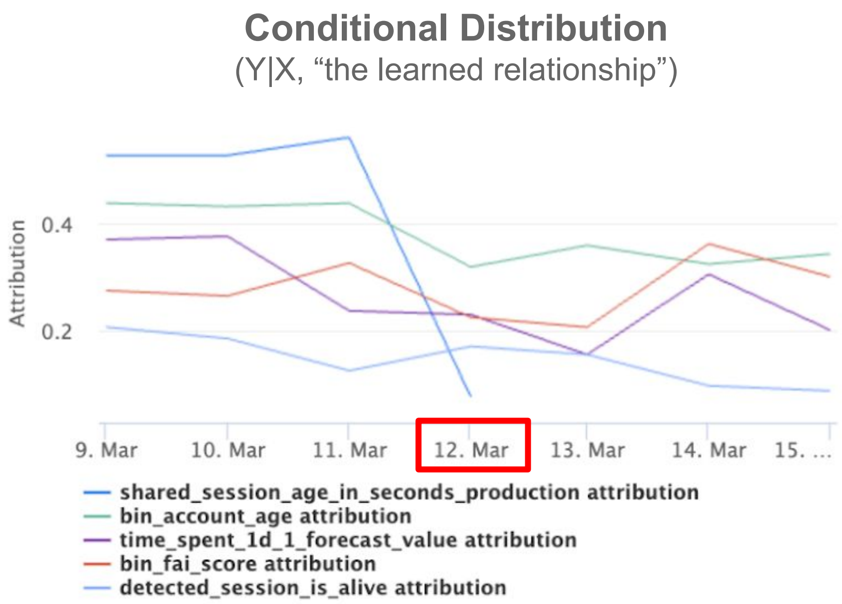


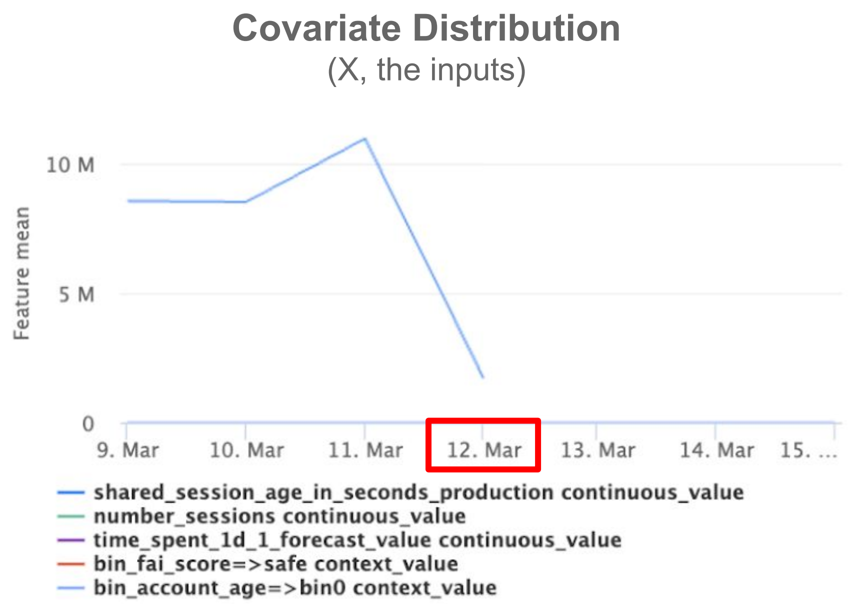
- Changes in the input distribution can be monitored using <u>feature mean</u>.
- Tracks top N features along time, plotting feature mean.
- We only track features that are important to avoid cluttering the plot

Messy input distribution



Why did responses spike on Mar 13?

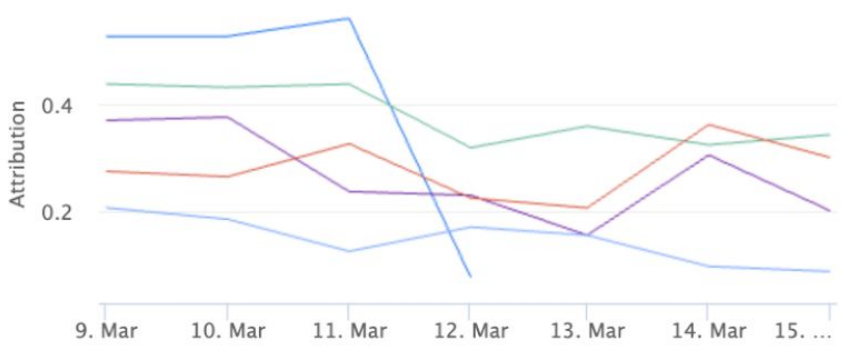




Covariate and Conditional Shift of “shared_session_age_in_seconds” feature on 12. Mar

Using Dashboard as a Starting Point for Further Investigation

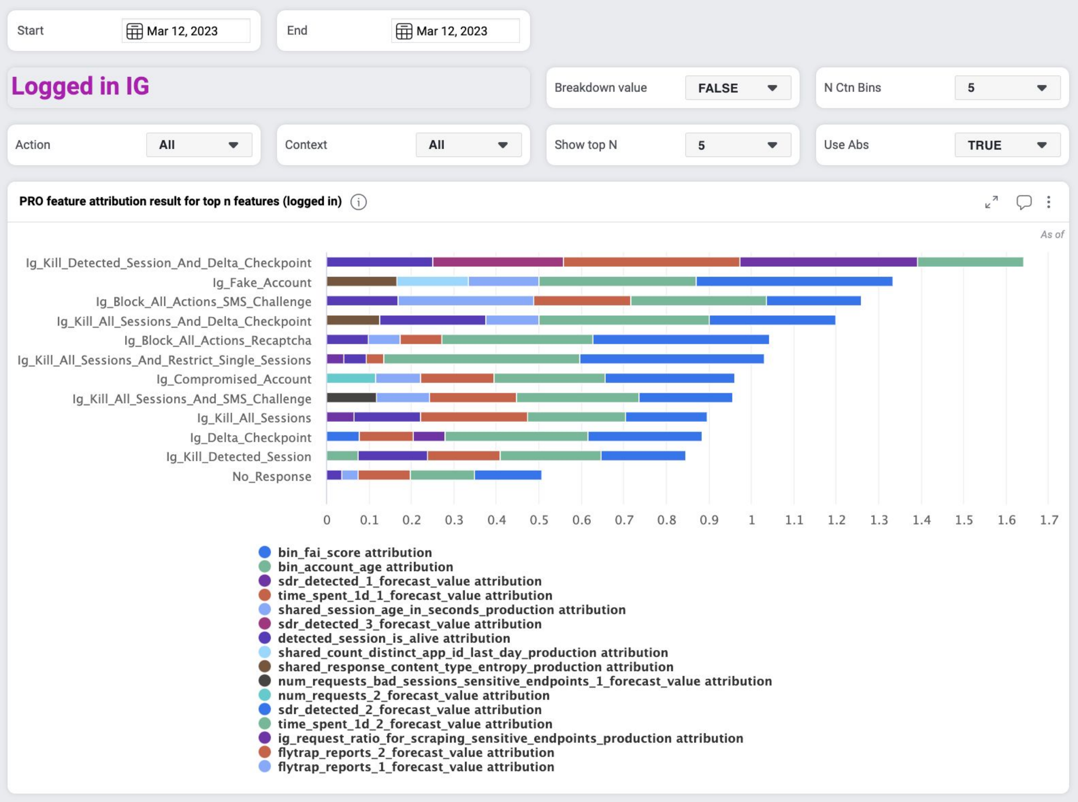
Conditional Distribution
(Y|X, “the learned relationship”)



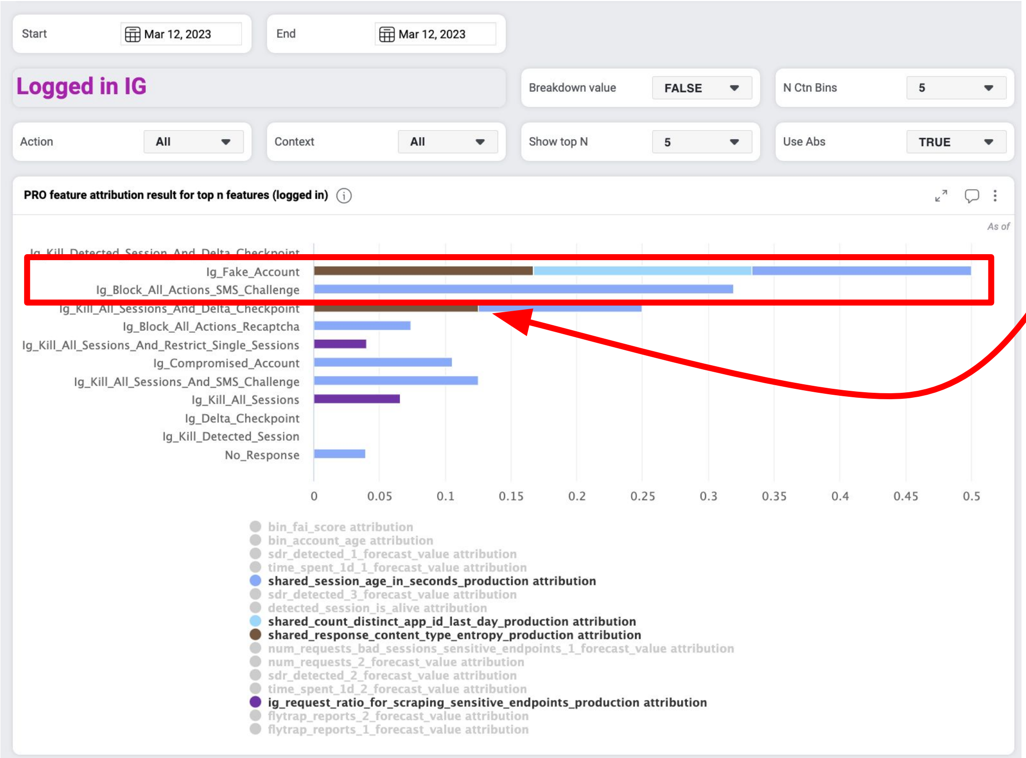
Covariate Distribution
(X, the inputs)

- “shared_session_age_in_seconds” was removed from the model due to capacity issues in training (D43843906) along with all other features starting with “shared”.
- The removal of the feature was made on 3/6/2023 but apparently the model was still using it until 3/12/2023.

Why did a response spike?



Why did only “IgFakeAccount” & “IgBlockAllActionsSMSChallenge” spike?



Response Specific Feature Attribution

“IgFakeAccount” and “IgBlockAllActionsSMSChallenge” rely on the deleted features the most.

Note: the tool currently cannot predict whether the response will spike up or down.

Oncall use

Reactive use (SEV):

- Check which important features shifted before and after the SEV using the dashboard
- Determine if it is conditional or covariate shift
  - *If covariate shift*.. investigate change in the features
    - Has it been deleted? Is the precision of the feature recently dropped?
  - *If conditional shift..* investigate changes in model training and deployment
    - e.g., checking changes in metric, model, objective etc.

Proactive use:

- Set up alert on the feature attribution or covariate shift plots.
- If out of the normal range, investigate as reactive use above.

Documentation

Wiki > Privacy > External Data Misuse (EDM) > Scraping Threat Mitigation > Intervention > Predictive Response Optimization > Monitoring > PRO feature attribution

Jiaxuan Wang
Page score: 100 · Updated on: March 22, 2023 · Viewers: 10

PRO feature attribution

PRO feature attribution dashboard user guide

Imported from

Overview

Predictive Response Optimization (PRO) is a machine learning system using reinforcement learning to fight unauthorized scraping. For IG logged in, PRO decides what responses to issue to users who are suspected to be scrapping. For IG logged out, it decides whether to block requests from an ip address. Despite its importance, for long, we treat PRO as a black box, making it hard to understand its logic to debug the system. Feature attribution is one approach to fill this void.

Feature attribution has the following benefits

- It improves model understanding and enhances defensibility of our decisions

  For assessor requests for safeguard evidencing we often get questions around why a particular response was chosen for a user and the most important features which influence PRO's decision. With feature attribution in place, we can answer those questions (see use case 1).

- It helps monitoring distribution shifts in the production traffic

  This will aid debugging when response distribution spikes or when online MSE degrades (focus our attention on the important shifting feature, see use case 2).

The purpose of this wiki is to a) explain how we compute feature attribution so that you have the necessary terminology to understand our feature attribution dashboard (methodology section), b) showcase a few use cases of the dashboard (example use cases section), and c) document all the tools provided by the dashboard (reference section).

Feature attribution dashboard ("go pro_attribution") currently supports

- **A**: Visualizing most important features
- **B**: Tracking feature attribution and value shifts
- **C**: Visualizing decision logic (attribution over value)
- **D**: Monitoring the volume of logged feature attribution

for both IG logged in and out. Furthermore each of the aforementioned features can be filtered by action and context features.

There are additional contexts on design choices and tools built around feature attribution besides the dashboard, which are tracked in T128501575.

Methodology

We use SHAP[1] to obtain local feature attribution, that is attributing PRO's decision for each sample to the features used by PRO. For example, to answer the question why an ip got blocked for IG logged out, local feature attribution could reveal that the ip has too many requests for the day and that's why PRO blocked it.

We implement SHAP online on WWW. That is feature attribution is computed at the same time a live decision is made by PRO. Due to the expensive nature of computing SHAP, we only compute it for a small proportion of input.

Q: Why do we use SHAP for feature attribution?
A: We choose SHAP because it is model agnostic. PRO not only has categorical input, but also the business logic on top of the model's output (e.g., cooldown, allowed_actions) makes PRO's decision surface non smooth. We therefore cannot turn to methods such as integrated gradient or CAM that assumes access to gradient. Furthermore, being model agnostic allows our approach to continue working when the underlying machine learning model is changed. In addition to being model agnostic, SHAP is widely used and has game theoretic interpretation stemming from Shapley Value.

Wiki (main resource)

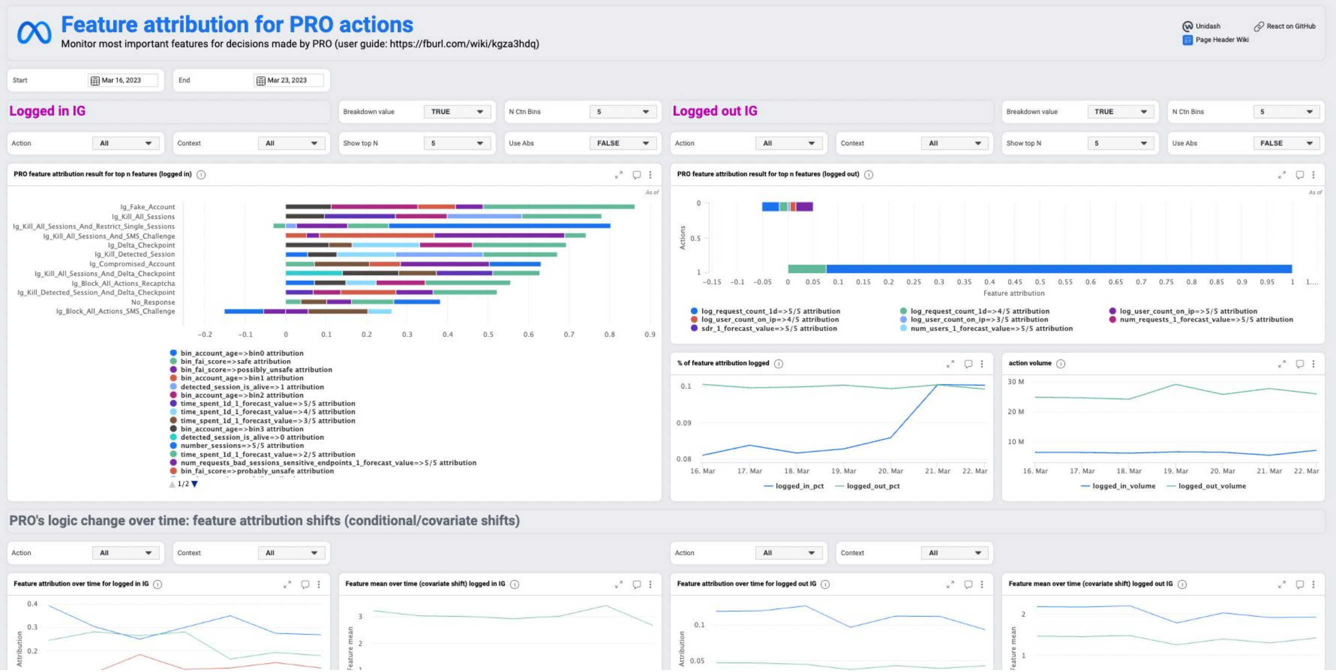
Bunnylol "go wiki_pro_attribution"

Covers

- Methodology: more details of SHAP
- Example usage
- Documentation for each panel
  - More detailed description
  - More panels not discussed today
- Future directions
- Q&A on miscellaneous topics

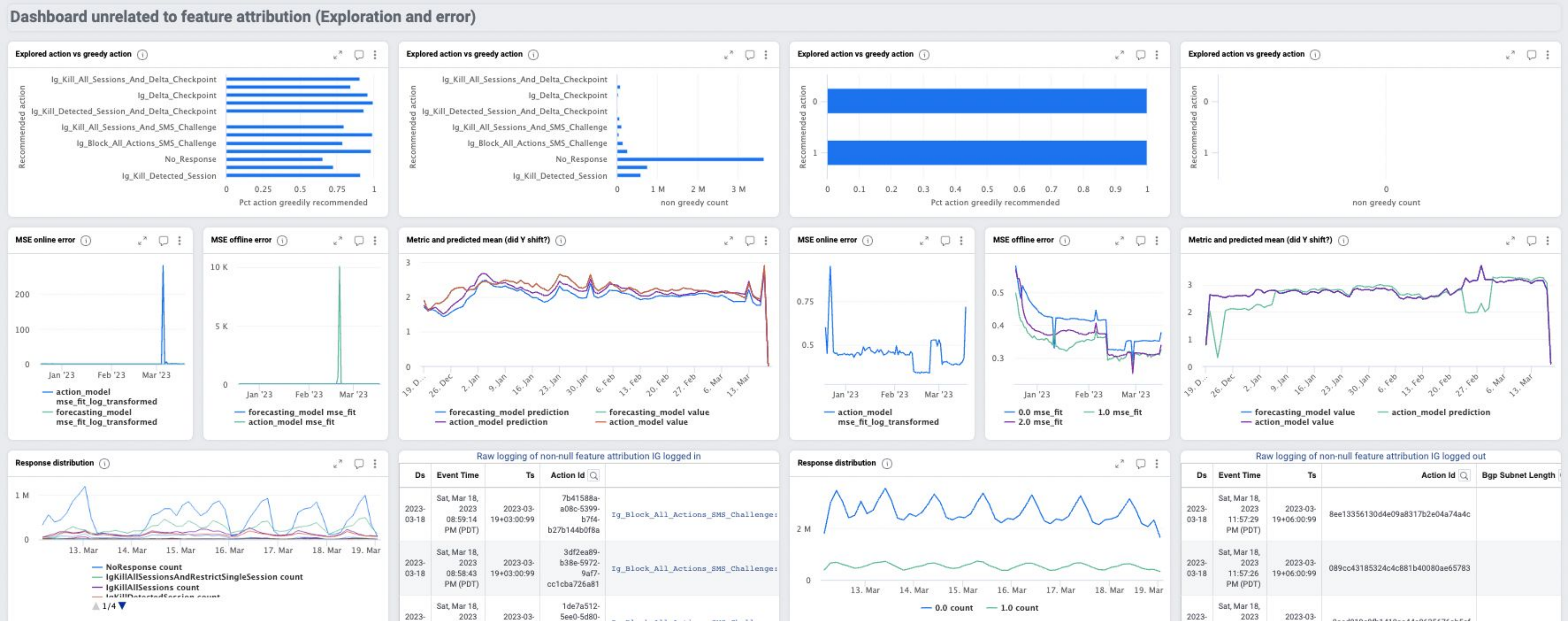
Demo

A tour of the panels: “go pro_attribution”



Questions

Set of other useful features



Example attribution: 40+ features (3/14)


```
{
  "context_features": {
    "bin_account_age": 0,
    "bin_fai_score": 0,
    "detected_session_is_alive": 0
  },
  "continuous_features": {
    "flytrap_reports_1_forecast_value": 0,
    "flytrap_reports_2_forecast_value": 0,
    "flytrap_reports_3_forecast_value": 0,
    "meaningful_engagement_score_1_forecast_value": 0,
    "meaningful_engagement_score_2_forecast_value": 0,
    "meaningful_engagement_score_3_forecast_value": 0,
    "meaningful_engagement_score_impressions_only_1_forecast_value": 0,
    "meaningful_engagement_score_impressions_only_2_forecast_value": 0
```