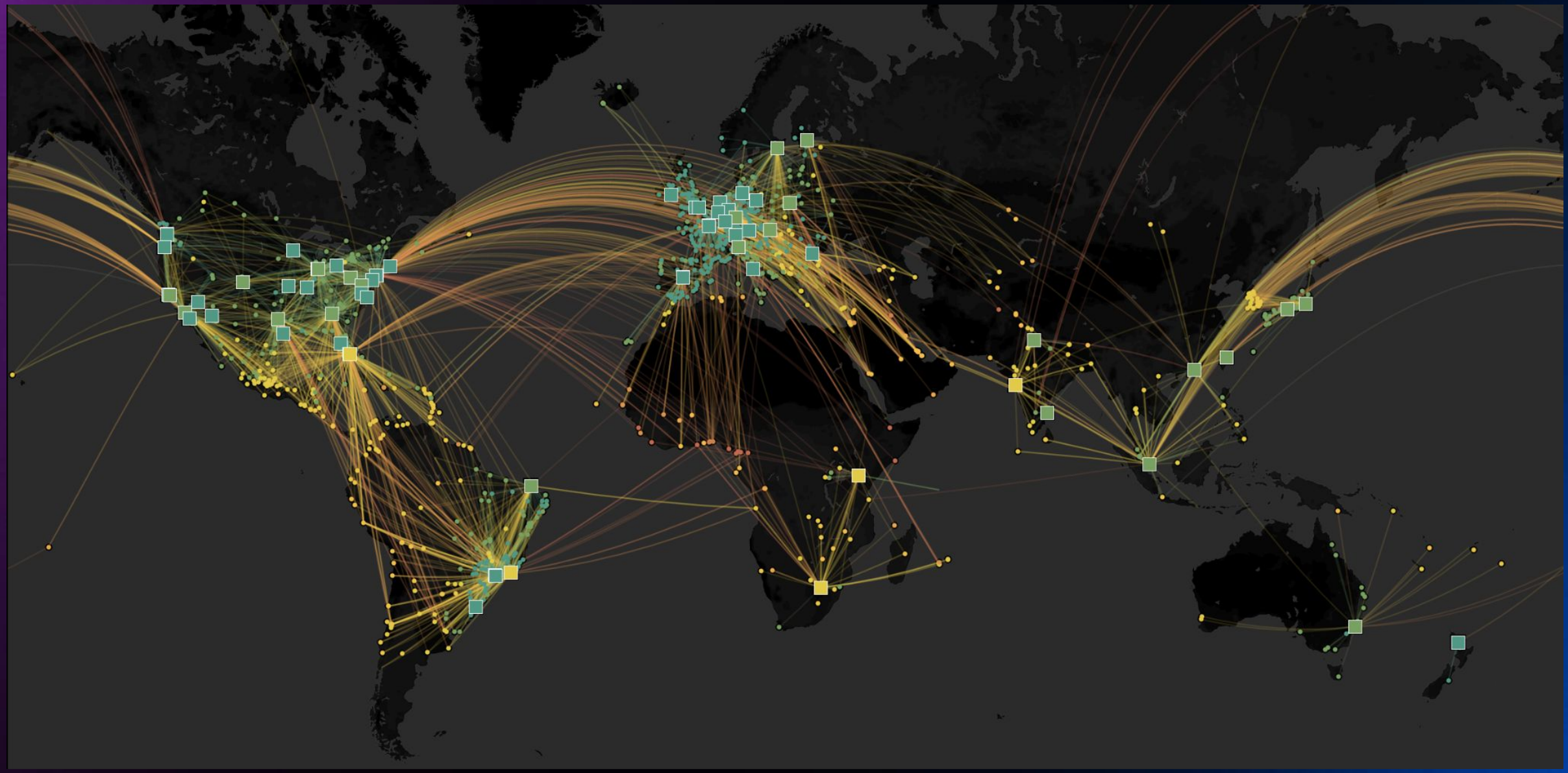
Using Statistical Techniques to Automatically Detect Game-Breaking Issues

Ian Neidel
CDN Reliability Engineer
March 25, 2025

CDN Reliability



Cloud Games

Unique Challenges

We’re used to measuring latency, bitrate, and dropped packets that measure how well we deliver content

But with cloud games, we must also be sure that *what* we deliver is fine as well

Motivation

An incident:
Our most popular game was broken in specific cases/countries for almost 3 weeks

An incident

An incident:
Our most popular game was broken in specific cases/countries for almost 3 weeks

Due to:

```
NetflixException: NGP Profiles response typename unknown
  at NextGames.Externals.Netflix.NetflixResultExtensions.GetOrThrow[T]
(Netflix.Games.NetflixResult`1[TResultType] result) [0x00000] in
<00000000000000000000000000000000>:0
```

An incident

How we found and root caused the issue:

Reading through free form feedback

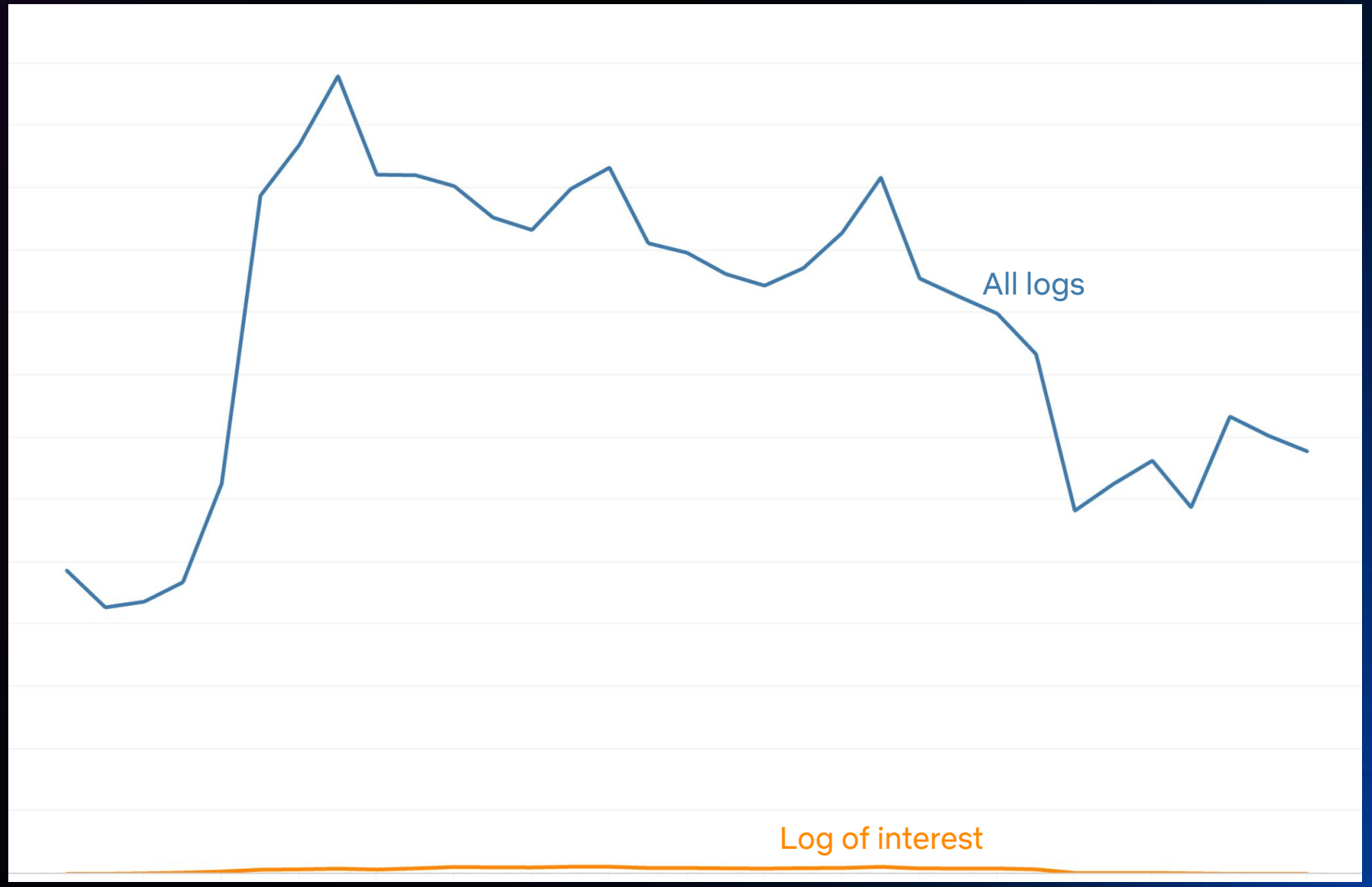
Free Form Response	Quality Score
Desespera que al pasar a otro nivel se desorganice todo	3
Improve adieo	1
It's cool	4
El juego no me carga	1
Nothing the game was great	5
Keep the games coming- love it	5
laggy	5
Glitch	5
No aparece el juego	1
Wouldn't load..	1
LE NOM DU GAME	2
Game didn't load	1

Then combing through error logs

log_message
UnityEngine.WorkRequest:Invoke() UnityEngine.UnitySynchronizationContext:Exec()
JSON (response): {"data":{"ngpProgress_slot":{"__typename":"NGPProgress_SlotNotFoundError"}}} Netflix.Games.Cloud.<>c__DisplayClass30_0`1:<CreateResponseCallback>b__0(String) Netflix.Games.Cloud.CloudNativeApi:HandleAsyncResponse(UInt64, String)
[725] Exception of type 'NextGames.Externals.Netflix.NetflixException' was thrown. Aqua.Client.StateManagement.UnityConsoleLogger:Debug(LogEntryDelegate) Aqua.Client.Booting.<CreateSession>d__9:MoveNext() System.Threading.ExecutionContext:RunInternal(ExecutionContext, ContextCallback, Object, Boolean) System.Runtime.CompilerServices.MoveNextRunner:Run() System.Threading.Tasks.AwaitTaskContinuation:RunCallback(ContextCallback, Object, Task&) System.Threading.Tasks.Task:FinishContinuations() System.Threading.Tasks.Task:TrySetException(Object)

An incident

The log in orange amidst all the rest



How can we do this better?

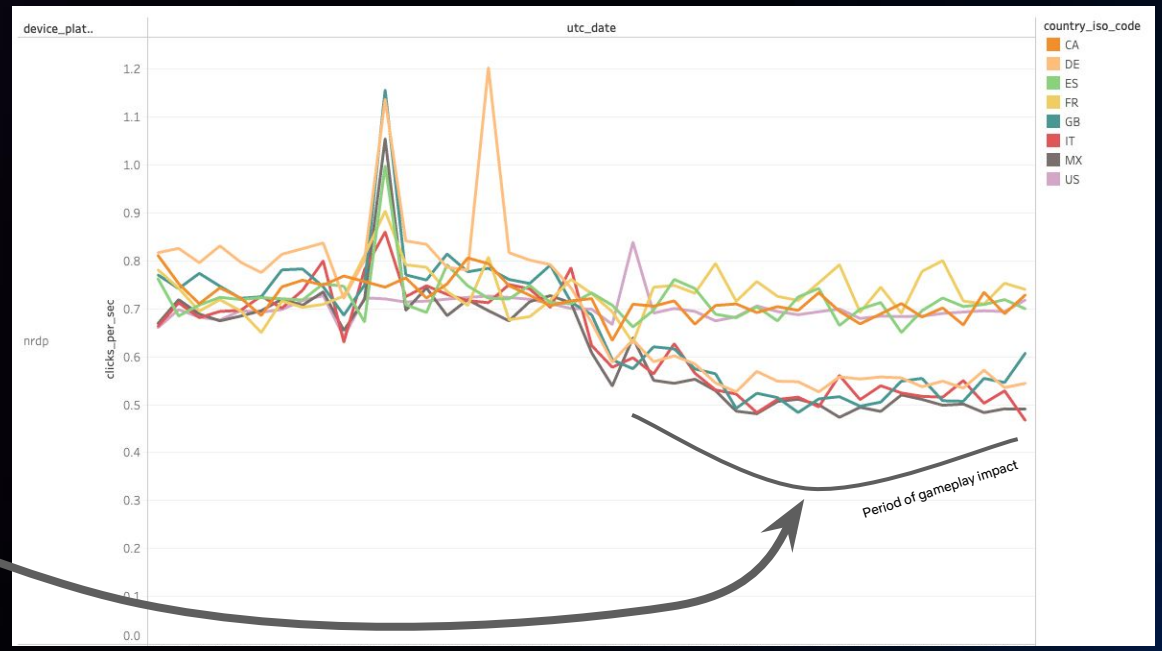
Given millions of error logs a day and no strong indicator a session is broken

How can we do this better?

Intersect broken session user behavior and potentially breaking logs to gain confidence in both

Indicators of a Broken Session

- Gameplay time
- Survey scores
- Freeform feedback
- Bitrates
- Click rates

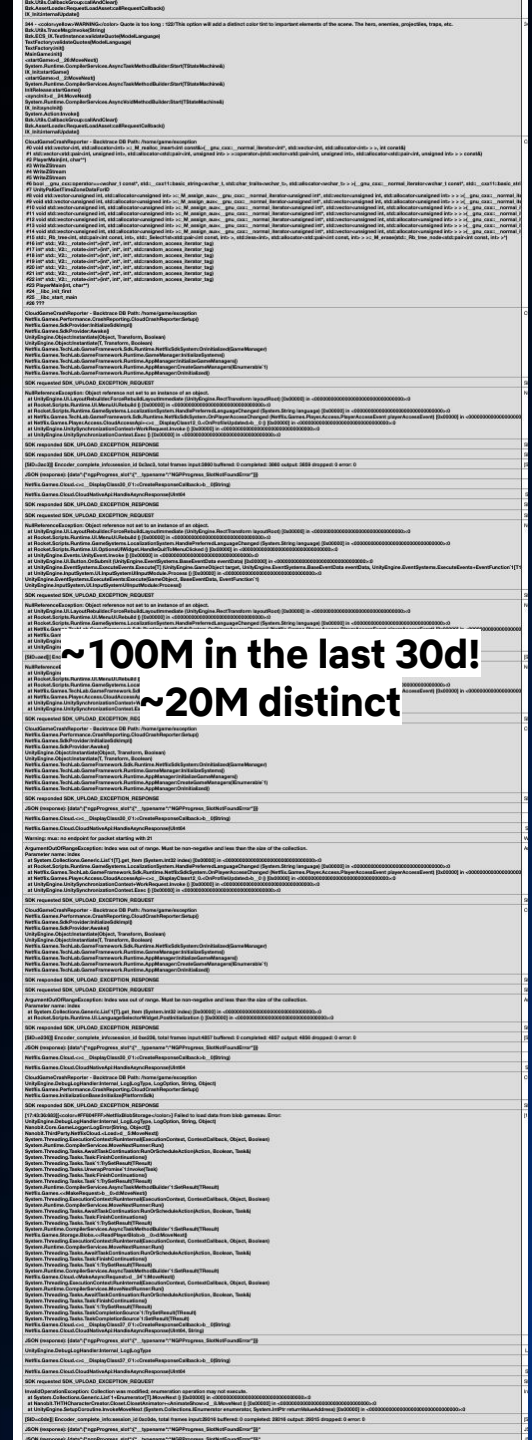


Chosen Indicator of a Broken Session

- Gameplay time (<120s)

Potential Causes

Error / Exception / Warning Logs



Clustering logs

Selection & Filtering

⇓

Preprocessing

⇓

Vectorizing

⇓

Generating Clusters

Selection & Filtering

We chose program outputs:

- Containing ‘exception’ / ‘error’ / ‘warning’
- Excluding several high volume non-useful logs

Preprocessing

```
'Transport receive task exception (id 1): System.Net.WebSockets.WebSocketException (0x80004005): Unable to connect to the remote server\n  at System.Net.WebSockets.WebSocketHandle.ParseAndValidateConnectResponseAsync (System.IO.Stream stream, System.Net.WebSockets.ClientWebSocketOptions options, System.String expectedSecWebSocketAccept, System.Threading.CancellationToken cancellationToken) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Threading.ExecutionContext.RunInternal (System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, System.Object state, System.Boolean preserveSyncCtx) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Runtime.CompilerServices.AsyncMethodBuilderCore+MoveNextRunner.Run () [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Threading.Tasks.AwaitTaskContinuation.RunOrScheduleAction (System.Action action, System.Boolean allowInlining, System.Threading.Tasks.Task& currentTask) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Threading.Tasks.Task.FinishContinuations () [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Threading.Tasks.Task`1[TResult].TrySetResult (TResult result) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Runtime.CompilerServices.AsyncTaskMethodBuilder`1[TResult].SetResult (TResult result) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Net.WebSockets.WebSocketHandle.ReadResponseHeaderLineAsync (System.IO.Stream stream, System.Threading.CancellationToken cancellationToken) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Threading.ExecutionContext.RunInternal (System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, System.Object state, System.Boolean preserveSyncCtx) [0x00000] in <00000000000000000000000000000000>:0 \n  at System.Runtime.CompilerServices.AsyncMethodBuilderCore+MoveNextRunner.Run () [0x00000] in <00000000000000000000000000000000>:0
```

⇓ restrict to first line ⇓

```
'Transport receive task exception (id 1): System.Net.WebSockets.WebSocketException (0x80004005): Unable to connect to the remote server'
```

⇓ replace numbers/ips/urls ⇓

```
'Transport receive task exception (id <NUM>): System.Net.WebSockets.WebSocketException (<NUM>): Unable to connect to the remote server'
```

Vectorizing

Capture the semantic meaning of the logs in a vector using Transformers

```
model = SentenceTransformer("all-MiniLM-L6-v2")
loglines = df['preprocessed'].unique().tolist()
embeddings = model.encode(loglines)
```

```
Last executed at 2025-03-19 21:42:38 in 320ms
```

Batches: 100%

[<NUM>] Pipeline error: Unknown, Failed to handle incoming message. Sys...
System.OperationCanceledException: The operation was canceled.
[<NUM>] Pipeline error: CreateSession, CreateSession InternalServerErro...
ArgumentOutOfRangeException while executing 'performed' callbacks of 'G...
NullReferenceException while executing 'performed' callbacks of 'Game/Y...
NetflixException: Encountered error code -1 with message: Response is null

⇒

```
array([[-0.04990379, -0.07574321, -0.05073855, ...,  0.05344664,
        -0.02589879,  0.04191371],
       [ 0.05790532, -0.05039235, -0.00142821, ...,  0.01898477,
        -0.01576353,  0.02862386],
       [ 0.01524874, -0.00824321, -0.01096346, ...,  0.05970234,
        -0.02260385, -0.01558303],
       ...,
       [ 0.03752244,  0.05013627, -0.07225624, ...,  0.05532859,
        -0.00288594, -0.03726526],
       [ 0.02289473,  0.08571737, -0.01283948, ...,  0.02961025,
         0.10395923, -0.03161674],
       [ 0.06632128,  0.00227036, -0.10962974, ...,  0.05059672,
        -0.00144654, -0.05374308]], dtype=float32)
```

Clustering

Clustering with DBSCAN

```
dbscan = DBSCAN(eps=0.2, min_samples=2, metric = 'cosine')
dbscan.fit(embeddings)
cluster_labels = dbscan.labels_
```

Error: Locale with code th not found

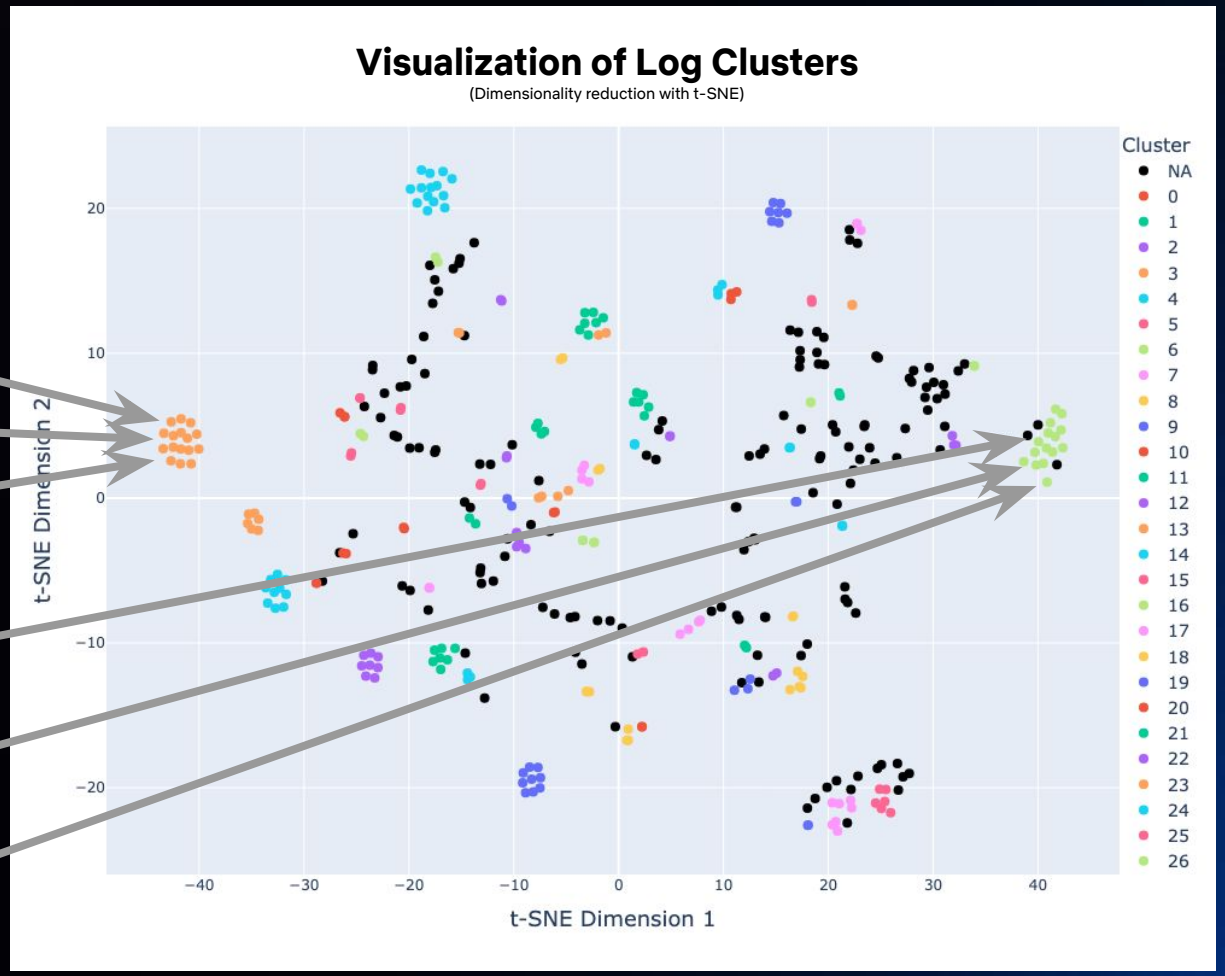
Error: Locale with code fi not found

Error: Locale with code id not found

NullReferenceException on Gamepad/Right/…

NullReferenceException on Gamepad/Select/…

NullReferenceException on Gamepad/Up/…



Results of clustering

20m+ unique logs

⇓

Selection & Filtering
(in SQL)

⇓

Preprocessing
(using regex_replace)

⇓

Vectorizing
(using SentenceTransformer)

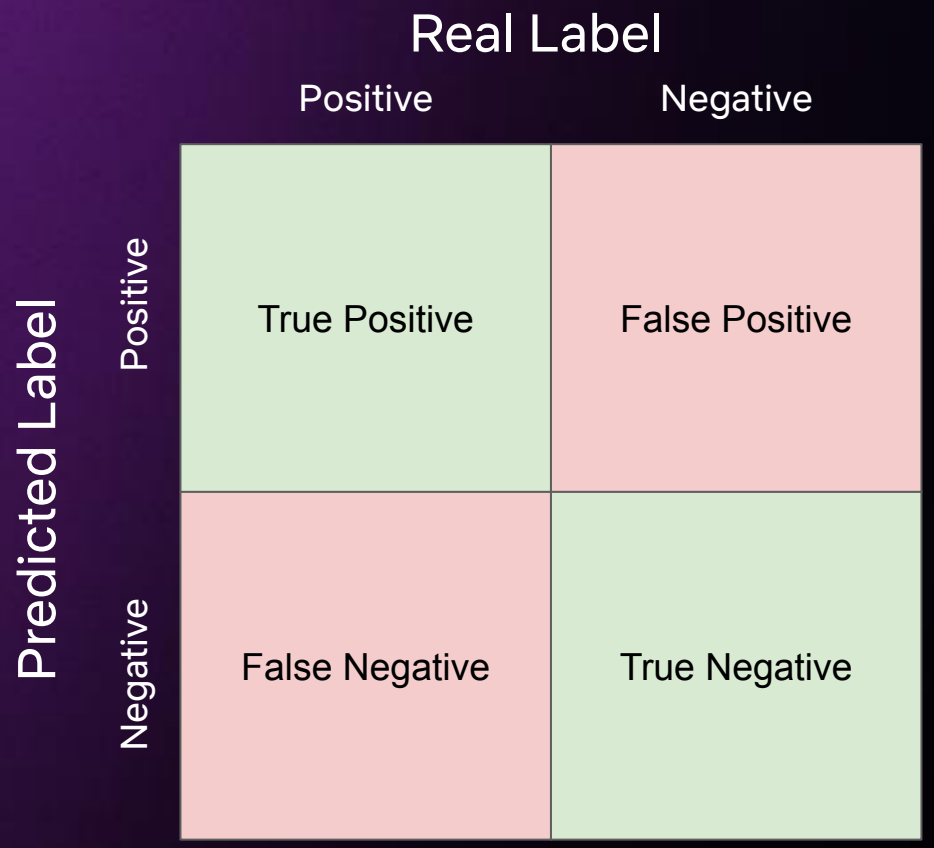
⇓

Generating Clusters
(using DBSCAN)

⇓

~135 unique logs
(with some in 26 log clusters)

Evaluating log impact

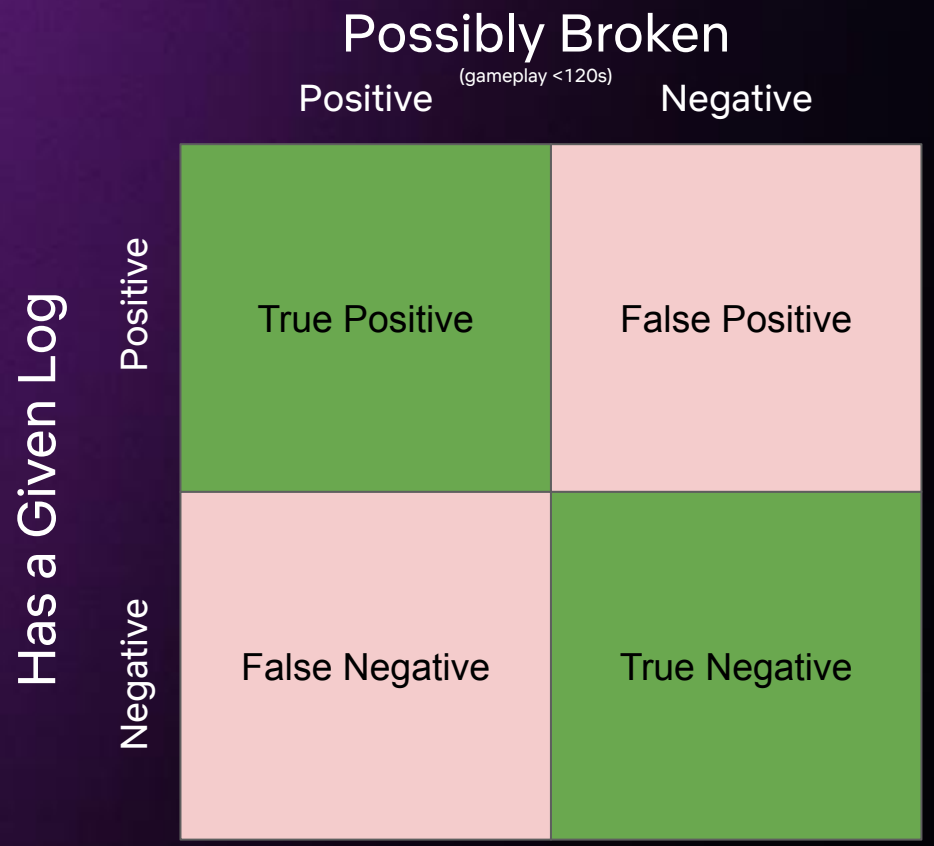


Precision: TP / TP+FP

Recall: TP / TP+FN

F1 score: harmonic mean of precision and recall

Evaluating log impact

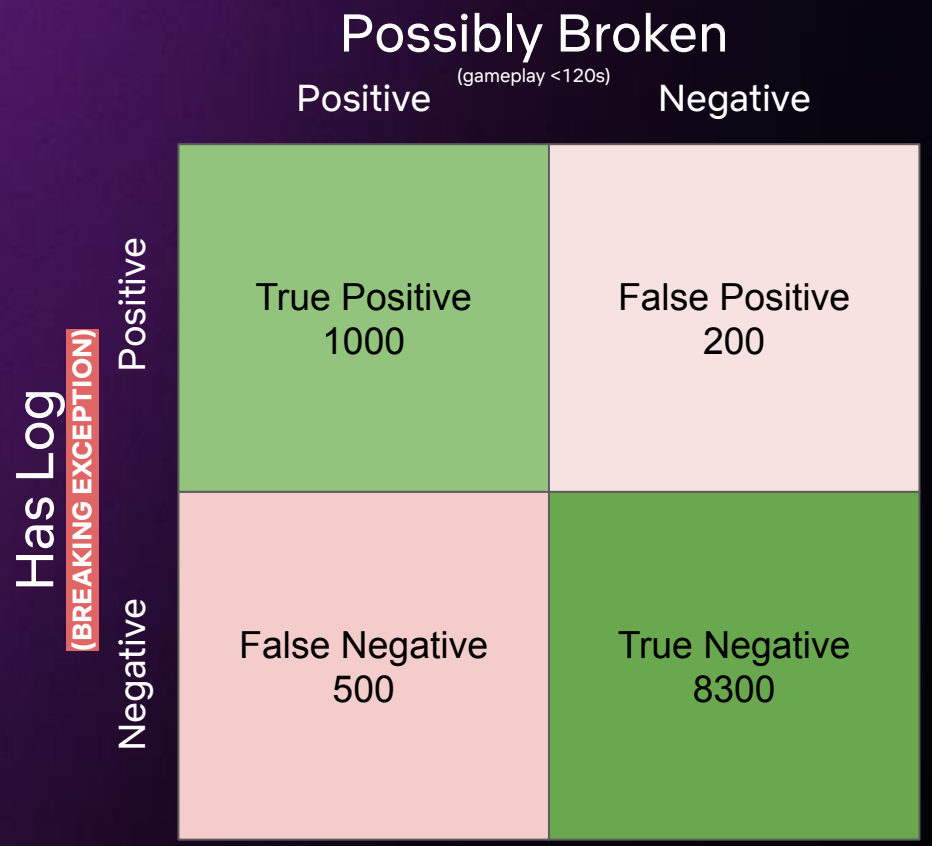


Precision: TP / TP+FP = what percent of sessions with a given log seem broken

Recall: TP / TP+FN = what percent of broken seeming sessions have a given log

F1 score: harmonic mean of precision and recall = a balance of the above

Evaluating log impact

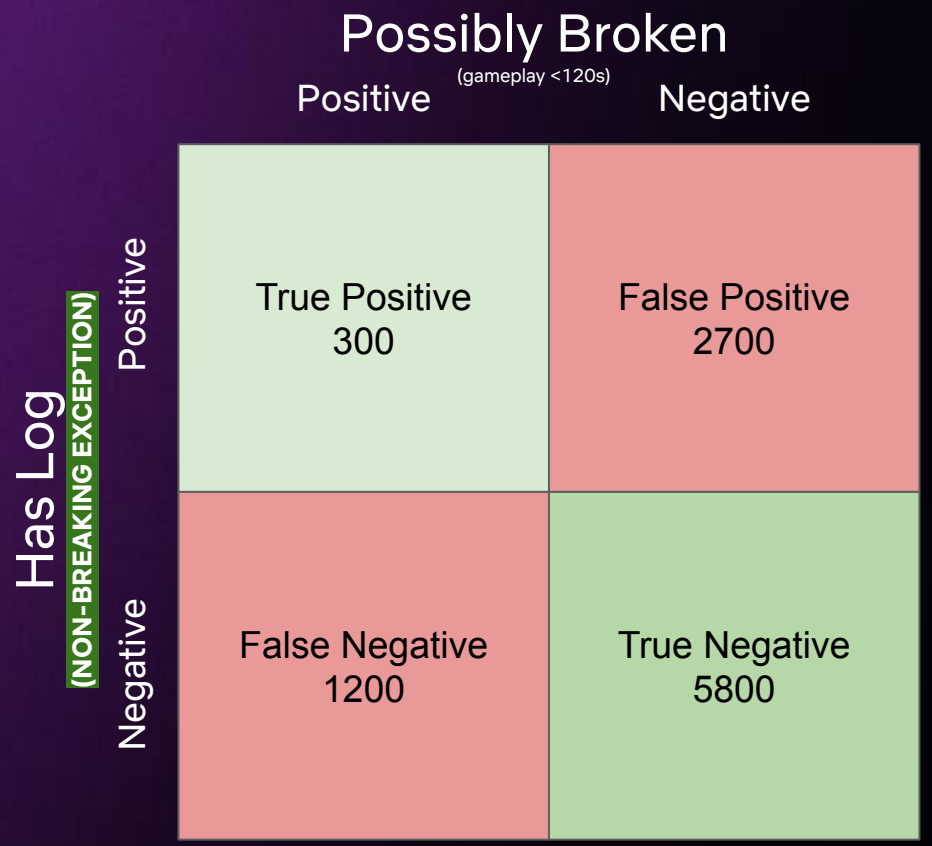


Precision: TP / TP+FP = what percent of sessions with a given log seem broken = 83%

Recall: TP / TP+FN = what percent of broken seeming sessions have a given log = 67%

F1 score: harmonic mean of precision and recall = a balance of the above = 74%

Evaluating log impact

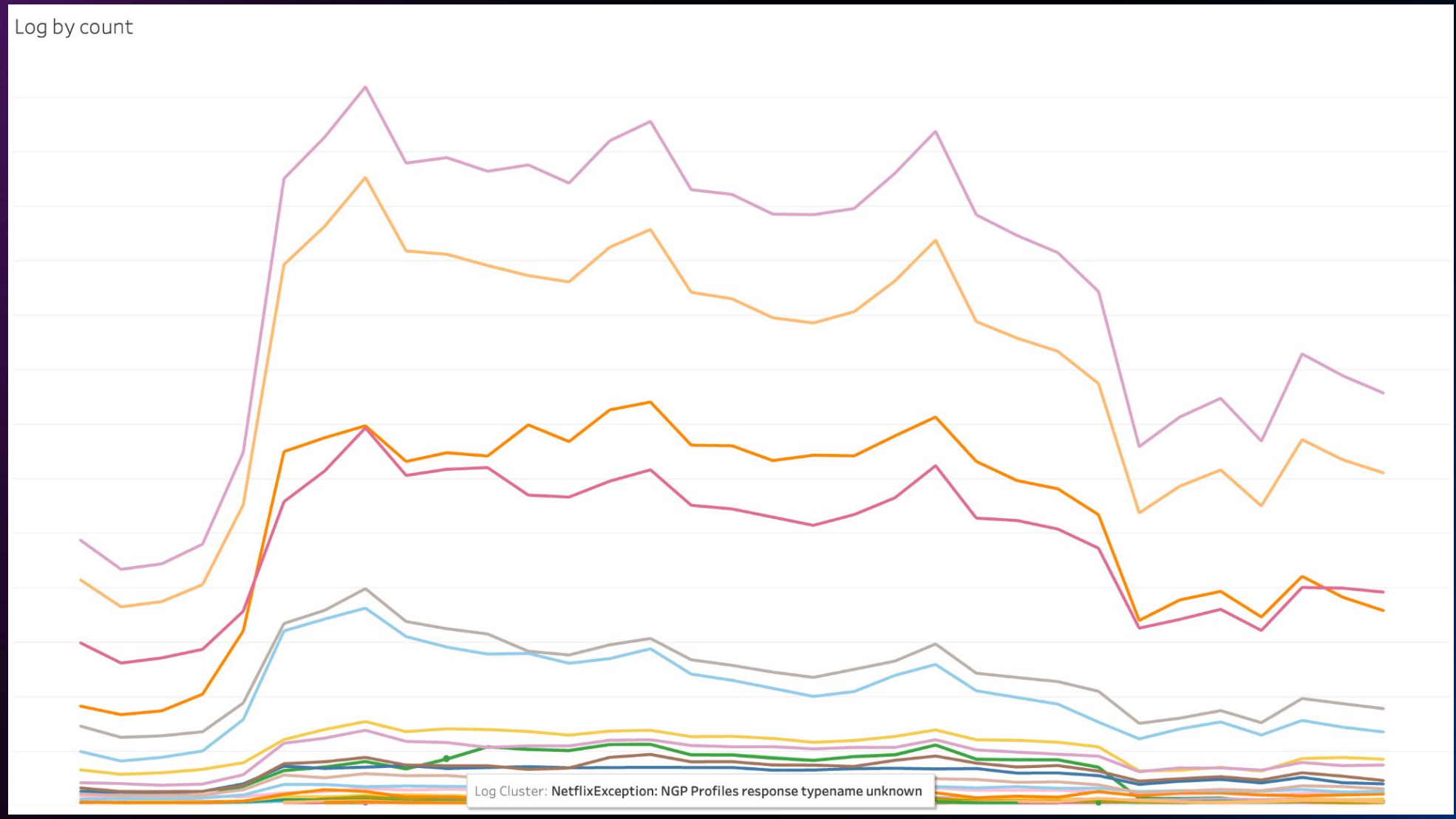


Precision: TP / TP+FP = what percent of sessions with a given log seem broken = 10%

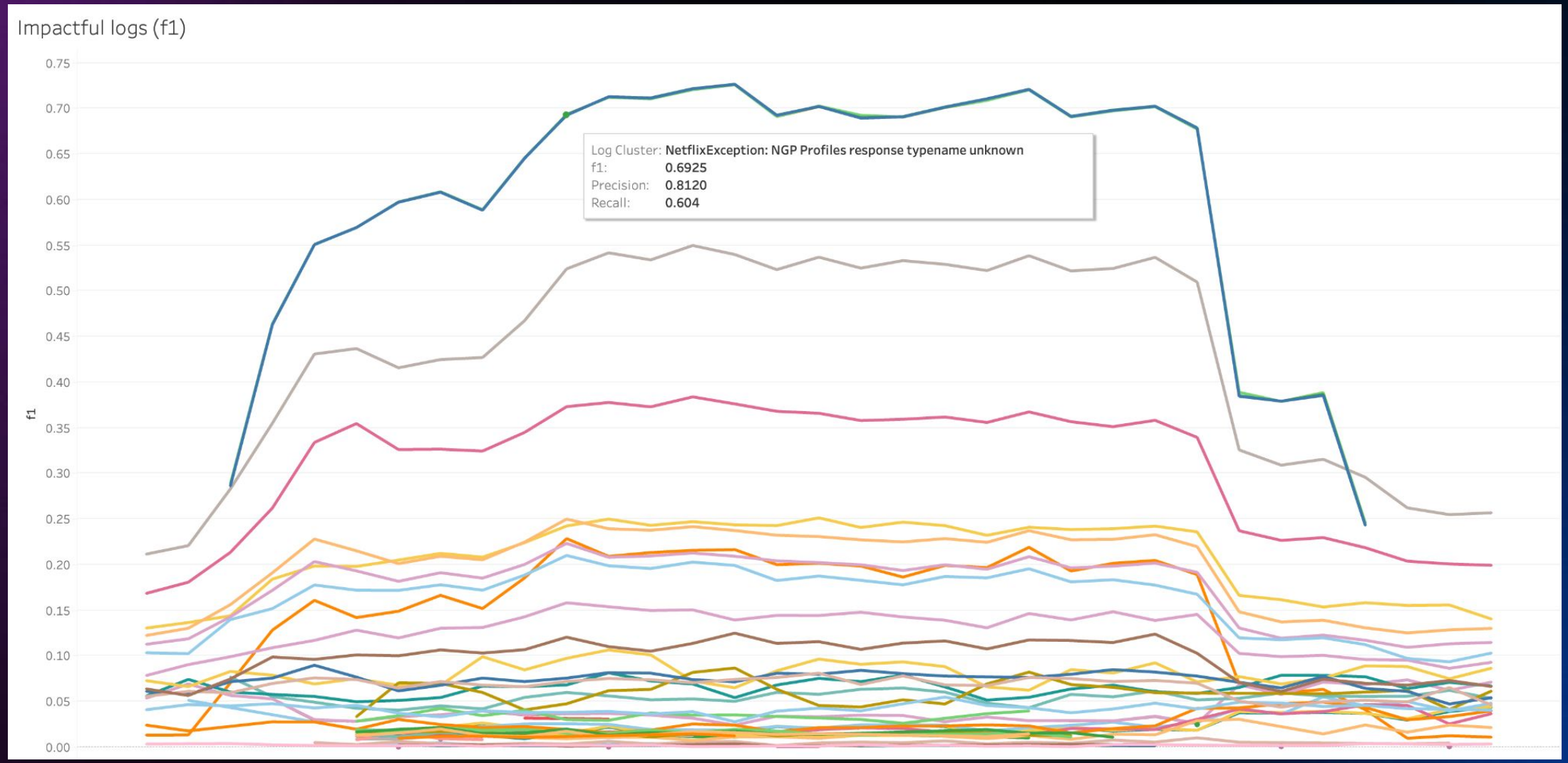
Recall: TP / TP+FN = what percent of broken seeming sessions have a given log = 20%

F1 score: harmonic mean of precision and recall = a balance of the above = 13%

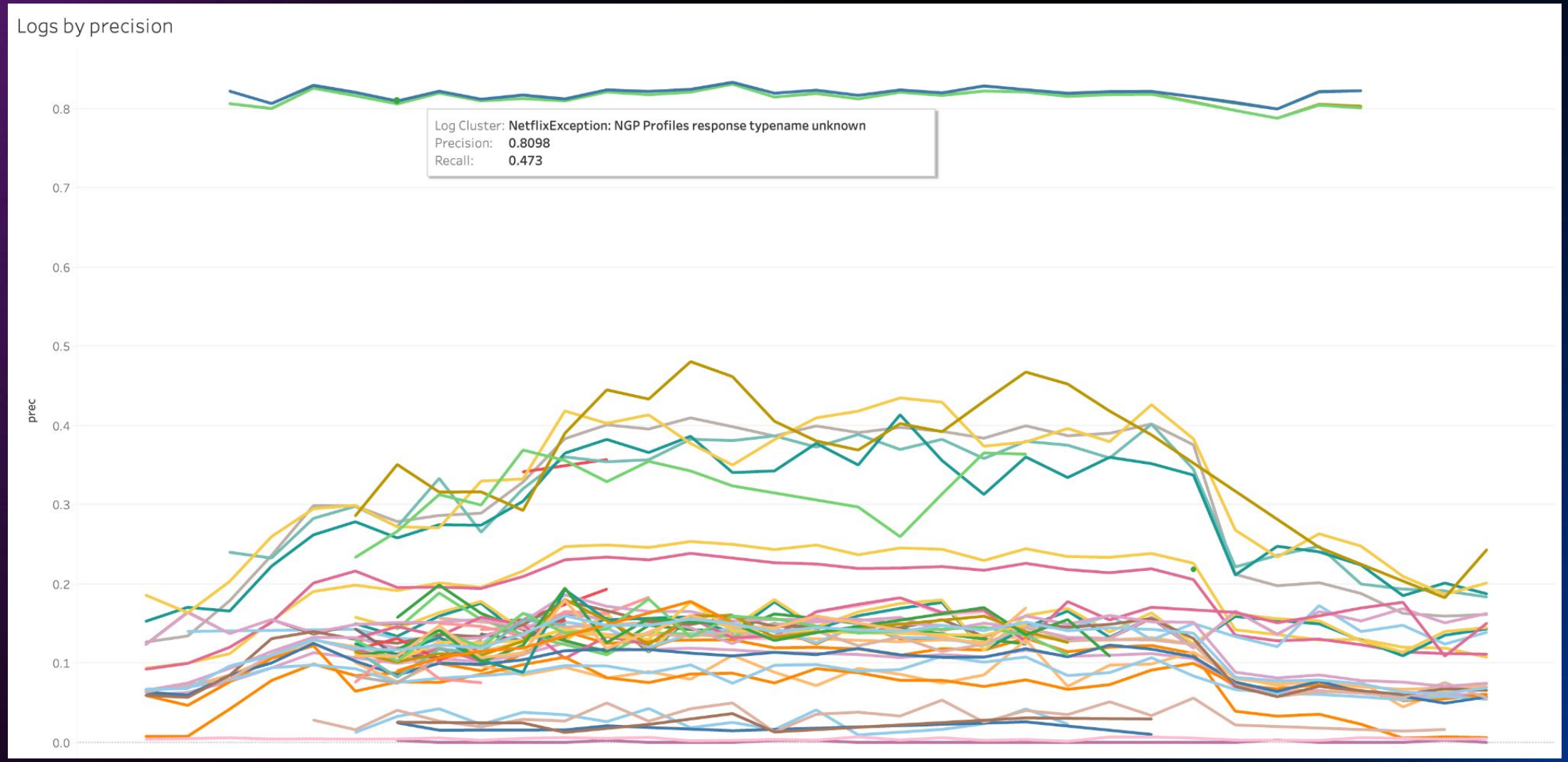
Incident Revisited



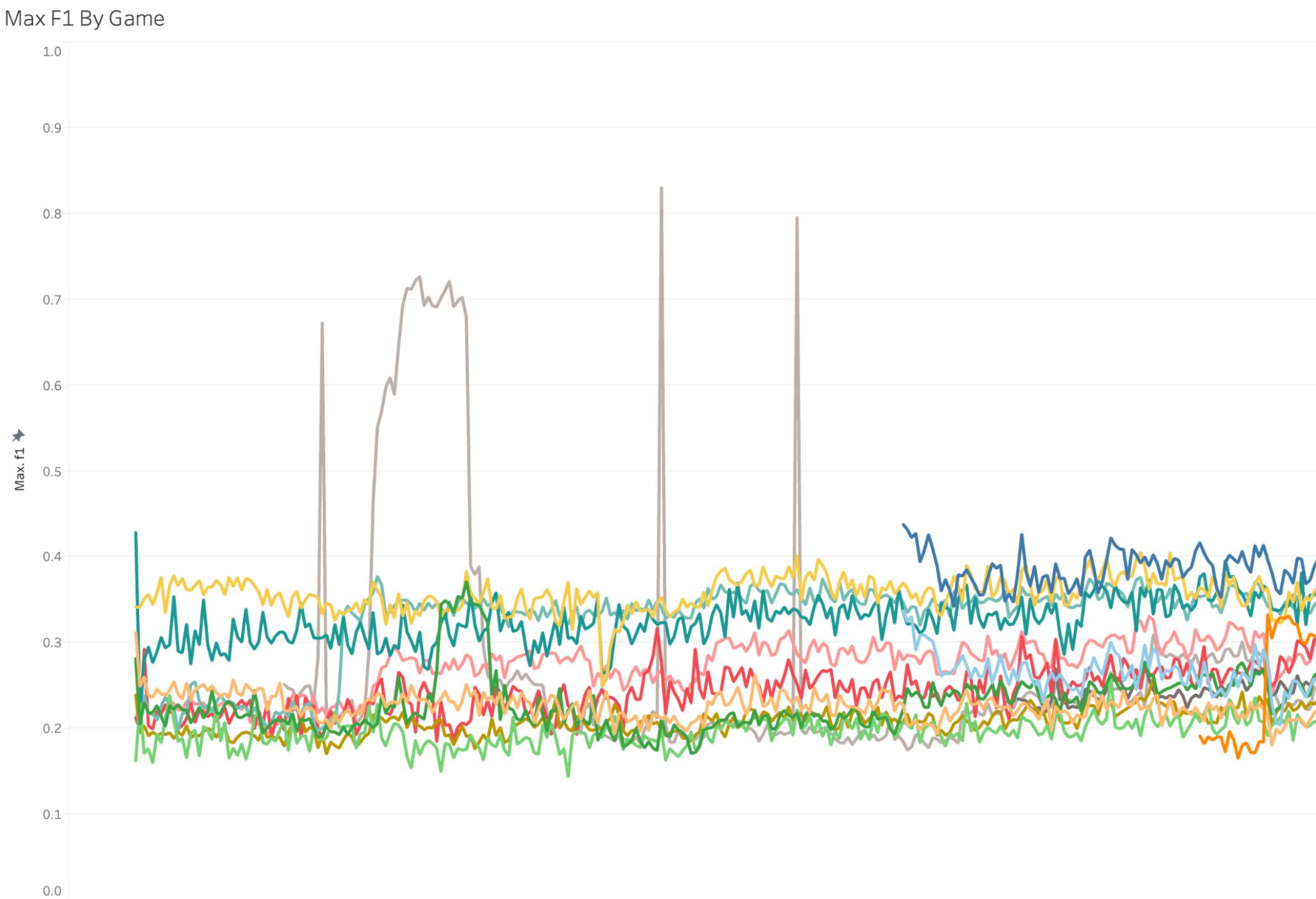
Incident Revisited



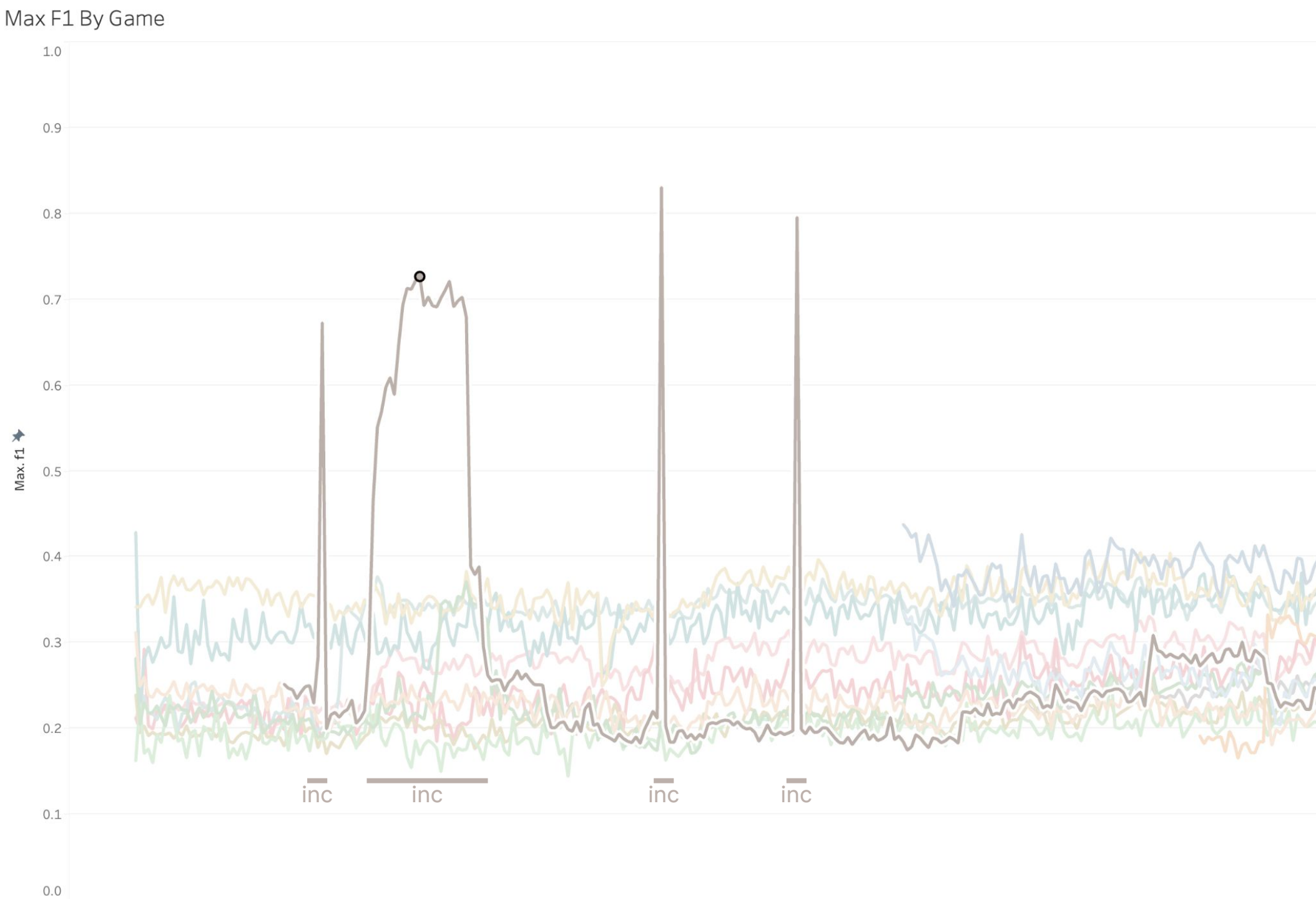
Incident Revisited



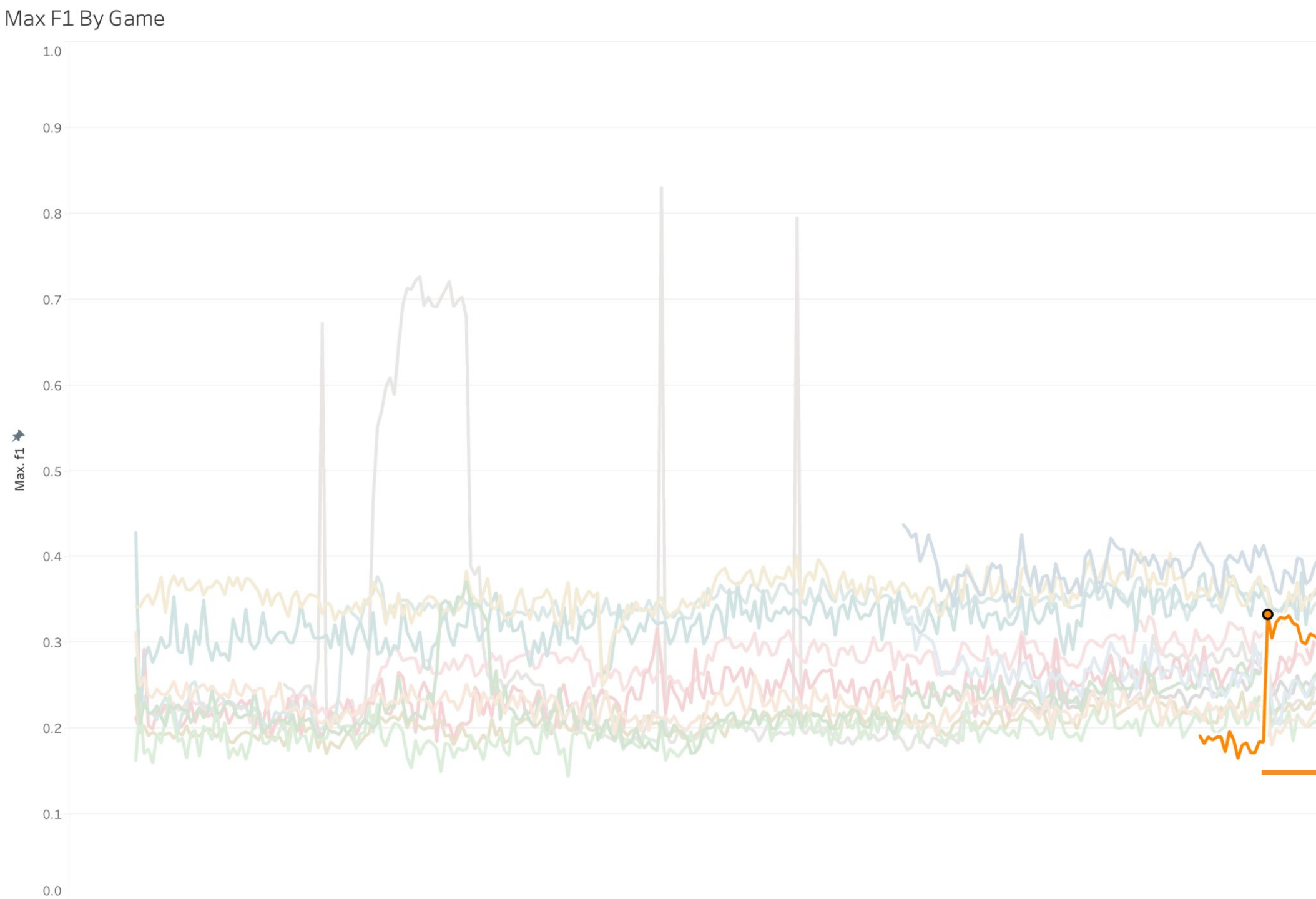
Now



Now


Max F1 By Game
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Max. f1
inc
inc
inc
inc

Now



Future

Speeding up our alerting processes

Using time to abort metrics to detect mid-game breaking issues

Improving testing, working with partners to get cleaner signals, etc.

Takeaways

New areas bring new seemingly intractable challenges

Statistical methods can make hard, high volume problems doable

Tons of statistical tools/concepts are readily available for integration into our workflows

Thank You

ineidel@netflix.com