## Introduction to Netflix

Netflix, the video streaming service that we all know and love, has 223 million subscribers in countries all around the world watching over 200 million hours of Netflix per day. If you assume that 1 hr of Netflix content in HD is about 3 GB of data, then Netflix is delivering over 6 petabytes of data to customers every single day. This data, once collected and aggregated, sheds light on the streaming experience from both the perspective of the viewer and that of the server.

Laura Pruitt is Director of Streaming, Platform, and Security Data Science & Engineering at Netflix, and this blog post covers some of her insights into how the company maintains the quality of this core dataset.

## How Netflix streaming works

Netflix has custom-built servers to hold video, audio, and subtitle files. These servers are distributed around the world, as close as possible to customers. The goal of this localization is so that when customers have to stream the data, it never has to go very far.

To outline the lifecycle of watching a TV show on Netflix: you’ve found something you want to watch, and your device sends a request to one of the servers asking for a piece of content. The server sends the first chunk of that particular video back to you, which your device then decodes and renders in real-time. As your device is decoding and rendering it, it also asks the server for more data, which it sends back. All of this is done in real-time.

While all of this is happening, Netflix is collecting a lot of information from both the device and the server.

From the device side:

* Who are you as a customer
* What device are you streaming on
* How quickly did it take for that video to load
* Did you experience any errors or interruptions during the course of this playback

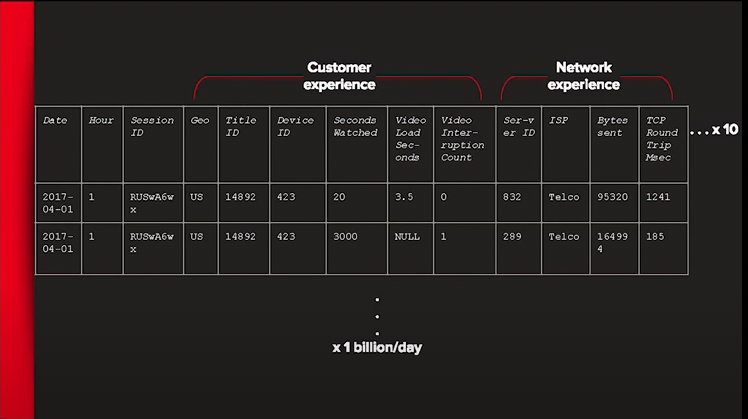
From the server side:

* What ISP was the server connected to deliver the content?
* How many bytes did the server transfer?
* How long did it take for those bytes to arrive at their destination?

All these raw logs are landing in Amazon S3, which is Netflix’s central data hub. (From S3 the data is directed into additional services like Redshift, Kinesis, etc).

## What Pruitt’s team does

Pruitt’s team subsequently runs ETL pipelines that use business logic, and windowing, to process these raw logs into a dataset that is a unified view into both the customer experience and the network experience. This dataset sees several billion new records every day, and is a core dataset at Netflix.



In putting anomaly detection and data integrity checks on this dataset, Pruitt’s team had the following considerations.

**Impact**

This dataset is a very important dataset for Netflix. It is used to answer questions like and make decisions about:

* Which partnerships to invest in
* Which ISPs or devices can bring valuable partnerships to Netflix
* Where to invest internal engineering resources
* Where the service is seeing the most performance issues

“Any dataset should have a bare minimum of checks in place, but this is one that is being used by many different people and we are making pretty important decisions with it, so it makes sense to make additional investments in making sure the data is of high quality,” Pruitt said.

**Data Integrity**

In addition to the devices and the servers, there are several more data sources in this pipeline. Each of these data sources is a place where things can go wrong. Examples of data integrity issues that might pop up include:

* Missing data
* Unexpected datatypes
* Unexpected NULLS
* Malformed records which means you can’t parse out key-value pairs

Pruitt’s team found that it’s best to detect these sorts of data integrity issues before the ETL process (Netflix, it seems, chooses to monitor their data at the source. See our blog post about whether to monitor at source or destination). They do via a metadata service that gives them high-level metadata metrics on their tables, including:

* Is the partition loaded?
* How many rows are there?
* What’s the min and max value that exists within that column
* What’s the cardinality of that column?
* If a certain amount of data is using thrown away during ETL processing, what is that percentage number?

Netflix has built reusable frameworks that are shared between data engineering teams and data platform teams to make sure that these basic, generic data quality issues are addressed on source table. For example, every time a service writes out data, the producer can audit it before it’s published to confirm that the main metadata metrics are looking good, before the data is made available for downstream consumption.

**Business metrics**

This data pipeline produces dozens of metrics that the company cares about, including things like:

* Error rates
* Customers’ consumption of Netflix

Additionally, these metrics often have extremely high dimensionality, due to the fact that Netflix operates in hundreds of countries and thousands of ISPs. This makes it challenging to figure out where things are when there are so many permutations.

For example, consider a business metric like the global playback error rate – the percentage of sessions that end in a fatal error for customers. Let’s say that the spike is actually caused only by Android phones in Brazil – Pruitt’s team needs to identify and annotate this before the CEO comes knocking on the door.

To deal with the high cardinality, Netflix relies on **anomaly detection**. Netflix pre-aggregates data to grains that they believe are meaningful (devices, countries) and sends that data to an anomaly detection service, which sends back data points they think are anomalous. This pre-aggregation is an effort to reduce the dimensionality of their metrics.

In terms of alerting, Pruitt's team started conservatively. It picked the top metrics that it cared about, and only alerted on those to the right people (over email).

**Conclusion**

At Netflix, data quality directly translates into informed decisions that affect our viewing experience and their business bottom line. As a result, the company has made a wise decision to invest in it.