



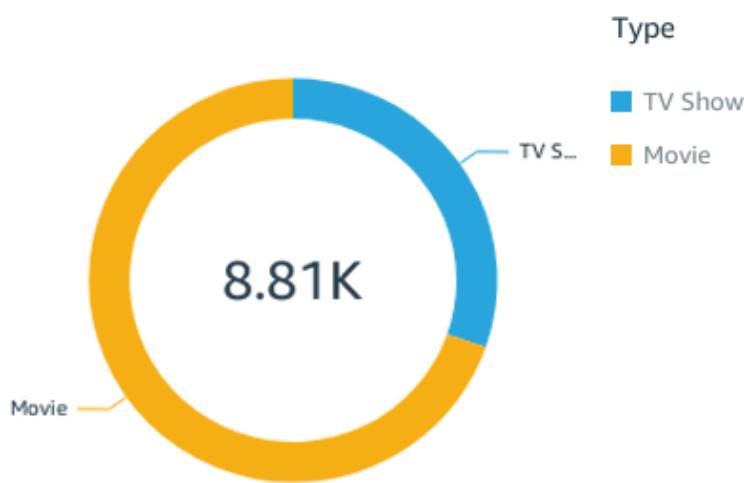
Visualize data with QuickSight



Vivek Murali

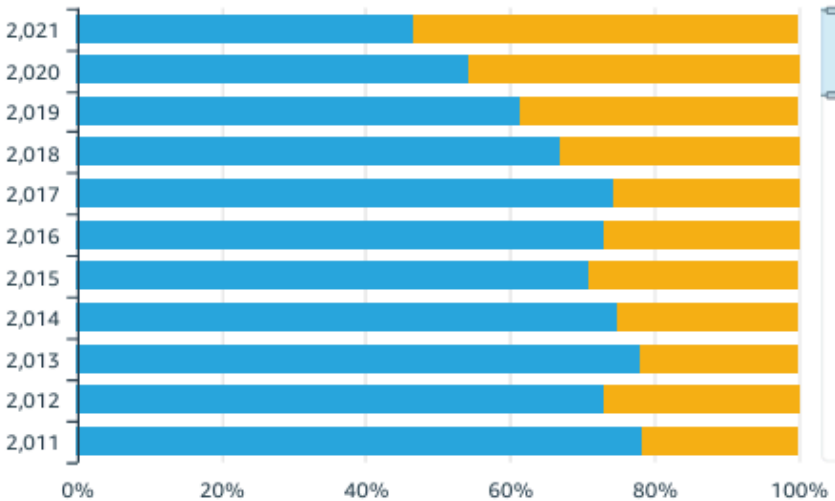


Records By Type



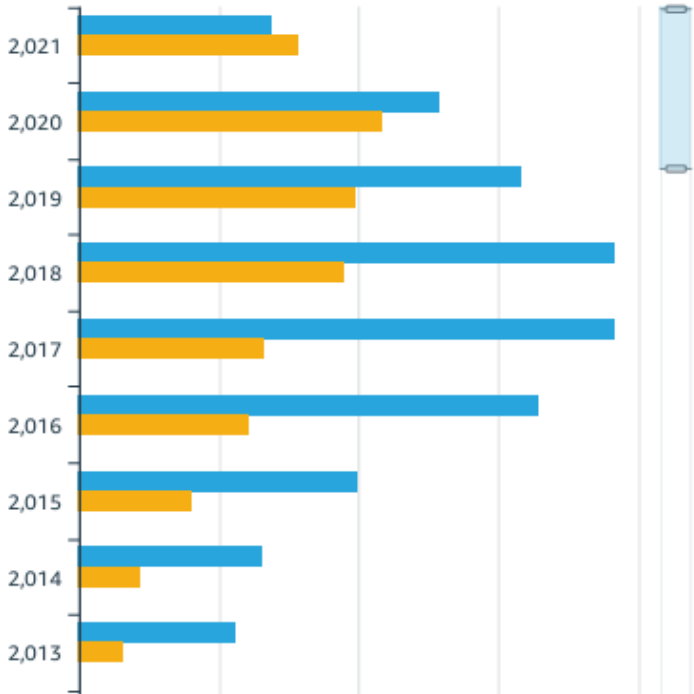
Movies vs TV shows by Release Year

SHOWING TOP 50 IN RELEASE_YEAR AND TOP 3 IN TYPE



Movies vs TV Shows By Released Year

SHOWING TOP 50 IN RELEASE_YEAR AND TOP 3 IN TYPE



Movies Vs TV Shows Titles by Release Year

Year	Type	
	TV Show	Movie
	Total	Total
1,925	1	
1,942		2
1,943		3
1,944		3
1,945	1	3
1,946	1	1
1,947		1
1,954		2
1,955		3
1,956		2
1,958		3

Rating





Introducing Amazon QuickSight!

What it does & how it's useful

Amazon QuickSight is a cloud-based BI service that helps create dashboards & visualizations from data, scaling to tens of thousands of users without infrastructure management. It's popular among developers & teams for its serverless auto-scaling, broad data source support, super-fast in-memory calculation engine, and multitenancy features, making it an ideal choice for enterprise workloads.

How I'm using it in today's project

I'm using Amazon QuickSight in this project to craft engaging visualizations of Netflix data through graphs, charts, and a customized dashboard, enabling a deeper understanding of this valuable dataset.

This project took me...

The project took me 50 minutes to complete, and writing the documentation took another 20-odd minutes.



Upload project files into S3

- S3 is used in this project to store two files, which are `netflix_titles.csv` and `manifest.json`.
- I edited the `manifest.json` file by specifying the S3 URL of `netflix_titles.csv`. This modification establishes the connection for QuickSight to access and utilize the dataset stored in S3.

Here's my bucket with the CSV file and manifest.json!

netflix-quicksight-project-vivek-murali [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	manifest.json	json	July 8, 2024, 20:59:51 (UTC+05:30)	310.0 B	Standard
<input type="checkbox"/>	netflix_titles.csv	csv	July 8, 2024, 20:58:09 (UTC+05:30)	3.2 MB	Standard



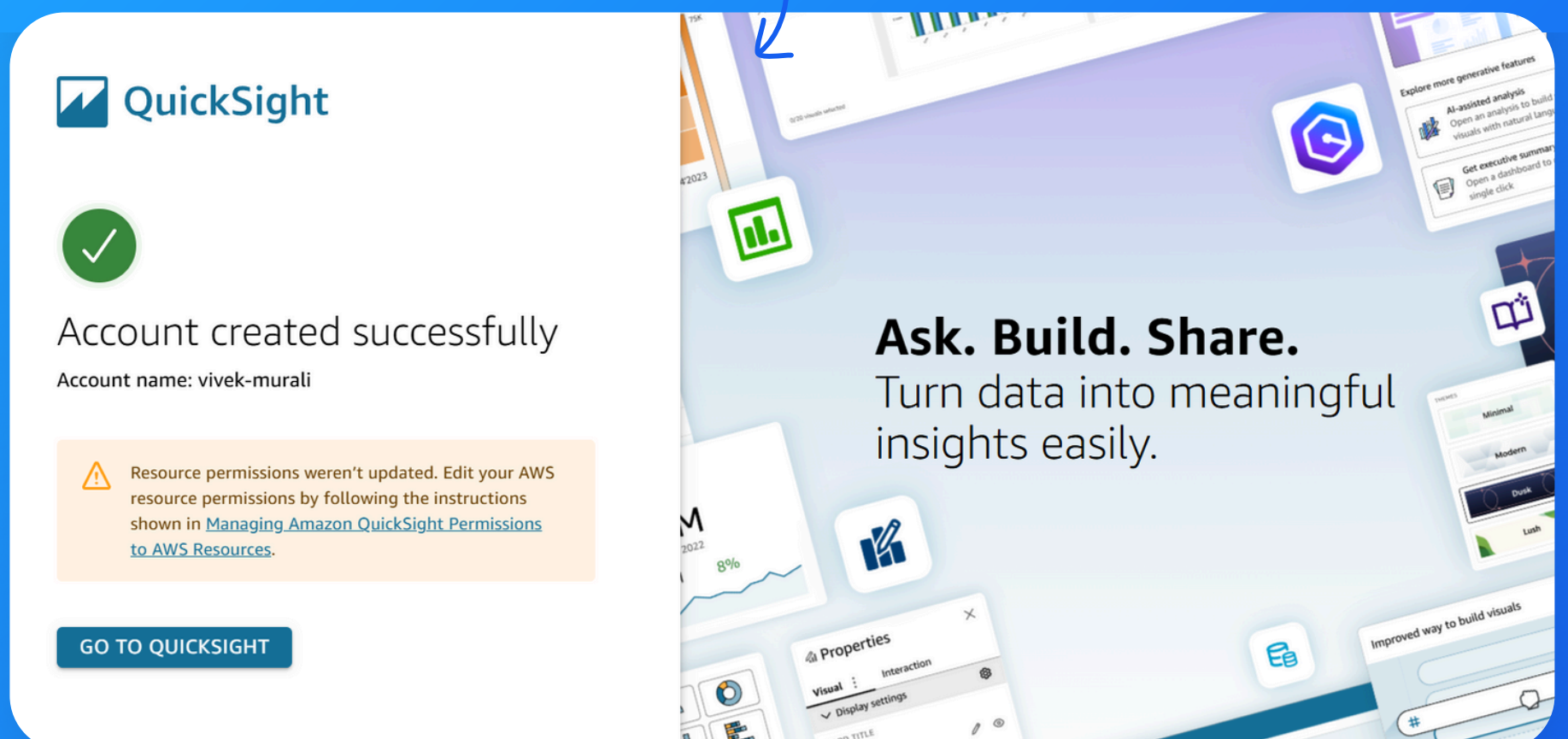
Vivek Murali
[linkedin.com/in/vivek-murali-de](https://www.linkedin.com/in/vivek-murali-de)

NextWork.org

Create QuickSight account

- It cost no money because I used a free trial
- Creating a QuickSight account took me almost 2 minutes
- I also had to enable QuickSight's access to S3 because the data needed for visualization is stored in an S3 bucket.

Voila! I created my QuickSight account successfully.





Connect S3 + QuickSight

- I connected the S3 bucket to QuickSight by configuring QuickSight to access the S3 URL associated with the manifest.json file.
- The manifest.json file was important in this step because it serves as a blueprint, defining the location and structure of data files in the S3 bucket, allowing QuickSight to retrieve and utilize the stored data.

Entering the manifest.json URL.

New S3 data source ×

Data source name

kaggle-netflix-data

Upload a **manifest file** ☒ URL ☐ Upload

s3://netflix-quicksight-project-vivek-murali/manifest.json

Connect



Let's make visualisations!

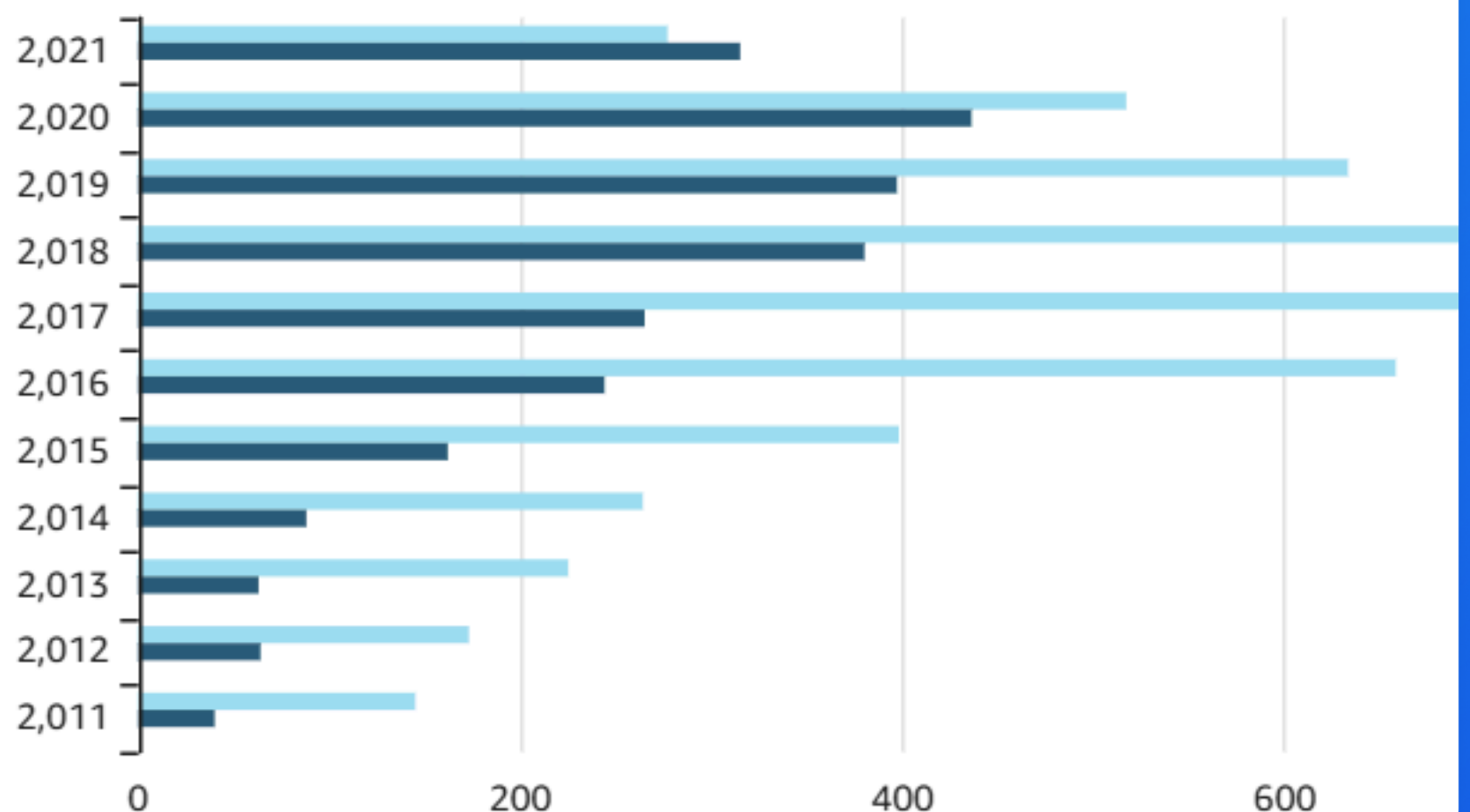
To create visualizations on Amazon QuickSight, I followed the steps:

- Selecting an existing dataset, creating a new visualization, and finally choosing the type of graphic display desired.
- The specific graph bar chart illustrates the relationship between release year and type.
- I constructed this chart by dragging the "release_year" dimension to the x-axis, and then adding the "type" attribute to the same graph, allowing me to explore these data points side-by-side.

One of my first visualisations.

Count of Records by Release_year and Type

SHOWING TOP 50 IN RELEASE_YEAR AND TOP 3 IN TYPE



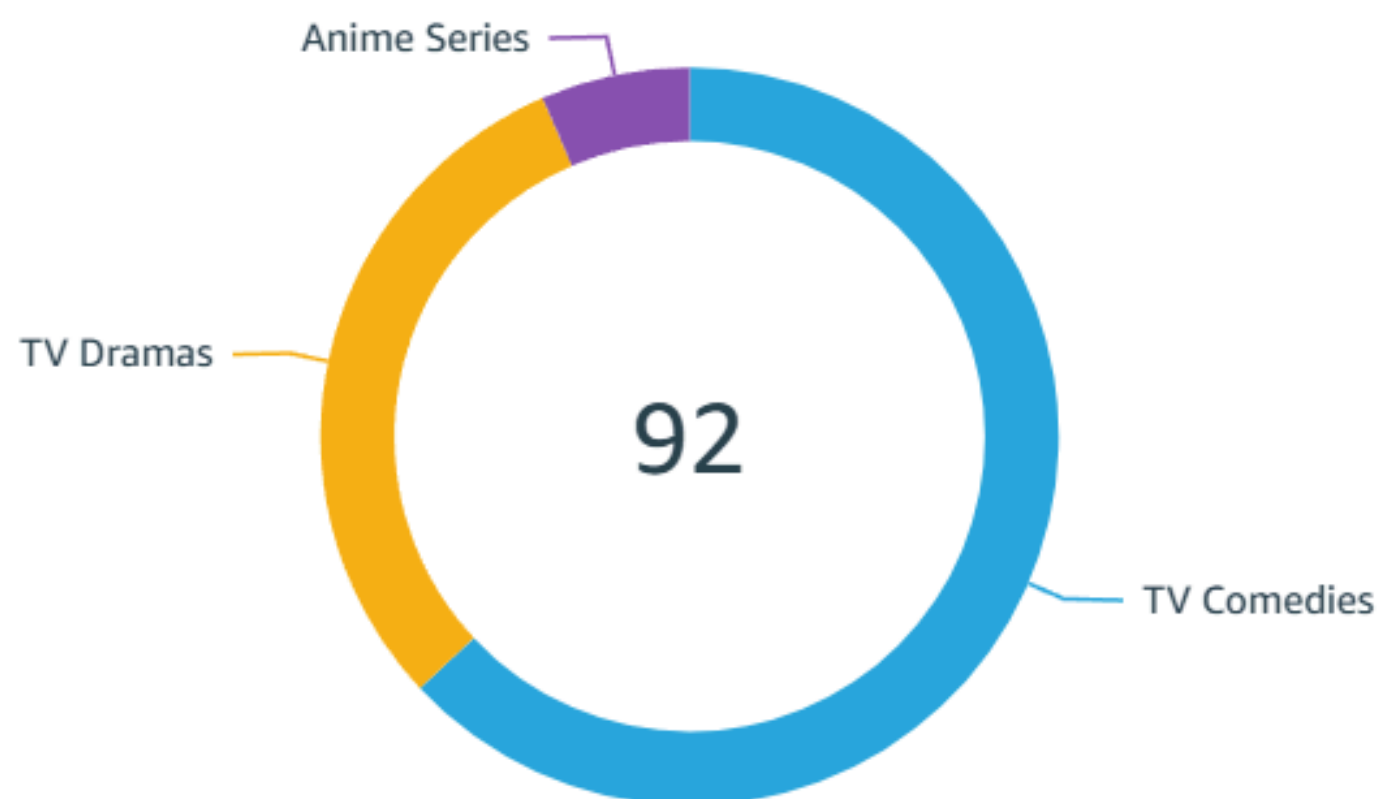


Using filters

- Filters are useful for refining large datasets by letting you choose specific values, simplifying complex data, and highlighting what matters
- Here I refined my visualization by adding two filters: one for the "Release Year" column, where I selected values beyond 2015, and another filter for the "Listed In" category, where I picked specific values from TV Drama, TV Comedies, and Anime Series.

A visualisation set up after adding filters.

Count of Listed_in by Listed_in

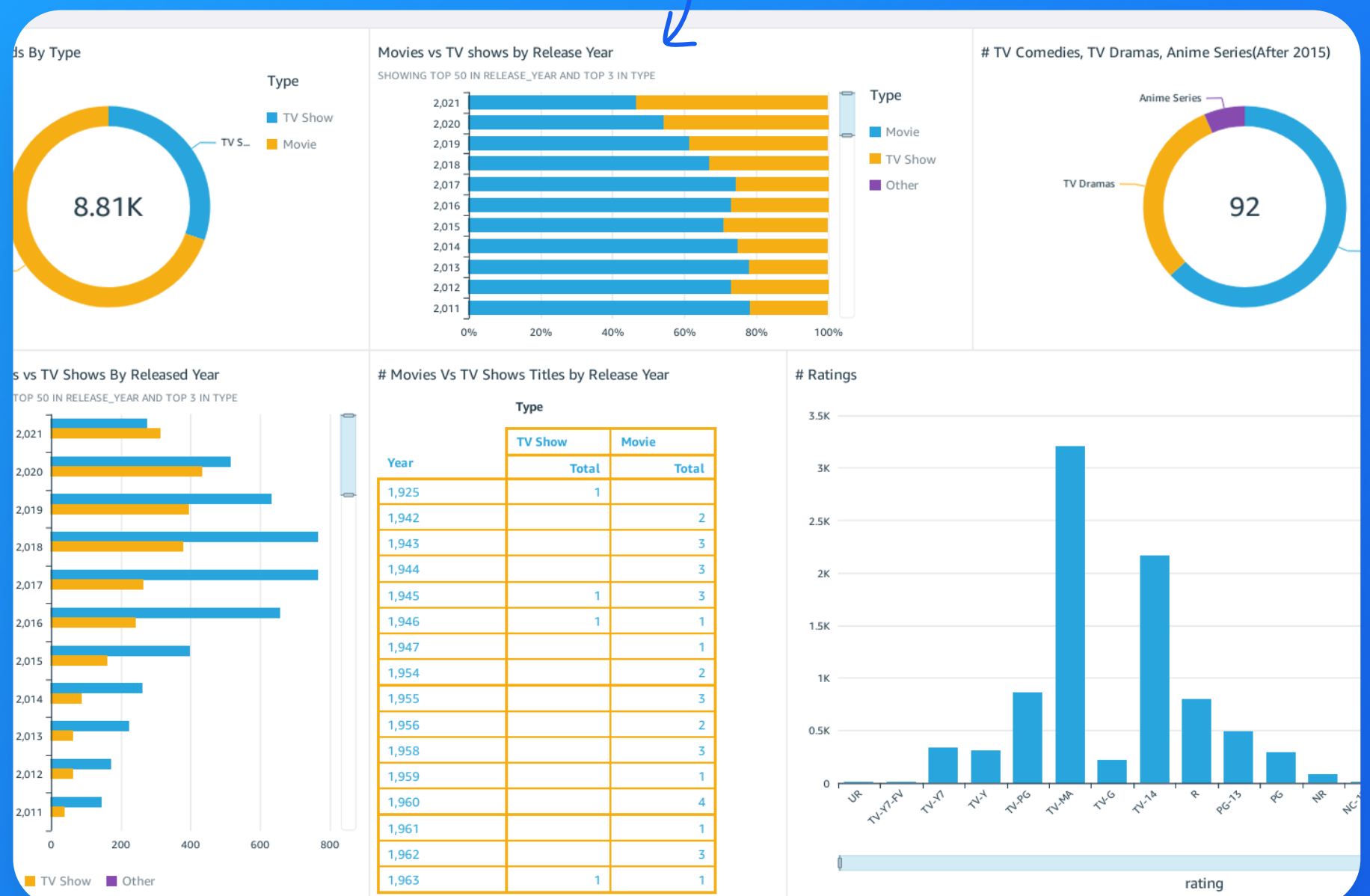




Set up your dashboard!

- As a finishing touch, I polished my dashboard by selecting a modern theme, renaming titles to provide context, and reorganizing visuals into a single sheet view for smooth scrolling.
- Did you know you could export your dashboard as PDFs too? I did this by selecting Export and Generate PDFs, it would notify you when it's done exporting.

Voila! Here's the finished dashboard!





My key learnings

- 1 An S3 bucket was used in this project to serve as a storage container, holding the essential data files that power visualizations in Amazon QuickSight.
- 2 To connect the data stored in S3 with QuickSight, I had to use the manifest.json file.
- 3 Creating visualizations on Amazon QuickSight was surprisingly easy and intuitive, able to create complex dashboards and visualize my data quickly and efficiently, exceeding my expectations for ease of implementation.
- 4 One thing I didn't expect was Amazon QuickSight's ability to rapidly process and visualize large datasets, streamlining the process of creating complex and insightful dashboards.