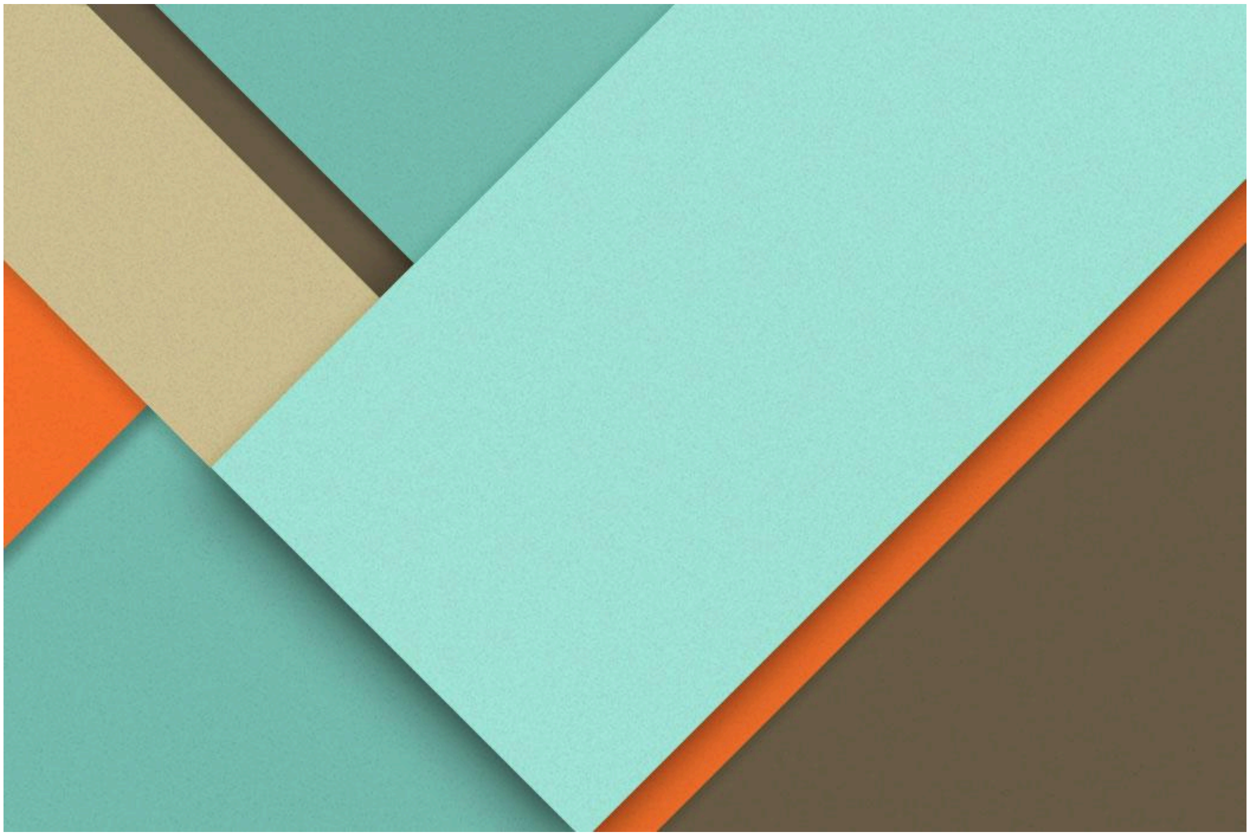

DV : StreamScope

Netflix Content Strategy Analyzer: Insights into Global Streaming Trends

📊 Milestone 1 – Netflix Data Cleaning & Insights Report

Project Scope & Success Metrics

Scope:
The goal of this milestone is to prepare and clean the Netflix Titles dataset for subsequent analysis and modeling. The dataset contains information about Netflix movies and TV shows, including metadata like title, cast, director, country, rating, and duration.

1. Importing and Describing the Dataset

Functions used:

- pd.read_csv() – to import the dataset.
- df.shape – to get the dimensions.
- df.size – to find the total number of cells.
- df.head() – to preview the data.

Output:

- **Shape:** (8807, 12) → 8,807 rows × 12 columns
- **Total cells:** 105,684
- Columns include: show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, description

Insight: The dataset is moderately sized with rich categorical information, suitable for both descriptive analysis and encoding for machine learning.

2. Handling Duplicates

Functions used:

- df.shape[0] (before and after)
- df.drop_duplicates() – to remove duplicate rows.

Output:

Rows before dropping duplicates: 8807

Rows after dropping duplicates: 8807

Total duplicates removed: 0

Insight: There were **no duplicate rows** in the dataset.

3. Identifying Missing Values

Functions used:

- df.info() – to check datatypes and non-null counts.
- df.isnull().sum() – to count missing values per column.

Output:

show_id 0

type 0

title 0

director 2634

cast 825

country 831

date_added 10

release_year 0

rating 4

duration 3

listed_in 0

description   0

Insight: Columns such as **Director**, **Cast**, **Country**, **Date Added**, **Rating**, and **Duration** contained missing values, which needed proper handling before analysis.

4. Handling Missing Values

Functions used:

- df.fillna() – to fill missing values with placeholders.

Imputations made:

- Director → “Unknown”
- Cast → “Not Available”
- Country → “Unknown”
- Date Added → “Unknown”
- Rating → “Unrated”
- Duration → “Unknown”

Insight: After filling, the dataset contained **0 missing values**, ensuring completeness for further processing.

5. Cleaning Text Columns

Functions used: str.strip() – to remove leading and trailing spaces from string columns like Title, Director, Cast, Country, Description, Listed_in.

7. Checking and Standardizing Column Formats

Functions used:

- df.info() → to check column datatypes and non-null counts.
- df.select_dtypes() → to select text-based columns.
- fillna() → to handle missing values.
- astype(), str.strip(), str.title() → to standardize text formatting.
- pd.to_datetime() & pd.to_numeric() → for proper conversion of date and numeric columns.

Process: After completing initial cleaning steps, we verified the **datatype consistency** of all columns using: df.info()

This revealed:

- **11 object columns** (text),
- **1 datetime64[ns]** column (date_added),
- **1 int64** column (release_year), and
- **1 float64** column (duration_num).

8. Data Normalization & Encoding

To make the dataset ready for numerical analysis and machine learning, different types of encoding and normalization were done on the columns. This helped convert text or categorical data into a format that models can understand.

a. Column Dropping

- **Columns Dropped:** rating 66min, rating 74 min, rating 84min
- **Function Used:** drop()
- **Insight:** These columns were not useful and contained incorrect data, so they were removed to keep the dataset clean.

b. Frequency Encoding

- **Columns:** country, listed_in (Genres)
- **Function Used:** value_counts() with map()
- **Insight:** Here, countries and genres were replaced with numbers based on how often they appeared. This way, categories that appear more frequently got higher values, which helps the model understand their importance.

c. Ordinal Encoding

- **Columns:** release_year, rating
- **Function Used:** map() with custom order
- **Insight:**
  - For **release_year**, values were converted into an increasing order to keep the timeline intact.
  - For **rating**, we gave each rating a number based on its maturity level (like G < PG < TV-MA), so the model understands the rating hierarchy.

d. One-Hot Encoding

- **Column:** type (Movie / TV Show)
- **Function Used:** get_dummies()
- **Insight:** This split the column into two separate columns — one for Movie and one for TV Show — using 0s and 1s. This avoids treating the two categories as if they have an order.

e. Duration Numeric Extraction & Normalization

- **Column:** duration
- **Functions Used:** str.extract() and Min-Max normalization
- **Insight:** Numbers were pulled out from the duration column (like “90 min” → 90), and then these numbers were scaled between 0 and 1. This makes the values easier to compare and helps during modeling.

Milestone -2

1. Netflix Content Growth Over Time

Steps:

- Counted titles added each year using:
  `content_per_year = df['year_added'].value_counts().sort_index()`
- Explored multiple visualizations:
  - Line Plot
  - Bar Plot
  - Pie Chart (for recent 5 years)
  - Histogram
  - Scatter Plot

Functions Used:

- `plt.plot()` → Line plot
- `plt.bar()` → Bar chart
- `plt.pie()` → Pie chart
- `plt.hist()` → Histogram
- `plt.scatter()` → Scatter plot

Insights:

- Netflix content grew rapidly after 2015, showing massive expansion.
- Peak content additions occurred around 2018–2020.
- Growth slowed slightly post-2021, possibly due to content saturation or pandemic effects.
- Most titles are recent, indicating Netflix's focus on continuous new releases.

2. Genre Distribution

Steps:

- Split multiple genres per title and counted their occurrences.
- Visualized top 20 genres.

Functions Used:

- .str.split(), .stack(), .str.strip() → to handle multiple genres in one cell
- .value_counts() → to count occurrences
- .plot(kind='barh') → horizontal bar chart for readability

Insights:

- International movies and dramas are the most common, showing Netflix's global reach.
- Comedies and documentaries are also very popular among viewers.
- There's a good mix of genres for all audiences — from kids to adults.
- Horror and reality TV are less focused, meaning Netflix prefers story-based content.

3. Content Type Distribution (Movies vs TV Shows)

Steps:

- Counted types using:
  `type_counts = df['type'].value_counts()`
- Plotted both bar and pie charts.

Functions Used:

- .value_counts() → Counts categories

- .plot(kind='bar'), .plot(kind='pie') → Visualization

Insights:

- Movies dominate Netflix (≈ 70%), while TV Shows make up ≈ 30%.

- This shows Netflix started as a movie-based platform but has diversified into series and shows for better engagement.

4. Ratings Distribution

Steps:

- Filled missing ratings and counted frequency:
- Plotted bar chart and stacked bar chart (Rating vs Type)

Functions Used:

- .fillna() → Handle missing data
- .value_counts() → Count ratings
- pd.crosstab() → Cross-tab for comparing two columns
- .plot(stacked=True) → Stacked bar visualization

Insights:

- Most Common Ratings: TV-MA, TV-14, and TV-PG.

- Majority content is rated TV-MA (Mature Audience) → Netflix targets adult viewers.

- Movies are mostly rated R/PG-13, while TV Shows are mostly TV-14 or TV-MA.

5. Country-Level Analysis

Steps:

- Handled multiple countries per show and counted.
- Plotted Top 10 Countries using a horizontal bar chart.

Functions Used:

- .dropna(), .str.split(), .stack(), .value_counts()
- .plot(kind='barh') → Horizontal bar chart

Insights:

- Top Contributors:
  - 🇺🇸 United States
  - 🇮🇳 India
  - 🇬🇧 United Kingdom
  - 🇨🇦 Canada
  - 🇫🇷 France
- Indicates Netflix's strong presence in the U.S. market, followed by emerging regions like India and Europe.
- Highlights Netflix's growing global content strategy.