**Netflix Insights**

**Data Cleaning Steps (using Pandas)**

1. Loaded the raw Netflix dataset into a Pandas DataFrame.

2. Removed duplicate records to ensure uniqueness of shows.

3. Handled missing values by either filling or removing them depending on the column.

4. Dropped unnecessary columns (e.g., 'Unnamed: 0' index column).

5. Standardized date formats in the 'date\_added' column.

6. Converted 'release\_year' to integer type for proper analysis.

7. Ensured 'duration' field consistency (minutes for Movies, seasons for TV Shows).

8. Cleaned and standardized text fields such as 'title', 'director', 'cast', and 'country'.

**Metrics and Insights**

• Movies vs TV Shows distribution:

- Movie: 5185

- TV Show: 147

• Top 5 most common ratings:

- TV-MA: 1822

- TV-14: 1214

- R: 778

- PG-13: 470

- TV-PG: 431

• Top 5 countries producing content:

- United States: 1846

- India: 875

- United Kingdom: 183

- Canada: 107

- Spain: 91

• Content release trend (last 5 years):

- 2017: 657

- 2018: 648

- 2019: 519

- 2020: 442

- 2021: 161

• Top 5 most popular genres:

- International Movies: 2369

- Dramas: 2293

- Comedies: 1553

- Action & Adventure: 806

- Independent Movies: 740

**SOME FUNCTIONS I HAVE USED TO CLEAN THE DATA**

**1 ) print("\nmissing values per coloumn:")**

**print(df\_read.isnull().sum())**

**df\_cleaned = df\_read.dropna()**

dropna() removes **all rows** that contain **any missing values**, regardless of the column.

**2)** import pandas as pd

df\_read = pd.read\_csv("/Volumes/workspace/default/infoysys/INFOYSYS.csv")

display(df\_read)

**print("\nmissing values per coloumn:")**

**print(df\_read.isnull().sum())**

The dataset contains incomplete records, especially in descriptive fields like director and cast.

Further cleaning is required before analysis to ensure accuracy and consistency.

**3) df\_read['type'] = df\_Read['type'].str.strip()**

**df\_read['title'] = df\_Read['rating'].str.strip()**

Removed leading and trailing whitespaces from key text columns (type, title, rating) using .str.strip().

**4) display(df\_read)**

**df\_read.country.value\_counts()**

Counts how many times each unique country appears in the country column.

**5)** **df\_read['duration\_minutes'] = df\_read['duration'].str.extract(r'(\d+)').astype(float)**

Extract only the numeric part of the duration (e.g., "90" from "90 min" or "1" from "1 Season").

**6) df\_read.type.value\_counts()**

Found that the dataset contains 6131 Movies and 2676 TV Shows.  
This shows that Movies make up the majority of the content in the dataset.

**NORMALIZED DATA STEPS**

**Summary of Data Cleaning and Normalization Steps**

1. **Data Loading:**
   * Loaded the raw Netflix dataset (INFOYSYS.csv) into a Pandas DataFrame.
2. **Missing Values Check:**
   * Identified columns with missing/null values such as director, cast, country, date\_added, and rating.
3. **Data Cleaning:**
   * Removed rows with missing values (dropna) to ensure complete records for analysis.
   * Stripped leading and trailing whitespaces from important text columns like type, title, and rating for consistency.
4. **Verification:**
   * Checked for remaining missing values to confirm all nulls were removed.
   * Verified the shape of the cleaned dataset (reduced number of rows after dropping nulls).
5. **Saving Cleaned Data:**
   * Saved the cleaned and normalized dataset as INFOYSYS\_NORMALIZED.csv for future use.

**1. One-Hot Encoding**

* **What it does:**  
  Converts each category in a feature into a new binary (0 or 1) column. For example, a type column with categories Movie and TV Show becomes two columns: type\_Movie and type\_TV Show where each row has a 1 in the column corresponding to its category and 0 elsewhere.
* df\_encoded = pd.get\_dummies(df\_cleaned, columns=['type'])

**2) Label Encoding**

* **What it does:**  
  Assigns each unique category a unique integer label. For example, rating categories like PG, R, TV-MA might be encoded as 0, 1, 2, etc.

le = LabelEncoder()

df\_cleaned['rating\_encoded'] = le.fit\_transform(df\_cleaned['rating'])

**Milestone 2: Week 3 & 4 - EDA & Feature Engineering**

EDA is the process of analyzing and summarizing a dataset to understand its main characteristics, detect patterns, spot anomalies, and check assumptions before applying modeling or machine learning.

**Key Objectives of EDA:**

1. Understand the data
2. Summarize the data
3. Visualize patterns
4. Detect anomalies or missing values
5. Generate insights

**Examples of EDA Tasks in our Netflix Dataset:**

* Distribution of release years.
* Count of Movies vs TV Shows.
* Popular genres.
* Content added per year or month.
* Country-wise content distribution.

**Libraries I have imported:**

* import pandas as pd
* import matplotlib.pyplot as plt
* import seaborn as sns
* import os

1. **Release Year Distribution**

Objective: Understand how Netflix content is distributed over the years.

Method: Plotted a histogram of the release\_year column using seaborn.histplot.

Insights: Helps identify which years had the most content releases.

Saved Plot: Release\_Year\_Distribution.png

1. Distribution of Content Type:

Counted number of Movies vs TV Shows using sns.countplot.

Saved as Content\_Type\_Distribution.png.

1. Distribution of Ratings

Plotted rating distribution to see how many titles belong to each rating category.

Saved as Ratings\_Distribution.png

1. Distribution of Genre

Split listed\_in column by commas to get individual genres.

Counted frequency of each genre and plotted with a bar chart.

Saved as Genre\_Distribution.png

1. Top 10 Countries by Content Contribution

Split country column and counted titles per country.

Plotted top 10 countries.

Saved as Top\_10\_Countries.png.

6) Content Type vs Rating

Analyzed combination of content type and rating using a count plot.

Saved as Content\_Type\_vs\_Rating.png.

7) Release Year vs Number of Titles by Type

Stacked histogram to see how content of different types (Movie/TV Show) varied over the years.

Saved as Release\_Year\_vs\_Type.png

**Feature Engineering**

**It helps to**

Convert textual or categorical data into meaningful numeric values.

Create new insights like content age or release trends.

Improve performance of models like clustering and classification.

Make visualizations and interpretations much more informative**.**

**content\_length\_category -** Categorized movies/shows based on their duration (short, medium, long)

**type\_rating** - Combined content type and rating (e.g., “Movie\_PG-13”, “TV-MA”).

**content\_age -** Computed how old the movie/show is using 2025 - release\_year.

**added\_year, added\_month, added\_day, added\_weekday** - Extracted from date\_added column using datetime conversion.

**duration\_mins** - Converted text like “90 min” or “2 Seasons” into numerical minutes.

**rating\_num** - Converted categorical ratings (“PG”, “TV-MA”, etc.) into numerical values (1–5).