Exploratory Data Analysis (EDA) And Feature Engineering

Week 3&4:

1. Introduction

This document outlines the complete process of Exploratory Data Analysis (EDA) and Feature Engineering performed on the Netflix dataset. It includes steps, explanations of functions used, and placeholders for visual outputs and insights.

1.1 What is EDA?

Exploratory Data Analysis (EDA) is the process of analyzing datasets to summarize their main characteristics using visual and statistical techniques. It helps in understanding data distribution, detecting anomalies, spotting patterns, and forming hypotheses.

1.2 What is Feature Engineering?

Feature Engineering is the process of using domain knowledge to create new meaningful features from raw data, which can improve model performance and insights.

2. Why EDA is Important Before Machine Learning

Before building any machine learning model, it is essential to understand the dataset. EDA helps identify data quality issues, understand patterns, detect outliers, and decide which features are relevant. Without EDA, models may produce misleading results due to hidden biases or noise.

3. Importance of Feature Engineering in Real Projects

Feature Engineering improves the predictive power of models by creating meaningful variables. In real-world projects, raw data is rarely perfect. Creating new features helps models capture deeper insights and relationships within the data, leading to better performance and interpretability.

4. Detailed EDA Process with Explanation

1. **Import Libraries** – We import pandas for data manipulation, matplotlib and seaborn for visualization. These libraries provide powerful functions to analyze and visualize datasets.
2. **Load Dataset** – The dataset is loaded using pd.read_csv(), which converts CSV data into a DataFrame that is easier to work with.
3. **Data Inspection** – Using head(), info(), and describe() helps us get a quick overview of dataset structure, column types, and summary statistics.

4. **Handle Datatypes** – Converting date_added to datetime allows us to extract year and perform time-based analysis.
5. **Missing Value Analysis** – Detecting null values is important to decide whether to drop, impute, or replace missing data.
6. **Univariate Analysis** – We analyze one feature at a time to understand its distribution, such as most common ratings or genres.
7. **Bivariate/Time Series Analysis** – Here, we analyze how content changes over time, which reveals growth trends on Netflix.
8. **Feature Engineering** – We create new features like Content Length Category to get more insights beyond raw data.
9. **Visualization** – Charts make it easier to interpret insights quickly and present findings clearly.
10. **Export Processed Dataset** – Finally, we save the enhanced dataset for future analysis or machine learning.

5. Code Functions Explanation

Function	Purpose
pd.read_csv()	Load CSV data into a DataFrame
df.head()	Display first 5 rows
df.info()	Show data types and memory usage
df.describe()	Summary statistics for numeric features
pd.to_datetime()	Convert string dates to datetime
df.isnull().sum()	Count missing values per column
sns.countplot()	Plot counts of categorical variables
df.groupby()	Aggregate data for trend analysis
plt.figure()	Create a new plot area
sns.heatmap()	Visualize correlation between numeric columns

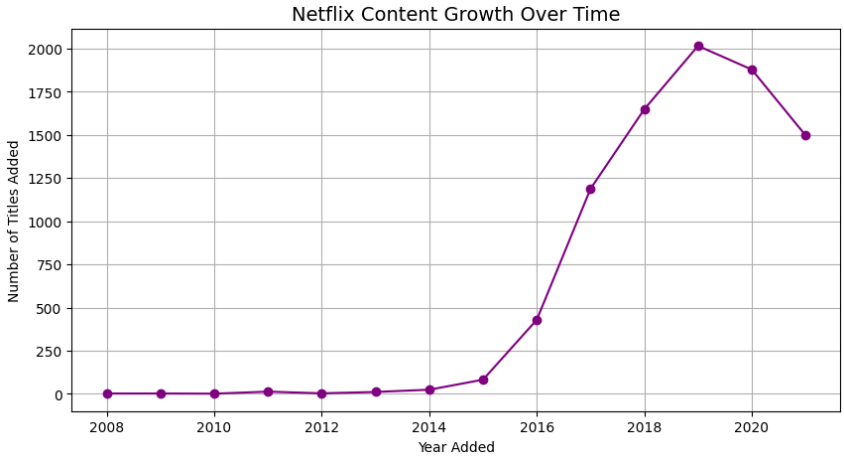
---

4. Netflix Content Growth Over Time

This step helps us see how Netflix content has grown over the years. By converting the date into proper datetime format, we can group content by year and easily visualize the trend. This gives a clear idea of whether Netflix is expanding its content library steadily or in sudden jumps.

What we are doing: Converting dates and counting how many titles were added each year to understand the growth trend.

Output :



Insight Example: "Here, we can observe that after 2016, Netflix started rapidly increasing the number of releases, showing platform expansion."

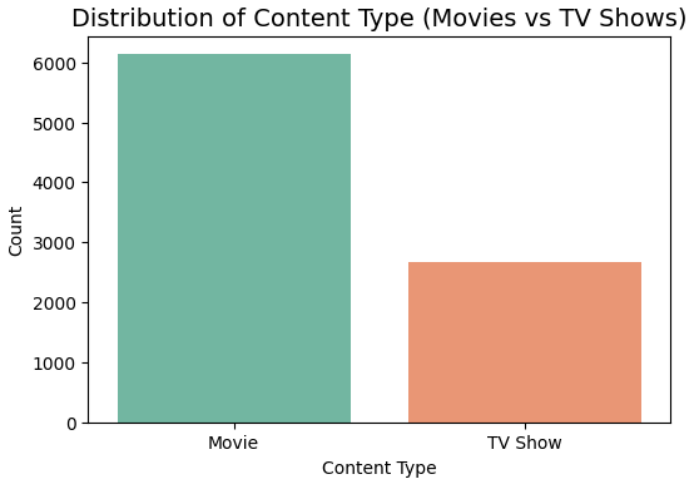
---

5. Distribution Analysis

Distribution analysis helps us understand what type of content Netflix favors the most.

a) Content Type Distribution (Movies vs TV Shows)

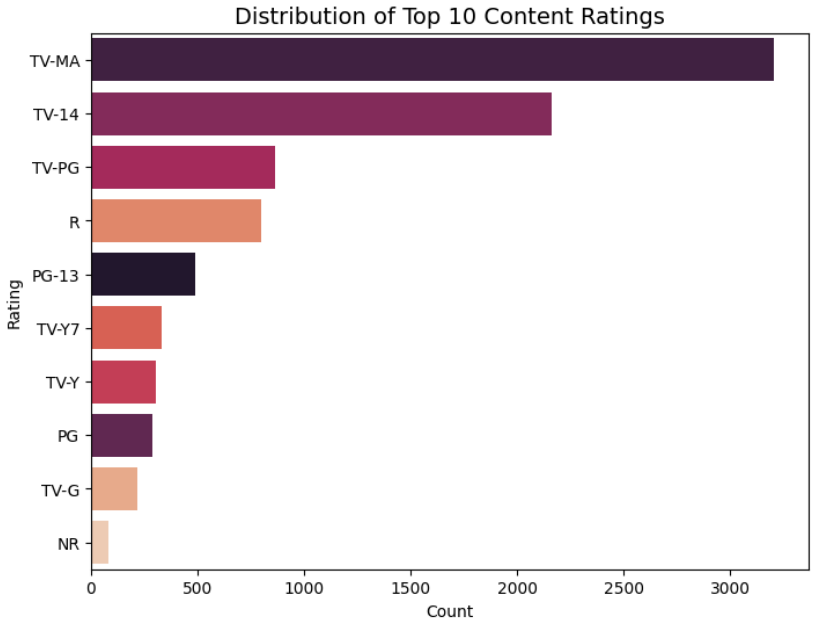
Here, we check whether Netflix focuses more on Movies or TV Shows. This gives a quick view of platform strategy.



Insight Example: "If Movies are dominating, it shows Netflix is more focused on film content than series."

b) Ratings Distribution

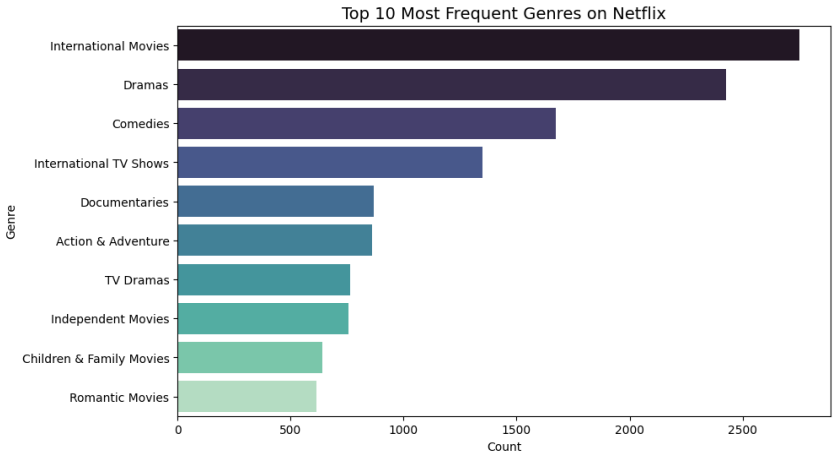
Ratings tell us what type of audience Netflix targets (Kids, Teens, Adults). This helps understand content maturity levels.



Insight Example: "If TV-MA is high, Netflix is pushing more adult-focused content, likely to attract mature viewers."

c) Genre Distribution

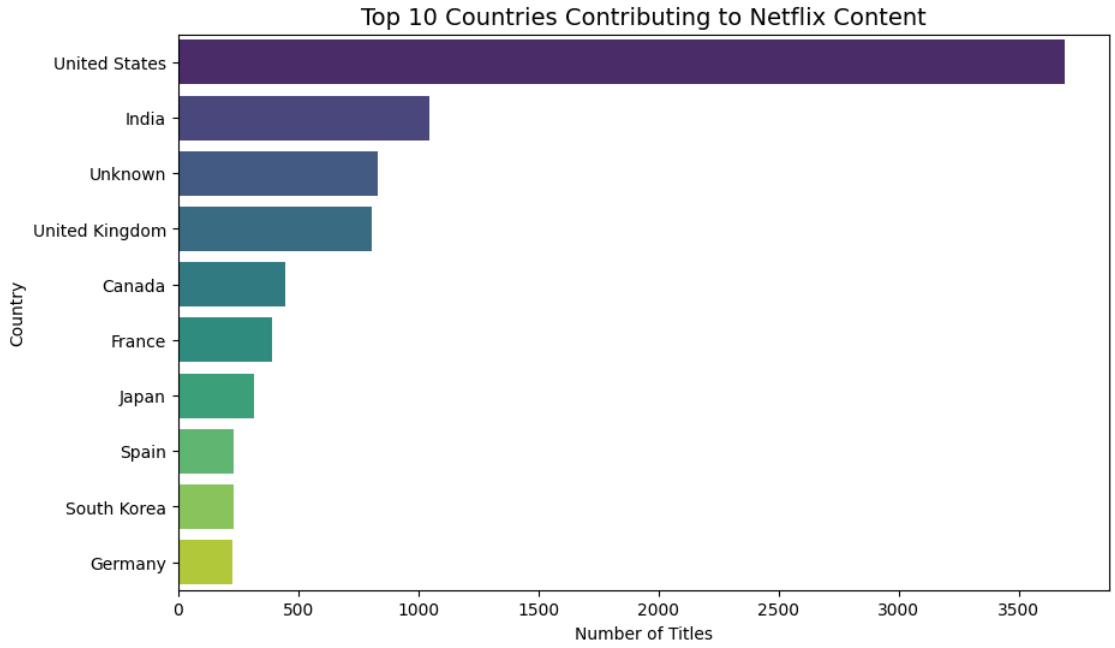
This helps us identify the most popular genre. Knowing the top genres tells us about audience preferences.



6. Country-Level Content Contribution

Finding which countries contribute the most content helps us understand Netflix's global reach. This also highlights regional strategy.

What we are doing: Counting which countries produce the highest number of titles.



Insight Example: "USA leads by a large margin, showing Netflix's strong presence in American entertainment, followed by India and UK."

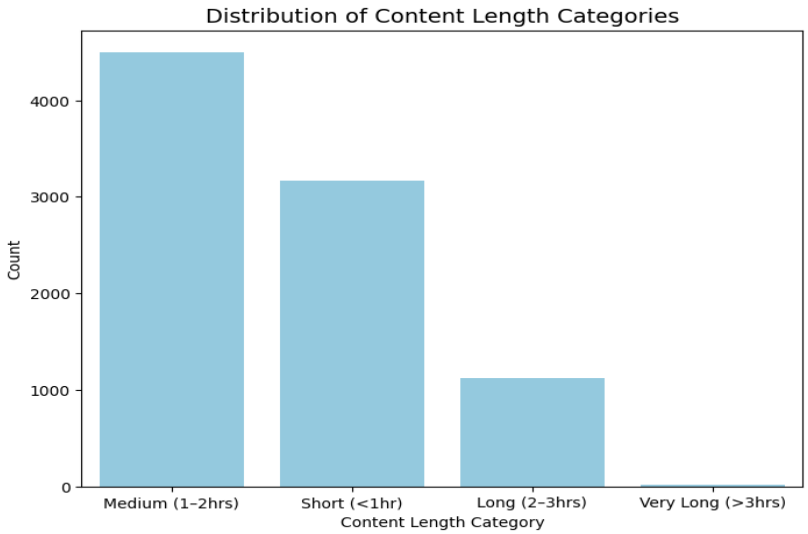
---

7. Feature Engineering

Feature Engineering helps us go beyond raw data and create smarter insights. It is the process of transforming raw data into meaningful features that better represent underlying patterns. It helps models understand data more effectively by creating new informative columns or modifying existing one's .Smart feature engineering often has a bigger impact on performance than choosing complex algorithms.

a) Content Length Category

We categorize movies by duration to identify viewer preferences (Short, Medium, Long format).

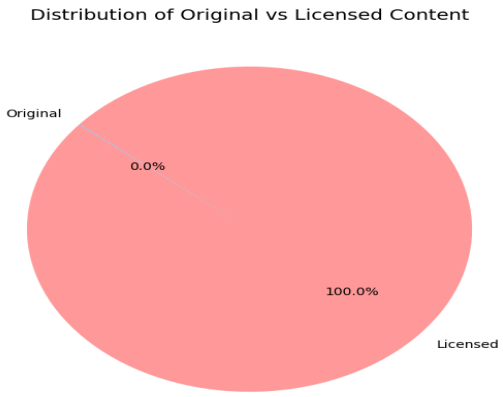


Insight Example: "If medium-length content is higher, it may mean users prefer quick-to-watch shows."

b) Original vs Licensed

This helps us differentiate between Netflix Originals and Third-party licensed content.

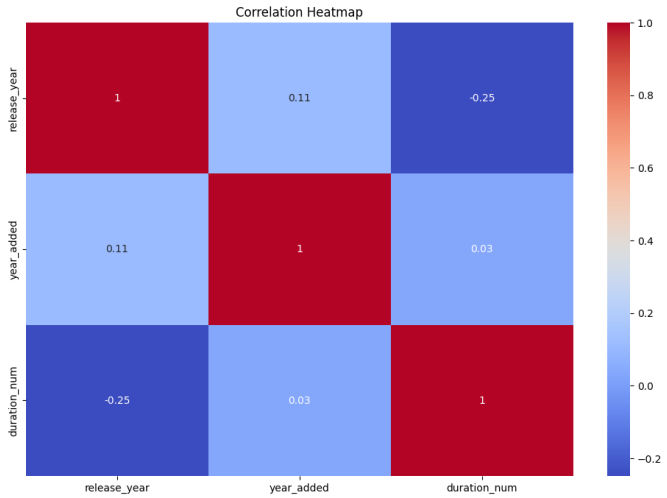
Insight Example: "If Originals are increasing, Netflix is strengthening its brand identity rather than depending on other production houses."



Pie Chart: Original vs Licensed Content

8. Correlation Heatmap (Numeric Features)

A correlation heatmap helps us understand which numeric features relate to each other. This is especially useful before modeling.



Insight Example: "If duration has no correlation with other fields, it means it's independent and might be used as a unique feature."

---

9. Exporting Final Dataset

- File saved as netflix_feature_eda.csv.

Code Insight:

df_read.to_csv("/Volumes/workspace/default/netflix/netflix_feature_eda.csv", index=False)

output: *Dataset saved successfully.*