1. Data Cleaning Process

Section	Description
Process	**Data Cleaning** is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database. It involves identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and replacing, modifying, or deleting the dirty data.
Steps	1. **Data Inspection and Loading:** Read the raw dataset into a DataFrame. 2. **Data Type Conversion:** Convert columns to the appropriate data type. For instance, the 'date_added' column was converted from an object type to the proper datetime64[ns] type to enable time-series analysis. 3. **Missing Value Handling :** Identify missing values using df.isna().sum(). All missing values in the dataset were replaced with the value 0 using df.fillna(0) to ensure a complete dataset for subsequent steps.

Section	Description
	4. **Handling of Duplicates:** Check for and remove duplicate rows to ensure data uniqueness.
Metrics / Outcome	1. **Missing Value Count:** The primary metric is the count of missing values per column, aiming for a count of **0** in all essential columns. 2. **Data Type Integrity:** Verification that columns like 'date_added' are successfully converted to the correct data type 3. **Data Completeness:** Final dataset contains the full expected number of rows and columns

2. Normalization Process

Section	Description
Process	**Normalization** in data preprocessing is the process of adjusting values measured on different scales to a notionally common scale, often to prevent features with larger ranges from dominating the model training

Section	Description
	process. (Note: The uploaded notebook also includes feature encoding steps).
Steps	1. **Data Loading:** Read the cleaned dataset for further transformation. 2. **Data Imputation:** Fill in missing values for text-based columns, such as replacing missing values in the 'director' column with a placeholder like "Unknown Director". 3. **Categorical Encoding:** Create a new feature by mapping a categorical variable to a numerical code. A feature called 'director_code' was created, where a **Movie** type is mapped to **1** and a **TV Show** type is mapped to **0**. 4. **Numerical Scaling/Standardization:** Apply techniques like Min-Max Scaling or Z-Score Standardization to numerical columns if they were not already handled in the Feature Engineering stage.

Section	Description
Metrics / Outcome	1. **Missing Value Resolution:** Confirmation that all placeholder values (e.g., "Unknown Director") have successfully replaced the original missing values. 2. **Encoding Consistency:** Verification that the encoding rule is correctly applied across all rows (e.g., every 'Movie' row has director_code = 1). 3. **Distribution Uniformity (Inferred):** For true normalization (scaling), the resulting numerical distribution should have a more uniform range (e.g., 0 to 1 for Min-Max Scaling) or mean 0 and standard deviation 1 (for Standardization).

3.Exploratory Data Analysis (EDA) Process

Section	Description
Process	**Exploratory Data Analysis (EDA)** is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. It is used to discover

Section	Description
	patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations.
Steps	1. **Load Cleaned Data:** Read and display the cleaned dataset (e.g., netflix_cleaned.csv). 2. **Structural Check:** Examine the dataset's basic structure, including the number of rows/columns and data types (df.info()) to confirm the data is ready for analysis. 3. **Summary Statistics:** Generate and print "Quick Insights" which include mode and frequency counts for key categorical columns. 4. **Visualization :** Create plots (e.g., bar charts, histograms, heatmaps) to visualize distributions and relationships between features.
Metrics / Outcome	1. **Data Count:** Total record count 2. **Feature Distribution (Counts):** Quantified distribution of data types (e.g., **Movies: 6130** and **TV Shows: 2676**).

Section	Description
	3. **Mode/Frequency:** Identification of the most frequent categories, such as the **Most common rating (TV-MA)**, **Most frequent country (United States)**, and **Most frequent genre (dramas, international movies)**. 4. **Data Quality Score (Inferred):** Insights into data consistency and potential outliers are implicitly measured.

4.Feature Engineering

Section	Description
Process	**Feature Engineering** is the process of using domain knowledge to select, transform, or create variables (features) that best represent the underlying problem to a predictive model, resulting in improved model accuracy.
Steps	1. **Categorical Encoding (Label Encoding):** Use techniques like LabelEncoder (from sklearn.preprocessing) to convert

Section	Description
	categorical columns, such as 'type', into numerical representations creating a new feature 'type_encoded'. 2. **Feature Creation (Binning/Grouping):** Create a new, derived feature by grouping values from an existing feature. A function was used to categorize the 'duration' column into four distinct bins: 'Short', 'Medium', 'Long', and 'Series', creating the feature 'Content_Length_Category'. 3**. Numerical Scaling (MinMaxScaler):** Scale a numerical column to a fixed range. The MinMaxScaler was applied to 'release_year' to scale its values to the range of 0.0 to 1.0, resulting in the new feature 'release_year_scaled'.
Metrics / Outcome	1. **Range Validation (Scaling):** Scaled numerical features (e.g., 'release_year_scaled') must fall within the defined range (e.g., 0.0 to 1.0).

Section	Description
	2. **Successful Categorization:** Verification that all values in the original column ('duration') are correctly mapped to one of the defined categories in the new feature ('Content_Length_Category'). 3. **Information Gain :** The success of Feature Engineering is ultimately measured by the improvement in the predictive model's performance metrics (e.g., accuracy, precision) when trained on the new features.