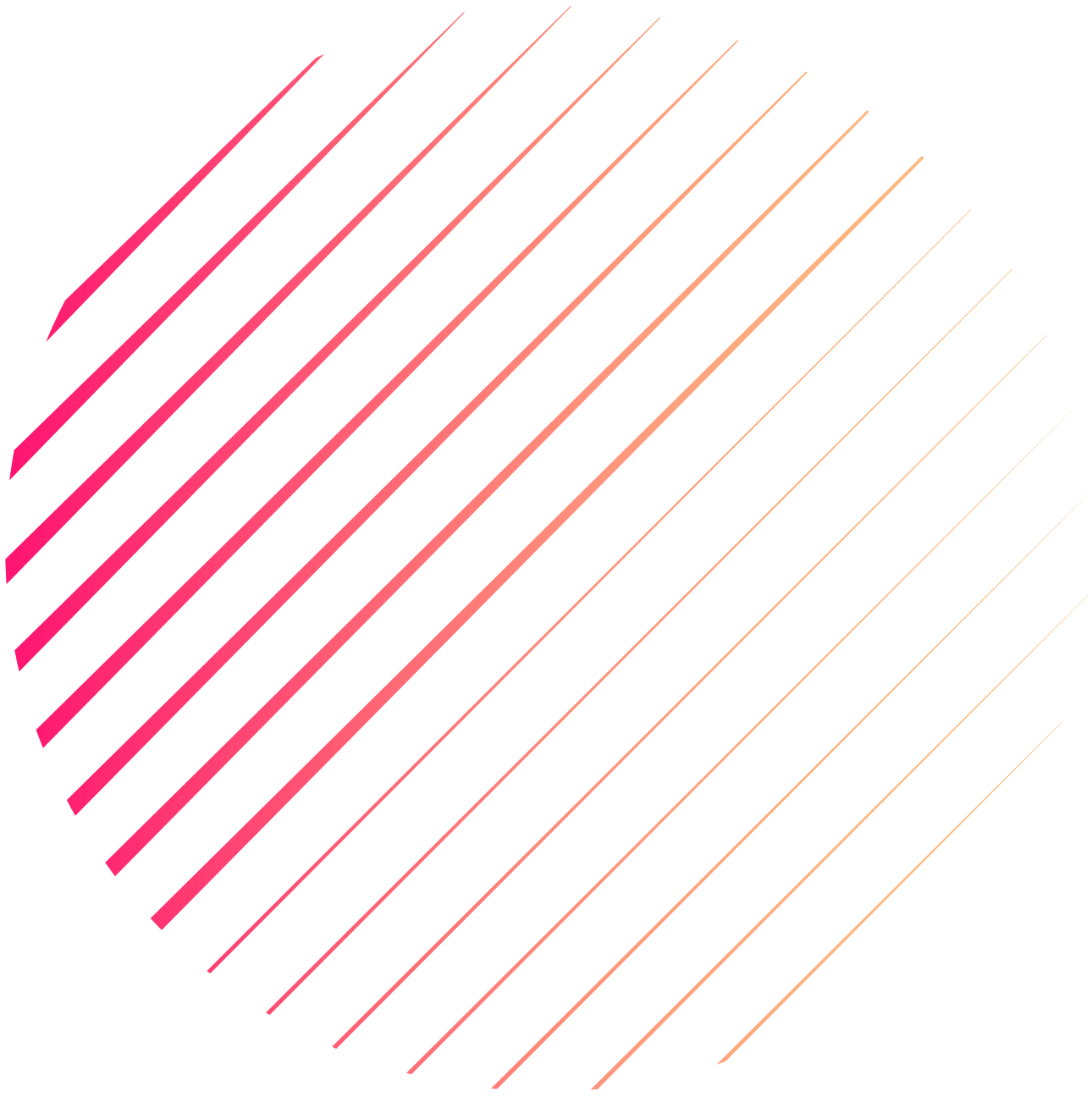
Preprocessing of Data



Student	Roll No
Amanat Ali Mirza	Bsmm-f21-004
Shanzay Iftihkar	Bsmm-f21-022
Danish Arooj	Bsmmm-f23-002
Nimra Sarwar	Su92-Bsmm-s23-001

Course:	Tools and Techniques for Data Science
Submitted to:	Dr. Nimra Tariq

I. Introduction

This report provides a comprehensive analysis of a Python code that explores and cleans a dataset named "titles.csv". The code leverages the capabilities of pandas (`pd`) for data manipulation, seaborn (`sns`) for visualization, and matplotlib (`plt`) for graphical representation.

II. Code Functionality

A. Data Loading and Initial Assessment

1. **Library Imports:** The code imports essential libraries to facilitate data analysis (`pandas`), data visualization (`seaborn` and `matplotlib`), and potentially others based on specific requirements.
2. **Data Loading:** The `pd.read_csv` function is employed to read the "titles.csv" file into a pandas DataFrame object named `df`. This DataFrame serves as the primary data structure for storing and manipulating the dataset.
3. **Data Shape:** The code retrieves the dimensions (number of rows and columns) of the DataFrame using `df.shape`. This provides insights into the overall size and structure of the data.
4. **Data Preview:** The code displays the initial state of the DataFrame using `print(df)`. This offers a rudimentary glimpse into the data's contents.

B. Exploratory Data Analysis

5. **Number of Shows and Movies:** The code isolates entries based on the "type" column to determine the counts of shows and movies. Boolean indexing and `shape[0]` are utilized for efficient filtering and counting.

C. Data Characteristics

6. **Data Types and Null Values:** The `df.info()` function provides a summary of data types present in each column and the number of null values encountered. This information is crucial for understanding the data's structure and potential data quality issues.

D. Data Cleaning Techniques

7. **Null Value Imputation:** The code addresses missing values (nulls) in specific columns using the `fillna` function with the `inplace=True` parameter to modify the DataFrame directly. Here's a breakdown of the imputation strategies:
    - `age_certification`: Replaced with "NO AC available" (assuming a placeholder for missing age certification information).
    - `description`: Replaced with "Not_available" (acting as a placeholder for entries lacking descriptions).
    - `seasons`: Replaced with 0.0 (assuming numerical representation of seasons).
    - `imdb_score`: Imputed with the median value, which is a robust measure for skewed distributions that can be less sensitive to outliers compared to the mean.
    - `imdb_votes`: Imputed with the mean value, representing the average number of votes across entries.
    - `tmdb_popularity`: Replaced with the maximum value, assuming higher values indicate greater popularity.
    - `tmdb_score`: Replaced with the minimum value, on the assumption that lower scores might be more indicative of missing data than higher scores (depending on the data).
    - `imdb_id`: Replaced with "Tm0000" as a placeholder for missing IDs.
8. **Identifying Null Values:** The code leverages `isna().any(axis=1)` to identify rows containing any null values. Additionally, `df[condition]` is used for targeted filtering based on specific null value conditions (e.g., rows with missing values in the "title" or "imdb_id" columns).
9. **Handling Rows with Null Values:** The code employs `dropna(inplace=True)` to eliminate rows from the DataFrame that contain any null values. This approach ensures a dataset free of missing entries, but it's essential to consider potential data loss and potential biases introduced by dropping rows.

E. Data Visualization (using Matplotlib and Seaborn)

10. **Count Plots:** The code creates count plots for the "type" and "age_certification" columns using `sns.countplot`. These visualizations effectively reveal the distribution of categorical data within these columns.
11. **Histogram:** The code generates a histogram for the "release_year" column using `sns.histplot` with `bins=30` and `kde=True`. This plot depicts the distribution of release years along with a kernel density curve, providing a smoothened representation of the data's underlying distribution.

III. Conclusion

The provided Python code effectively explores and cleans the "titles.csv" dataset. It identifies null values, replaces them with appropriate strategies, and generates visualizations to understand the data's characteristics. This report serves as a documentation of the code's functionality and the resulting cleaned DataFrame.