**Experiment 1**

**Shashwat Tripathi**

**D15A Batch C**

**Roll No: 64  
  
AIM : Data preparation using NumPy and Pandas**

**THEORY :**

**1. NumPy: Numerical Computing in Python**

NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these elements. In the context of data preparation, NumPy is often used for:

- Data Representation: NumPy arrays are efficient data structures for storing and manipulating numerical data. They offer advantages in terms of memory efficiency and computation speed compared to native Python lists.

- Data Cleaning: NumPy provides functions for handling missing data, filtering, and cleaning operations. For example, you can use functions like `np.isnan()` to identify missing values and `np.where()` to replace them.

- Data Transformation: NumPy allows for various array manipulations, such as reshaping, merging, and splitting arrays. These operations are valuable when transforming raw data into a format suitable for analysis or modeling.

**2. Pandas: Data Manipulation and Analysis**

Pandas is built on top of NumPy and provides high-level data structures, such as Series and DataFrame, along with tools for data manipulation and analysis. Key aspects of Pandas in data preparation include:

- Data Loading and Cleaning: Pandas simplifies the process of reading data from different file formats (e.g., CSV, Excel, SQL databases) into DataFrames. It provides methods for handling missing data, duplications, and outliers.

- Data Exploration: Pandas offers functions for basic statistical analysis, summarization, and exploration of data. The `describe()` function, for instance, provides summary statistics, and `value\_counts()` helps understand the distribution of categorical variables.

- Data Transformation: Pandas facilitates data transformation through operations like filtering, grouping, merging, and pivoting. The `groupby()` function is particularly useful for aggregating data based on specific criteria.

- Feature Engineering: Pandas allows for the creation of new features by combining or transforming existing ones. This is a crucial step in preparing data for machine learning models.

**3. Workflow: Combining NumPy and Pandas**

The typical data preparation workflow involves using NumPy for low-level numerical operations and array manipulations and Pandas for higher-level data manipulation and analysis tasks. For instance, you might use NumPy to handle missing values at a lower level, and then switch to Pandas to leverage its powerful tools for cleaning, transforming, and exploring tabular data.

**OUTPUT :**

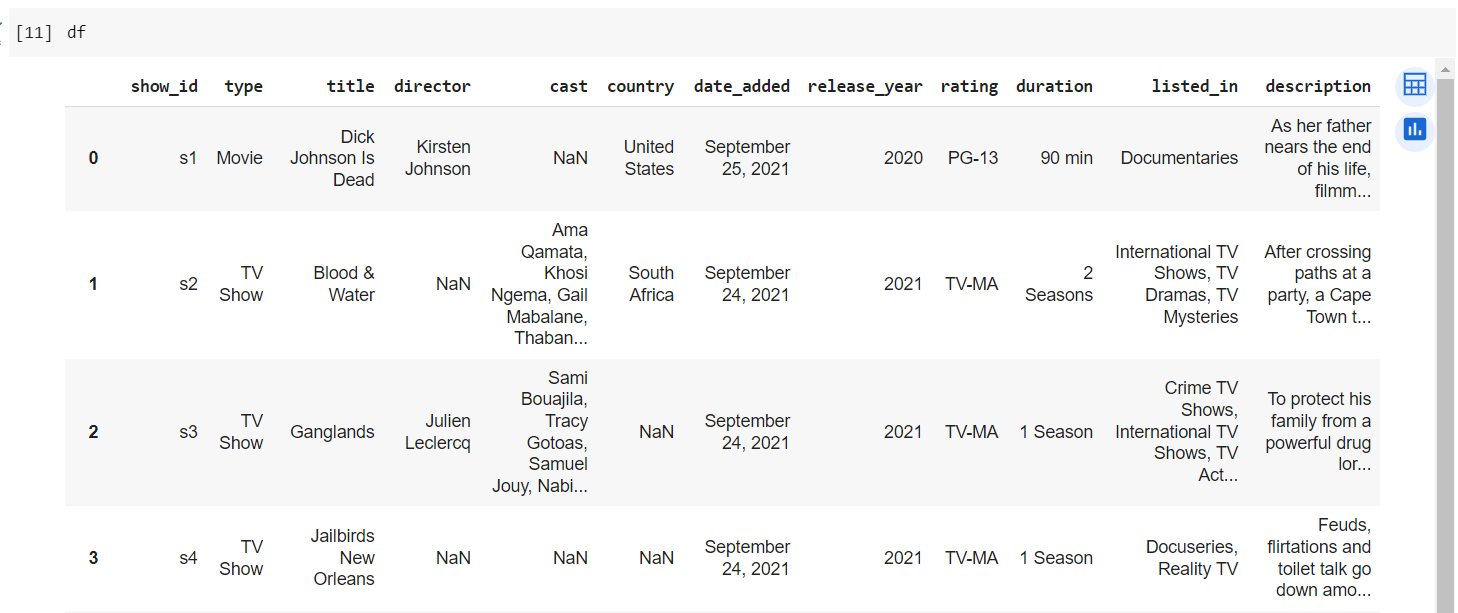
**1. Load data in pandas**

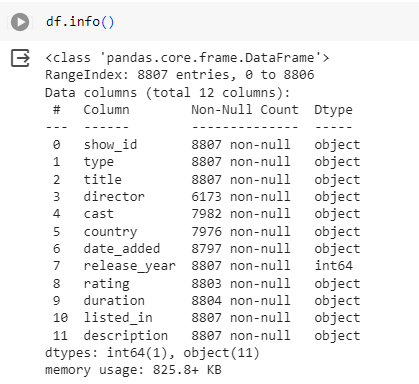
Loading data into Pandas is a crucial initial step in the data analysis process. The pandas library provides functions like read\_csv(), read\_excel(), or read\_sql() to read data from different file formats or databases into a Pandas DataFrame. Once loaded, a DataFrame provides a structured and flexible representation of the data, making it easy to perform various operations.

****

**2. Description of the dataset.**

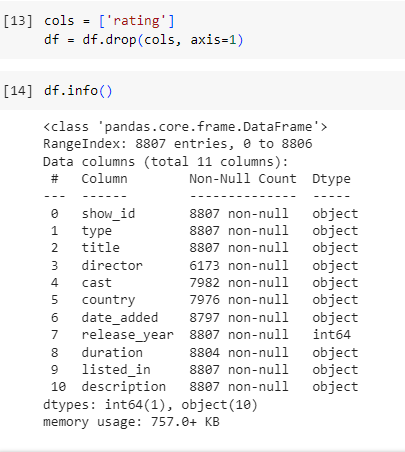
Understanding the structure and characteristics of the dataset is important for effective analysis. The info(), describe(), and head() methods in Pandas provide information about the data types, summary statistics, and a preview of the dataset, respectively.

****

****

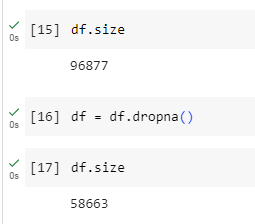
**3. Drop columns that aren’t useful**

Not all columns in a dataset are relevant for analysis or modeling. You can use the drop() method to eliminate columns that are not useful.

****

**4. Drop rows with maximum missing values.**

Handling missing data is crucial. You can identify and drop rows with a significant number of missing values using the dropna() method.

****

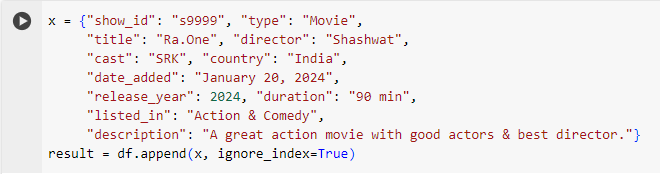
**5. Take care of missing data**

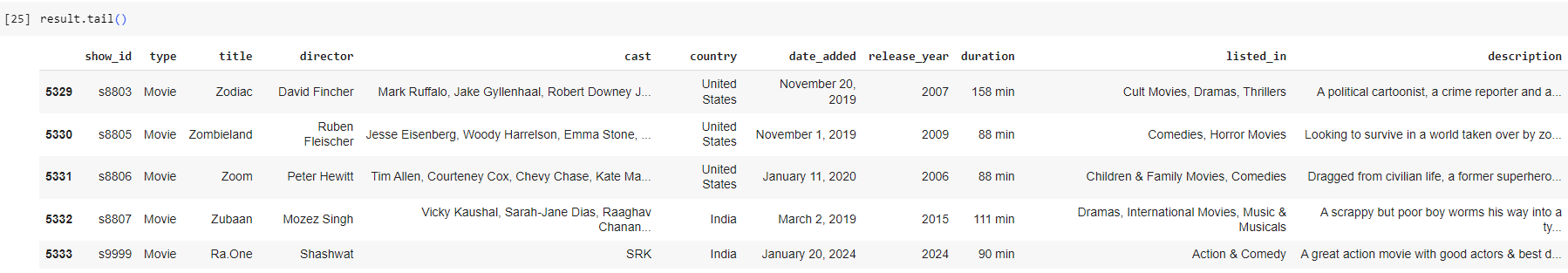
For columns with missing values, you can choose to fill them with a specific value or use methods like mean, median, or interpolation to impute missing values.

****

**6. Create dummy variables**

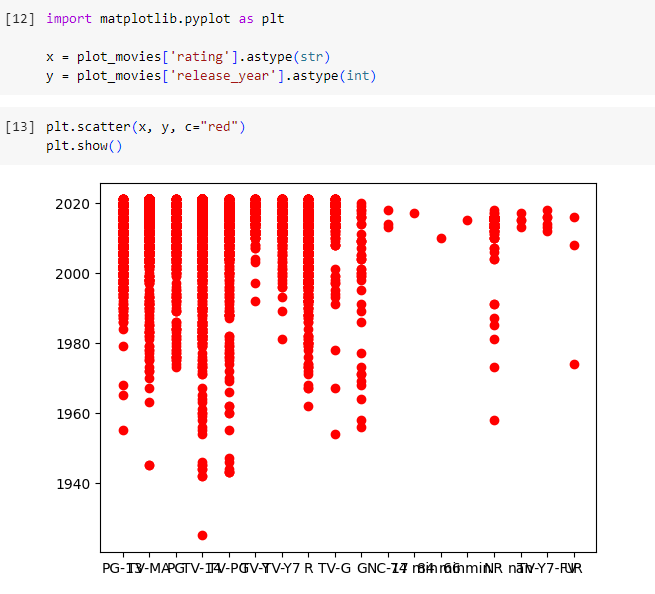
For categorical variables, creating dummy variables is essential for including them in machine learning models.

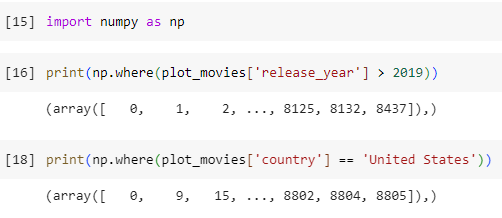
****

****

**7. Find out outliers (manually)**

Outliers can be detected visually or using statistical methods. Visual inspection through box plots or scatter plots is a common approach.

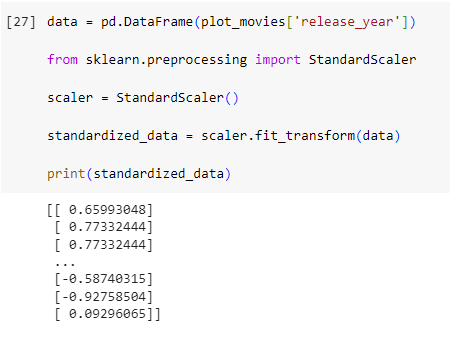
****

****

**8. Standardization of columns**In statistics and machine learning, data standardization is a process of converting data to z-score values based on the mean and standard deviation of the data.

The resulting standardized value shows the number of standard deviations the raw value is away from the mean.

Basically, each value of a given feature of a dataset will be converted to a representative number of standard deviations that it’s away from the mean of the feature.

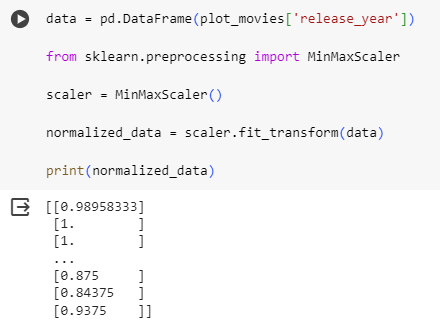
****

**9. Normalization of columns**

In statistics and machine learning, the min-max normalization of data is a process of converting the original range of data to the range between 0 and 1.

The resulting normalized values represent the original data on a 0-1 scale.

This will allow us to compare multiple features together and get more relevant information since now all the data will be on the same scale.

****

**CONCLUSION:** In conclusion, the series of experiments performed on data preparation using NumPy and Pandas, along with various data preprocessing techniques, is essential for ensuring the data is in a suitable and meaningful form for analysis or machine learning applications.