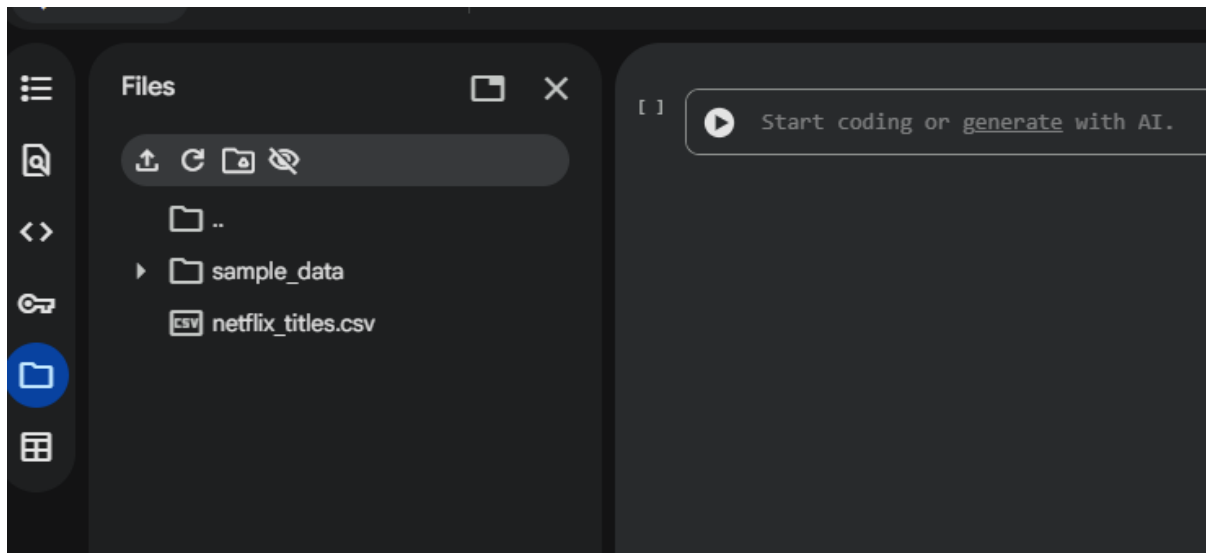


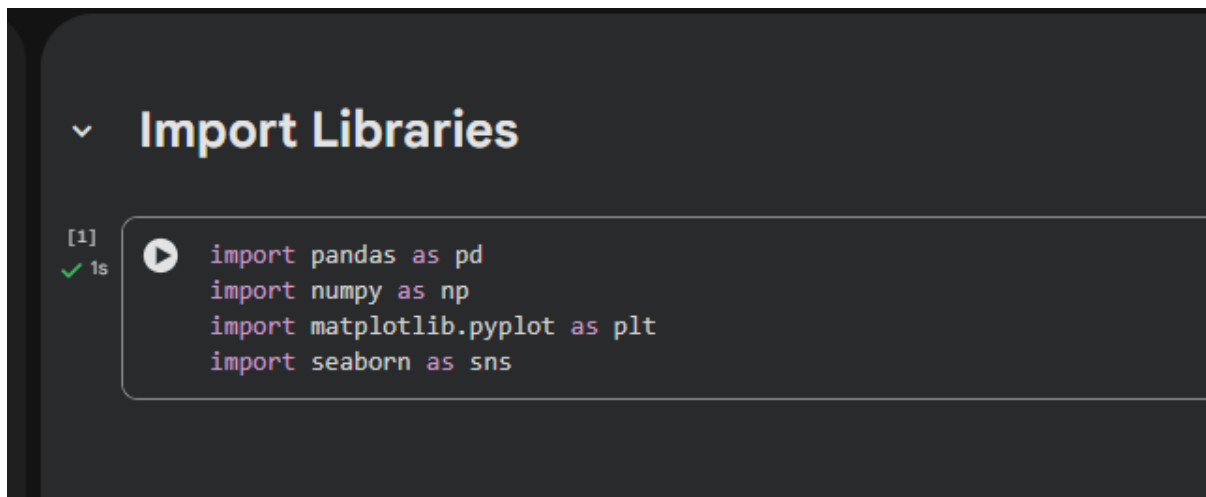
Name: **Sumbul**  
**Abdul wahid**

**Data Science**  
(internship)

**Introduction:** In this task, advanced data cleaning and preprocessing techniques were applied to a real-world dataset as part of Week 1 of the Data Science Internship. The goal was to transform raw and messy data into a clean, analysis-ready format using Python



**Import Libraries:**



**Dataset Description:** The Netflix Movies and TV Shows dataset was selected for this task. It contains information such as title, type, director, cast, country, release year, rating, and duration. The dataset consists of more than 8,000 records and includes missing values and inconsistent data types.

Name: **Sumbul**

**Abdul wahid**

**Data Science**

(internship)

### Load Dataset

```
df = pd.read_csv("netflix_titles.csv")
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mablane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town L...
2	s3	TV Show	Ganglands	Julien	Tracy Gotoas, NaN	NaN	September 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New	+					Gemini 2.5 Flash	Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go

**Initial Diagnostics:** Initial data inspection was performed to understand the structure, size, and data types of the dataset. This step helped identify missing values and columns requiring preprocessing.

### Initial Diagnostics

```
df.shape
df.info()
df.describe(include='all')
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 8807 entries, 0 to 8806  
Data columns (total 12 columns):  
# Column Non-Null Count Dtype  
---  
0 show\_id 8807 non-null object  
1 type 8807 non-null object  
2 title 8807 non-null object  
3 director 6173 non-null object  
4 cast 7982 non-null object  
5 country 7976 non-null object  
6 date\_added 8797 non-null object  
7 release\_year 8807 non-null int64  
8 rating 8803 non-null object  
9 duration 8804 non-null object  
10 listed\_in 8807 non-null object  
11 description 8807 non-null object  
dtypes: int64(1), object(11)  
memory usage: 825.8+ KB

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	NaN	17	220	514	8775
top	s8807	Movie	Zubaan	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	NaN	3207	1793	362	4
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	NaN	NaN	NaN

**Missing Value Analysis:** Missing values were analyzed using both tabular and visual methods. Several columns such as director, cast, country, and rating contained missing data.

Missing Values Analysis

```
[4] df.isnull().sum()
```

	0
show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3

```
5] missing_percent = (df.isnull().sum() / len(df)) * 100  
missing_percent
```

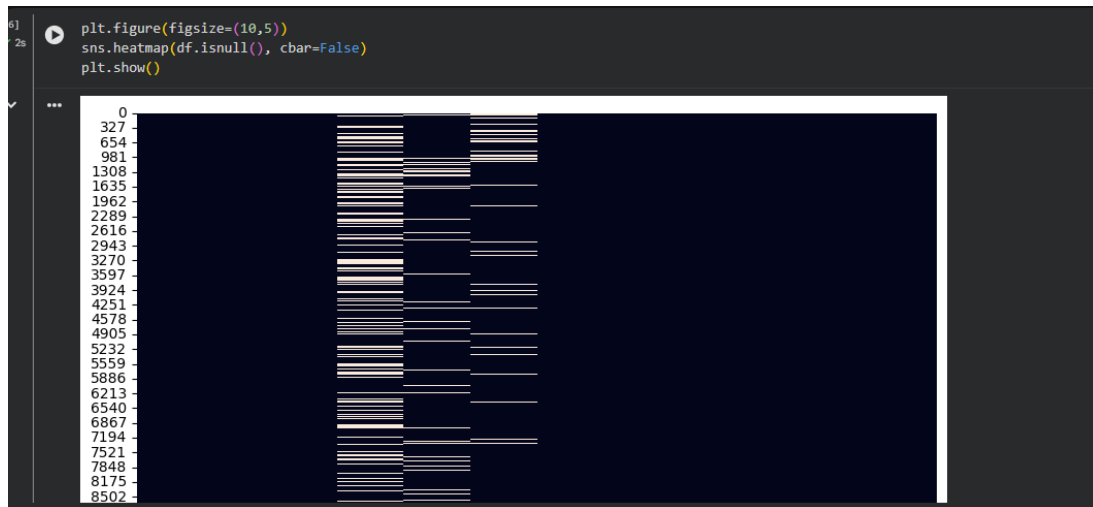
	0
show_id	0.000000
type	0.000000
title	0.000000
director	29.908028
cast	9.367549
country	9.435676
date_added	0.113546
release_year	0.000000
rating	0.045418
duration	0.034064
listed_in	0.000000
description	0.000000

Name: **Sumbul**

**Abdul wahid**

**Data Science**

(internship)



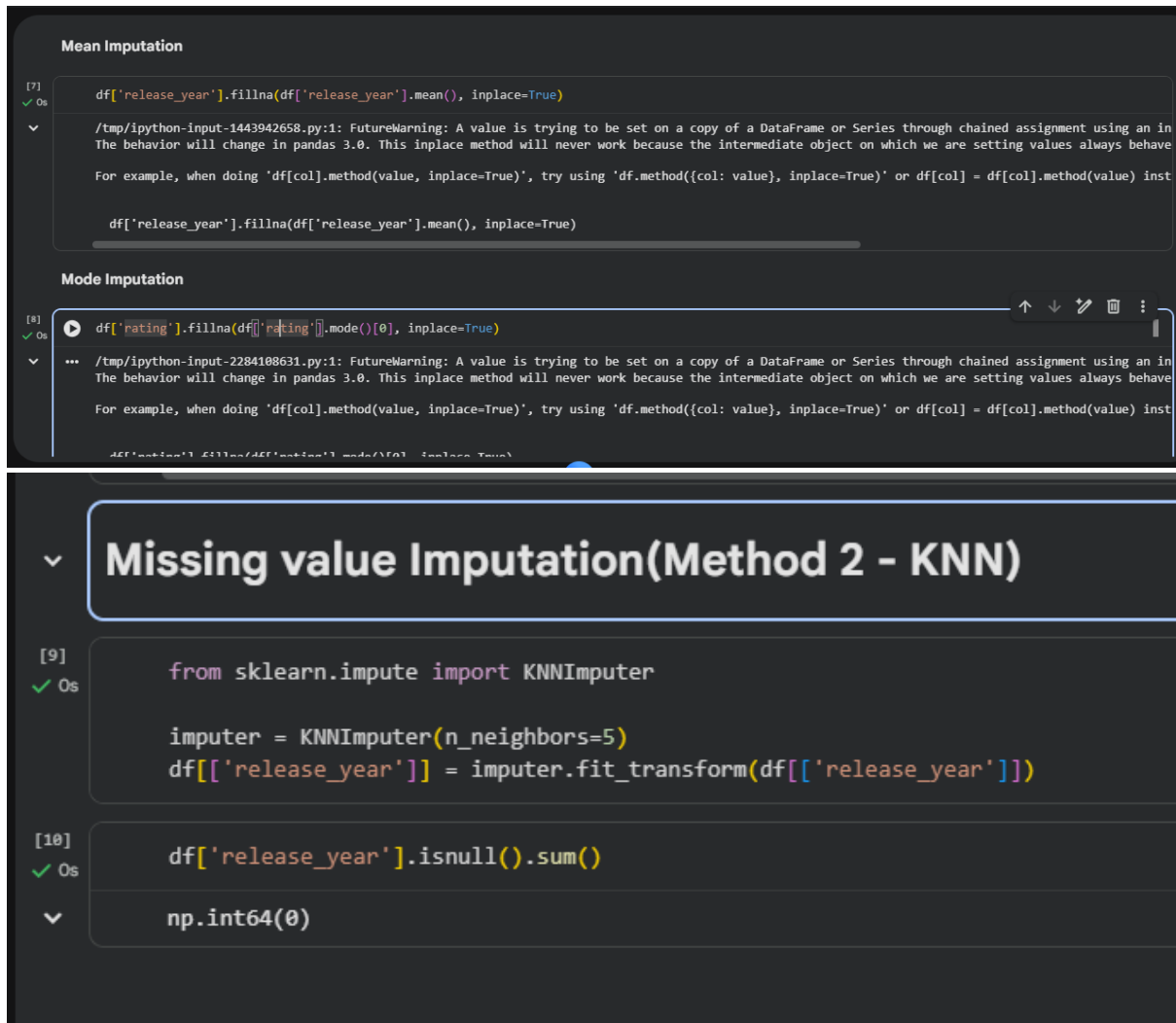
**Missing Value Imputation:** Two different imputation techniques were applied. Statistical methods such as mean and mode were used for numerical and categorical features. Additionally, KNN Imputation was applied as an advanced machine-learning-based technique to handle missing values more effectively.

Name: **Sumbul**

**Abdul wahid**

**Data Science**

(internship)



```
[7] ✓ Os df['release_year'].fillna(df['release_year'].mean(), inplace=True)

/tmp/ipython-input-1443942658.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behave
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst

df['release_year'].fillna(df['release_year'].mean(), inplace=True)

Mode Imputation

[8] ✓ Os df['rating'].fillna(df['rating'].mode()[0], inplace=True)

... /tmp/ipython-input-2284108631.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behave
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst

df['rating'].fillna(df['rating'].mode()[0], inplace=True)

Missing value Imputation(Method 2 - KNN)

[9] ✓ Os from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=5)
df[['release_year']] = imputer.fit_transform(df[['release_year']])

[10] ✓ Os df['release_year'].isnull().sum()

np.int64(0)
```

**Data Type Correction & Date Parsing:** Data type corrections were performed to ensure consistency across the dataset. The date\_added column was converted into a proper datetime format for better usability.

## ▼ Data Parsing Data Type Correction

```
[11] ✓ 0s df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')  
df.info()
```

```
▼ ... <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 8807 entries, 0 to 8806  
Data columns (total 12 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   show_id                8807 non-null   object   
1   type                   8807 non-null   object   
2   title                  8807 non-null   object   
3   director               6173 non-null   object   
4   cast                   7982 non-null   object   
5   country                7976 non-null   object   
6   date_added             8709 non-null   datetime64[ns]  
7   release_year           8807 non-null   float64   
8   rating                 8807 non-null   object   
9   duration               8804 non-null   object   
10  listed_in              8807 non-null   object   
11  description            8807 non-null   object   
dtypes: datetime64[ns](1), float64(1), object(10)  
memory usage: 825.8+ KB
```

**Outlier Detection & Treatment:** Outliers in the release\_year column were detected using the Interquartile Range (IQR) method. Identified outliers were removed to improve data quality and reliability.

## ▼ IQR Method(Outlier Detection)

```
[13] ✓ 0s Q1 = df['release_year'].quantile(0.25)  
Q3 = df['release_year'].quantile(0.75)  
  
IQR = Q3 - Q1  
lower = Q1 - 1.5 * IQR  
upper = Q3 + 1.5 * IQR
```

## ▼ Outlier Removal

```
[14] ✓ 0s df = df[(df['release_year'] >= lower) & (df['release_year'] <= upper)]  
df.shape
```

```
▼ ... (8088, 12)
```

