

Data and Artificial Intelligence

Cyber Shujaa Program

Week 2 Assignment

Netflix Data Wrangling

Student Name: Salome Kungu

Student ID: CS-DA03-26054

Contents

Data and Artificial Intelligence	1
Cyber Shujaa Program.....	1
Week 2 Assignment Netflix Data Wrangling	1
Introduction	2
The key objectives of the assignment were to:.....	2
1. Load and explore the Netflix dataset.....	2
2. Perform data discovery and assess data quality.	2
3. Clean and structure the dataset using data wrangling techniques.	2
4. Validate the dataset for consistency and accuracy.	2
5. Export and publish the final cleaned dataset and notebook on Kaggle.	2
Tasks Completed	2
Conclusion.....	3
Screenshots of the project.....	3
Step 1: Load the Dataset.....	3
Step 2: Data Discovery	5
Step 3: Data Structuring.....	8
Step 5: Handle Missing Values (Imputation & Labeling)	11
Step 6: Remove Records with Critical Missing Data.....	12
Step 7: Error and Logic Validation	13
Step 8: Final Validation and Cleanup	14

Introduction

The objective of this assignment was to apply data wrangling techniques to a real-world dataset sourced from Kaggle, specifically the Netflix Movies and TV Shows dataset. The task focused on developing practical skills in data discovery, structuring, cleaning, transformation, validation, and publication using Python and the Pandas library within a Kaggle Notebook environment.

Through this assignment, I gained hands-on experience in identifying data quality issues such as missing values, duplicates, and inconsistent formatting, and resolving them using systematic and reproducible methods. The final outcome is a cleaned and structured dataset suitable for further analysis and visualization.

The key objectives of the assignment were to:

1. Load and explore the Netflix dataset.
2. Perform data discovery and assess data quality.
3. Clean and structure the dataset using data wrangling techniques.
4. Validate the dataset for consistency and accuracy.
5. Export and publish the final cleaned dataset and notebook on Kaggle.

Tasks Completed

The assignment began with importing the Netflix dataset from a CSV file into a Pandas DataFrame. An initial data discovery phase was conducted to understand the dataset's structure, including the number of records, column data types, missing values, and duplicate entries.

The dataset was then structured by converting date fields into appropriate datetime formats and extracting numeric and categorical components from complex fields such as the duration column. Additional formatting improvements were applied to ensure consistency across columns.

Data cleaning was performed by removing duplicate records, dropping unnecessary columns, and handling missing values. Missing director values were inferred using recurring director–cast relationships, while country values were populated using director–country associations where possible. Remaining missing values were clearly labeled as “Not Given” to maintain dataset completeness without introducing false assumptions. Records with critical missing fields such as date_added, rating, and duration were removed.

Logical validation checks were carried out to identify inconsistencies, such as cases where the date added to Netflix occurred before the release year. The dataset was further validated by confirming correct data types, ensuring completeness, and sampling records for visual inspection. Temporary columns created during wrangling were removed, and the index was reset.

Finally, the cleaned dataset was exported as a CSV file and the complete Kaggle Notebook was published with public access enabled.

Ensure you allow ‘Anyone with the link’ to be a Viewer

Link to Code: <https://www.kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054>

Conclusion

This assignment provided a comprehensive introduction to practical data wrangling using Python and Pandas. By working with a real-world dataset, I developed a deeper understanding of data quality challenges and the importance of systematic cleaning and validation processes. The skills acquired during this exercise—particularly in handling missing data, enforcing logical consistency, and preparing datasets for analysis—form a strong foundation for future work in data analytics and artificial intelligence. The completed notebook and dataset also contribute to my growing professional portfolio.

Screenshots of the project

The following screenshots are taken from my Kaggle Notebook and document the data wrangling process applied to the Netflix movies dataset. They show key steps such as data cleaning, handling missing values, filtering and transforming variables, and verifying the dataset structure. The accompanying explanations demonstrate my active engagement with the lab tasks and confirm the successful completion of the assignment.

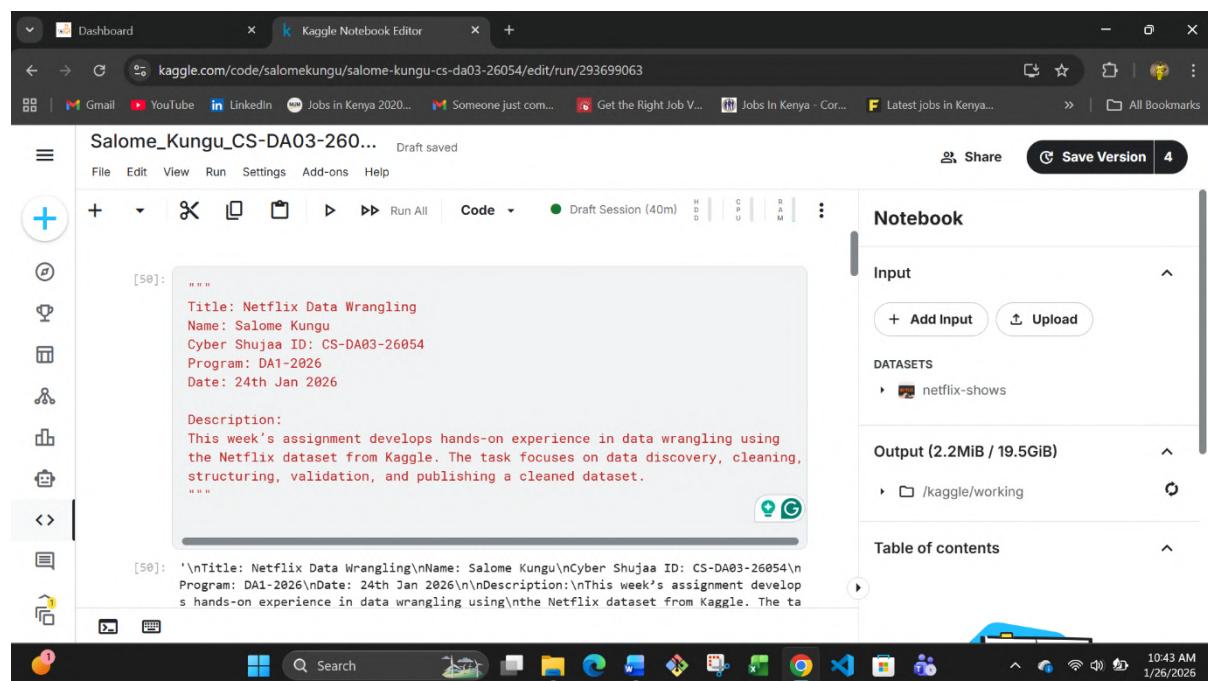
Step 1: Load the Dataset

What:

Import the Netflix CSV file into a Pandas DataFrame using `pd.read_csv()`.

Why:

Loading the data into Pandas enables structured inspection, manipulation, and cleaning using efficient data-handling tools.



The screenshot shows a Kaggle Notebook Editor window titled "Salome_Kungu_CS-DA03-26054". The notebook contains the following code and its output:

```
[50]: """
Title: Netflix Data Wrangling
Name: Salome Kungu
Cyber Shujaa ID: CS-DA03-26054
Program: DA1-2026
Date: 24th Jan 2026

Description:
This week's assignment develops hands-on experience in data wrangling using the Netflix dataset from Kaggle. The task focuses on data discovery, cleaning, structuring, validation, and publishing a cleaned dataset.
"""

[50]: '\nTitle: Netflix Data Wrangling\nName: Salome Kungu\nCyber Shujaa ID: CS-DA03-26054\nProgram: DA1-2026\nDate: 24th Jan 2026\n\nDescription:\nThis week's assignment develops hands-on experience in data wrangling using the Netflix dataset from Kaggle. The ta
```

The notebook interface includes a sidebar with various icons for file operations, a toolbar with buttons for code, run, and settings, and a right panel titled "Notebook" showing sections for Input, Datasets, Output, and Table of contents.

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260... Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (40m) H D C U R A M

```
# Step 1: Load the Dataset

#What:
#Import the Netflix CSV file into a Pandas DataFrame using pd.read_csv().

#Why:
#Loading the data into Pandas enables structured inspection, manipulation, and c

import pandas as pd
import numpy as np
```

+ Code + Markdown

[47]:

Input

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

10:44 AM 1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260... Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (40m) H D C U R A M

```
[47]: #Load the Dataset

#What:
#Import the Netflix CSV file into a Pandas DataFrame using pd.read_csv().

#Why:
#Loading the data into Pandas enables structured inspection, manipulation, and c

import os
os.listdir('/kaggle/input')

!ls /kaggle/input/netflix-shows
```

netflix_titles.csv

Input

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

10:44 AM 1/26/2026

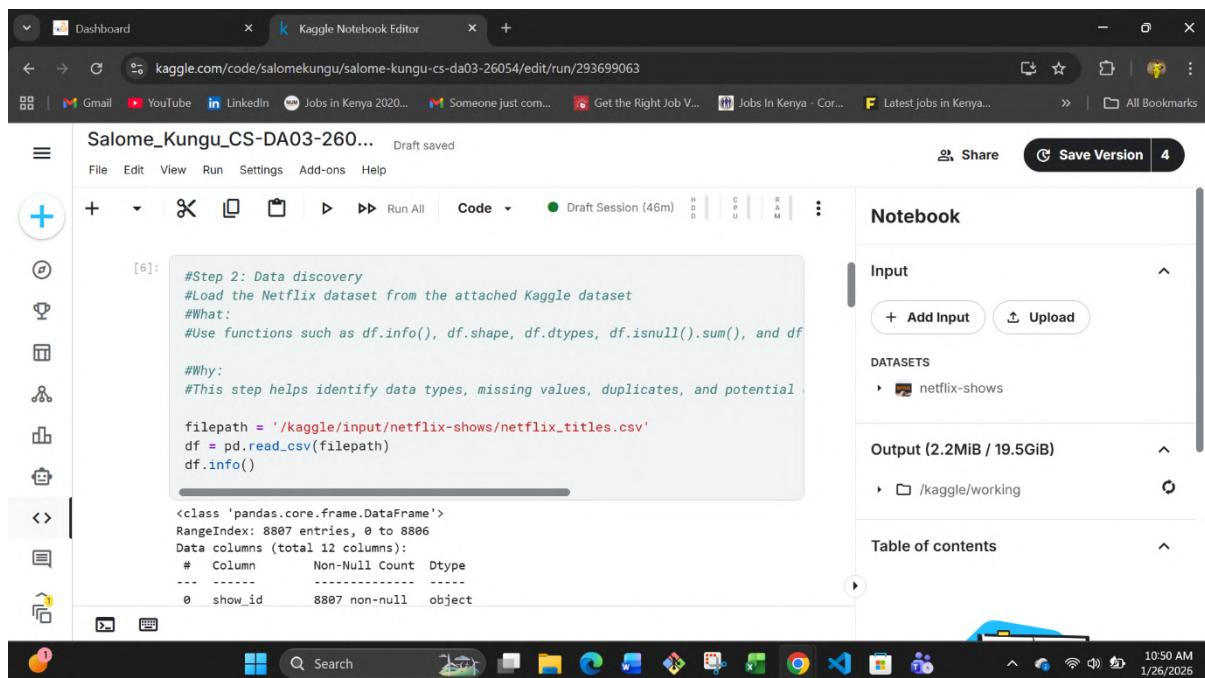
Step 2: Data Discovery

What:

Use functions such as df.info(), df.shape, df.dtypes, df.isnull().sum(), and df.duplicated().sum() to explore the dataset.

Why:

This step helps identify data types, missing values, duplicates, and potential quality issues before any cleaning is performed.



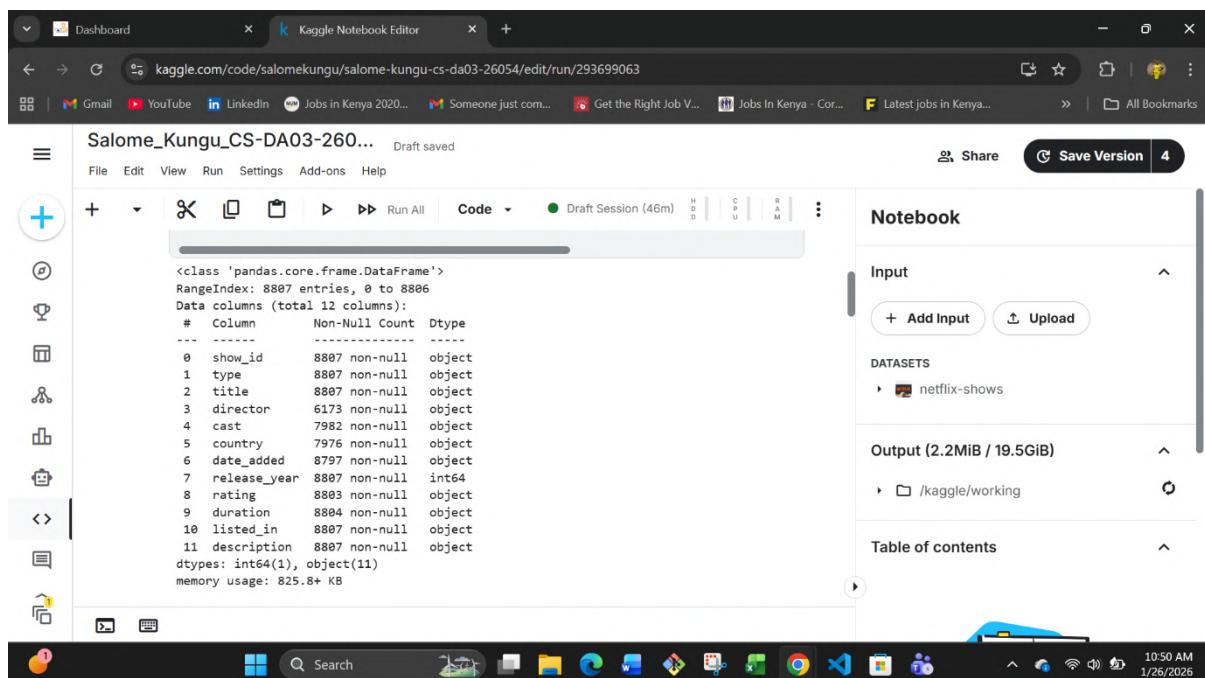
```
[6]: #Step 2: Data discovery
#Load the Netflix dataset from the attached Kaggle dataset
#What:
#Use functions such as df.info(), df.shape, df.dtypes, df.isnull().sum(), and df.duplicated().sum()

#Why:
#This step helps identify data types, missing values, duplicates, and potential quality issues before any cleaning is performed.

filepath = '/kaggle/input/netflix-shows/netflix_titles.csv'
df = pd.read_csv(filepath)
df.info()
```

Output:

```
<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
```



```
[6]: #Step 2: Data discovery
#Load the Netflix dataset from the attached Kaggle dataset
#What:
#Use functions such as df.info(), df.shape, df.dtypes, df.isnull().sum(), and df.duplicated().sum()

#Why:
#This step helps identify data types, missing values, duplicates, and potential quality issues before any cleaning is performed.

filepath = '/kaggle/input/netflix-shows/netflix_titles.csv'
df = pd.read_csv(filepath)
df.info()
```

Output:

```
<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
```

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-26054/edit/run/293699063

Notebook

```
[7]: #Shape of dataset
#What:
#Display the number of rows and columns in the DataFrame using df.shape.

#Why:
#The shape attribute reveals the dataset's dimensions (rows, columns), helping a
print("Dataset Shape (Rows, Columns):", df.shape)

Dataset Shape (Rows, Columns): (8807, 12)
```

```
[8]: # List of all column names
print("Columns in the dataset:\n", df.columns.tolist())

Columns in the dataset:
```

10:50 AM 1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-26054/edit/run/293699063

Notebook

```
[8]: # List of all column names
print("Columns in the dataset:\n", df.columns.tolist())

Columns in the dataset:
['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_ye
ar', 'rating', 'duration', 'listed_in', 'description']
```

```
[9]: # Data types of each column
print("Data types:\n", df.dtypes)

Data types:
show_id          object
type            object
title           object
director        object
```

10:50 AM 1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260... Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (47m) H D C U R A

```
[9]: # Data types of each column
print("Data types:\n", df.dtypes)
```

Data types:

show_id	object
type	object
title	object
director	object
cast	object
country	object
date_added	object
release_year	int64
rating	object
duration	object
listed_in	object
description	object
dtype:	object

Notebook

Input

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

Share Save Version 4

10:51 AM 1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260... Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (47m) H D C U R A

```
[10]: # Missing values per column
print("Missing values per column:\n", df.isnull().sum())
```

Missing values per column:

show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3
listed_in	0
description	0
dtype:	int64

Notebook

Input

+ Add Input Upload

DATASETS

- netflix-shows

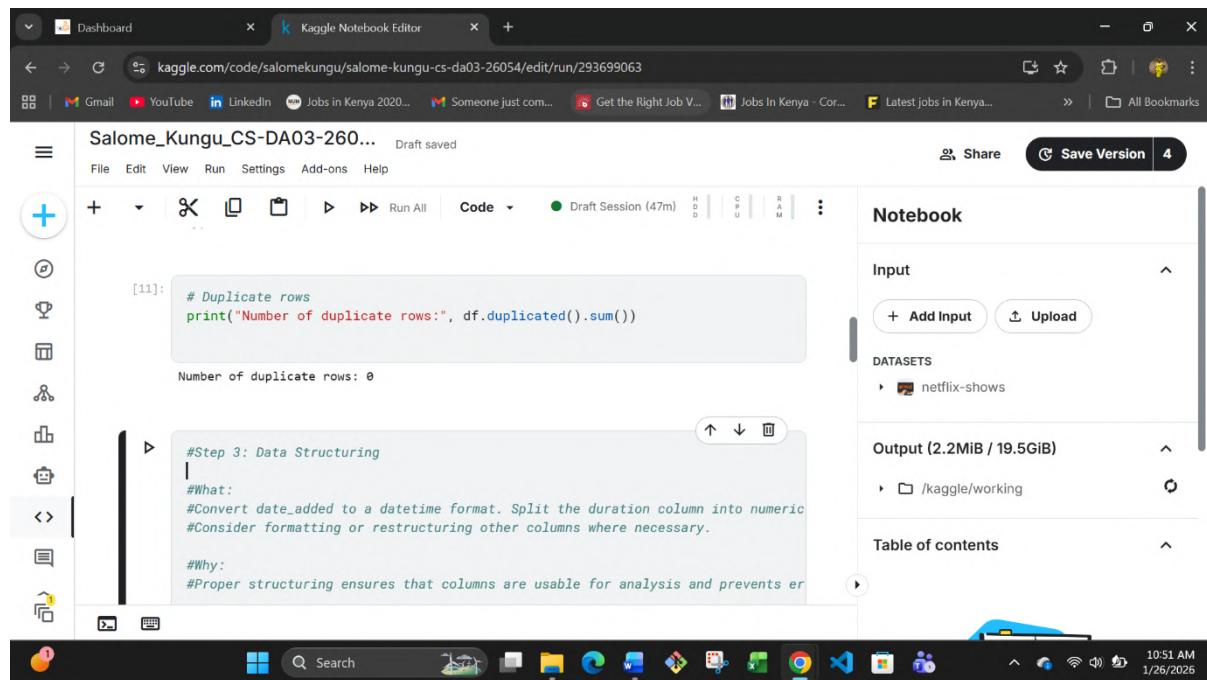
Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

Share Save Version 4

10:51 AM 1/26/2026



The screenshot shows a Kaggle Notebook Editor window. In the code editor, cell [11] contains the following Python code:

```
# Duplicate rows
print("Number of duplicate rows:", df.duplicated().sum())
```

The output of this code is "Number of duplicate rows: 0". To the right of the code editor is the Notebook sidebar, which includes sections for Input, Datasets, Output, and Table of contents.

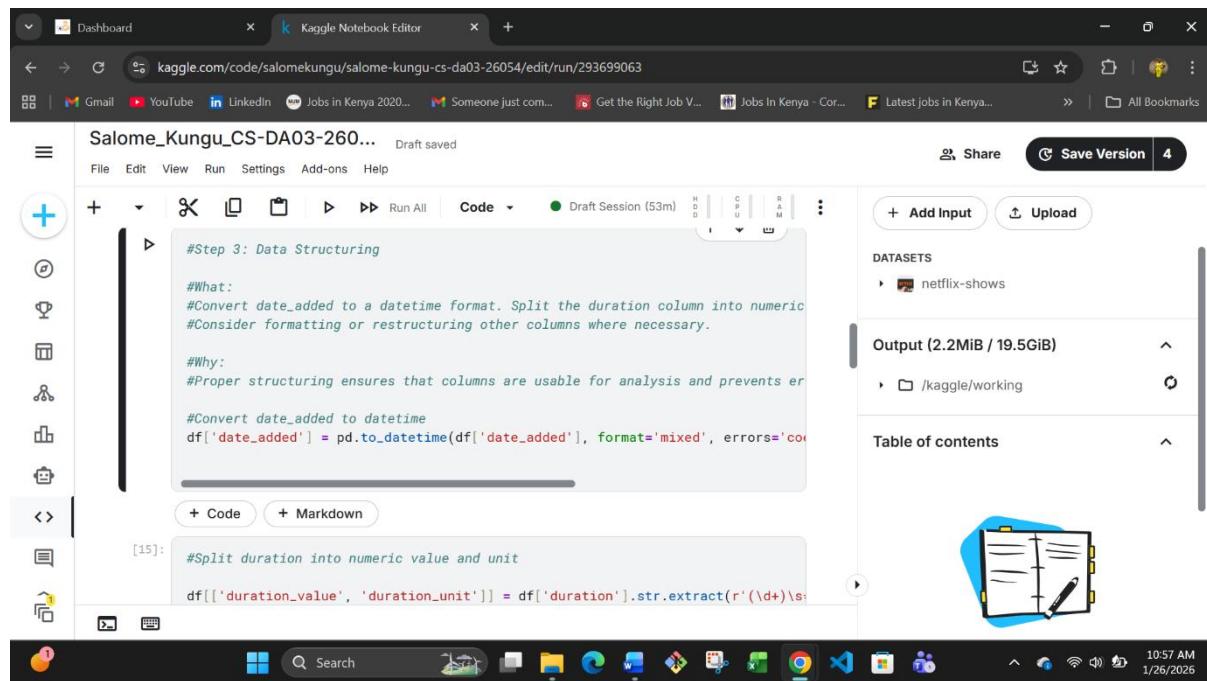
Step 3: Data Structuring

What:

- Convert date_added to a datetime format.
- Split the duration column into numeric and unit components.
- Consider formatting or restructuring other columns where necessary.

Why:

Proper structuring ensures that columns are usable for analysis and prevents errors during filtering, grouping, or validation.



The screenshot shows a Kaggle Notebook Editor window. In the code editor, cell [11] contains the code from the previous screenshot. Below it, cell [15] contains the following Python code:

```
#Step 3: Data Structuring

#What:
#Convert date_added to a datetime format. Split the duration column into numeric
#Consider formatting or restructuring other columns where necessary.

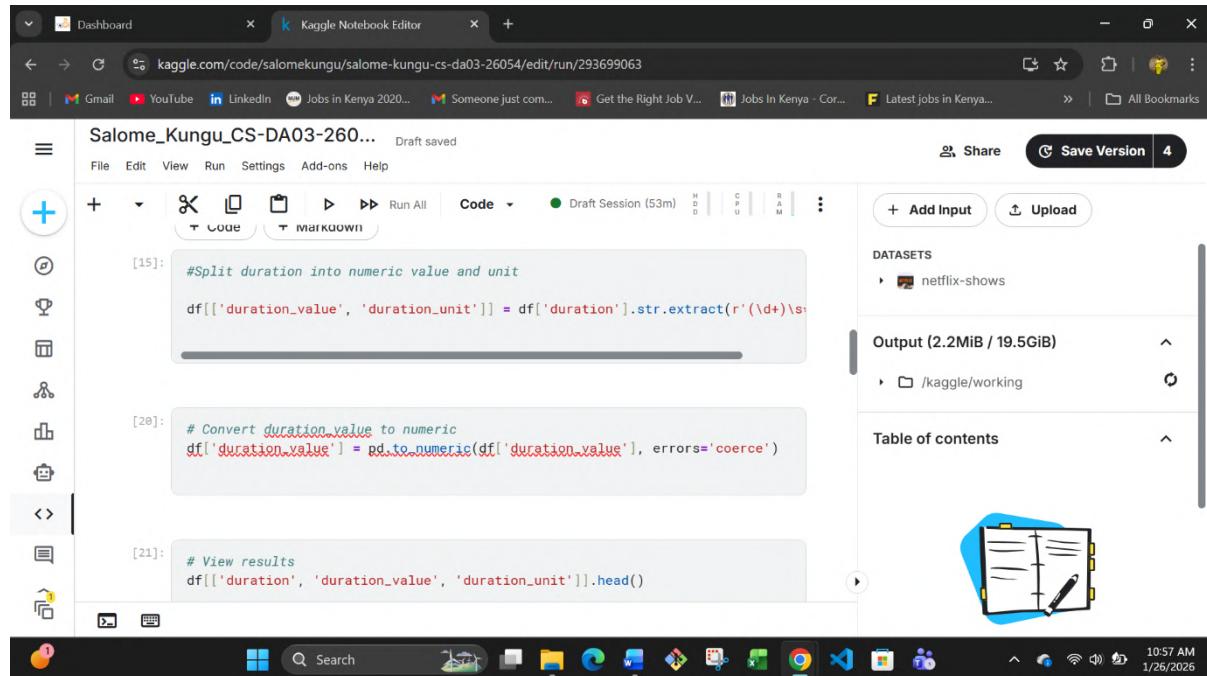
#Why:
#Proper structuring ensures that columns are usable for analysis and prevents errors

#Convert date_added to datetime
df['date_added'] = pd.to_datetime(df['date_added'], format='mixed', errors='coerce')
```

Below this, cell [15] continues with:

```
#Split duration into numeric value and unit
df[['duration_value', 'duration_unit']] = df['duration'].str.extract(r'(\d+)\s:(\w+)')
```

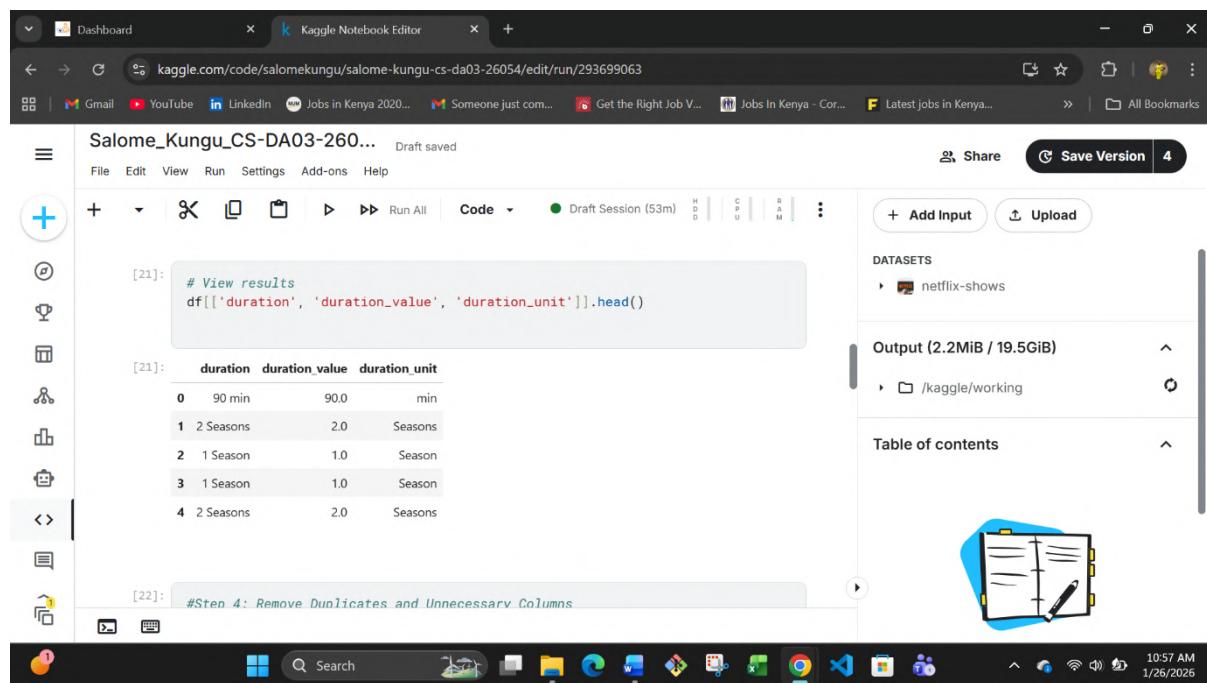
The Notebook sidebar is visible on the right.



```
[15]: #Split duration into numeric value and unit
df[['duration_value', 'duration_unit']] = df['duration'].str.extract(r'(\d+)\s*(\w*)')

[20]: # Convert duration_value to numeric
df['duration_value'] = pd.to_numeric(df['duration_value'], errors='coerce')

[21]: # View results
df[['duration', 'duration_value', 'duration_unit']].head()
```



```
[21]: # View results
df[['duration', 'duration_value', 'duration_unit']].head()
```

	duration	duration_value	duration_unit
0	90 min	90.0	min
1	2 Seasons	2.0	Seasons
2	1 Season	1.0	Season
3	1 Season	1.0	Season
4	2 Seasons	2.0	Seasons

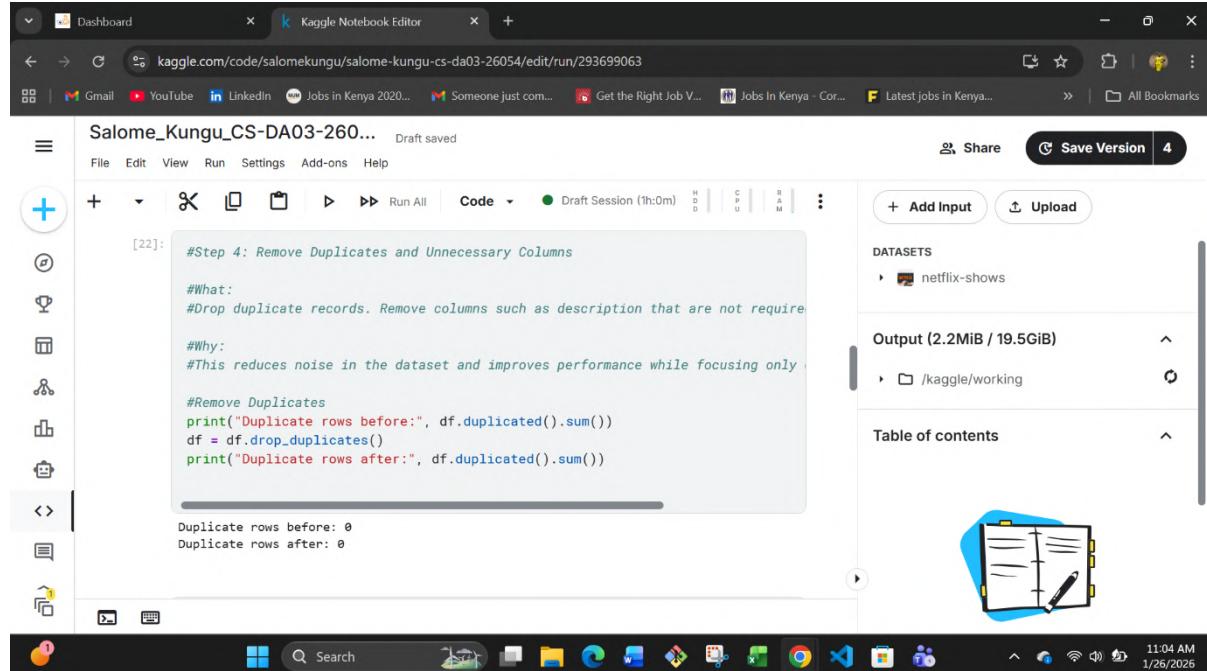
Step 4: Remove Duplicates and Unnecessary Columns

What:

- Drop duplicate records.
- Remove columns such as description that are not required for analysis.

Why:

This reduces noise in the dataset and improves performance while focusing only on relevant attributes.



The screenshot shows a Kaggle Notebook Editor window. The code in cell [22] is as follows:

```
#Step 4: Remove Duplicates and Unnecessary Columns

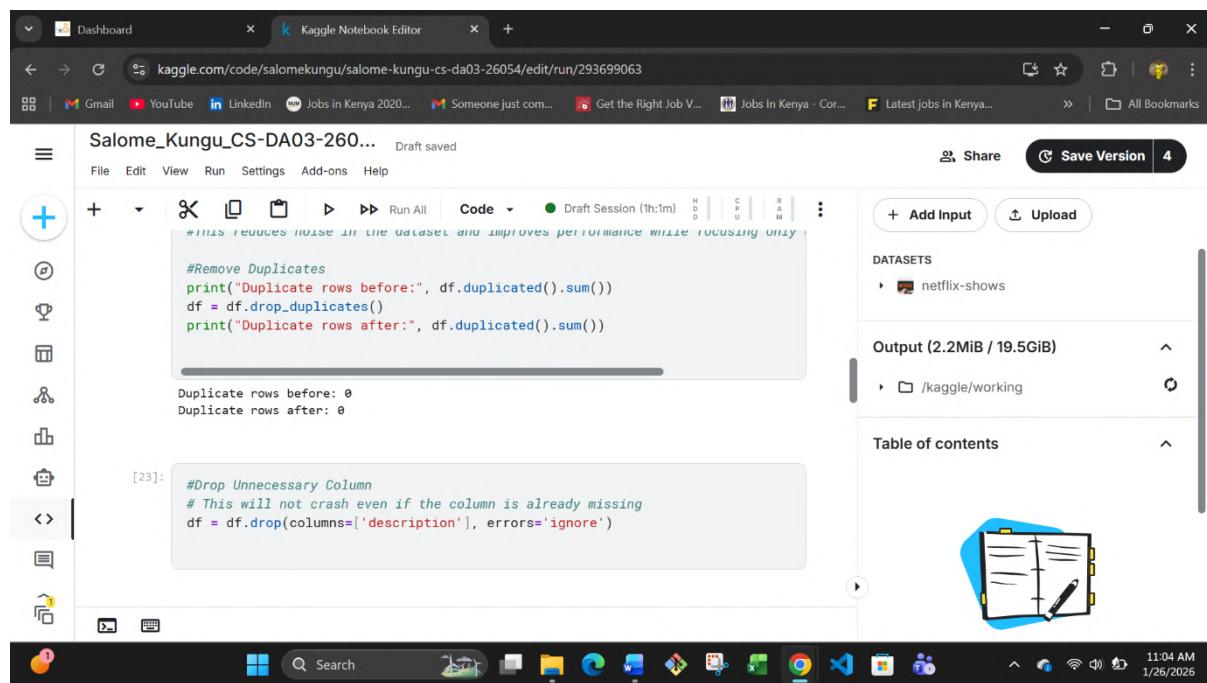
#What:
#Drop duplicate records. Remove columns such as description that are not required

#Why:
#This reduces noise in the dataset and improves performance while focusing only on relevant attributes

#Remove Duplicates
print("Duplicate rows before:", df.duplicated().sum())
df = df.drop_duplicates()
print("Duplicate rows after:", df.duplicated().sum())

Duplicate rows before: 0
Duplicate rows after: 0
```

The notebook also displays a sidebar with datasets and output information, and a Windows taskbar at the bottom.



The screenshot shows a Kaggle Notebook Editor window. The code in cell [22] is identical to the previous one. In cell [23], the following code is added:

```
#Drop Unnecessary Column
# This will not crash even if the column is already missing
df = df.drop(columns=['description'], errors='ignore')
```

The notebook also displays a sidebar with datasets and output information, and a Windows taskbar at the bottom.

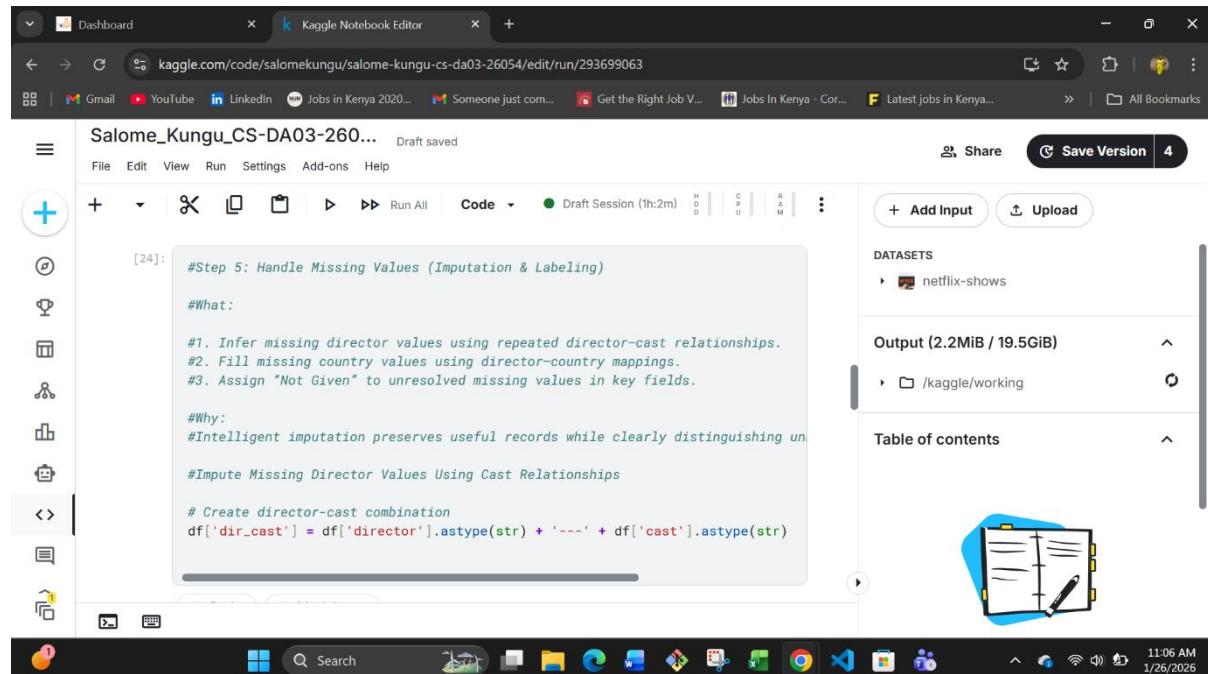
Step 5: Handle Missing Values (Imputation & Labeling)

What:

- Infer missing director values using repeated director–cast relationships.
- Fill missing country values using director–country mappings.
- Assign “Not Given” to unresolved missing values in key fields.

Why:

Intelligent imputation preserves useful records while clearly distinguishing unknown data from valid values.



The screenshot shows the Kaggle Notebook Editor interface. The notebook title is "Salome_Kungu_CS-DA03-260...". The current cell, [24], contains the following code and comments:

```
#Step 5: Handle Missing Values (Imputation & Labeling)

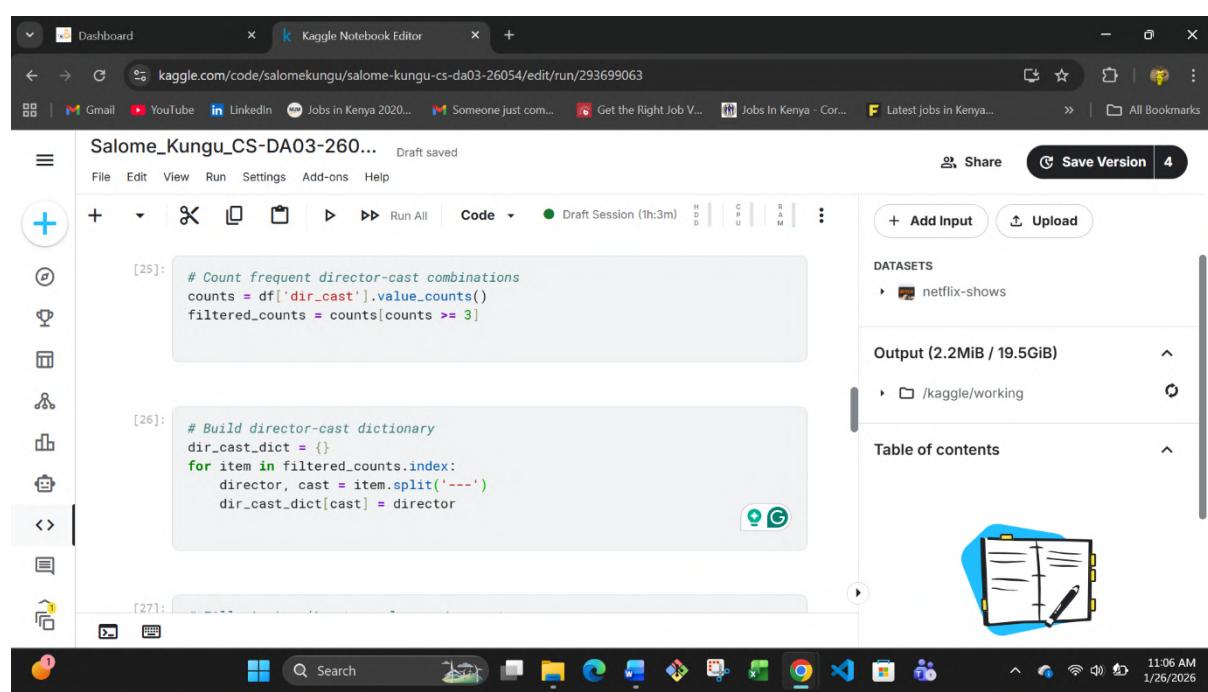
#What:
#1. Infer missing director values using repeated director–cast relationships.
#2. Fill missing country values using director–country mappings.
#3. Assign "Not Given" to unresolved missing values in key fields.

#Why:
#Intelligent imputation preserves useful records while clearly distinguishing un

#Impute Missing Director Values Using Cast Relationships

# Create director–cast combination
df['dir_cast'] = df['director'].astype(str) + '---' + df['cast'].astype(str)
```

The right sidebar shows datasets ("netflix-shows"), output ("2.2MiB / 19.5GiB"), and a table of contents. The status bar at the bottom indicates "11:06 AM 1/26/2026".



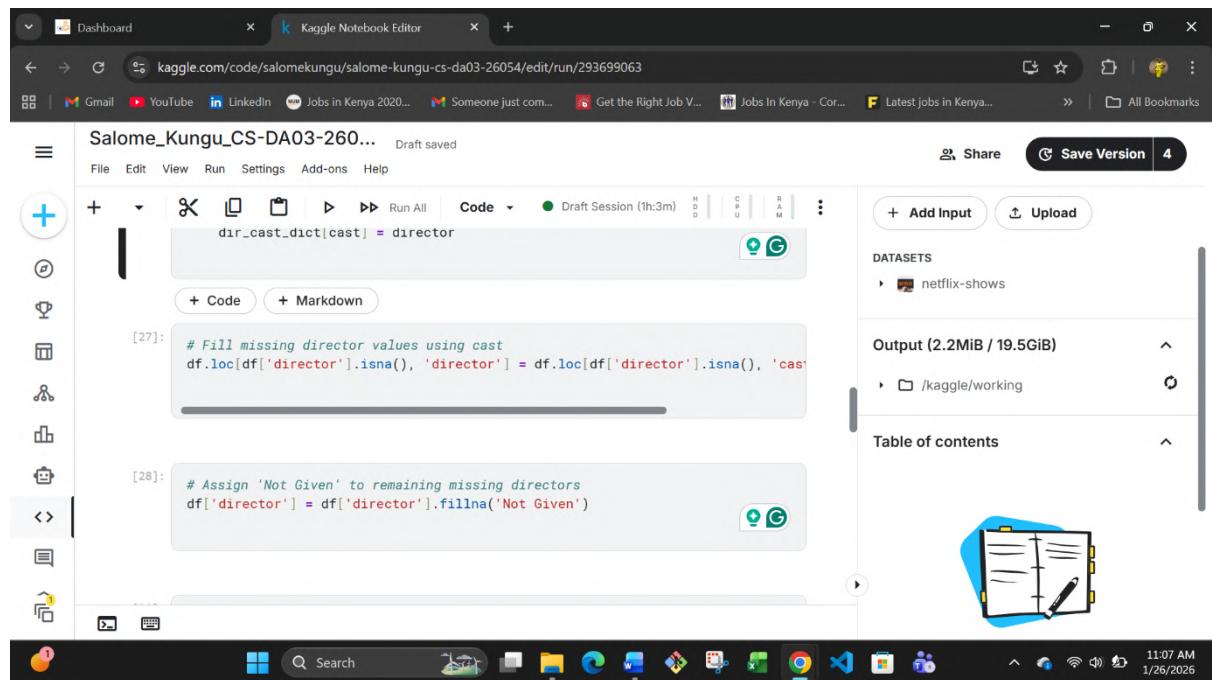
The screenshot shows the Kaggle Notebook Editor interface. The notebook title is "Salome_Kungu_CS-DA03-260...". The current cell, [25], contains the following code:

```
# Count frequent director–cast combinations
counts = df['dir_cast'].value_counts()
filtered_counts = counts[counts >= 3]
```

The next cell, [26], contains the following code:

```
# Build director–cast dictionary
dir_cast_dict = {}
for item in filtered_counts.index:
    director, cast = item.split('---')
    dir_cast_dict[cast] = director
```

The right sidebar shows datasets ("netflix-shows"), output ("2.2MiB / 19.5GiB"), and a table of contents. The status bar at the bottom indicates "11:06 AM 1/26/2026".



The screenshot shows a Kaggle Notebook Editor window. The notebook title is "Salome_Kungu_CS-DA03-260...". The code in the editor is:

```

dir_cast_dict[cast] = director

[27]: # Fill missing director values using cast
df.loc[df['director'].isna(), 'director'] = df.loc[df['director'].isna(), 'cas']

[28]: # Assign 'Not Given' to remaining missing directors
df['director'] = df['director'].fillna('Not Given')

```

The right sidebar shows datasets and output files.

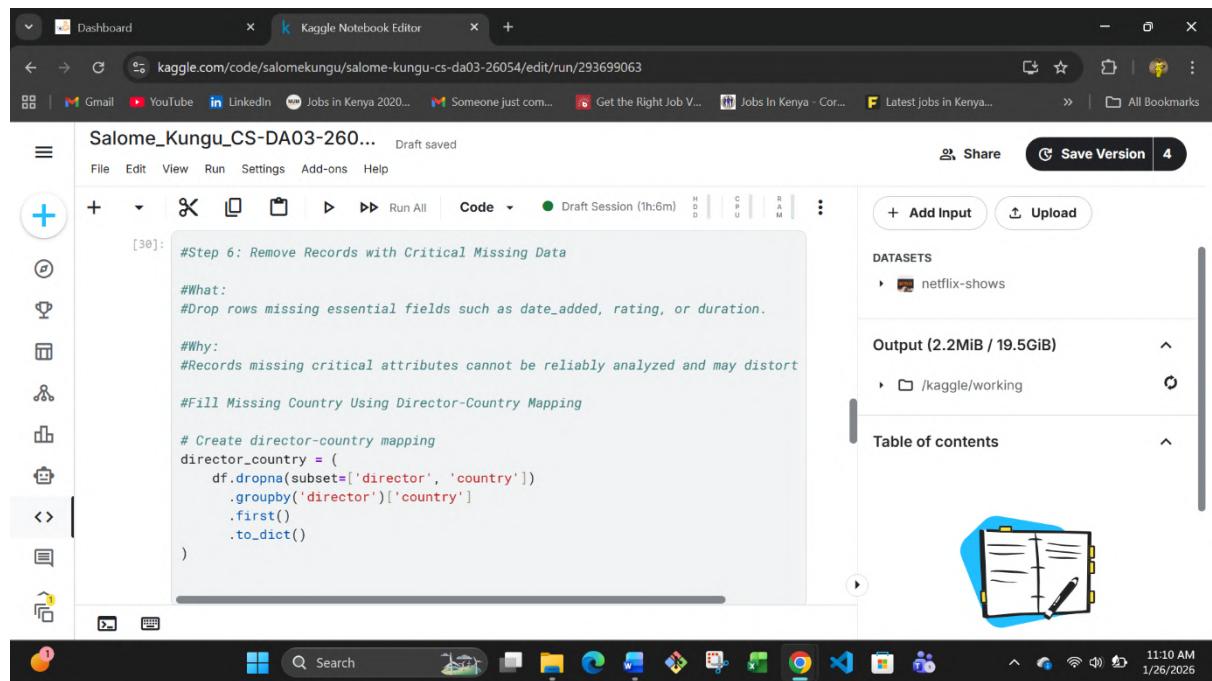
Step 6: Remove Records with Critical Missing Data

What:

Drop rows missing essential fields such as date_added, rating, or duration.

Why:

Records missing critical attributes cannot be reliably analyzed and may distort insights.



The screenshot shows a Kaggle Notebook Editor window. The notebook title is "Salome_Kungu_CS-DA03-260...". The code in the editor is:

```

#Step 6: Remove Records with Critical Missing Data

#What:
#Drop rows missing essential fields such as date_added, rating, or duration.

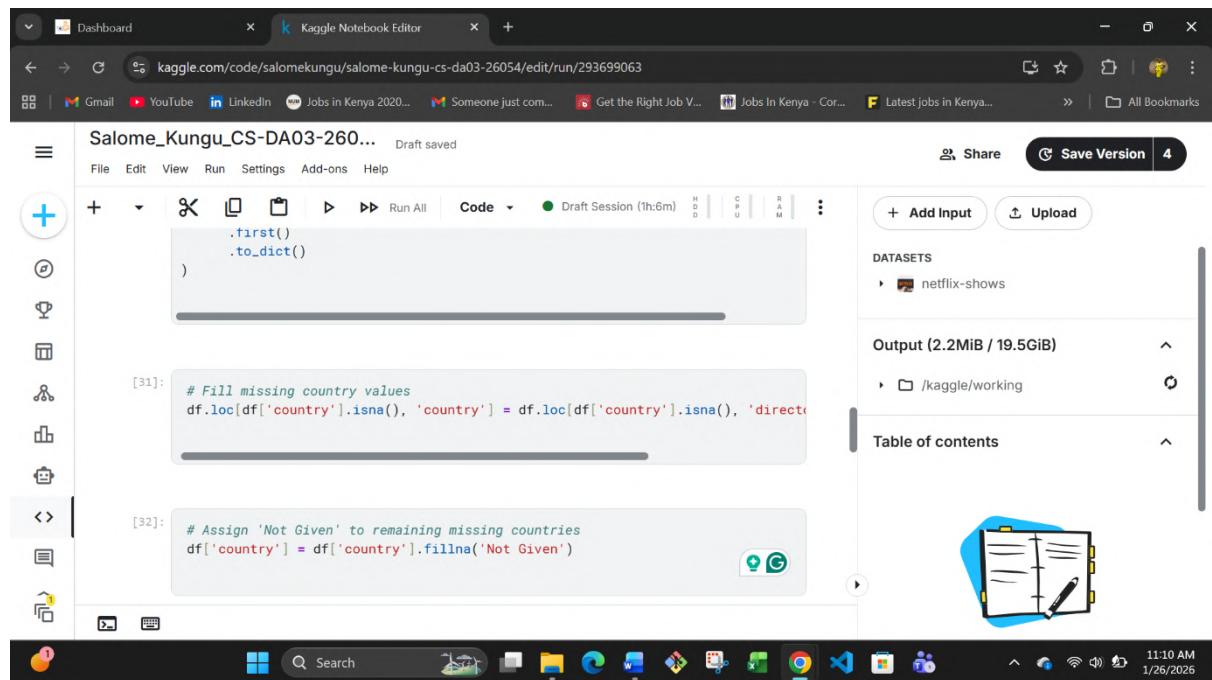
#Why:
#Records missing critical attributes cannot be reliably analyzed and may distort

#Fill Missing Country Using Director-Country Mapping

# Create director-country mapping
director_country = (
    df.dropna(subset=['director', 'country'])
    .groupby('director')['country']
    .first()
    .to_dict()
)

```

The right sidebar shows datasets and output files.



The screenshot shows a Kaggle Notebook Editor window. The code in the editor is:

```

[31]: .first()
       .to_dict()
    )

# Fill missing country values
df.loc[df['country'].isna(), 'country'] = df.loc[df['country'].isna(), 'directo

```

Output (2.2MiB / 19.5GiB) shows a folder named /kaggle/working.

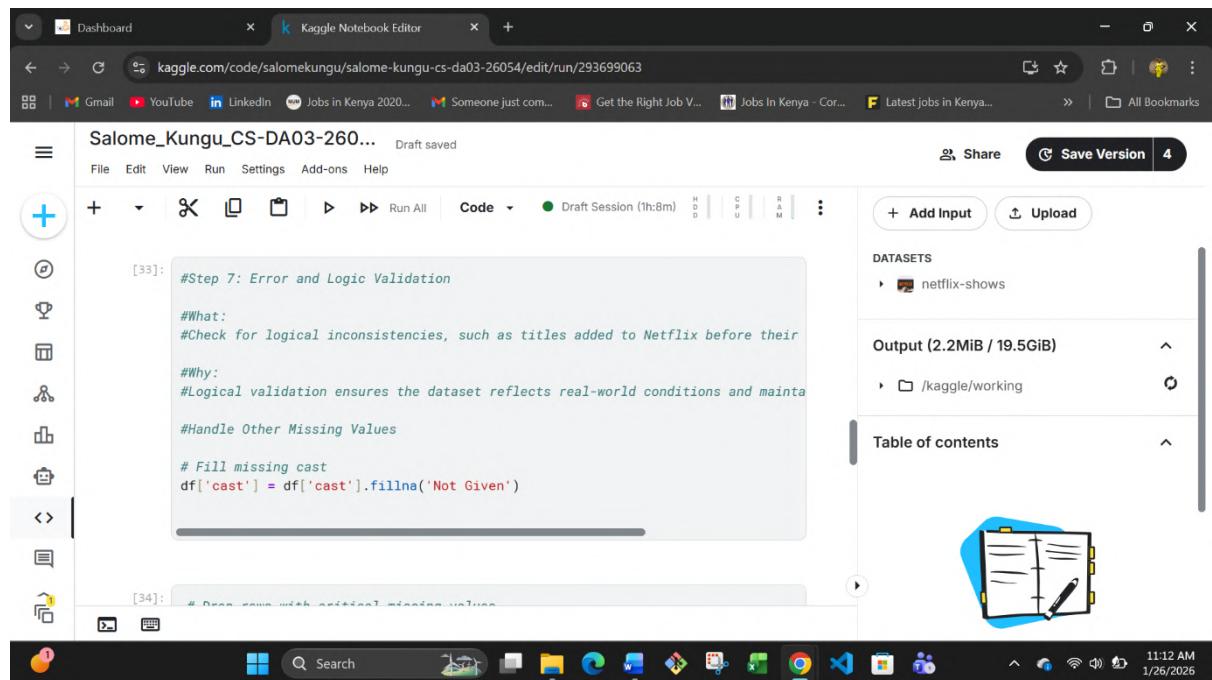
Step 7: Error and Logic Validation

What:

Check for logical inconsistencies, such as titles added to Netflix before their release year.

Why:

Logical validation ensures the dataset reflects real-world conditions and maintains analytical integrity.



The screenshot shows a Kaggle Notebook Editor window. The code in the editor is:

```

[33]: #Step 7: Error and Logic Validation

#What:
#Check for logical inconsistencies, such as titles added to Netflix before their

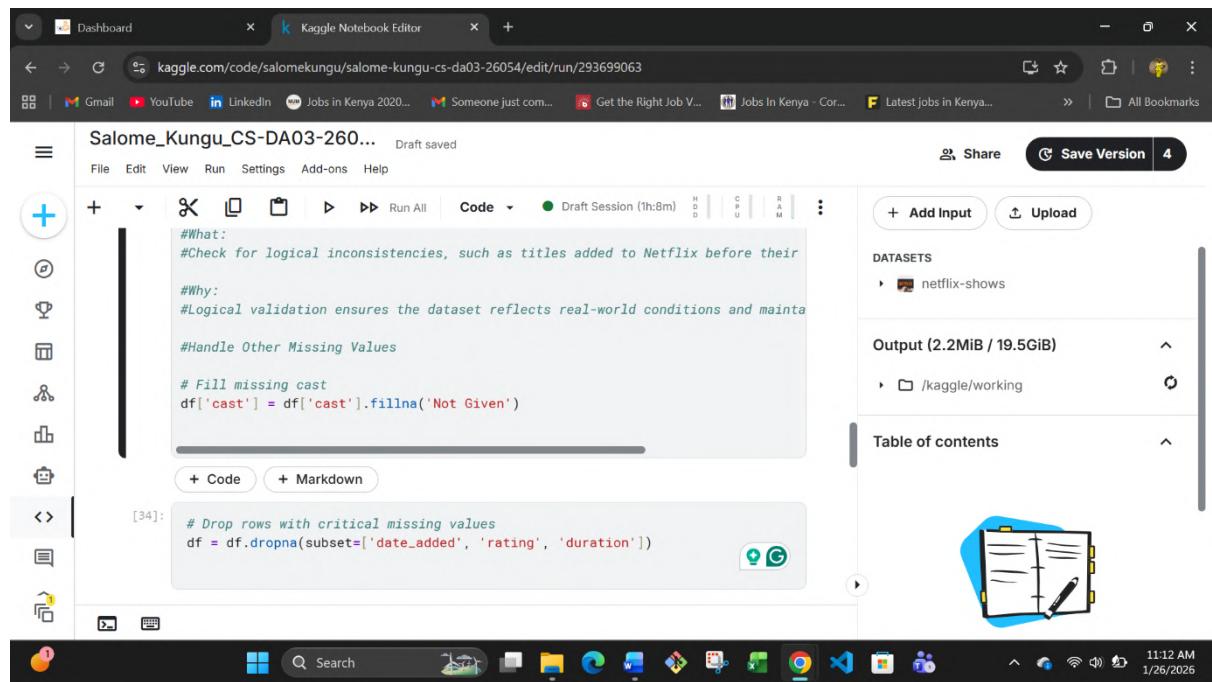
#Why:
#Logical validation ensures the dataset reflects real-world conditions and mainta

#Handle Other Missing Values

# Fill missing cast
df['cast'] = df['cast'].fillna('Not Given')

```

Output (2.2MiB / 19.5GiB) shows a folder named /kaggle/working.



The screenshot shows a Kaggle Notebook Editor window. The code in the main pane is as follows:

```

#What:
#Check for logical inconsistencies, such as titles added to Netflix before their
#Why:
#Logical validation ensures the dataset reflects real-world conditions and maintains
#Handle Other Missing Values

# Fill missing cast
df['cast'] = df['cast'].fillna('Not Given')

[34]: # Drop rows with critical missing values
df = df.dropna(subset=['date_added', 'rating', 'duration'])

```

The notebook has a draft session of 1h:8m. On the right side, there are sections for Datasets (netflix-shows), Output (2.2MiB / 19.5GiB, /kaggle/working), and a Table of contents. A sidebar on the left contains various icons for file operations. The taskbar at the bottom shows several open applications, and the system tray indicates the date and time as 1/26/2026, 11:12 AM.

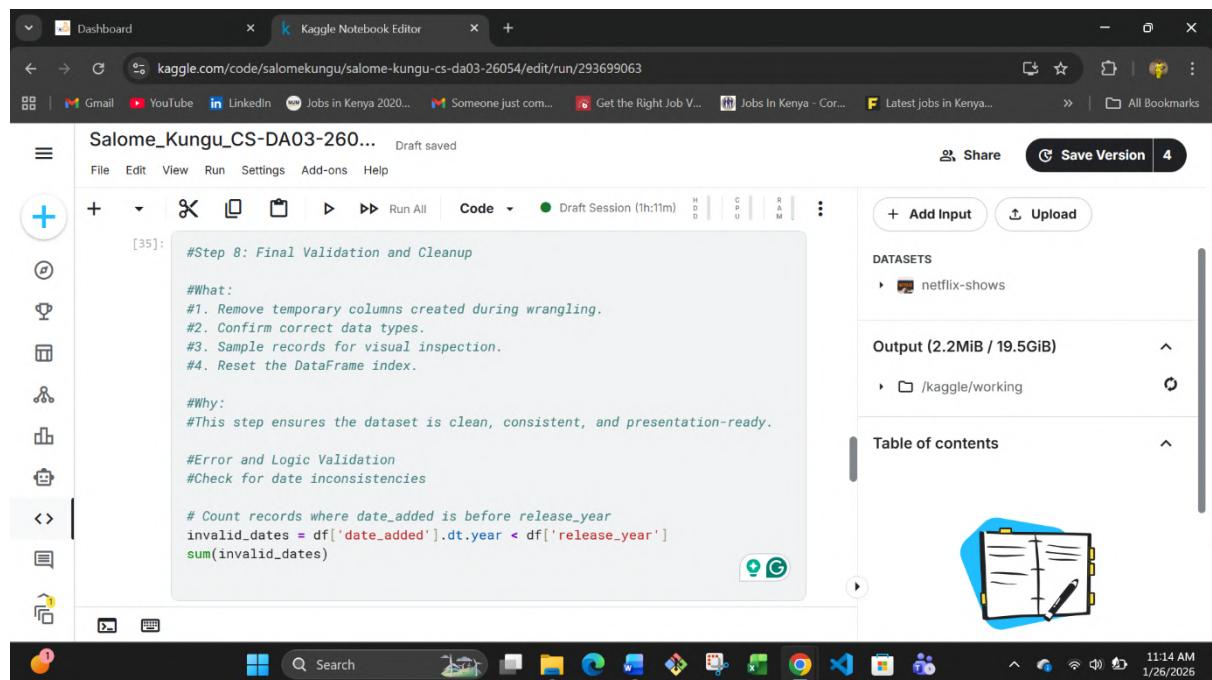
Step 8: Final Validation and Cleanup

What:

- Remove temporary columns created during wrangling.
- Confirm correct data types.
- Sample records for visual inspection.
- Reset the DataFrame index.

Why:

This step ensures the dataset is clean, consistent, and presentation-ready.



The screenshot shows the same Kaggle Notebook Editor window, but now with additional code in the main pane:

```

#Step 8: Final Validation and Cleanup

#What:
#1. Remove temporary columns created during wrangling.
#2. Confirm correct data types.
#3. Sample records for visual inspection.
#4. Reset the DataFrame index.

#Why:
#This step ensures the dataset is clean, consistent, and presentation-ready.

#Error and Logic Validation
#Check for date inconsistencies

# Count records where date_added is before release_year
invalid_dates = df['date_added'].dt.year < df['release_year']
sum(invalid_dates)

```

The notebook now has a draft session of 1h:11m. The rest of the interface remains the same, including the sidebar, datasets section, output section, and system status bar.

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260...

Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (1h:11m) N D C U R A M

```
#Error and Logic Validation
#Check for date inconsistencies

# Count records where date_added is before release_year
invalid_dates = df['date_added'].dt.year < df['release_year']
sum(invalid_dates)
```

[35]: 14

```
[36]: # View inconsistent records
df.loc[invalid_dates, ['title', 'date_added', 'release_year']]
```

[36]:

	title	date_added	release_year
1551	Hilda	2020-12-14	2021
1696	Polly Pocket	2020-11-15	2021
2920	Love Is Blind	2020-02-13	2021
3168	Fuller House	2019-12-06	2020
3287	Maradona in Mexico	2019-11-13	2020
3369	BoJack Horseman	2019-10-25	2020
3433	The Hook Up Plan	2019-10-11	2020
4844	Unbreakable Kimmy Schmidt	2018-05-30	2019
4845	Arrested Development	2018-05-29	2019
5394	Hans Tjeuwen: Real Rancour	2017-07-01	2018

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents



11:15 AM
1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-260...

Draft saved

File Edit View Run Settings Add-ons Help

Code Draft Session (1h:11m) N D C U R A M

```
df.loc[invalid_dates, ['title', 'date_added', 'release_year']]
```

[36]:

	title	date_added	release_year
1551	Hilda	2020-12-14	2021
1696	Polly Pocket	2020-11-15	2021
2920	Love Is Blind	2020-02-13	2021
3168	Fuller House	2019-12-06	2020
3287	Maradona in Mexico	2019-11-13	2020
3369	BoJack Horseman	2019-10-25	2020
3433	The Hook Up Plan	2019-10-11	2020
4844	Unbreakable Kimmy Schmidt	2018-05-30	2019
4845	Arrested Development	2018-05-29	2019
5394	Hans Tjeuwen: Real Rancour	2017-07-01	2018

+ Add Input Upload

DATASETS

- netflix-shows

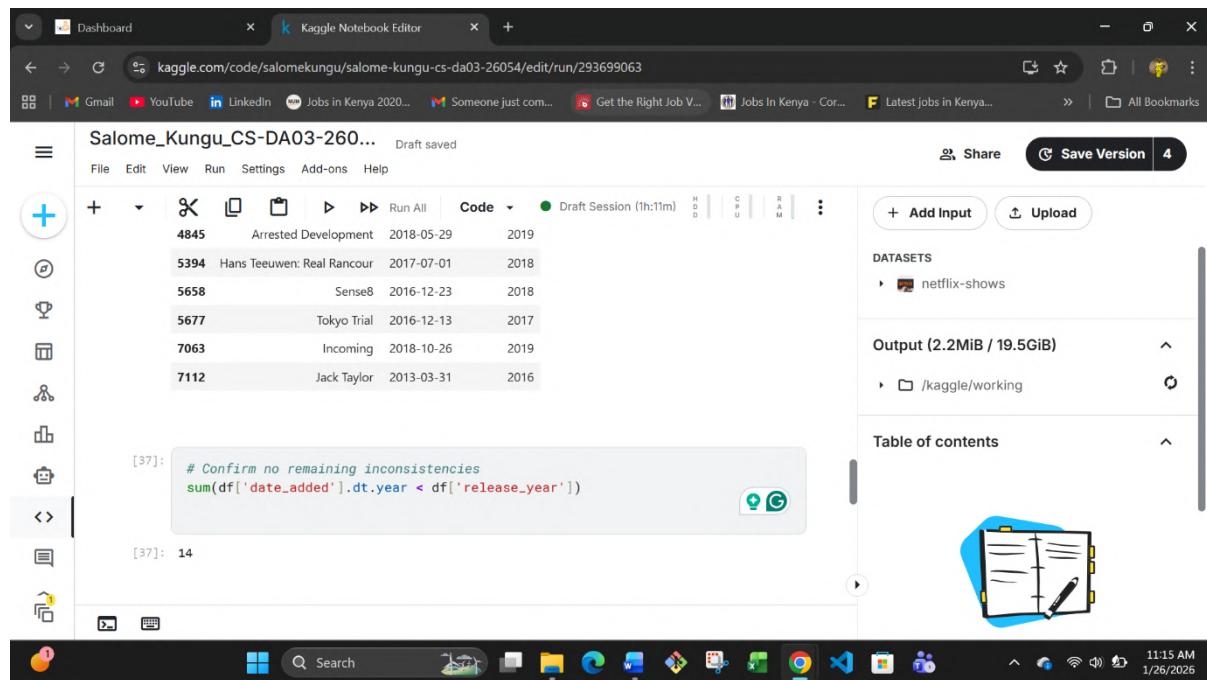
Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents



11:15 AM
1/26/2026



A screenshot of the Kaggle Notebook Editor interface. The notebook is titled "Salome_Kungu_CS-DA03-260...". The code cell at [37] contains the following Python code:

```
# Confirm no remaining inconsistencies
sum(df['date_added'].dt.year < df['release_year'])
```

The cell has a green checkmark icon indicating it has been run successfully. The status bar at the bottom right shows the time as 11:15 AM and the date as 1/26/2026.

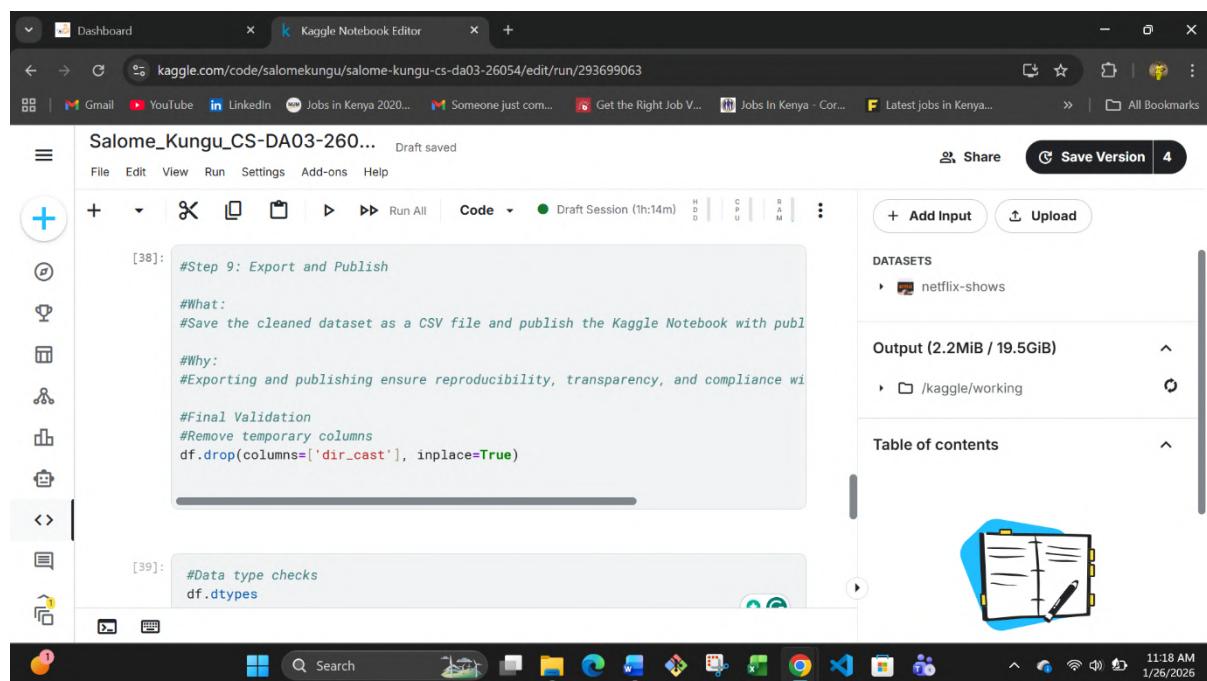
Step 9: Export and Publish

What:

Save the cleaned dataset as a CSV file and publish the Kaggle Notebook with public access enabled.

Why:

Exporting and publishing ensures reproducibility, transparency, and compliance with submission requirements.



A screenshot of the Kaggle Notebook Editor interface. The notebook is titled "Salome_Kungu_CS-DA03-260...". The code cell at [38] contains the following Python code:

```
#Step 9: Export and Publish

##What:
#Save the cleaned dataset as a CSV file and publish the Kaggle Notebook with publ

##Why:
#Exporting and publishing ensure reproducibility, transparency, and compliance wi

##Final Validation
#Remove temporary columns
df.drop(columns=['dir_cast'], inplace=True)
```

The cell has a green checkmark icon indicating it has been run successfully. Below it, another code cell at [39] contains:

```
#Data type checks
df.dtypes
```

The status bar at the bottom right shows the time as 11:18 AM and the date as 1/26/2026.

Dashboard Kaggle Notebook Editor

kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054/edit/run/293699063

File Edit View Run Settings Add-ons Help

Code Draft Session (1h:14m) H CPU RAM

[39]: #Data type checks
df.dtypes

show_id	object
type	object
title	object
director	object
cast	object
country	object
date_added	datetime64[ns]
release_year	int64
rating	object
duration	object
listed_in	object
duration_value	float64
duration_unit	object
dtype:	object

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

11:18 AM 1/26/2026

Dashboard Kaggle Notebook Editor

kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054/edit/run/293699063

File Edit View Run Settings Add-ons Help

Code Draft Session (1h:15m) H CPU RAM

[40]: #Business sanity rule (Netflix founded in 1997)
df[df['release_year'] < 1997]

show_id	type	title	director	cast	country	date_added	release_year	rating	
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA
22	s23	Movie	Avai Shanmughi	K.S. Ravikumar	Kamal Hassan, Meena, Gemini Ganesan,	Not Given	2021-09-21	1996	TV-PG

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

11:18 AM 1/26/2026

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-26054/edit/run/293699063

Draft Session (1h:15m) N D C U R A M

Code

Share Save Version 4

File Edit View Run Settings Add-ons Help

+ Add Input Upload

DATASETS netflix-shows

Output (2.2MiB / 19.5GiB) /kaggle/working

Table of contents

412 rows x 13 columns

11:19 AM 1/26/2026

show_id	type	title	director	cast	country	date_added	release_year	rating
8763	s8764	Movie	Duigan	Downes, Baz Luh...	Australia	2010-11-01	1943	TV-PG
8764	s8765	Movie	WWII: Report from the Aleutians	John Huston	Not Given	United States	2020-01-01	1994 PG-13
8792	s8793	Movie	Wyatt Earp	Lawrence Kasdan	Kevin Costner, Dennis Quaid, Gene Hackman, Dav...	United States	2016-11-01	1973 NR
				Qiu Yuen, Charlie Chin, Jackie Chan, Hu Chin, ...	Hong Kong			

Kaggle Notebook Editor

Salome_Kungu_CS-DA03-26054/edit/run/293699063

Draft Session (1h:15m) N D C U R A M

Code

Share Save Version 4

File Edit View Run Settings Add-ons Help

+ Add Input Upload

DATASETS netflix-shows

Output (2.2MiB / 19.5GiB) /kaggle/working

Table of contents

[41]: #Sample records df.sample(5)

11:19 AM 1/26/2026

show_id	type	title	director	cast	country	date_added	release_year	rating
5800	s5801	TV Show	Kulipari: An Army of Frogs	Not Given	Mark Hamill, Keith David, Wendie Malick, Josh ...	United States	2016-09-02	2016 TV-Y'
3790	s3791	Movie	Mere Pyare Prime Minister	Rakeysh Omprakash Mehra	Anjali Patil, Om Kanjiya, Niteesh Wadhwa, Mak...	India	2019-05-31	2019 TV M
				Captain Lou				

Dashboard Kaggle Notebook Editor

kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054/edit/run/293699063

File Edit View Run Settings Add-ons Help

Share Save Version 4

Salome_Kungu_CS-DA03-260... Draft saved

Code Draft Session (1h:15m) H D C U R M

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

Minister Mehra Wadhwa, Mak...

8523 s8524 TV Show The Super Mario Bros. Super Show! Not Given Captain Lou Albano, Danny Wells, Jeannie Elias...

8604 s8605 TV Show Top 10 Secrets and Mysteries Not Given Not Given United Kingdom, United States, Czech Republic

2659 s2660 TV Show Middleditch & Schwartz Ryan Polito Thomas Middleditch, Ben Schwartz United States 2020-04-21 2020 TV M/

11:19 AM 1/26/2026

Dashboard Kaggle Notebook Editor

kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054/edit/run/293699063

File Edit View Run Settings Add-ons Help

Share Save Version 4

Salome_Kungu_CS-DA03-260... Draft saved

Code Draft Session (1h:16m) H D C U R M

+ Add Input Upload

DATASETS

- netflix-shows

Output (2.2MiB / 19.5GiB)

- /kaggle/working

Table of contents

2659 s2660 TV Show Middleditch & Schwartz Ryan Polito Middleditch, Ben Schwartz United States 2020-04-21 2020 TV M/

[43]: #Reset index
df = df.reset_index(drop=True)

[48]: #Publish (Export Cleaned Dataset)
df.to_csv('/kaggle/working/cleaned.netflix.csv', index=False)

11:20 AM 1/26/2026

Kaggle Notebook Editor

kaggle.com/code/salomekungu/salome-kungu-cs-da03-26054/edit/run/293699063

Dashboard | Kaggle Notebook Editor | +

File Edit View Run Settings Add-ons Help

Share Save Version 4

[48]: #Publish (Export Cleaned Dataset)
df.to_csv('/kaggle/working/cleaned_netflix.csv', index=False)

[49]: #verify the export explicitly to see if the operation succeeds.
import os
os.listdir('/kaggle/working')

[49]: ['.virtual_documents', 'cleaned_netflix.csv']

DATASETS
netflix-shows

Output (2.2MiB / 19.5GiB)
/kaggle/working

Table of contents

11:20 AM
1/26/2026