

Data and Artificial Intelligence

Cyber Shujaa Program

Week 2 Assignment

Data Wrangling

Student Name: Austin Githinji

Student ID: CS-EH03-25417

Introduction

The purpose of this assignment was to gain hands-on experience in data wrangling by using the Netflix dataset from Kaggle. Data wrangling is an essential skill in data science that involves cleaning, structuring, and enriching raw data into a format suitable for analysis and visualization. This project involved working with the Netflix Movies and TV Shows dataset by Shivam Bansal on Kaggle. The dataset includes metadata about Netflix titles such as type, director, cast, country, release year, duration, and genres.

Tasks Completed

1. Loading and Exploring the Dataset

I loaded the dataset using pandas and checked its structure, shape, and missing values.

Code:

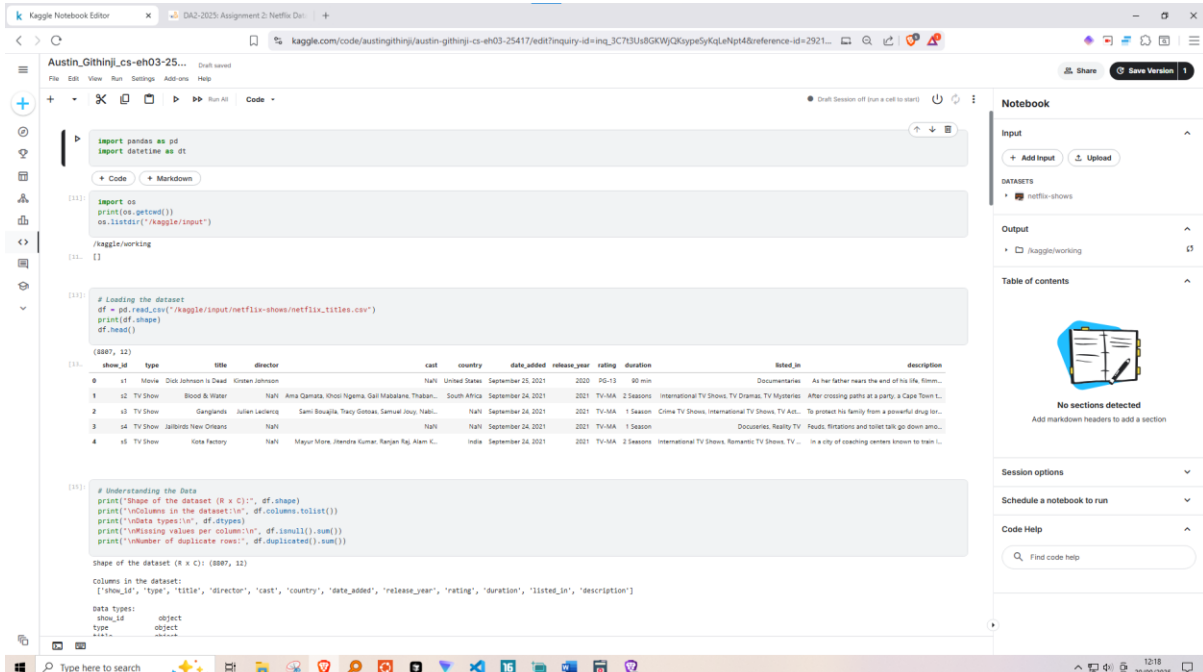
```
import pandas as pd
import datetime as dt

# Load dataset
df = pd.read_csv("/kaggle/input/netflix-shows/netflix_titles.csv")

# Explore structure
print("Shape of the dataset (R x C):", df.shape)
print("\nColumns in the dataset:\n", df.columns.tolist())
print("\nData types:\n", df.dtypes)
```

```
print("\nMissing values per column:\n", df.isnull().sum())
```

```
print("\nNumber of duplicate rows:", df.duplicated().sum())
```



The screenshot shows a Kaggle Notebook titled "Austin_Githini_cs-eh03-25...". The code cell contains the following Python code:

```
import pandas as pd
import datetime as dt

[ ]:
# Code
# Markdown

[ ]:
import os
print(os.getcwd())
os.listdir("/kaggle/input")

/kaggle/working
[ ]:

[ ]:
# Loading the dataset
df = pd.read_csv("/kaggle/input/netflix-shows/netflix_titles.csv")
print(df.shape)
df.head()
```

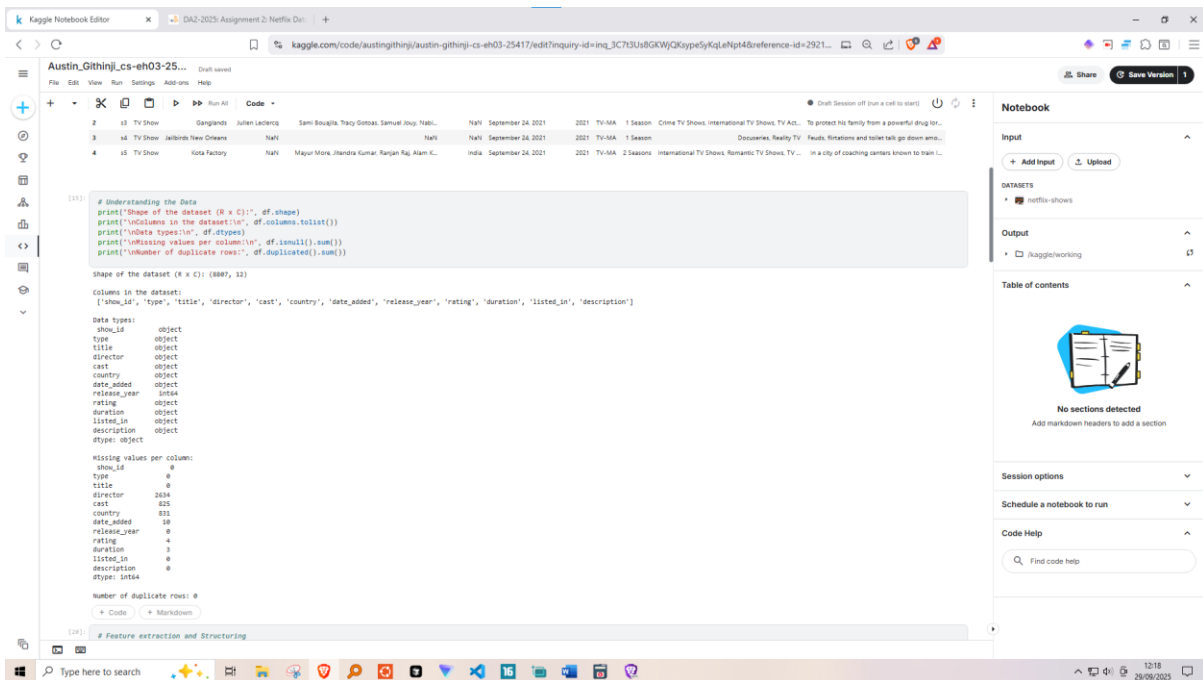
The output shows the shape of the dataset as (8807, 12) and a preview of the first few rows of the dataset:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
1	Movie	Dick Johnson Is Dead	Glenn Johnson	NaN	United States	September 23, 2021	2020	PG-13	80 min	Documentaries	As her father nears the end of his life, Ellen...
2	TV Show	Blood & Water	NaN	Anna Opemka, Khosi Ngema, Gail Mabatane, Thabani...	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...
3	TV Show	Ganglands	Julien Lacrocq	Sami Bouagila, Tracy Gotsas, Samuel Jouy, Nabil...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug bar...
4	TV Show	Jallibre New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Documentaries, Reality TV	Fights, flirtations and toilet talk go down amo...
5	TV Show	Kota Factory	NaN	Major Moore, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train L...

The code cell also includes a section for understanding the data:

```
[ ]:
# Understanding the Data
print("Shape of the dataset (R x C):", df.shape)
print("\nColumns in the dataset:\n", df.columns.tolist())
print("\nData types:\n", df.dtypes)
print("\nMissing values per column:\n", df.isnull().sum())
print("\nNumber of duplicate rows:", df.duplicated().sum())
```

The output shows the shape of the dataset as (8807, 12) and the columns in the dataset: ['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_year', 'rating', 'duration', 'listed_in', 'description']. The data types are: 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. The missing values per column are: show_id: 0, type: 0, title: 0, director: 2634, cast: 625, country: 631, date_added: 10, release_year: 0, rating: 4, duration: 3, listed_in: 0, description: 0. The number of duplicate rows is 0.



The screenshot shows the same Kaggle Notebook, but with the code cell updated to include the following Python code:

```
[ ]:
# Understanding the Data
print("Shape of the dataset (R x C):", df.shape)
print("\nColumns in the dataset:\n", df.columns.tolist())
print("\nData types:\n", df.dtypes)
print("\nMissing values per column:\n", df.isnull().sum())
print("\nNumber of duplicate rows:", df.duplicated().sum())
```

The output shows the shape of the dataset as (8807, 12) and the columns in the dataset: ['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_year', 'rating', 'duration', 'listed_in', 'description']. The data types are: 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. The missing values per column are: show_id: 0, type: 0, title: 0, director: 2634, cast: 625, country: 631, date_added: 10, release_year: 0, rating: 4, duration: 3, listed_in: 0, description: 0. The number of duplicate rows is 0.

The code cell also includes a section for feature extraction and structuring:

```
[ ]:
# Feature extraction and Structuring
```

2. Data Discovery

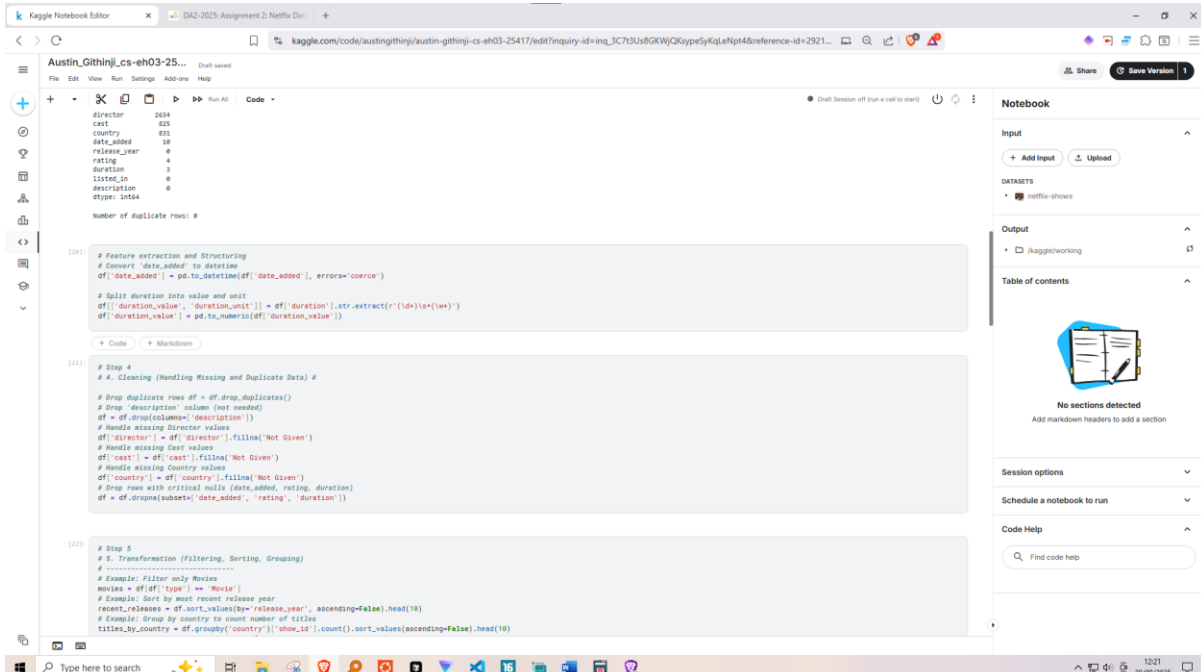
I checked data types, identified missing values, and looked for duplicates and inconsistencies.

Code:

```
df.info()
```

`df.isnull().sum()`

`df.duplicated().sum()`



```

# Feature extraction and Structuring
# Convert 'date_added' to datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

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

# Step 4
# 4. Cleaning (Handling Missing and Duplicate Data) #
# Drop duplicate rows df = df.drop_duplicates()
# Drop 'description' column (not needed)
df = df.drop(columns='description')
# Handle missing Director values
df['director'] = df['director'].fillna('Not Given')
# Handle missing Cast values
df['cast'] = df['cast'].fillna('Not Given')
# Handle missing Country values
df['country'] = df['country'].fillna('Not Given')
# Drop rows with critical nulls (date_added, rating, duration)
df = df.dropna(subset=['date_added', 'rating', 'duration'])

# Step 5
# 5. Transformation (Filtering, Sorting, Grouping)
# -----
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
  
```

3. Structuring the Data

I converted `date_added` to datetime format and extracted numeric values and units from the duration column.

Code:

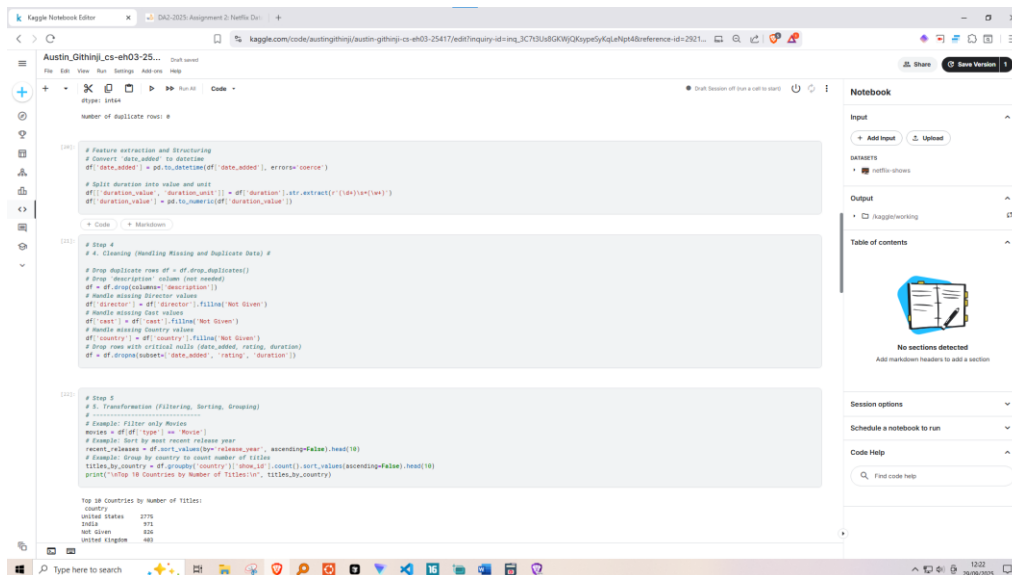
Convert 'date_added' to datetime

```
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
```

Split duration into value and unit

```
df[['duration_value', 'duration_unit']] = df['duration'].str.extract(r'(\d+)\s*(\w+)')
```

```
df['duration_value'] = pd.to_numeric(df['duration_value'])
```



4. Cleaning the Data

I removed duplicate rows, dropped unnecessary columns, and handled missing values.

Code:

Drop duplicates

```
df = df.drop_duplicates()
```

Drop 'description' column

```
df = df.drop(columns=['description'])
```

Fill missing values

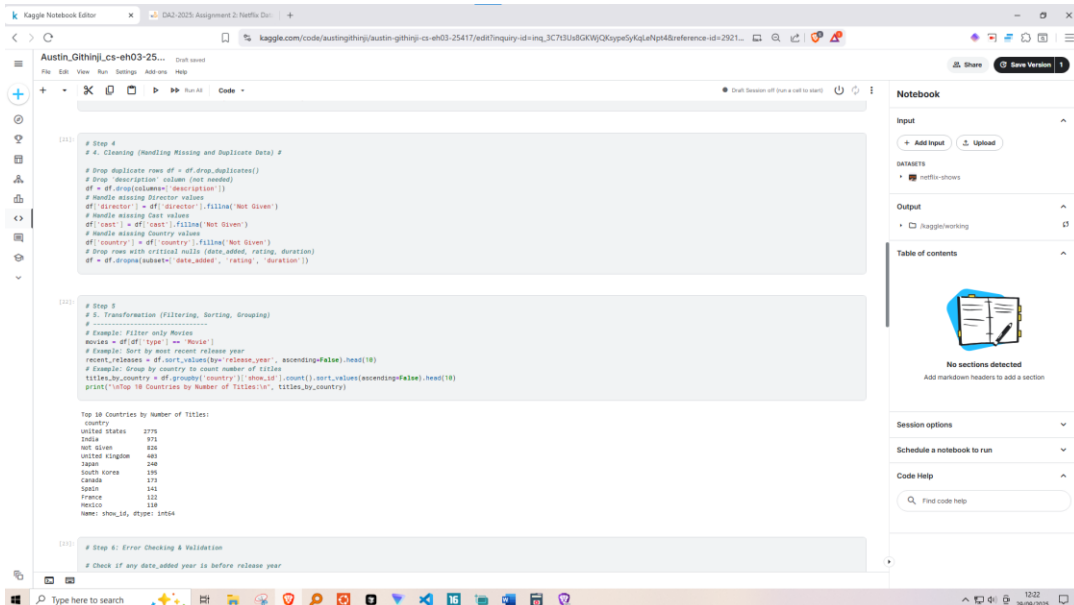
```
df['director'] = df['director'].fillna('Not Given')
```

```
df['cast'] = df['cast'].fillna('Not Given')
```

```
df['country'] = df['country'].fillna('Not Given')
```

Drop rows with critical nulls

```
df = df.dropna(subset=['date_added', 'rating', 'duration'])
```



```

# Step 4: Cleaning (Handling Missing and Duplicate Data) #
# Drop duplicate rows df = df.drop_duplicates()
# Drop 'description' column (not needed)
df = df.drop(columns='description')
# Handle missing Director values
df['director'] = df['director'].fillna('Not Given')
# Handle missing Cast values
df['cast'] = df['cast'].fillna('Not Given')
# Handle missing Country values
df['country'] = df['country'].fillna('Not Given')
# Drop rows with critical nulls (date_added, rating, duration)
df = df.dropna(subset=['date_added', 'rating', 'duration'])

# Step 5: Transformation (Filtering, Sorting, Grouping)
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
print("\nTop 10 Countries by Number of Titles:\n", titles_by_country)

# Step 6: Error Checking & Validation
# Check if any date_added year is before release year

```

5. Transformation and Enrichment

I applied filtering, sorting, and grouping to gain insights.

Code:

Filter only Movies

```
movies = df[df['type'] == 'Movie']
```

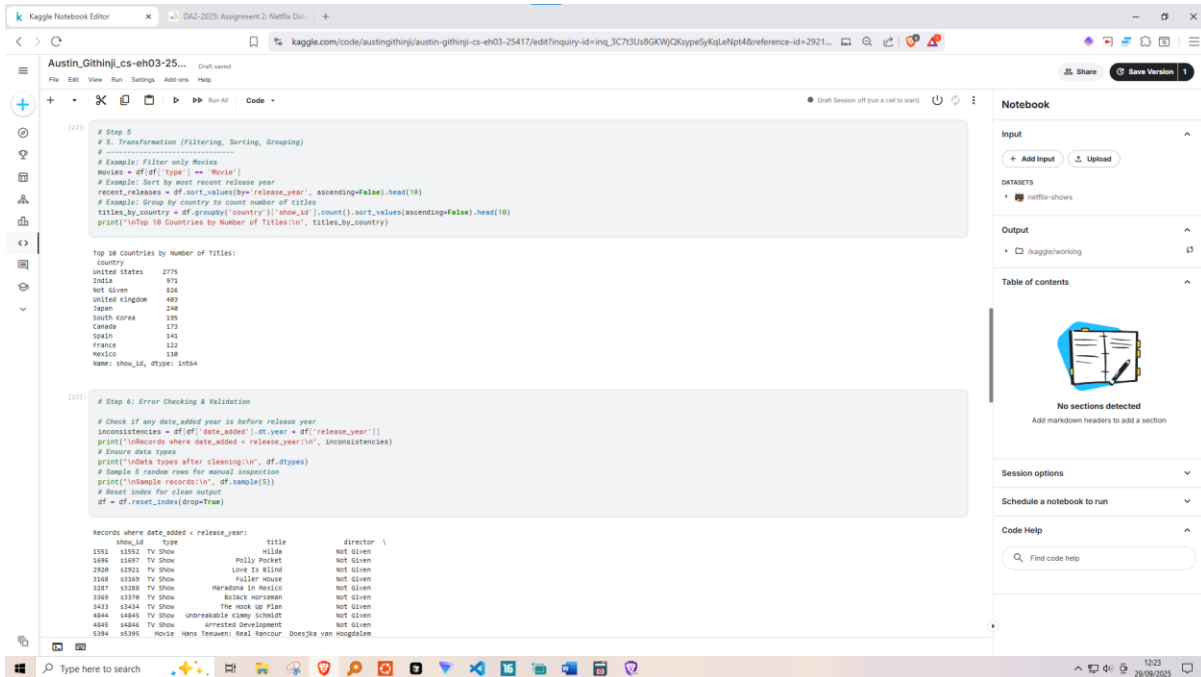
Sort by most recent release year

```
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
```

Group by country and count number of titles

```
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
```

```
print(titles_by_country)
```



The screenshot shows a Kaggle Notebook with the following content:

```
# Step 5
# 5. Transformation (Filtering, Sorting, Grouping)
# -----
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
print("\nTop 10 Countries by Number of Titles:\n", titles_by_country)
```

Top 10 Countries by Number of Titles:

country	count
United States	2775
India	971
Not given	826
United Kingdom	483
Japan	246
South Korea	195
Canada	173
Spain	141
France	122
Mexico	118

Name: show_id, dtype: int64

```
# Step 6: Error Checking & Validation
# Check if any date_added year is before release year
inconsistencies = df[df['date_added'].dt.year < df['release_year']]
print("\nRecords where date_added < release_year:\n", inconsistencies)
# Ensure data types
print("\nData types after cleaning:\n", df.dtypes)
# Sample 3 random rows for manual inspection
print("\nSample records:\n", df.sample(3))
# Reset index for clean output
df = df.reset_index(drop=True)
```

Records where date_added < release_year:

show_id	type	title	director
1551	TV Show	Willie	Not given
1696	TV Show	Polly Pocket	Not given
2328	TV Show	Love Is Blind	Not given
3348	TV Show	Puller House	Not given
3287	TV Show	Maradona in Mexico	Not given
3369	TV Show	Bohica Harrison	Not given
3433	TV Show	The Hook Up Plan	Not given
4844	TV Show	Unbreakable Jimmy Schmitt	Not given
4846	TV Show	Arrested Development	Not given
5394	Movie	Hans Tesaari: Real Rancour	Doesjka van Hoogdalen

6. Validation

I checked for inconsistencies, confirmed data types, and sampled random rows.

Code:

Check logical consistency

```
inconsistencies = df[df['date_added'].dt.year < df['release_year']]
```

```
print(inconsistencies)
```

Check data types

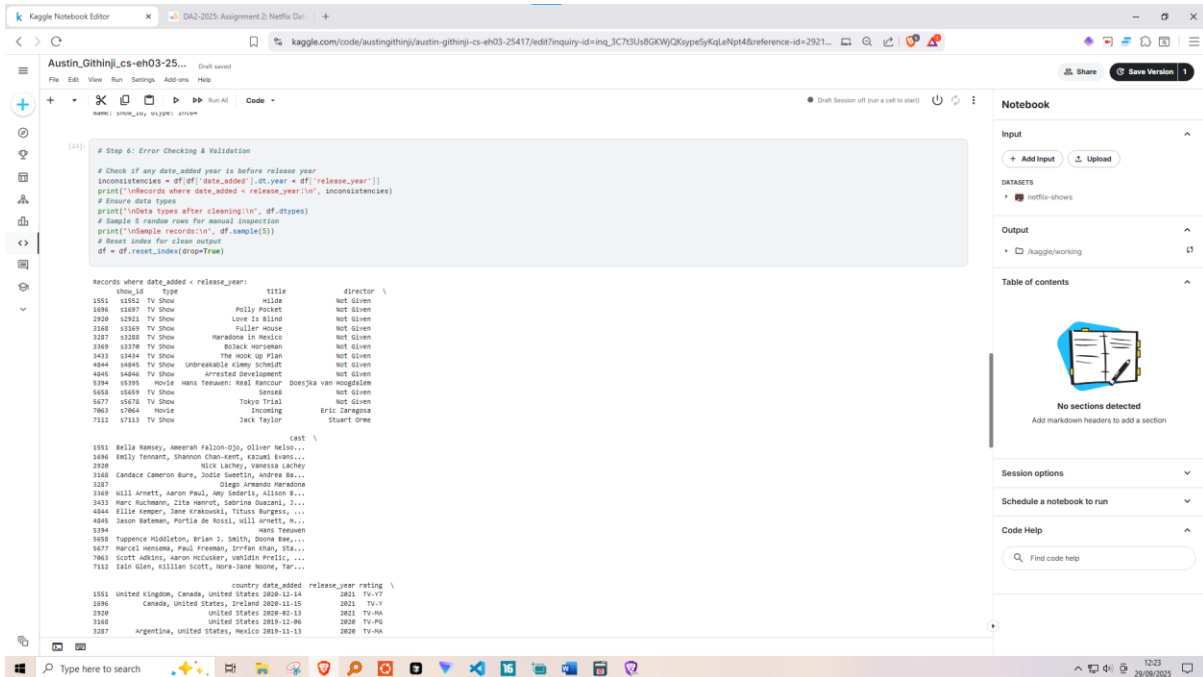
```
print(df.dtypes)
```

Sample 5 random rows

```
print(df.sample(5))
```

Reset index

```
df = df.reset_index(drop=True)
```



The screenshot shows a Kaggle Notebook titled "Austin_Githini_cs-eh03-25...". The code in the cell [241] performs the following steps:

- Step 6: Error Checking & Validation**
- Check if any `date_added` year is before `release_year`.
- Print records where `date_added < release_year`.
- Ensure data types.
- Print data types after cleaning.
- Sample 5 random rows for manual inspection.
- Reset index for clean output.

The output shows two tables:

show_id	Type	title	director
1551	TV Show	Willie	Not Given
1596	TV Show	Polly Pocket	Not Given
2928	TV Show	Love Is Blind	Not Given
3168	TV Show	Fuller House	Not Given
3287	TV Show	Maradona in Mexico	Not Given
3369	TV Show	Boback Horsman	Not Given
3433	TV Show	The Hook Up Plan	Not Given
4844	TV Show	Unbreakable Kimmy Schmidt	Not Given
4846	TV Show	Arrested Development	Not Given
5394	Movie	Hans Feuwerkerl: Real Rancour	Doesjka van Hoogdalen
5658	TV Show	Sensad	Not Given
5677	TV Show	Tokyo Trial	Not Given
7863	Movie	Incoming	Eric Zaragoza
7812	TV Show	Jack Taylor	Stuart Omer

show_id	country	date_added	release_year	rating
1551	United Kingdom, Canada	United States 2020-11-14	2021	TV-17
1596	Canada, United States, Ireland	2020-11-15	2021	TV-14
2928	United States	2020-02-13	2021	TV-MA
3168	United States	2019-12-06	2020	TV-PG
3287	Argentina, United States, Mexico	2019-11-13	2020	TV-MA

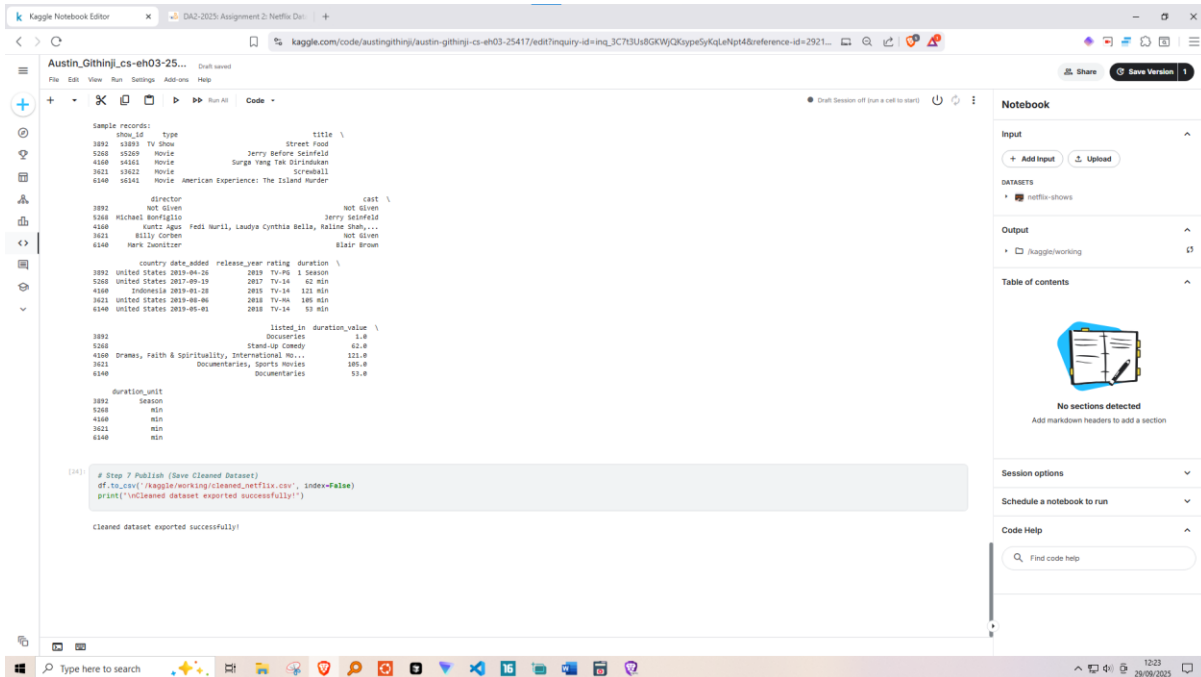
7. Exporting the Dataset

Finally, I exported the cleaned dataset as a CSV.

Code:

```
df.to_csv('/kaggle/working/cleaned_netflix.csv', index=False)
```

```
print("Cleaned dataset exported successfully!")
```



```

sample records:
  show_id  type  title
3892  33893  TV Show  Street Food
5268  35269  Movie  Jerry Before Seinfeld
4368  34561  Movie  Surge Yang Tak Sirindaman
3621  33622  Movie  Screemall
6148  35342  Movie  American Experience: The Island Hunter

director
3892  not given
5268  Michael Bonfiglio
4368  Kurtz Agos, Rodi Huril, Laudye Cynthia Bella, Baline Shuh,...
3621  Billy Corben
6148  Mark Zandiger

cast
3892  not given
5268  Jerry Seinfeld
4368  Jerry Seinfeld
3621  not given
6148  Blair Brown

country,date_added,release_year,rating,duration
3892  United States 2019-04-26  2019  TV-14  1 Season
5268  United States 2017-09-19  2017  TV-14  62 min
4368  Indonesia 2019-01-28  2019  TV-14  123 min
3621  United States 2019-08-06  2018  TV-14  186 min
6148  United States 2019-09-01  2018  TV-14  53 min

listed_in,duration_value
3892  Documentaries  1.0
5268  Stand-up Comedy  62.0
4368  Dramas, Faith & Spirituality, International, M...  123.0
3621  Documentaries, Sports Movies  186.0
6148  Documentaries  53.0

duration_unit
3892  Season
5268  min
4368  min
3621  min
6148  min

[24]: # Step 7 Publish (Save Cleaned Dataset)
df.to_csv('kaggle/working/cleaned_netflix.csv', index=False)
print('UnCleaned dataset exported successfully!')

Cleaned dataset exported successfully!

```

Conclusion

This project covered the full data wrangling workflow: loading, exploring, cleaning, structuring, transforming, validating, and exporting. I learned how to handle missing values, remove duplicates, split and reformat columns, and apply filtering, sorting, and grouping.

Kaggle Notebook Link

<https://www.kaggle.com/code/austingithinji/austin-githinji-cs-eh03-25417>

Conclusion

This week I gained a good grounding on the introductory concepts relating to data science and artificial intelligence. I am getting a better understanding that I can build on as we work on more advanced concepts in later weeks. I have posted my writeup on my blog and I look forward to building a portfolio that I can showcase on my CV as I look for jobs in Data and AI.