

Data and Artificial Intelligence

Cyber Shujaa Program

Week 2 Assignment

Data Wrangling

Student Name: Jasper Ngunjiri Wambugu

Student ID: CS-DA01-25045

Introduction

This week's assignment was to perform data wrangling using python programming language which I was new to. I had to create an account on Kaggle.com and got to join the amazing community. The assignment given was on data wrangling of a Netflix Data set.

The objectives of the assignment were:

- Explain data wrangling concepts and their importance in the data science workflow.
- Load and inspect various datasets using Python.
- Identify and handle missing values using techniques such as dropping, filling, and imputing.
- Detect and correct inconsistencies in data.
- Transform and reshape data using various techniques.
- Apply the entire data wrangling process to a real-world dataset and present a clean, analysis-ready version in a Jupyter notebook

Tasks Completed

Below are all the listed steps and procedures that were to be carried out:

1. **Data Discovery:** This initial step involves understanding the data, its format, and its potential issues. That is, explore the data, identify patterns, trends, and missing or incomplete information.
2. **Data Structuring:** This step focuses on organizing the data into a more usable format. This might involve converting data types, handling missing values, and creating new variables.
3. **Data Cleaning:** This step aims to address data inconsistencies, inaccuracies, and errors. It might involve removing duplicates, handling missing values, and correcting errors.

4. **Data Enrichment:** This step involves adding more context or information to the data. This could include integrating data from external sources or creating new variables based on existing ones.
5. **Data Validation:** This step focuses on ensuring the quality and integrity of the data. It might involve checking for data types, ranges, and other rules to ensure that the data meets the requirements for analysis.
6. **Data Publishing:** The final step involves making the cleaned and validated data available for analysis or other uses.

Link to the Netflix Data Set: [Netflix Data Set](#)

Link to Kaggle Site: [Kaggle Site](#)

link to Kaggle Project Notebook: [Notebook Link](#)

Stater Code



The screenshot shows a Jupyter Notebook titled "netflix-datawrangling-notebookaba848f2...". The notebook is in a "Draft Session (28m)" state. The code in the cell is as follows:

```
[1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

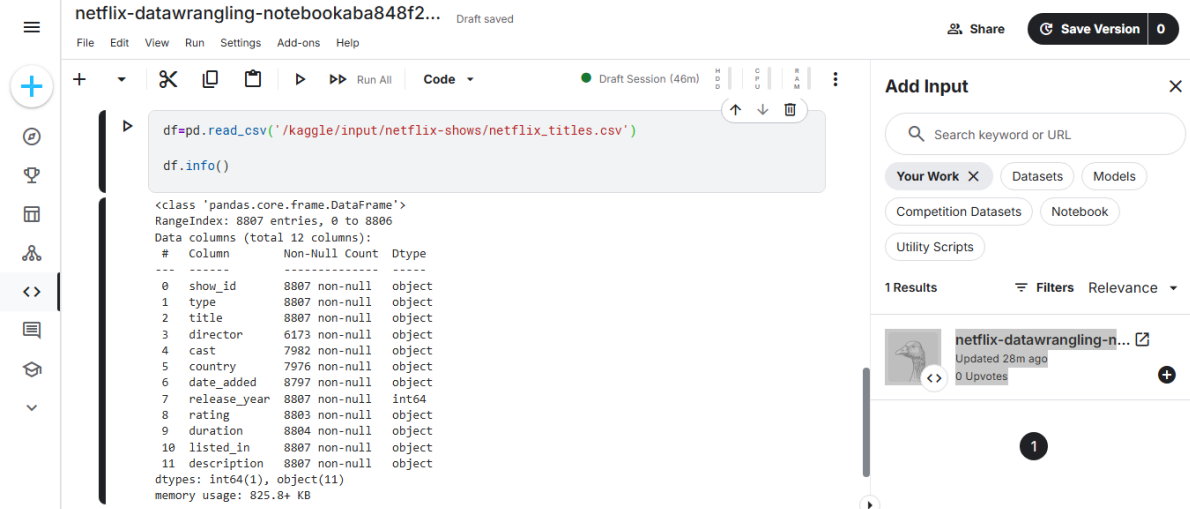
# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the session

/kaggle/input/netflix-shows/netflix_titles.csv
```

The right sidebar shows the "Output (76KIB / 19.5GiB)" section with a file explorer showing "/kaggle/working". Below that is the "Table of contents" section with a list of items: "Data Science Project: Data Wrangling", "STEP 1: DISCOVERY OF DATA", "Number of rows and columns", and "STEP 2: STRUCTURING OF DATA". At the bottom of the sidebar are "Session options", "Schedule a notebook to run", and "Code Help".

1. Data Discovery

Import data to a pandas data frame and have a quick overview of the data. **df.info()**



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

+ Draft Session (46m)

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

Add Input

Search keyword or URL

Your Work X Datasets Models

Competition Datasets Notebook

Utility Scripts

1 Results Filters Relevance

netflix-datawrangling-n... Updated 28m ago 0 Upvotes

Get the number of rows and columns in the dataset provided

Print("Number of Rows and Columns (R x C): \n", df.shape)



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

+ Draft Session (33m)

Number of rows and columns

The total count of rows and columns

```
print("Shape of the dataset (R x C): ", df.shape)
```

+ Code + Markdown

```
[5]: print("Shape of the dataset (R x C):", df.shape)
```

```
Shape of the dataset (R x C): (8807, 12)
```

Display a list of all names

```
print("List all columns in dataset:\n", df.columns.tolist())
```

Output (76KiB / 19.5GiB)

/kaggle/working

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns

STEP 2: STRUCTURING OF DATA

Session options

Schedule a notebook to run

Code Help

Print a list of all columns and data types

Print ("Print a list of all columns:", df.columns.tolist())

Print ("Print the dataframes for the data types", df.dtypes)

netflix-datawrangling-notebookaba848f2...

Draft saved

File Edit View Run Settings Add-ons Help

+

✂

📄

📄

▶

▶▶

Run All

Code

Draft Session (34m)

0

1

2

3

4

5

6

7

8

9

0

.

,

;

:'"

{}

[]

~

⋮

Display a list of all names

```
print("List all columns in dataset:\n", df.columns.tolist())
```

[6]:

```
print("List all columns in dataset:\n", df.columns.tolist())
```

```
List all columns in dataset:
['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_year', 'rating', 'duration', 'listed_in', 'description']
```

+ Code

+ Markdown

Get data types for each columns

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

[9]:

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

Output (76KiB / 19.5GiB)

▶

/kaggle/working

🔗

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns

STEP 2: STRUCTURING OF DATA

Session options

▼

Schedule a notebook to run

▼

Code Help

^

netflix-datawrangling-notebookaba848f2...

Draft saved

File Edit View Run Settings Add-ons Help

+

✂

📄

📄

▶

▶▶

Run All

Code

Draft Session (40m)

0

1

2

3

4

5

6

7

8

9

0

.

,

;

:'"

{}

[]

~

⋮

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

[9]:

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

```
Data types of each column:
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
```

+ Code

+ Markdown

Output (76KiB / 19.5GiB)

▶

/kaggle/working

🔗

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns

STEP 2: STRUCTURING OF DATA

Session options

▼

Schedule a notebook to run

▼

Code Help

^

Print and count missing values in each column

Print ("Print and count missing values in each column: ", df.isnull().sum())

netflix-datawrangling-notebookaba848f2...

Draft saved

File Edit View Run Settings Add-ons Help

+

✂

📄

📄

▶

▶▶

Run All

Code

Draft Session (43m)

0

1

2

3

4

5

6

7

8

9

0

.

,

;

:'"

{}

[]

~

⋮

Group and count Missing values in each column

```
print("Get Missing values for each column: \n ", df.isnull().sum())
```

[10]:

```
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
```

+ Code

+ Markdown

Output (76KiB / 19.5GiB)

▶

/kaggle/working

🔗

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns

STEP 2: STRUCTURING OF DATA

Session options

▼

Schedule a notebook to run

▼

Code Help

^

Group and count duplicated data

Print ("Duplicated data from dataset: ", df.duplicated().sum())



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

cast 825
country 831
date_added 10
release_year 0
rating 4
duration 3
listed_in 0
description 0
dtype: int64

Group and count duplicate rows

```
print("Number of duplicated rows: \n", df.duplicated().sum())
```

[11]: print("Number of duplicated rows: \n", df.duplicated().sum())

Number of duplicated rows:
0

Output (76KiB / 19.5GiB)
/kaggle/working

Table of contents
Data Science Project: Data Wrangling
STEP 1: DISCOVERY OF DATA
Number of rows and columns
STEP 2: STRUCTURING OF DATA

Session options
Schedule a notebook to run
Code Help

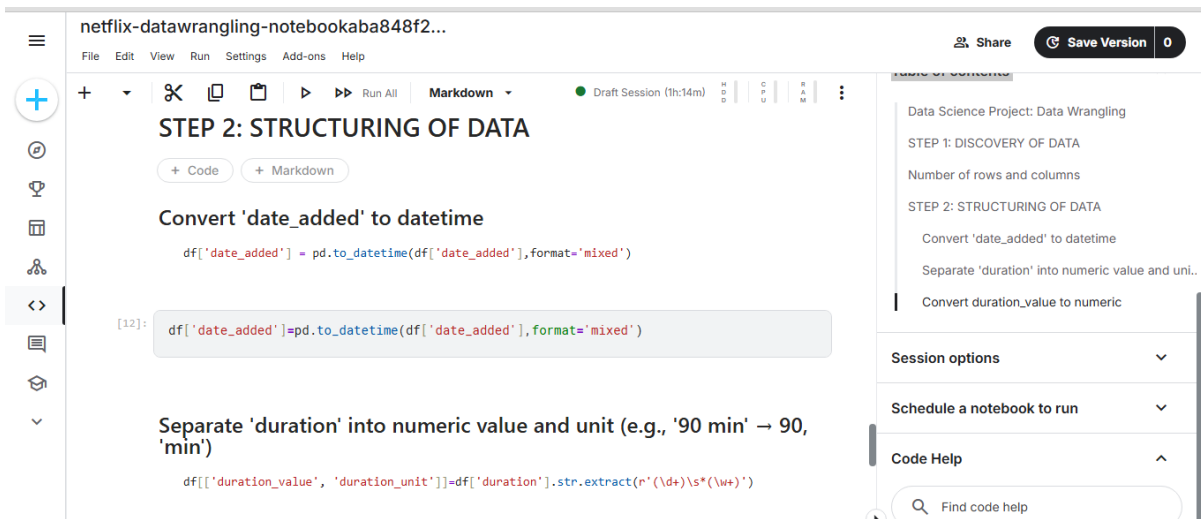
2. Data Structuring

Convert "date_added" to datetime

df["date_added"] = pd.to_datetime(df["date_added"], format='mixed')

Separate 'duration' into numeric value and unit (e.g., '90 min' → 90, 'min')

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



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

STEP 2: STRUCTURING OF DATA

Convert 'date_added' to datetime

```
df['date_added'] = pd.to_datetime(df['date_added'], format='mixed')
```

[12]: df['date_added'] = pd.to_datetime(df['date_added'], format='mixed')

Separate 'duration' into numeric value and unit (e.g., '90 min' → 90, 'min')

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

Output (76KiB / 19.5GiB)
/kaggle/working

Table of contents
Data Science Project: Data Wrangling
STEP 1: DISCOVERY OF DATA
Number of rows and columns
STEP 2: STRUCTURING OF DATA
Convert 'date_added' to datetime
Separate 'duration' into numeric value and uni..
Convert duration_value to numeric

Session options
Schedule a notebook to run
Code Help

Convert duration_value to numeric

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

View Resulting columns

print(df[['duration_value', 'duration_unit']])



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

Convert duration_value to numeric

```
df["duration_value"] = df.to_numeric(df["duration_value"])

[ ]: df["data_value"] = df.to_numeric(df["duration_value"])

## View Resulting columns
python
print(df[["data_value", "data_unit"]])

+ Code + Markdown

[ ]: print(df[["data_value", "data_unit"]])
```

Session options

Schedule a notebook to run

Code Help

Find code help

3. Cleaning Data

Check for duplicated rows

Print("Print duplicated rows before", df.duplicated().sum())



netflix-datawrangling-notebookaba848f2... Draft Session (1h:42m)

File Edit View Run Settings Add-ons Help

STEP 3. CLEANING

Check for duplicate rows

```
print("Duplicated rows before: ", df.duplicated().sum())

Duplicated rows before: 0

+ Code + Markdown
```

Add Input

Search keyword or URL

Your Work X Datasets Models

Competition Datasets Notebook

Utility Scripts

1 Results Filters Relevance

netflix-datawrangling-n... Updated 2h ago 0 Upvotes

Impute Director values by using relationship between cast and director

List of Director-Cast pairs and the number of times they appear

```
df['dir_cast'] = df['director'] + '---' + df['cast']
```

```
counts = df['dir_cast'].value_counts()
```

```
filtered_counts = counts[counts >= 3]
```

```
filtered_values = filtered_counts.index
```

```
lst_dir_cast = list(filtered_values)
```

```
dict_direcast = dict()
```

```
for i in lst_dir_cast :
```

```
director,cast = i.split('---')
```

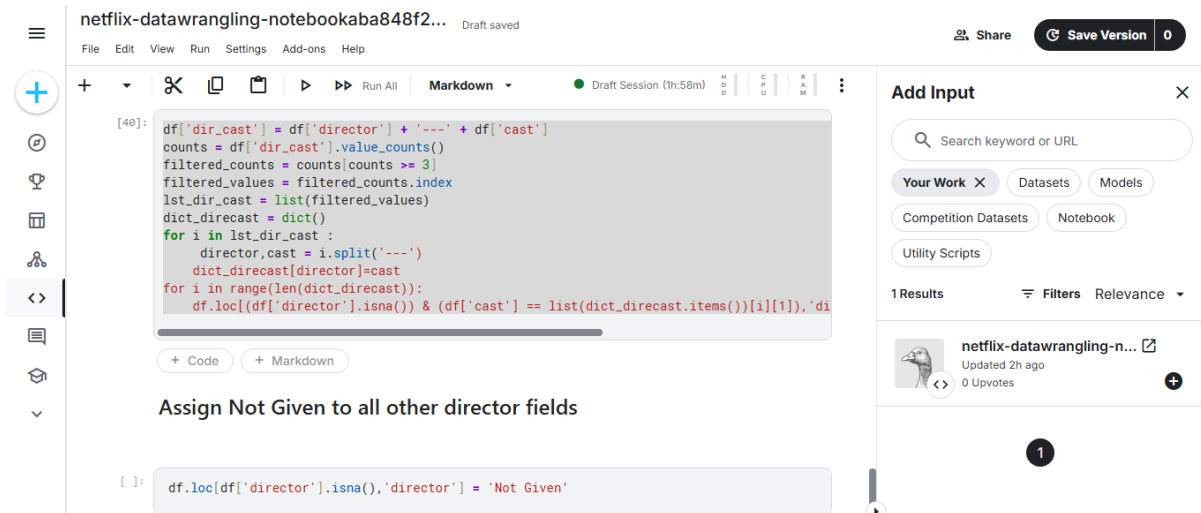
```
dict_directcast[director]=cast
```

```
for i in range(len(dict_directcast)):
```

```
    df.loc[(df['director'].isna()) & (df['cast'] == list(dict_directcast.items())[i][1]),'director'] =  
    list(dict_directcast.items())[i][0]
```

Assign not given to every other directory field

```
df.loc[df['director'].isna(),'director'] = 'Not Given'
```



The screenshot shows a Jupyter Notebook titled "netflix-datawrangling-notebookaba848f2...". The code in the cell is as follows:

```
[40]: df['dir_cast'] = df['director'] + '---' + df['cast']
counts = df['dir_cast'].value_counts()
filtered_counts = counts[counts >= 3]
filtered_values = filtered_counts.index
lst_dir_cast = list(filtered_values)
dict_directcast = dict()
for i in lst_dir_cast:
    director,cast = i.split('---')
    dict_directcast[director]=cast
for i in range(len(dict_directcast)):
    df.loc[(df['director'].isna()) & (df['cast'] == list(dict_directcast.items())[i][1]),'di
```

Below the code cell, there is a text prompt: "Assign Not Given to all other director fields". Below this, another code cell contains the following code:

```
[ ]: df.loc[df['director'].isna(),'director'] = 'Not Given'
```

The right sidebar shows the "Add Input" section with a search bar and buttons for "Your Work", "Datasets", "Models", "Competition Datasets", and "Notebook". It also shows "1 Results" and a "Filters" button.

Use directors to fill missing countries

#Use directors to fill missing countries

```
directors = df['director']
```

```
countries = df['country']
```



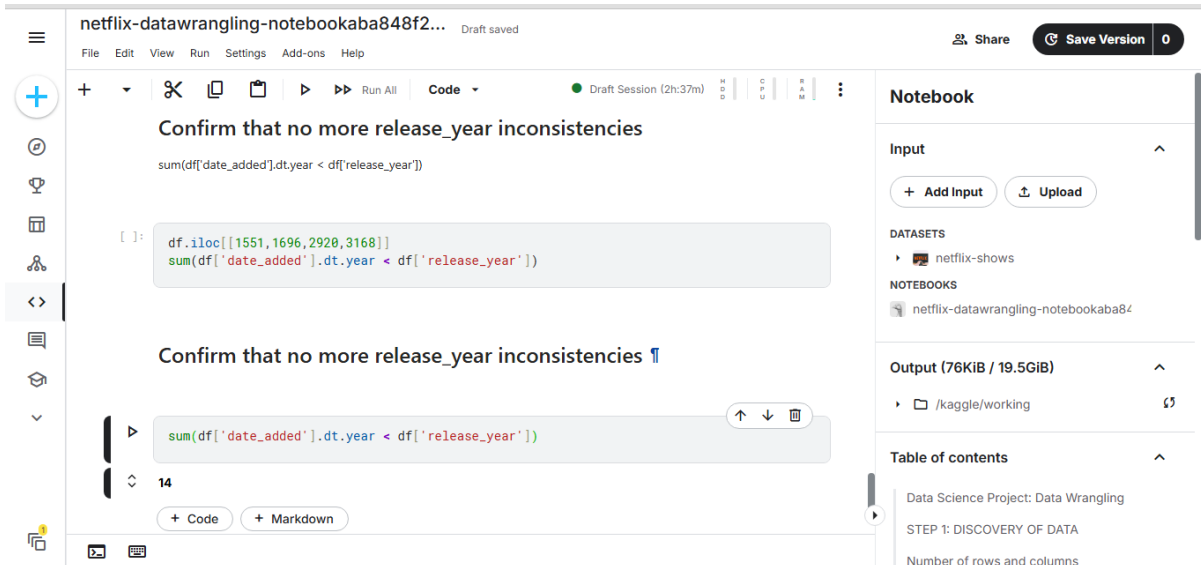
The screenshot shows a Jupyter Notebook titled "netflix-datawrangling-notebookaba848f2...". The code in the cell is as follows:

```
sum(df['date_added'].dt.year < df['release_year'])
df.loc[(df['date_added'].dt.year < df['release_year']),['date_added','release_year']]
```

Below the code cell, there is a table with the following data:

	date_added	release_year
1551	2020-12-14	2021
1696	2020-11-15	2021
2920	2020-02-13	2021
3168	2019-12-06	2020
3287	2019-11-13	2020
3369	2019-10-25	2020
3433	2019-10-11	2020
4844	2018-05-30	2019
4845	2018-05-29	2019
5394	2017-07-01	2018
5658	2016-12-23	2018
5677	2016-12-13	2017

The right sidebar shows the "Notebook" section with "Input" and "Add Input" buttons. It also shows "DATASETS" and "NOTEBOOKS" sections. The "Output" section shows "76KiB / 19.5GiB" and a link to "/kaggle/working". The "Table of contents" section shows "Data Science Project: Data Wrangling".



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

Run All Code Draft Session (2h:37m)

Confirm that no more release_year inconsistencies

```
sum(df['date_added'].dt.year < df['release_year'])
```

```
[ ]: df.iloc[[1551,1696,2928,3168]]
      sum(df['date_added'].dt.year < df['release_year'])
```

Confirm that no more release_year inconsistencies

```
sum(df['date_added'].dt.year < df['release_year'])
```

14

+ Code + Markdown

Share Save Version 0

Notebook

Input

+ Add Input Upload

DATASETS

netflix-shows

NOTEBOOKS

netflix-datawrangling-notebookaba84

Output (76KiB / 19.5GiB)

/kaggle/working

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns

STEP 4: Validation

```
date_issues = (df['date_added'].dt.year < df['release_year']).sum()
```

```
print(f"Found {date_issues} records where added date precedes release year")
```

```
df.loc[df['date_added'].dt.year < df['release_year'], 'date_added'] = pd.to_datetime(
```

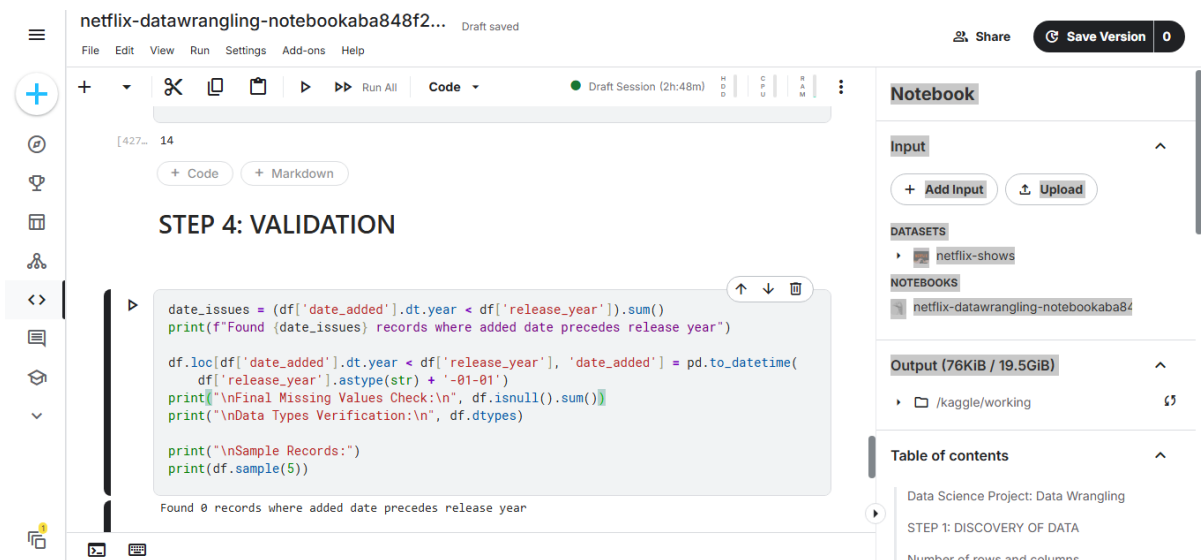
```
df['release_year'].astype(str) + '-01-01')
```

```
print("\nFinal Missing Values Check:\n", df.isnull().sum())
```

```
print("\nData Types Verification:\n", df.dtypes)
```

```
print("\nSample Records:")
```

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



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

Run All Code Draft Session (2h:48m)

[427] 14

+ Code + Markdown

STEP 4: VALIDATION

```
date_issues = (df['date_added'].dt.year < df['release_year']).sum()
print(f"Found {date_issues} records where added date precedes release year")

df.loc[df['date_added'].dt.year < df['release_year'], 'date_added'] = pd.to_datetime(
df['release_year'].astype(str) + '-01-01')
print("\nFinal Missing Values Check:\n", df.isnull().sum())
print("\nData Types Verification:\n", df.dtypes)

print("\nSample Records:")
print(df.sample(5))
```

Found 0 records where added date precedes release year

Share Save Version 0

Notebook

Input

+ Add Input Upload

DATASETS

netflix-shows

NOTEBOOKS

netflix-datawrangling-notebookaba84

Output (76KiB / 19.5GiB)

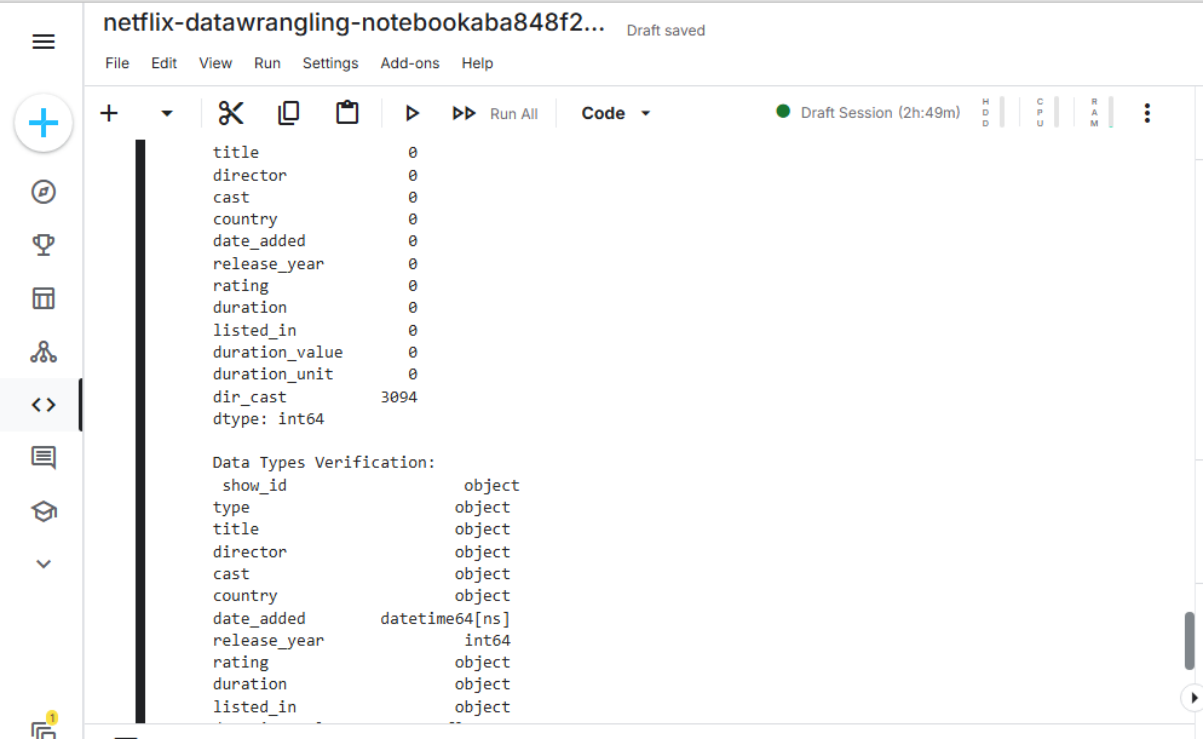
/kaggle/working

Table of contents

Data Science Project: Data Wrangling

STEP 1: DISCOVERY OF DATA

Number of rows and columns



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

+ Draft Session (2h:49m)

```

title      0
director   0
cast       0
country    0
date_added 0
release_year 0
rating     0
duration   0
listed_in  0
duration_value 0
duration_unit 0
dir_cast   3094
dtype: int64

Data Types Verification:
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
  
```

STEP 5: PUBLISHING

Save cleaned data



netflix-datawrangling-notebookaba848f2... Draft saved

File Edit View Run Settings Add-ons Help

+ Draft Session (2h:52m)

Share Save Version 0

STEP 5: DATA PUBLISHING

```

+ Code + Markdown
output_path = '/kaggle/working/cleaned_netflix.csv'
df.to_csv(output_path, index=False)
print(f'\n=== CLEANED DATASET SAVED TO: {output_path} ===')

=== CLEANED DATASET SAVED TO: /kaggle/working/cleaned_netflix.csv ===
  
```

Assign Not Given to all other country fields

Assign Not Given to all other fields

dropping other row records that are null

check if there are any added_dates that come .

sample some of the records and check that..

Confirm that no more release_year inconsisten.

STEP 4: VALIDATION

STEP 5: DATA PUBLISHING

Session options

Schedule a notebook to run

Code Help

Find code help

link to Kaggle Notebook: [Notebook Link](#)

Conclusion

This week I go to grasp on the importance of data cleaning and validation as well as all the steps carried out in data wrangling while as well got to do hands on practices. With this information, I am confident that with time I will build on my skill set while working on more advanced concepts. I look forward to building a portfolio which I can showcase on my CV as I look for jobs in Data and AI.