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

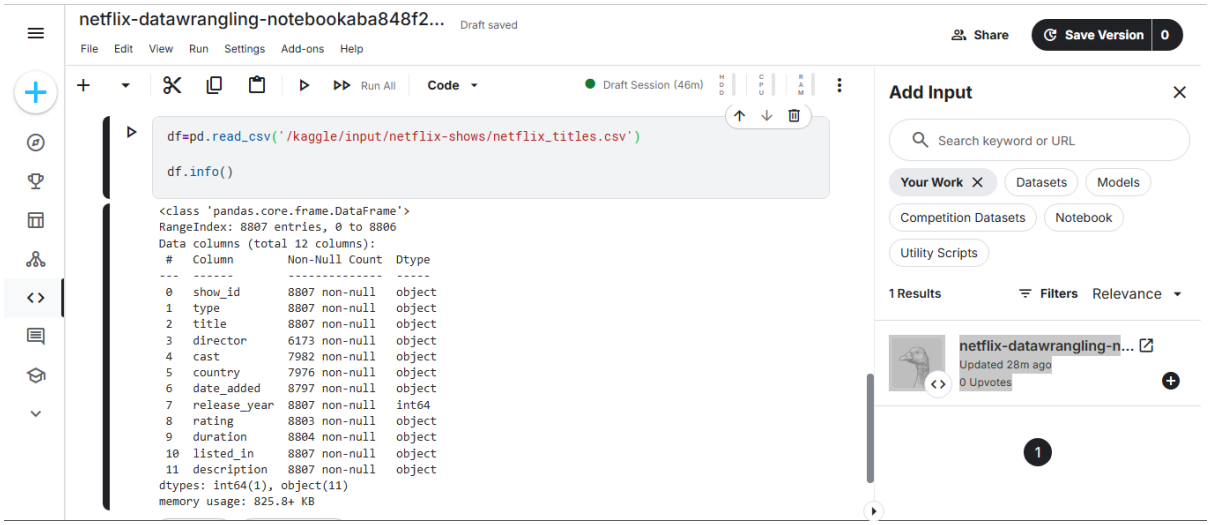
Link to Kaggle Site: Kaggle Site

link to Kaggle Project Notebook: Notebook Link

Stater Code

1. Data Discovery

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



Get the number of rows and columns in the dataset provided

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

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)

Print and count missing values in each column

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

Goup and count duplicated data

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

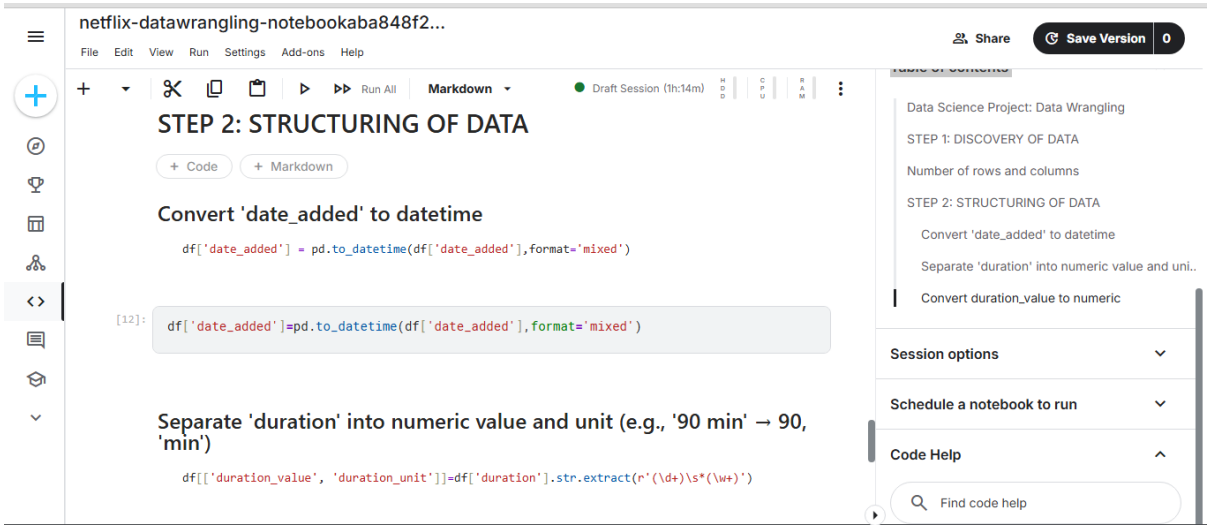
2. Data Structuring

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+)')*

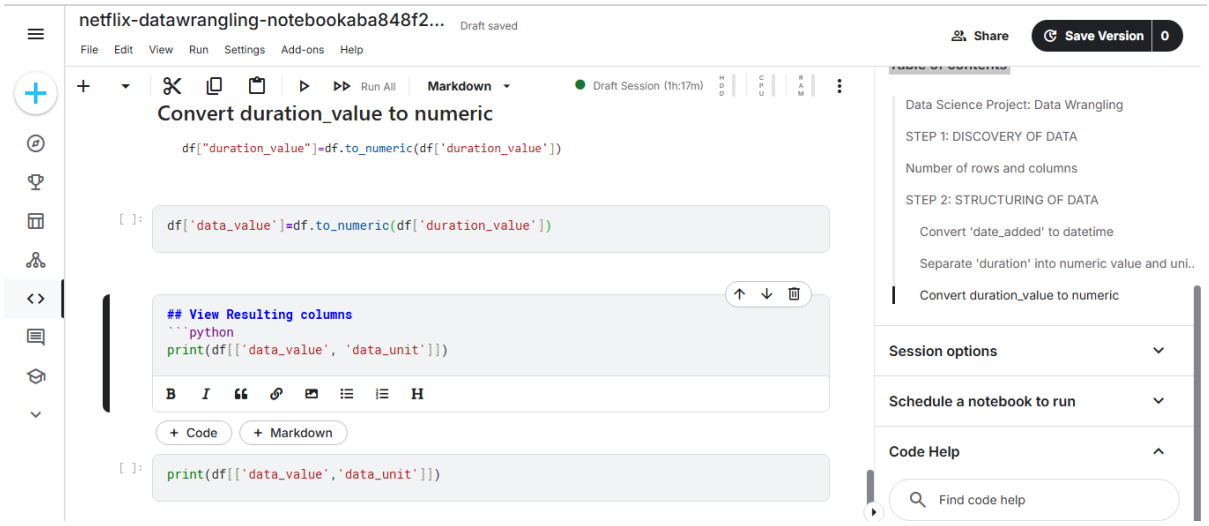


Convert duration_value to numeric

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

View Resulting columns

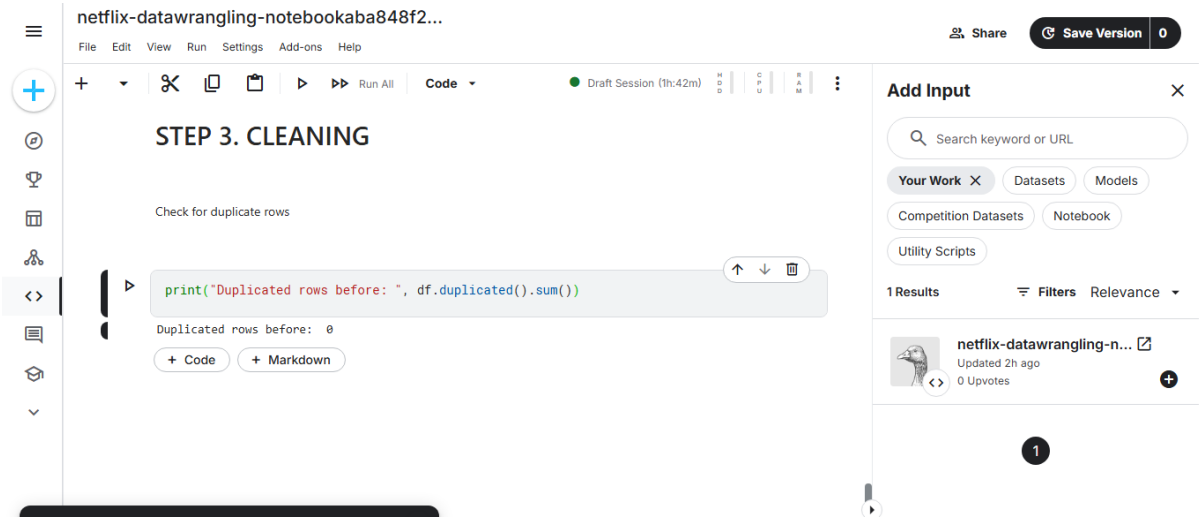
print(df[['duration_value', 'duration_unit']])



3. Cleaning Data

Check for duplicated rows

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



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('---')

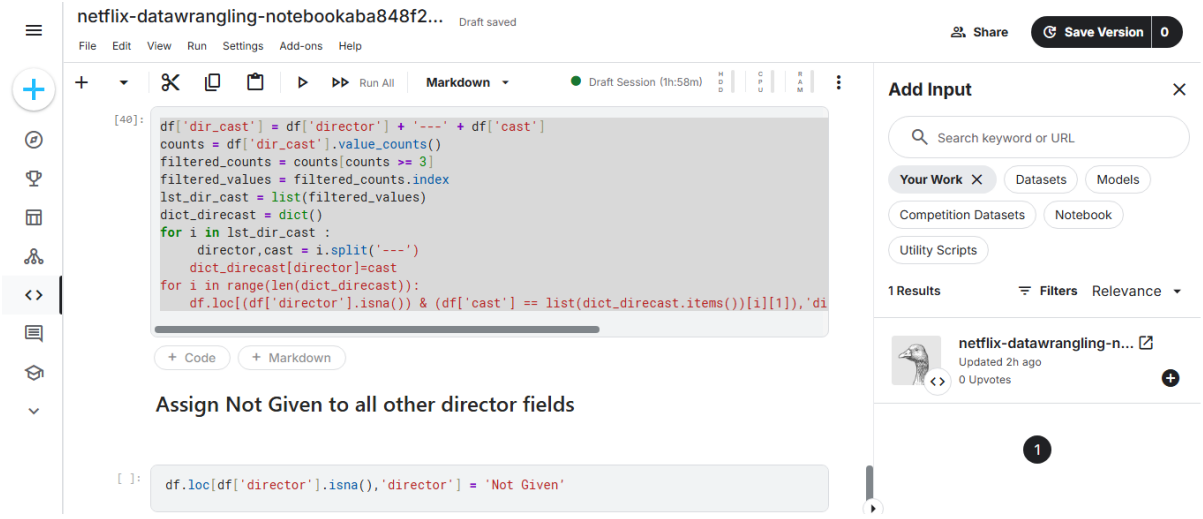
dict_direcast[director]=cast

for i in range(len(dict_direcast)):

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

Assign not given to every other directory field

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

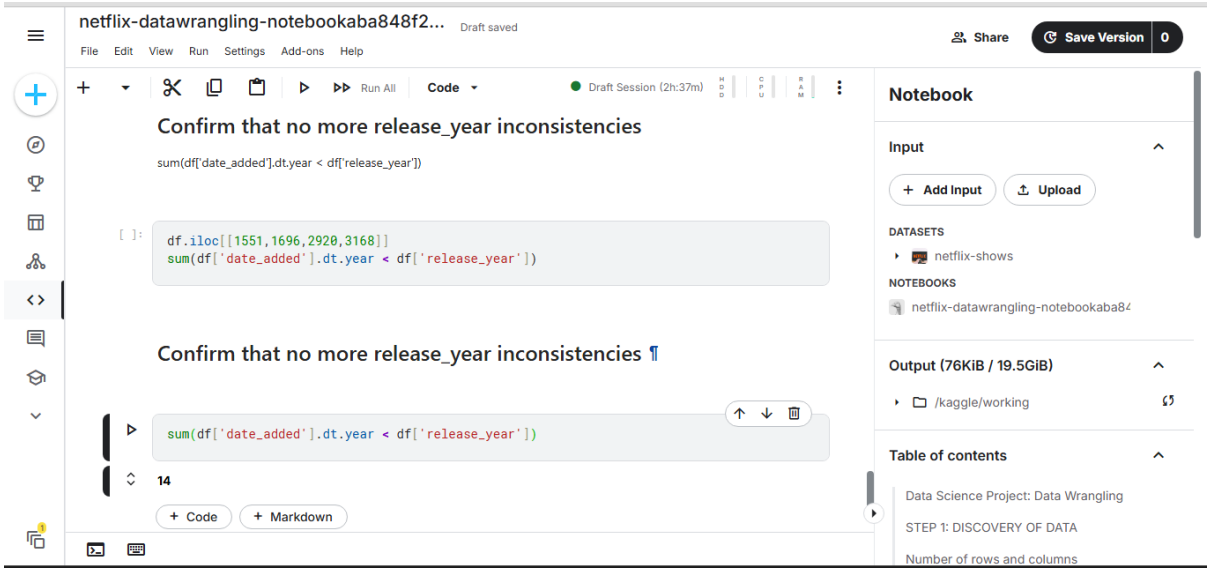


Use directors to fill missing countries

#Use directors to fill missing countries

directors = df['director']

countries = df['country']

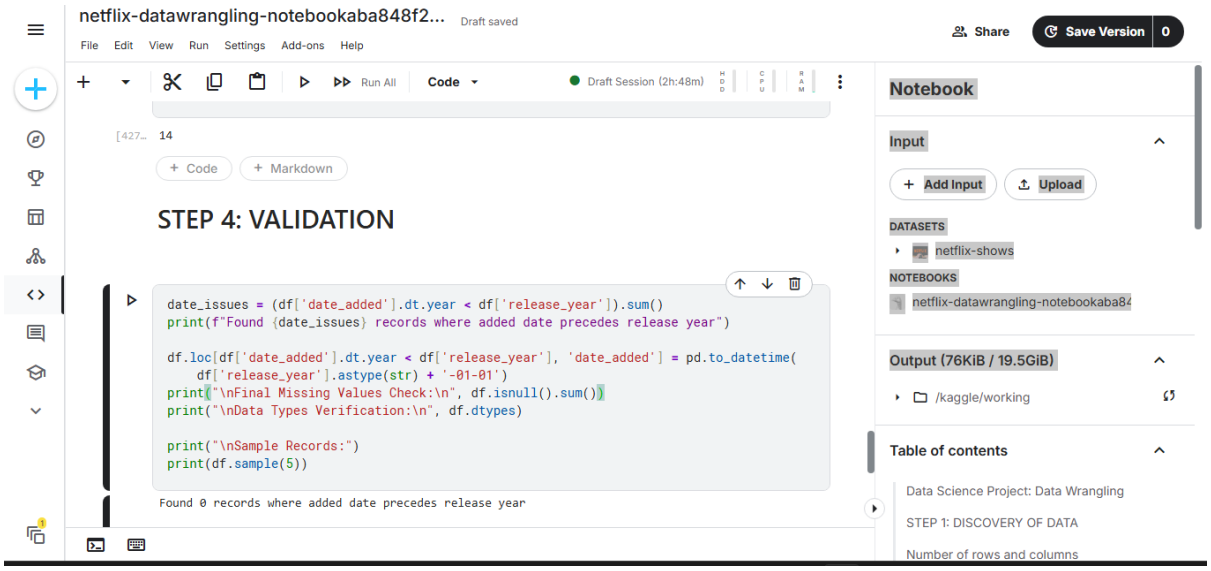


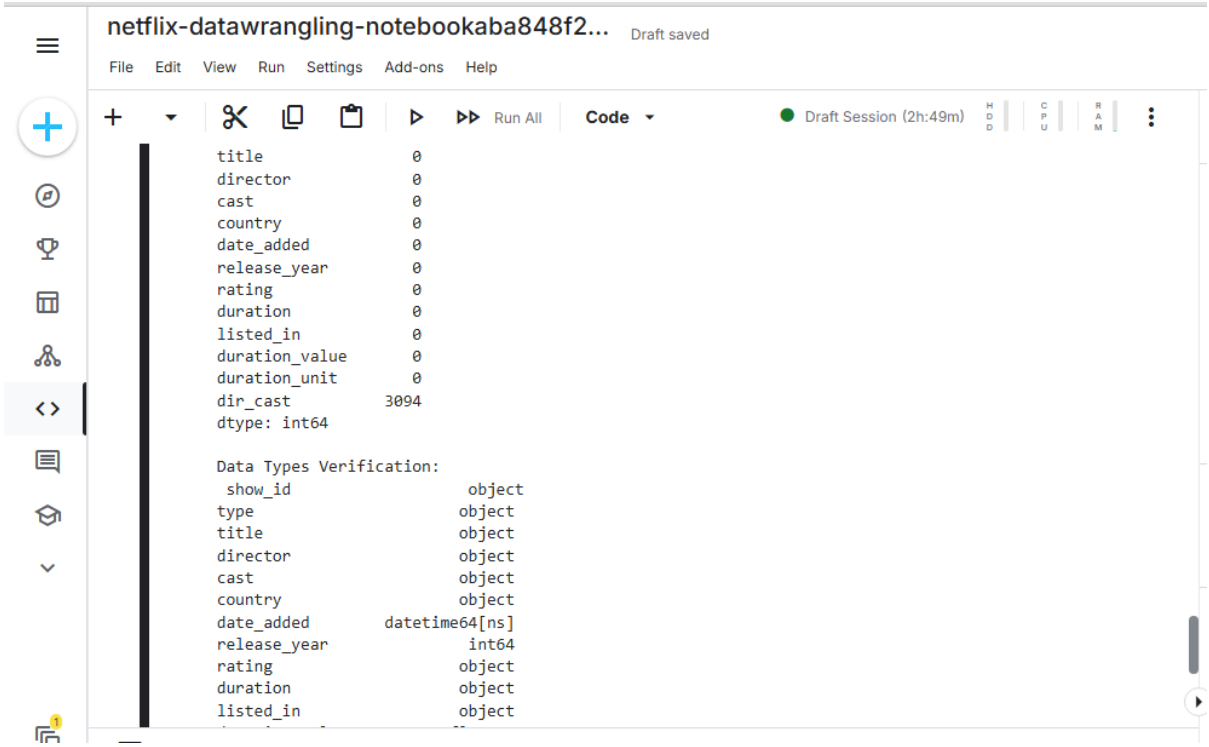
STEP 4: Validation

```
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))
```





STEP 5: PUBLISHING

Save cleaned data

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.