Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

1. KNIME Analytics

- At first, I tried searching which algorithm is mostly suitable for Movie recommendation system. I found out there are couple of algorithms which are commonly used such as Association Mining rule (Content based Recommender), Collaborative Filtered, K Nearest Neighbor. After I struggled with 3types of recommendations, I finally chose to use K Nearest Neighbor algorithm since it Identifies the k-nearest neighbors and the k training data points closest to the test point. Moreover, it is simple to understand and implement where no training phase making it fast for small datasets.

So, let's begin my journey with KNIME analytics in developing Movie Recommendation Systems with K Nearest Neighbor.

Before that, I would like to show the struggles that I did before heading towards the right path of the journey.

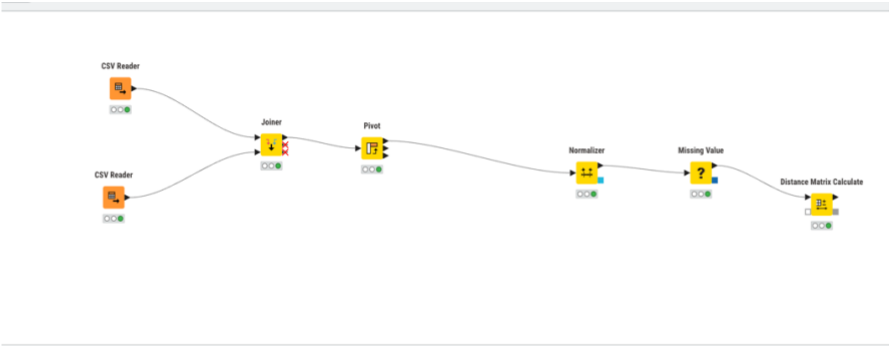


Figure1:

Here I first tried the collaborative filtered algorithm but end of the huge time consumption I failed to figure out the final output.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

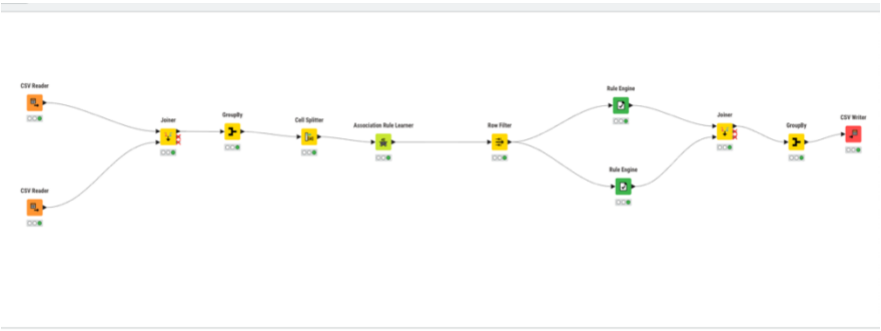


Figure2:

Here again I used association rule algorithm. But it did not give me the expected dataset for visualizing the datasets in the power BI. And I tried very hard to find the result but it was not the algorithm for movie recommendation that I wanted to get insights from. Since it dealt with Confidence, support, and lift variables which did not make me satisfy for making a great insightful analysis of this recommendation system.

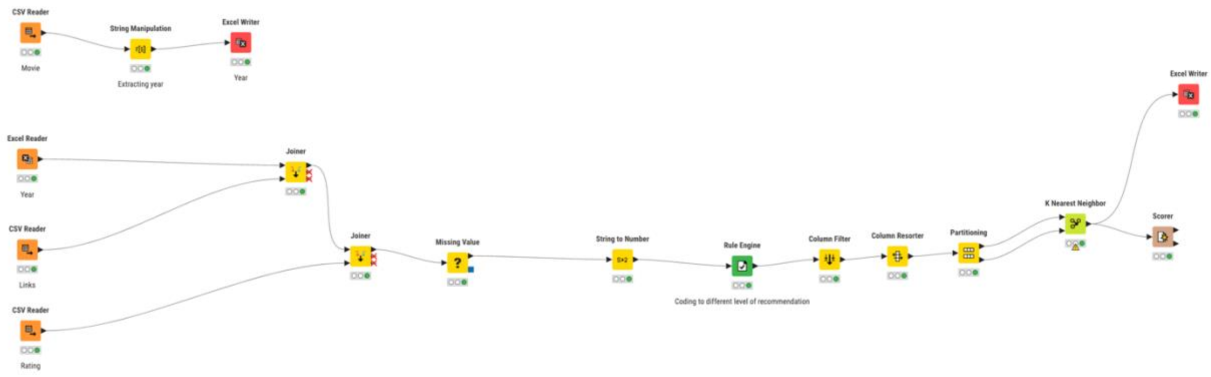


Figure 3:

Finally, I found out finally with the help of researching some articles and google search engine, I got to know about K nearest neighbor. Then I studied about it on YouTube and google online materials. Then I started to implement it step by step.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

The datasets I studied carefully, I noticed that in each title there is year attached. So, as I must get to figure out the time series analysis with span of time, So I decided to extract the year from the Movies dataset the column named title with string manipulation node. I got the formula from google searching for extracting the year.

The formula: regexReplace($title$,"[^0-9]" ,"")

After that I exported the excel file by Excel writer node.

Then I imported the Links and Ratings and joined with the Year excel file by Joiner node. In the joiner node, I matched the two columns with movieID and inner join. Further, I excluded unnecessary timestamp column from the datasets. After all configuration, I got the a combined dataset for my further analysis. Then I realized is there any missing value or not. For that purpose, I used missing value node and I configured the String and number integer in most frequent value and Number double value with fixed value which is 0. Then I noticed the data type of year column in string which is not fit for data analysis. I used string to number node for converting the string data type into number. As I planned before that I will develop the system based on Ratings by users. So I programmed the rule engine node with ratings following:

The coding:

```
 $rating$  = 0 => "Low"

$rating$  = 0.5 => "Low"

$rating$  = 1 => "Low"

$rating$  = 1.5 => "Low"
```

```
$rating$  = 2 => "Medium"

$rating$  = 2.5 => "Medium"

$rating$  = 3 => "Medium"

$rating$  = 3.5 => "Medium"

$rating$  = 4 => "High"

$rating$  = 4.5 => "High"

$rating$  = 5 => "High"
```

Then I renamed the prediction result as Level of recommendation as for specific column.

As I moved on, I saw some columns are unimportant to this analysis. So, I removed these columns such as imdbID, tmdbId. Then I arrange the columns as a to z. I selected Partitioning node for showing my maximum data in the first table the rest of 2nd table so I setup by 70% relative percentage so that I can get the most data in my first table. Then I finally brought the algorithm K class node which takes the closest numbers of the data identified and I checked the wrighted neighbors by distance and numbers of neighbors to consider(K) is 21. Chose the Column Level of recommendation. I used the scorer node for the accuracy level which up to the half that is 56%.

Then I exported the output file with excel writer node for Visualization on Power BI desktop.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

2. Power BI Dashboard

Let us now commence my Data Visualization journey step by step and the difficulties I faced will be described in detail.

At first, I imported the excel file into the power BI through Data source option where I chose the excel file option and then I transformed the data with Power Query the following: I analyzed the data with some common errors. For example, the data type, null value, missing value, and data quality. After all checking the data, I found out the Year column has some string rows along with the years. I tried to remove the string rows but no way found to eradicate the odd rows. Then I researched online about how to clean column from string value and keep only number values. Then I came across to a solution which I am going to explain here. At first, I went to add column tab and then chose custom column and apply the following formula:

```
Text.Select([Year], {"0","1","2","3","4","5","6","7","8","9"})
```

However, it did not work then I struggled a lot to find out the root cause of this problem. Then after I searched about the problem in detail online, I discovered the column is supposed to be in text data type then I should apply the formula. Then I can remove the non-numeric value from the column. I used the above-mentioned formula and then I removed the string value with a result of only numeric value then I changed the data type into Whole number. I also used the home tab remove rows option with error rows and black rows. Now my data is ready to be visualized in the power BI dashboard.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

First, I decided some metrics to be displayed as KPI, such as,

- Yearly filtered data
- Rating based data filtered
- Recommendation movies to top users who gave rating 3 to 5 (Medium to High) by identifying with Users by Ratings line chart
- Displaying card named User Recommendation for finding the top users according to the ratings
- Recommending the high rated genres to the specific users with line chart called Users by ratings
- Popular Genre among users by line chart
- Number of users watched movies per year
- Total number of movies per ratings (1 to 5)
- Percentage of users per Level of Recommendation (High, Medium, Low)
- Displaying the movie title and genre for recognizing the list of movies and genre according to the needs of movie recommendation.
- Top ten genres and users

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

I had to go through the datasets again and again for analyzing the insights I should show on Dashboard. After a while of scrutinizing the datasets, I used Pie chart for displaying the total movie per ratings and donut chart to show percentage of users based on level of recommendation, Movie watched per year by users with a area chart, a two slicers for time series analysis between a period of years, and for rating base movies and users data. I felt the necessity of user slicers for recognizing which users are the top ratings giver to which genre and movies. I used a card for numbers of users. Using line charts for identifying the popular genre among users and the individual user preferences according to ratings respectively. Lastly, I showed the list of movie name and genre for better visualization of specific data insights. And I added a bookmark named clear filter for better user interaction with the dashboard

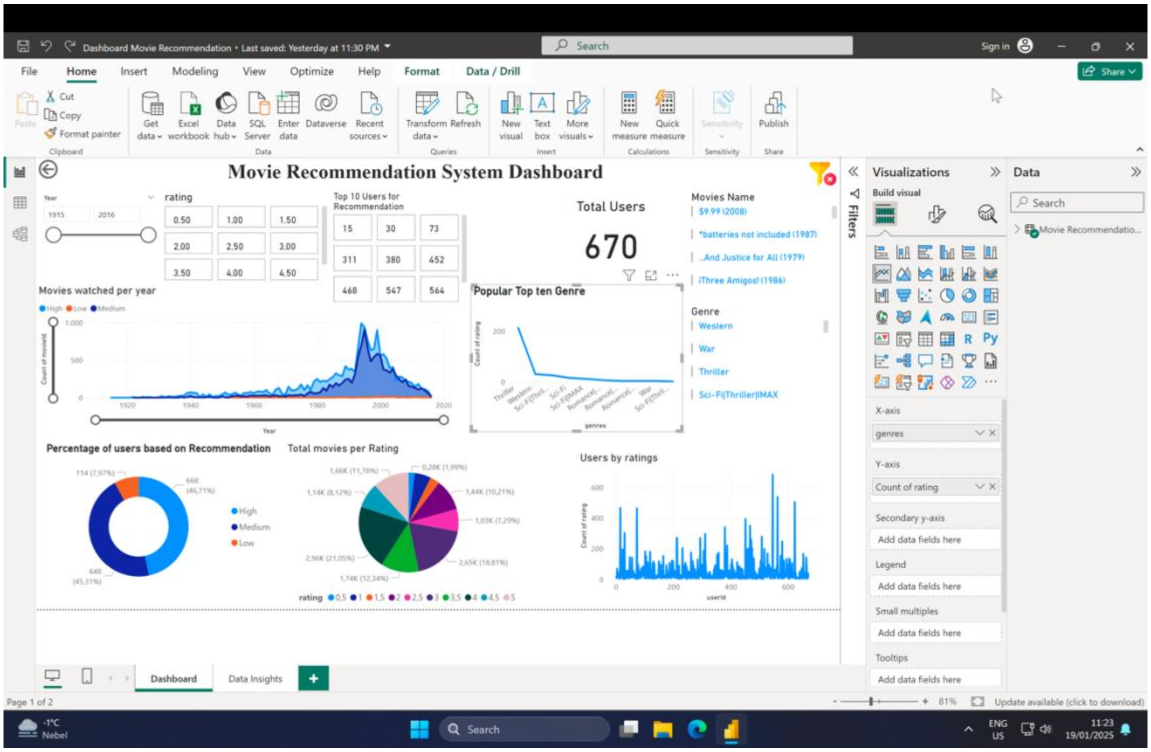


Figure 4:

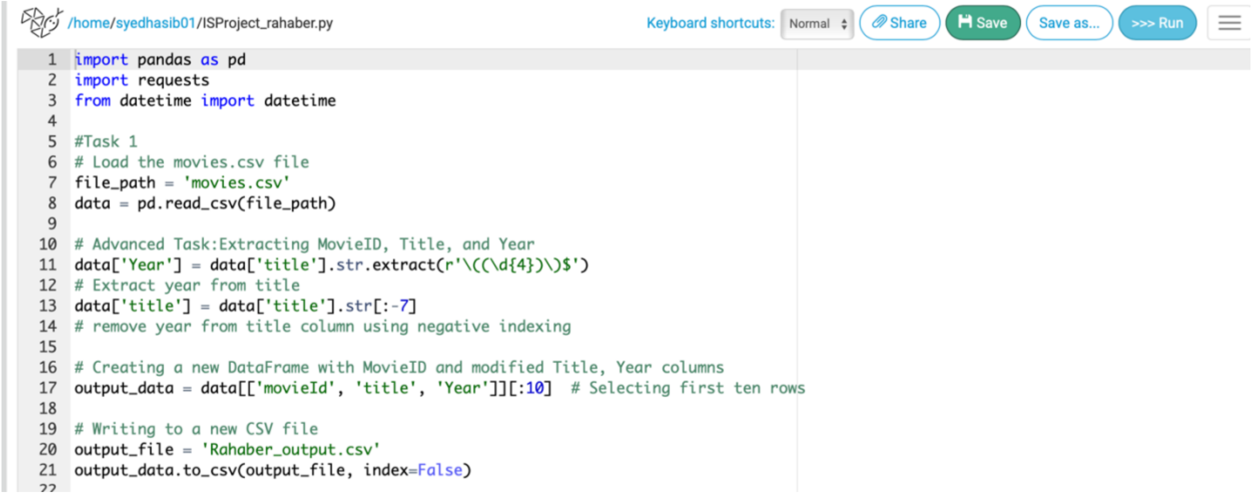
Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

3. Python task in PythonAnywhere

At first, I uploaded the movies.csv file in the file section and created a file named as ISProject_Rahaber.py according to the instruction given in the Moodle file project task. Then I followed the instruction mentioned in the project task.

1. At first, I implemented the local library pandas by writing the following code below-



```python
import pandas as pd
import requests
from datetime import datetime

#Task 1
# Load the movies.csv file
file_path = 'movies.csv'
data = pd.read_csv(file_path)

# Advanced Task:Extracting MovieID, Title, and Year
data['Year'] = data['title'].str.extract(r'\((\d{4})\)$')
# Extract year from title
data['title'] = data['title'].str[:-7]
# remove year from title column using negative indexing

# Creating a new DataFrame with MovieID and modified Title, Year columns
output_data = data[['movieId', 'title', 'Year']][:10]  # Selecting first ten rows

# Writing to a new CSV file
output_file = 'Rahaber_output.csv'
output_data.to_csv(output_file, index=False)
```

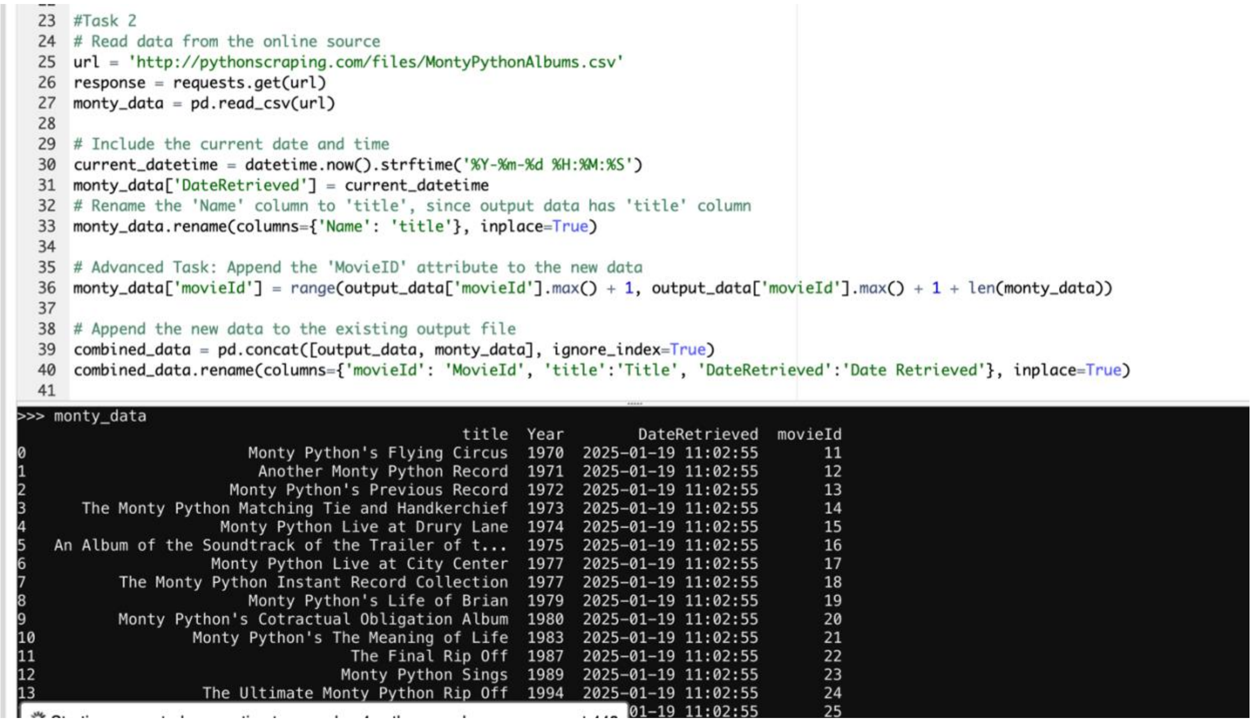
Here, I struggled in advanced task so much. Then I investigated online for suggestions regarding this. I found out how the year is extracted from string column. Applied the code and then then I must remove column from the string title column by coding a different expression with negative 7 since the year and parentheses take up the last 6 characters, with one space before them.

Then constructed a new Data Frame output data containing only the movieId, modified title, and Year from the first ten rows of the original data. Then the script writes this new DataFrame to a CSV file named 'Rahaber_output.csv', without including the index column.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

Task 2:



```python
#Task 2
# Read data from the online source
url = 'http://pythonscraping.com/files/MontyPythonAlbums.csv'
response = requests.get(url)
monty_data = pd.read_csv(url)

# Include the current date and time
current_datetime = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
monty_data['DateRetrieved'] = current_datetime
# Rename the 'Name' column to 'title', since output data has 'title' column
monty_data.rename(columns={'Name': 'title'}, inplace=True)

# Advanced Task: Append the 'MovieID' attribute to the new data
monty_data['movieId'] = range(output_data['movieId'].max() + 1, output_data['movieId'].max() + 1 + len(monty_data))

# Append the new data to the existing output file
combined_data = pd.concat([output_data, monty_data], ignore_index=True)
combined_data.rename(columns={'movieId': 'MovieId', 'title':'Title', 'DateRetrieved':'Date Retrieved'}, inplace=True)

```

```
>>> monty_data
                                          title  Year        DateRetrieved  movieId
0                  Monty Python's Flying Circus  1970  2025-01-19 11:02:55       11
1                   Another Monty Python Record  1971  2025-01-19 11:02:55       12
2                  Monty Python's Previous Record  1972  2025-01-19 11:02:55       13
3       The Monty Python Matching Tie and Handkerchief  1973  2025-01-19 11:02:55       14
4                Monty Python Live at Drury Lane  1974  2025-01-19 11:02:55       15
5   An Album of the Soundtrack of the Trailer of t...  1975  2025-01-19 11:02:55       16
6               Monty Python Live at City Center  1977  2025-01-19 11:02:55       17
7        The Monty Python Instant Record Collection  1977  2025-01-19 11:02:55       18
8                   Monty Python's Life of Brian  1979  2025-01-19 11:02:55       19
9        Monty Python's Cotractual Obligation Album  1980  2025-01-19 11:02:55       20
10           Monty Python's The Meaning of Life  1983  2025-01-19 11:02:55       21
11                          The Final Rip Off  1987  2025-01-19 11:02:55       22
12                         Monty Python Sings  1989  2025-01-19 11:02:55       23
13             The Ultimate Monty Python Rip Off  1994  2025-01-19 11:02:55       24
                                                           01-19 11:02:55       25
```

Here, I handled the online data with an online source using the requests.get() method. However, directly reading into pandas with pd.read_csv(url). Then I added the current date and time as a new column DateRetrieved to the online data and assigns new movieId values to the online data starting from one more than the maximum movieId in the output data from Task 1.

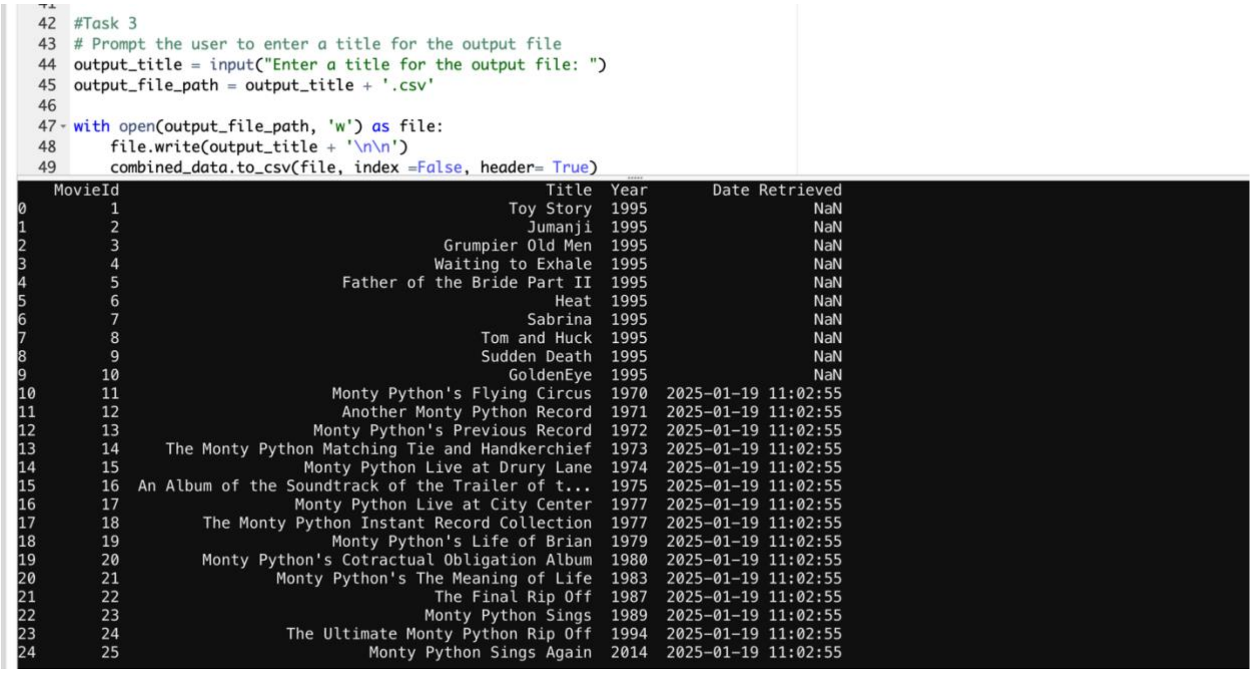
Finally, combined the local data (first 10 movies) with the online data into a single DataFrame called combined_data.

Documentation for IS Project

Name: Syed Hasib Rahaber ID: 30302237

Task 3:

Used this title to create a filename and write the DataFrame combined_data to this new CSV file.



```python
#Task 3
# Prompt the user to enter a title for the output file
output_title = input("Enter a title for the output file: ")
output_file_path = output_title + '.csv'

with open(output_file_path, 'w') as file:
    file.write(output_title + '\n\n')
    combined_data.to_csv(file, index =False, header= True)
```

```
   MovieId                                             Title  Year       Date Retrieved
0        1                                         Toy Story  1995                  NaN
1        2                                           Jumanji  1995                  NaN
2        3                                  Grumpier Old Men  1995                  NaN
3        4                                 Waiting to Exhale  1995                  NaN
4        5                       Father of the Bride Part II  1995                  NaN
5        6                                              Heat  1995                  NaN
6        7                                           Sabrina  1995                  NaN
7        8                                      Tom and Huck  1995                  NaN
8        9                                      Sudden Death  1995                  NaN
9       10                                         GoldenEye  1995                  NaN
10      11                      Monty Python's Flying Circus  1970  2025-01-19 11:02:55
11      12                       Another Monty Python Record  1971  2025-01-19 11:02:55
12      13                    Monty Python's Previous Record  1972  2025-01-19 11:02:55
13      14    The Monty Python Matching Tie and Handkerchief  1973  2025-01-19 11:02:55
14      15                   Monty Python Live at Drury Lane  1974  2025-01-19 11:02:55
15      16  An Album of the Soundtrack of the Trailer of t...  1975  2025-01-19 11:02:55
16      17                  Monty Python Live at City Center  1977  2025-01-19 11:02:55
17      18        The Monty Python Instant Record Collection  1977  2025-01-19 11:02:55
18      19                      Monty Python's Life of Brian  1979  2025-01-19 11:02:55
19      20        Monty Python's Cotractual Obligation Album  1980  2025-01-19 11:02:55
20      21                Monty Python's The Meaning of Life  1983  2025-01-19 11:02:55
21      22                                 The Final Rip Off  1987  2025-01-19 11:02:55
22      23                                Monty Python Sings  1989  2025-01-19 11:02:55
23      24                 The Ultimate Monty Python Rip Off  1994  2025-01-19 11:02:55
24      25                          Monty Python Sings Again  2014  2025-01-19 11:02:55
```

Finally, I downloaded the ISProject_rahaber.py file and uploaded the KNIME, POWER BI, and Python file into the moodle.

Thank you