Cloud Computing

Assignment2

Nureen Ehab Mahmoud Mohamed Barakat

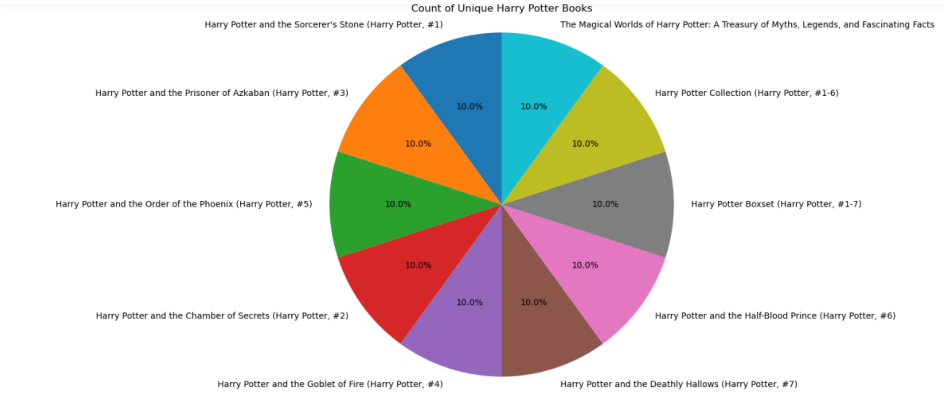
20221465124

Intelligent Systems

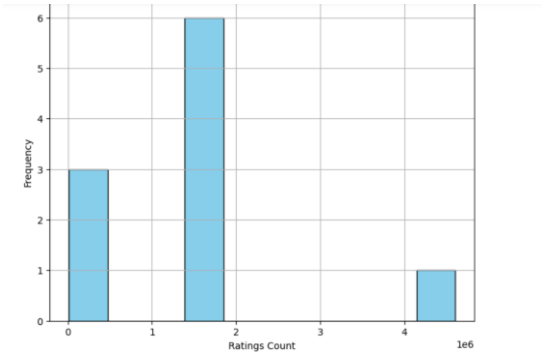
Analysis of Popular Books dataset

Note: plots are more obvious in notebook.

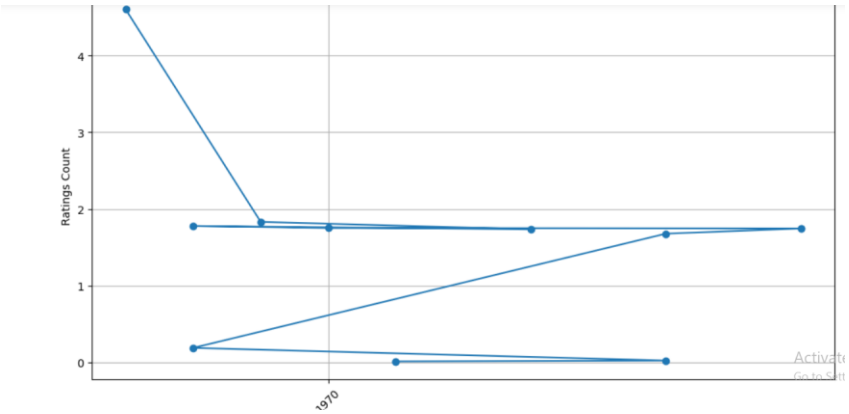
1) Shape of dataset: **(1354, 23)**.
2) Missing values were detected and dropped; number of rows become **1153**.
3) There are no duplicates.
4) Get the following statistical values of each numerical column: count, mean, standard deviation, min, max, The first quartile (Q1), The second quartile (Q2), The third quartile (Q3).
5) Get the following statistical values of each categorical column: count, top, unique, frequency.
6) Harry potter books analysis:
   The most selling Harry Potter book depending on rating count is **Harry Potter and the Sorcerer's Stone** book followed by **Harry Potter and the Prisoner of Azkaban** book, and so on.
   A. The average rating of Harry Potter books by calculating the mean of average rate column is **4.4910000000000005**.
   B. The count of unique Harry Potter books is **10**.
   C. Plot the unique values of Harry Potter books:



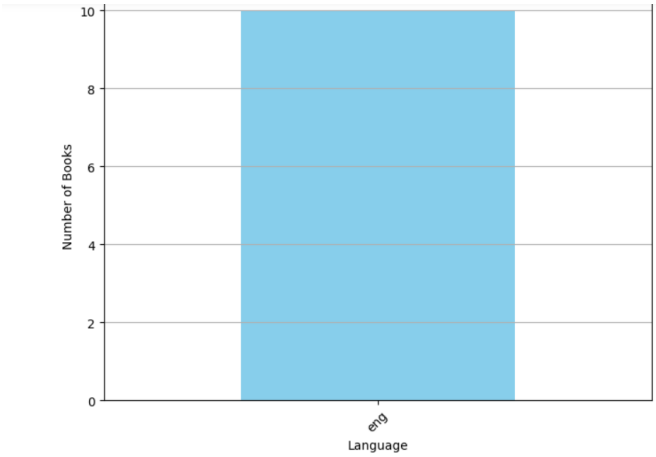
   D. The distribution of ratings count for Harry Potter books:



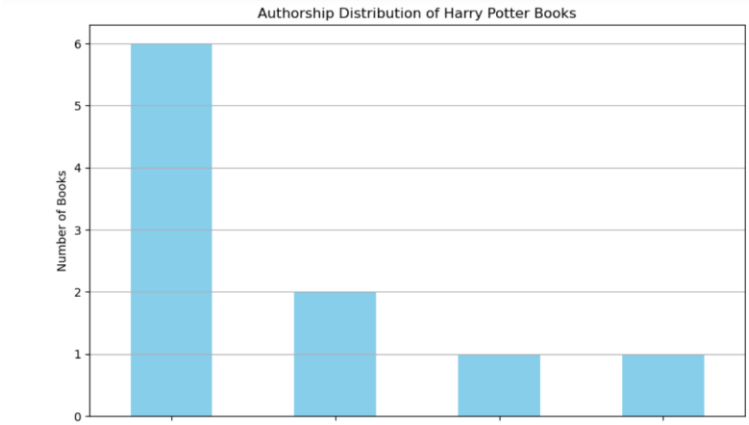
E. Analyze the sales trend over time by plotting sales against the original publication year:



F. Plot the distribution of languages in which the Harry Potter books were published:



G. Plot the authorship of the Harry Potter books and see if there are multiple authors:



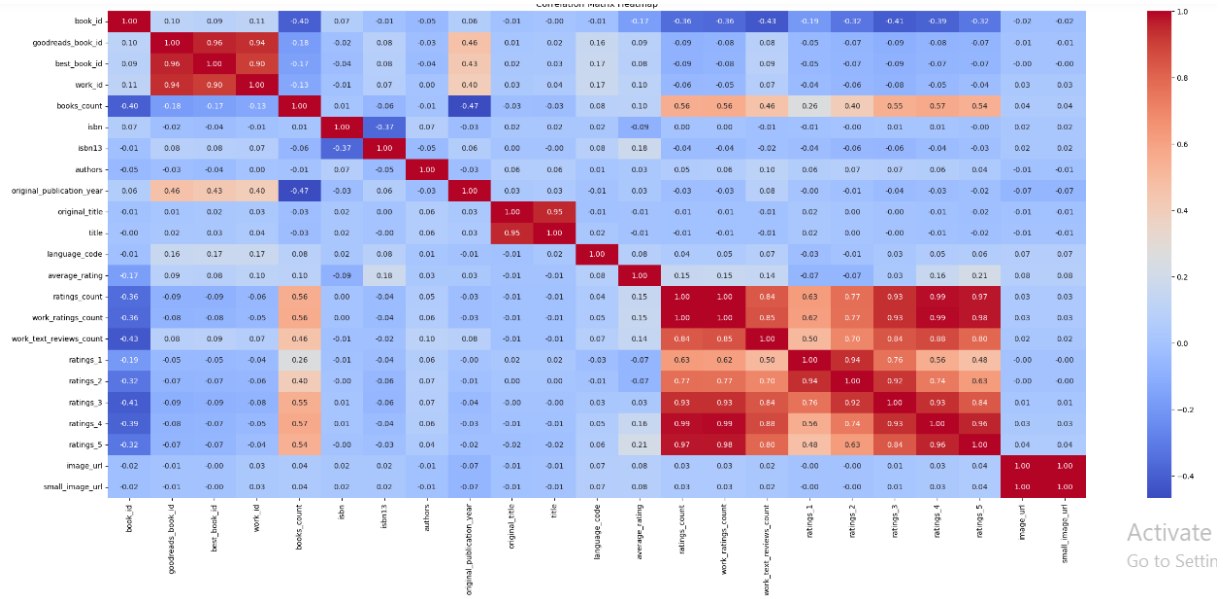
7) Encode categorical columns.

8) Calculate correlation_matrix which is a matrix or a data structure that contains correlations between numerical columns and 'ratings_count'

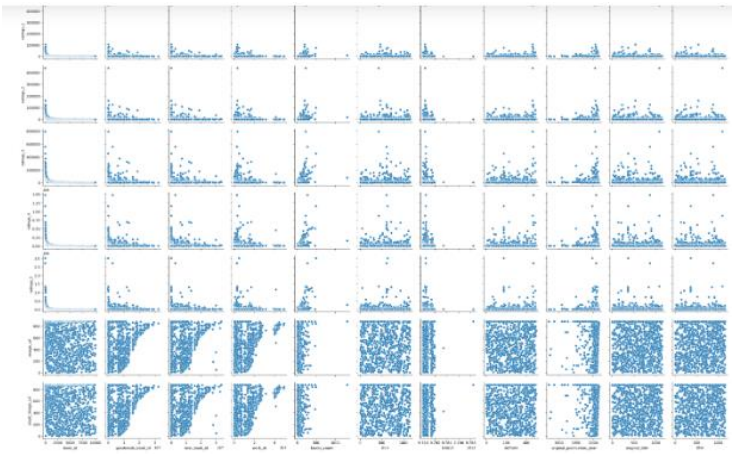
   Top 5 correlated features are:

   1. **work_ratings_count** with 0.998813
   2. **ratings_4** with 0.986129
   3. **ratings_5** with 0.974663
   4. **ratings_3** with 0.933926
   5. **work_text_reviews_count** with 0.838081

9) Plot the correlation with heatmap where each cell's color represents the strength and direction of correlation between the variables:

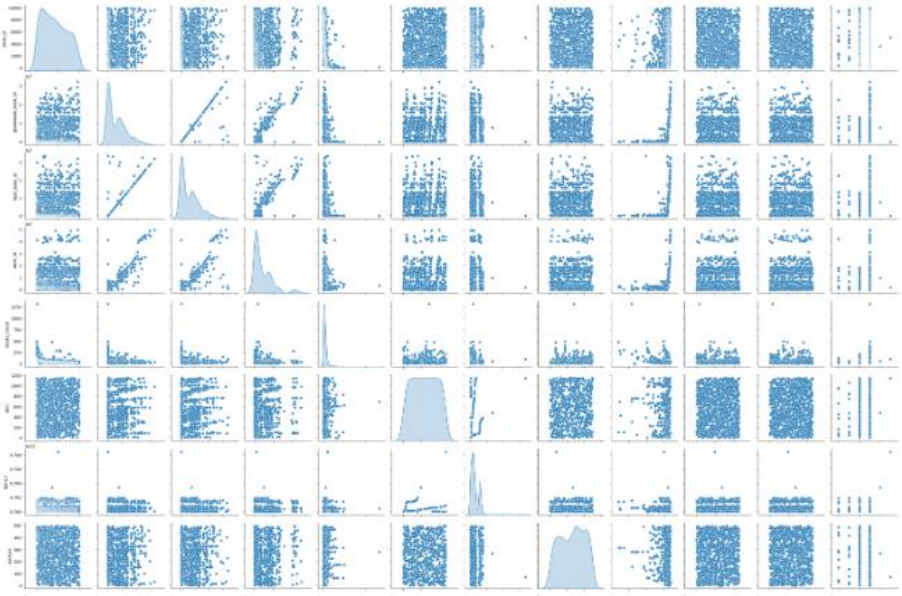


10) Display pair plot which is a grid of scatterplots where each variable is plotted against every other variable, forming a matrix of plots. It allows us to visualize the relationships between pairs of variables and identify potential patterns or correlations. Sample of output:



11) Display scatter plots for pairs of numerical variables with KDE plots along the diagonal, providing additional insights into the distribution of each numerical variable:

Sample of output:



12) Outliers were detected and treated using the Interquartile Range (IQR) method by checking which data points fall below the lower bound or above the upper bound.

- After removing the outliers, the shape of dataset becomes **(932, 23)**.
- Last iteration after removing the outliers:

```
Iteration 3: Outliers after removal
Column 'book_id' has 0 outliers after removal.
Column 'goodreads_book_id' has 0 outliers after removal.
Column 'best_book_id' has 0 outliers after removal.
Column 'work_id' has 0 outliers after removal.
Column 'books_count' has 0 outliers after removal.
Column 'isbn' has 0 outliers after removal.
Column 'isbn13' has 0 outliers after removal.
Column 'authors' has 0 outliers after removal.
Column 'original_publication_year' has 0 outliers after removal.
Column 'original_title' has 0 outliers after removal.
Column 'title' has 0 outliers after removal.
Column 'language_code' has 0 outliers after removal.
Column 'average_rating' has 0 outliers after removal.
Column 'ratings_count' has 0 outliers after removal.
Column 'work_ratings_count' has 0 outliers after removal.
Column 'work_text_reviews_count' has 1 outliers after removal.
Column 'ratings_1' has 0 outliers after removal.
Column 'ratings_2' has 0 outliers after removal.
```

- Example of lower and upper bounds for 'ratings_count' column:

```
Lower bound for ratings_count: -53777.0
Upper bound for ratings_count: 139183.0
```

- Example of outliers found before removal for 'ratings_count' column. Sample of output (last iteration):

```
Iteration 3: Outliers before removal
     book_id  goodreads_book_id  best_book_id  work_id  books_count  isbn  \
135      615              22205         22205   132402           42   251
145      667            5664985       5664985  5836517           41  1001

           isbn13  authors  original_publication_year  original_title  ...  \
135  9.780143e+12      437                     2002.0            1046  ...
145  9.780670e+12      437                     2009.0              50  ...

     ratings_count  work_ratings_count  work_text_reviews_count  ratings_1  \
135         151829              155107                     4597       2714
145         151721              156330                     6238       2588

     ratings_2  ratings_3  ratings_4  ratings_5  image_url  small_image_url
135       8175      32346      49750      62122        667              667
145       7073      30314      52171      64184        859              859

[2 rows x 23 columns]
------------------------------------------------
```

- Example of outliers found after removal for 'ratings_count' column. Sample of output (last iteration):

```
------------------------------------------------
Iteration 3: Outliers after removal
Empty DataFrame
Columns: [book_id, goodreads_book_id, best_book_id, work_id, books_count, isbn, isbn13, authors, original_publication_year, ori
ginal_title, title, language_code, average_rating, ratings_count, work_ratings_count, work_text_reviews_count, ratings_1, ratin
gs_2, ratings_3, ratings_4, ratings_5, image_url, small_image_url]
Index: []

[0 rows x 23 columns]
------------------------------------------------
```

13) Numeric features were scaled using MinMaxScaler to ensure uniformity of scale for modeling.

Now the dataset is cleaned, scaled, and ready for modeling.

Steps of creating Dockerfile to run Jupyter Notebook when the container launches

14) Pull the image (Jupyter Notebook)

```
PS C:\Users\merna\jupyter> docker pull jupyter/datascience-notebook
Using default tag: latest
latest: Pulling from jupyter/datascience-notebook
Digest: sha256:476c6e673e7d5d8b5059f8680b1c6a988942a79263da651bf302dc696ab311f2
Status: Image is up to date for jupyter/datascience-notebook:latest
docker.io/jupyter/datascience-notebook:latest
```

☐	jupyter/datascience-notebook f78a42f3bc9a	latest	Unused	6 months ago	5.92 GB	▶ ⋮ 🗑

15) Create a Dockerfile

1. Create a directory

```
PS C:\Users\merna> mkdir jupyter


    Directory: C:\Users\merna


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/23/2024  10:17 PM                jupyter
```

2. Create a Dockerfile inside the directory
3. Add commands inside Dockerfile

Dockerfile

```
C: > Users > merna > Dockerfile > ...
# Use an official Python runtime as a parent image
FROM python:3.8

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Make port 8888 available to the world outside this container
EXPOSE 8888

# Define environment variable
ENV NAME World

# Run Jupyter Notebook when the container launches
CMD ["jupyter", "notebook", "--ip='0.0.0.0'", "--port=8888", "--no-browser", "--allow-root"]
```

4. Create a `requirements.txt` file for adding dependences

```
pandas
seaborn
matplotlib==3.6.2
numpy==1.23.4
scikit-learn
```

5. Change directory to the created one

```
PS D:\FCDS\Sixth Term\Cloud Computing\Docker> cd "C:\Users\merna\jupyter"
```

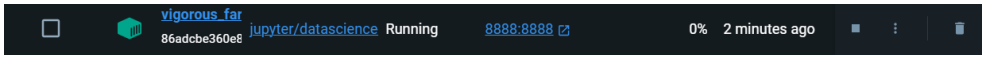
6. Run docker container

```
PS C:\Users\merna\jupyter> docker run -p 8888:8888 jupyter/datascience-notebook
Entered start.sh with args: jupyter lab
Running hooks in: /usr/local/bin/start-notebook.d as uid: 1000 gid: 100
Done running hooks in: /usr/local/bin/start-notebook.d
Running hooks in: /usr/local/bin/before-notebook.d as uid: 1000 gid: 100
Done running hooks in: /usr/local/bin/before-notebook.d
Executing the command: jupyter lab
[I 2024-04-23 22:12:12.636 ServerApp] Package jupyterlab took 0.0000s to import
[I 2024-04-23 22:12:12.651 ServerApp] Package jupyter_lsp took 0.0137s to import
[W 2024-04-23 22:12:12.651 ServerApp] A `_jupyter_server_extension_points` function was not found in jupyter_lsp. Instead, a `_jupyter_server_extens
ion_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-04-23 22:12:12.655 ServerApp] Package jupyter_server_mathjax took 0.0032s to import
[I 2024-04-23 22:12:12.767 ServerApp] Package jupyter_server_proxy took 0.1115s to import
[I 2024-04-23 22:12:12.776 ServerApp] Package jupyter_server_terminals took 0.0085s to import
[I 2024-04-23 22:12:12.848 ServerApp] Package jupyterlab_git took 0.0713s to import
[I 2024-04-23 22:12:12.852 ServerApp] Package nbclassic took 0.0035s to import
[W 2024-04-23 22:12:12.856 ServerApp] A `_jupyter_server_extension_points` function was not found in nbclassic. Instead, a `_jupyter_server_extensio
n_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-04-23 22:12:12.857 ServerApp] Package nbdime took 0.0000s to import
[I 2024-04-23 22:12:12.858 ServerApp] Package notebook took 0.0000s to import
[I 2024-04-23 22:12:12.863 ServerApp] Package notebook_shim took 0.0000s to import
[W 2024-04-23 22:12:12.863 ServerApp] A `_jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `_jupyter_server_exte
nsion_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-04-23 22:12:12.863 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-04-23 22:12:12.870 ServerApp] jupyter_server_mathjax | extension was successfully linked.
[I 2024-04-23 22:12:12.870 ServerApp] jupyter_server_proxy | extension was successfully linked.
[I 2024-04-23 22:12:12.878 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-04-23 22:12:12.889 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-04-23 22:12:12.889 ServerApp] jupyterlab_git | extension was successfully linked.
[I 2024-04-23 22:12:12.896 ServerApp] nbclassic | extension was successfully linked.
[I 2024-04-23 22:12:12.896 ServerApp] nbdime | extension was successfully linked.
[I 2024-04-23 22:12:12.907 ServerApp] notebook | extension was successfully linked.
[I 2024-04-23 22:12:12.915 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2024-04-23 22:12:13.317 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-04-23 22:12:13.350 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-04-23 22:12:13.355 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-04-23 22:12:13.356 ServerApp] jupyter_server_mathjax | extension was successfully loaded.
[I 2024-04-23 22:12:13.380 ServerApp] jupyter_server_proxy | extension was successfully loaded.
[I 2024-04-23 22:12:13.382 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-04-23 22:12:13.420 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.11/site-packages/jupyterlab
```



```
[I 2024-04-23 22:12:13.355 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-04-23 22:12:13.356 ServerApp] jupyter_server_mathjax | extension was successfully loaded.
[I 2024-04-23 22:12:13.380 ServerApp] jupyter_server_proxy | extension was successfully loaded.
[I 2024-04-23 22:12:13.382 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-04-23 22:12:13.420 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.11/site-packages/jupyterlab
[I 2024-04-23 22:12:13.420 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 2024-04-23 22:12:13.422 LabApp] Extension Manager is 'pypi'.
[I 2024-04-23 22:12:13.426 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-04-23 22:12:13.436 ServerApp] jupyterlab_git | extension was successfully loaded.
[I 2024-04-23 22:12:13.443 ServerApp] nbclassic | extension was successfully loaded.
[I 2024-04-23 22:12:13.595 ServerApp] nbdime | extension was successfully loaded.
[I 2024-04-23 22:12:13.599 ServerApp] notebook | extension was successfully loaded.
[I 2024-04-23 22:12:13.601 ServerApp] Serving notebooks from local directory: /home/jovyan
[I 2024-04-23 22:12:13.601 ServerApp] Jupyter Server 2.8.0 is running at:
[I 2024-04-23 22:12:13.602 ServerApp] http://30f9cf11438f:8888/lab?token=7f3128de9204fde36b2f8f09a95b42afb9605c47a1953ac9
[I 2024-04-23 22:12:13.602 ServerApp]     http://127.0.0.1:8888/lab?token=7f3128de9204fde36b2f8f09a95b42afb9605c47a1953ac9
[I 2024-04-23 22:12:13.602 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2024-04-23 22:12:13.608 ServerApp]

    To access the server, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/jpserver-8-open.html
    Or copy and paste one of these URLs:
        http://30f9cf11438f:8888/lab?token=7f3128de9204fde36b2f8f09a95b42afb9605c47a1953ac9
        http://127.0.0.1:8888/lab?token=7f3128de9204fde36b2f8f09a95b42afb9605c47a1953ac9
[I 2024-04-23 22:12:19.409 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescrip
t-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server
, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languagese
rver-bin, yaml-language-server
```



7. Access Jupyter Notebook by the defined port (http://localhost:8888)