

**Name: Bassant Mohamed**

**ID:20221376715      Department: Ai**

**First:** creating docker file that contains: 1) base image which is python

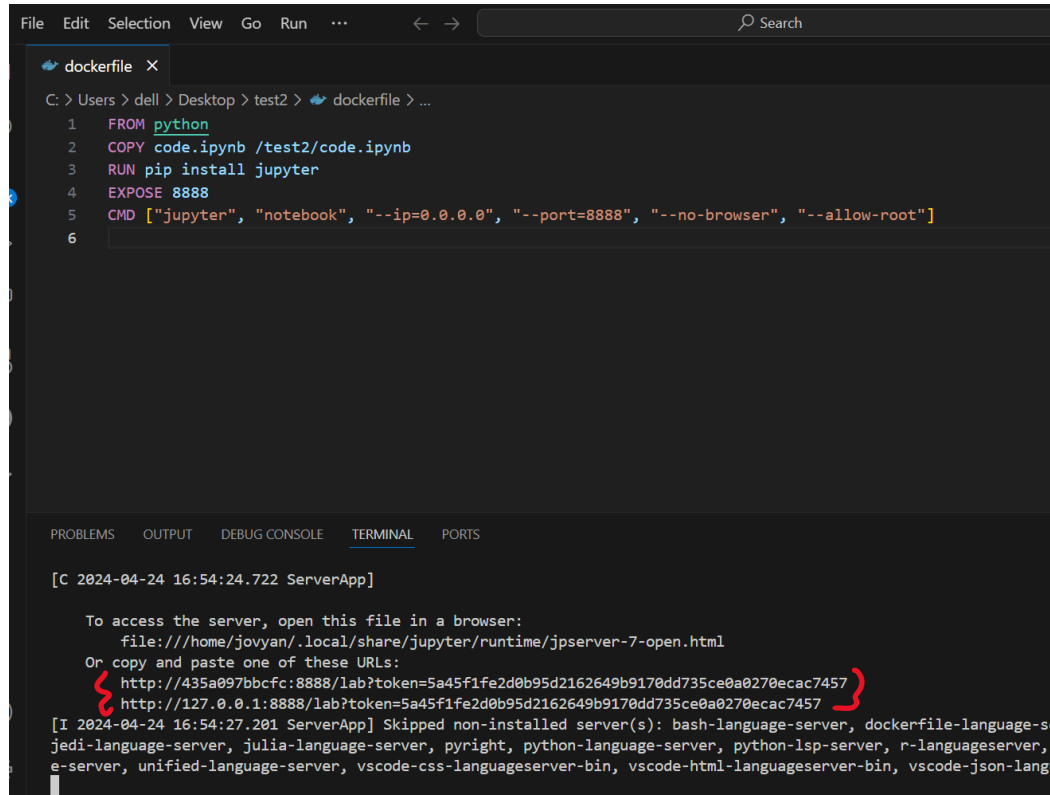
2) At first we don't have the notebook but after creating the notebook I copied it in the doker file.

3) installing the jupyter

4) expose that contains the port we will use

5)cmd that contains the environment we will use , port number , ip address

**Second:** open the terminal and build the image if not build then run it on the port you want so the container will be build and will give you the token of the jupyter notebook.



The screenshot shows a VS Code editor with a file named 'dockerfile'. The code in the file is as follows:

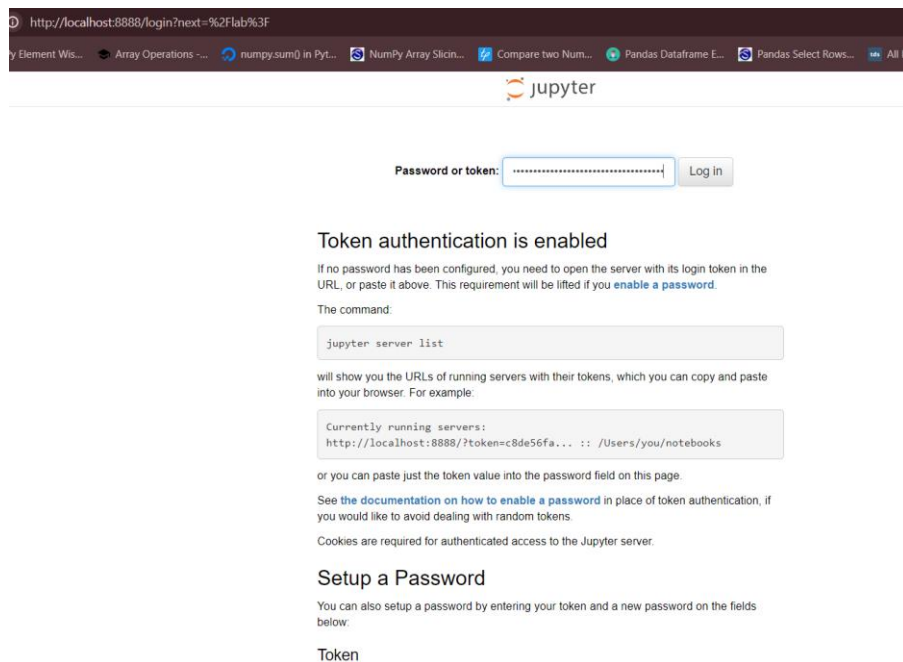
```
1 FROM python
2 COPY code.ipynb /test2/code.ipynb
3 RUN pip install jupyter
4 EXPOSE 8888
5 CMD ["jupyter", "notebook", "--ip=0.0.0.0", "--port=8888", "--no-browser", "--allow-root"]
6
```

The terminal output at the bottom shows the command execution and the resulting URLs to access the Jupyter server:

```
[C 2024-04-24 16:54:24.722 ServerApp]

To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
Or copy and paste one of these URLs:
http://435a097bbcfc:8888/lab?token=5a45f1fe2d0b95d2162649b9170dd735ce0a0270ecac7457
http://127.0.0.1:8888/lab?token=5a45f1fe2d0b95d2162649b9170dd735ce0a0270ecac7457
[I 2024-04-24 16:54:27.201 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-se
jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver,
e-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-langu
```

Then copy the token and put it to open jupyter in your localhost:



**Third :** now you can write your python code and start the analysis.

**Fourth:** I have uploaded the data on the jupyter notebook ten read the first 5 rows.

```
# Load the dataset
df = pd.read_csv('books.csv')
df.head()
```

	book_id	goodreads_book_id	best_book_id	work_id	books_count	isbn	isbn13	authors	original_publication_year	original_title	...	ratings_count
0	1	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins	2008.0	The Hunger Games	...	4780653
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	...	4602479
2	3	41865	41865	3212258	226	316015849	9.780316e+12	Stephenie Meyer	2005.0	Twilight	...	3866839
3	6	11870085	11870085	16827462	226	525478817	9.780525e+12	John Green	2012.0	The Fault in Our Stars	...	2346404
4	12	13335037	13335037	13155899	210	62024035	9.780062e+12	Veronica Roth	2011.0	Divergent	...	1903563

5 rows × 23 columns

**Fifth:** I have start the cleaning step first of all I checked for the missing values and found that there is four columns have missing values so, I decided to remove the missing values , lets see after removing the missing values.

<pre># Check for missing values missing_values = df.isna().sum()  print("Missing values in each column:") print(missing_values)</pre>	Before	<pre>df.dropna(inplace=True)</pre>	After
<pre># Check for missing values after removing missing values missing_values = df.isna().sum()  print("Missing values in each column:") print(missing_values)</pre>			
Missing values in each column:		Missing values in each column:	
book_id	0	book_id	0
goodreads_book_id	0	goodreads_book_id	0
best_book_id	0	best_book_id	0
work_id	0	work_id	0
books_count	0	books_count	0
isbn	52	isbn	0
isbn13	44	isbn13	0
authors	0	authors	0
original_publication_year	3	original_publication_year	0
original_title	52	original_title	0
title	0	title	0
language_code	109	language_code	0
average_rating	0	average_rating	0
ratings_count	0	ratings_count	0
work_ratings_count	0	work_ratings_count	0
work_text_reviews_count	0	work_text_reviews_count	0
ratings_1	0	ratings_1	0
ratings_2	0	ratings_2	0
ratings_3	0	ratings_3	0
ratings_4	0	ratings_4	0
ratings_5	0	ratings_5	0
image_url	0	image_url	0
small_image_url	0	small_image_url	0
dtype: int64		dtype: int64	

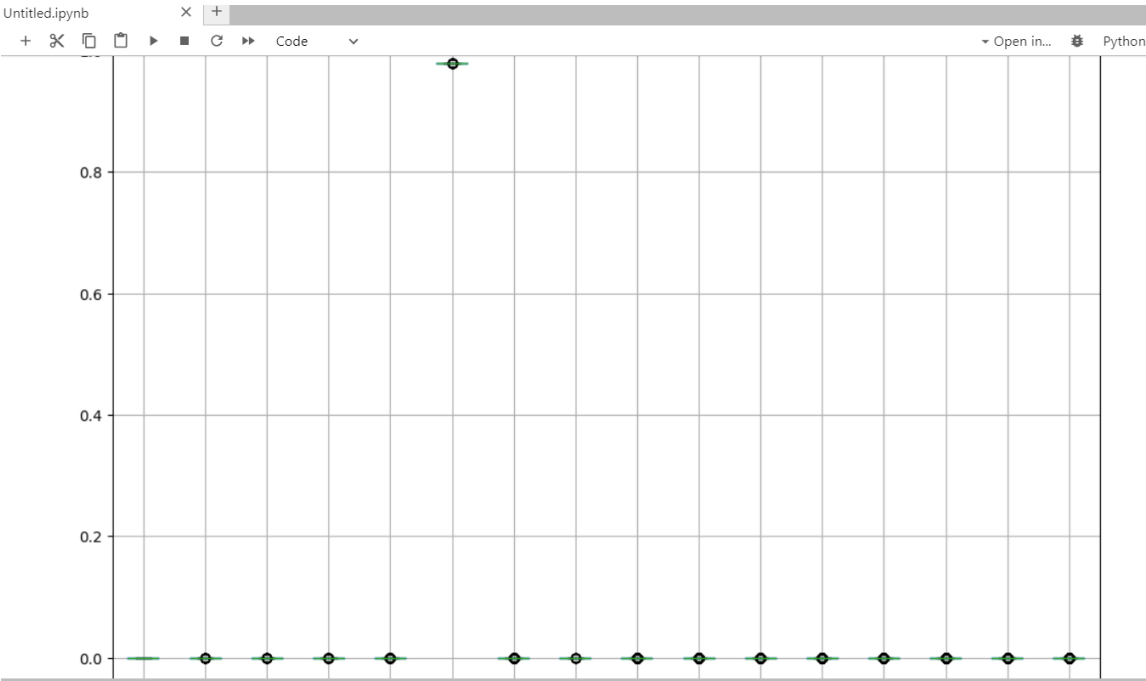
**Six:** checking for any duplicates and remove it, but there wasn't:

```
Duplicate rows:
Empty DataFrame
Columns: [book_id, goodreads_book_id, best_book_id, work_id, books_count, isbn, isbn13, authors, original_publication_year, original_title, title, lang
uage_code, average_rating, ratings_count, work_ratings_count, work_text_reviews_count, ratings_1, ratings_2, ratings_3, ratings_4, ratings_5, image_ur
l, small_image_url]
Index: []

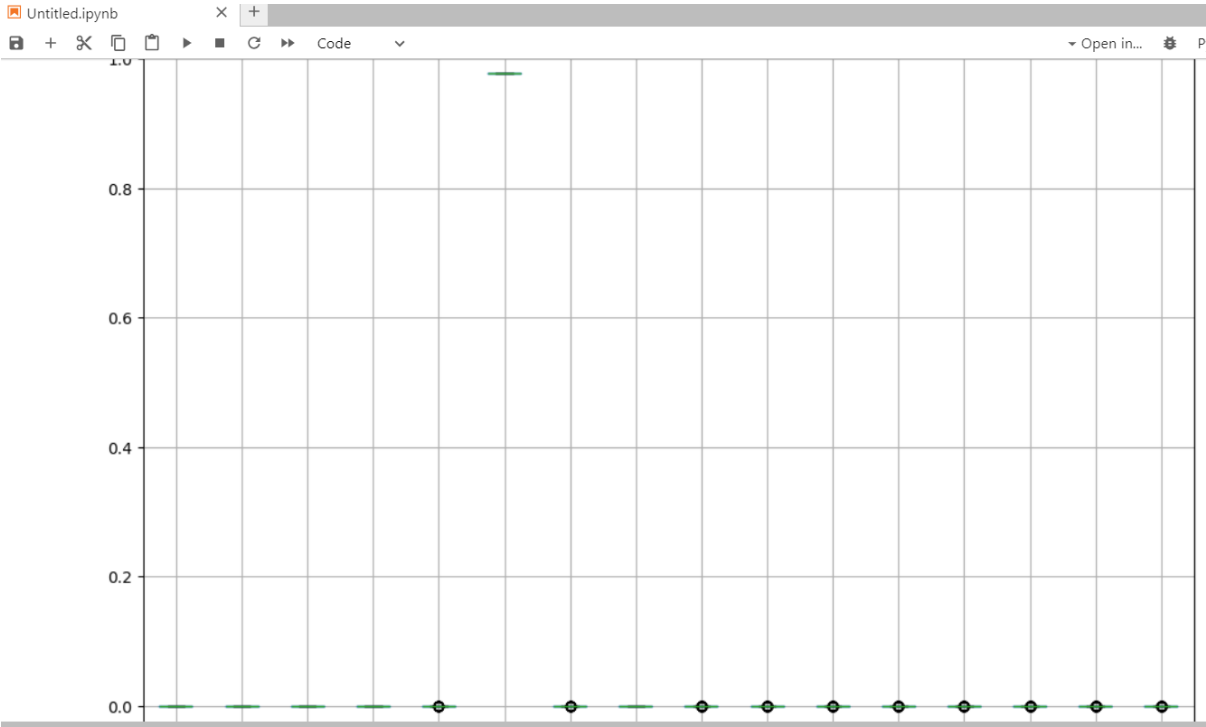
[0 rows x 23 columns]
```

**Seventh:** checking for outliers in each col then remove them through IQR method lets see before and after:

**Before:**



**After:**



Now lets filter the data to include harry potter books series only and do some analysis on it:

```
: # Filter the dataset to include only books from the Harry Potter series
harry_potter_books = df[df['title'].str.contains('Harry Potter', case=False)]
```

```
: # Now you have a DataFrame containing only the Harry Potter book series
```

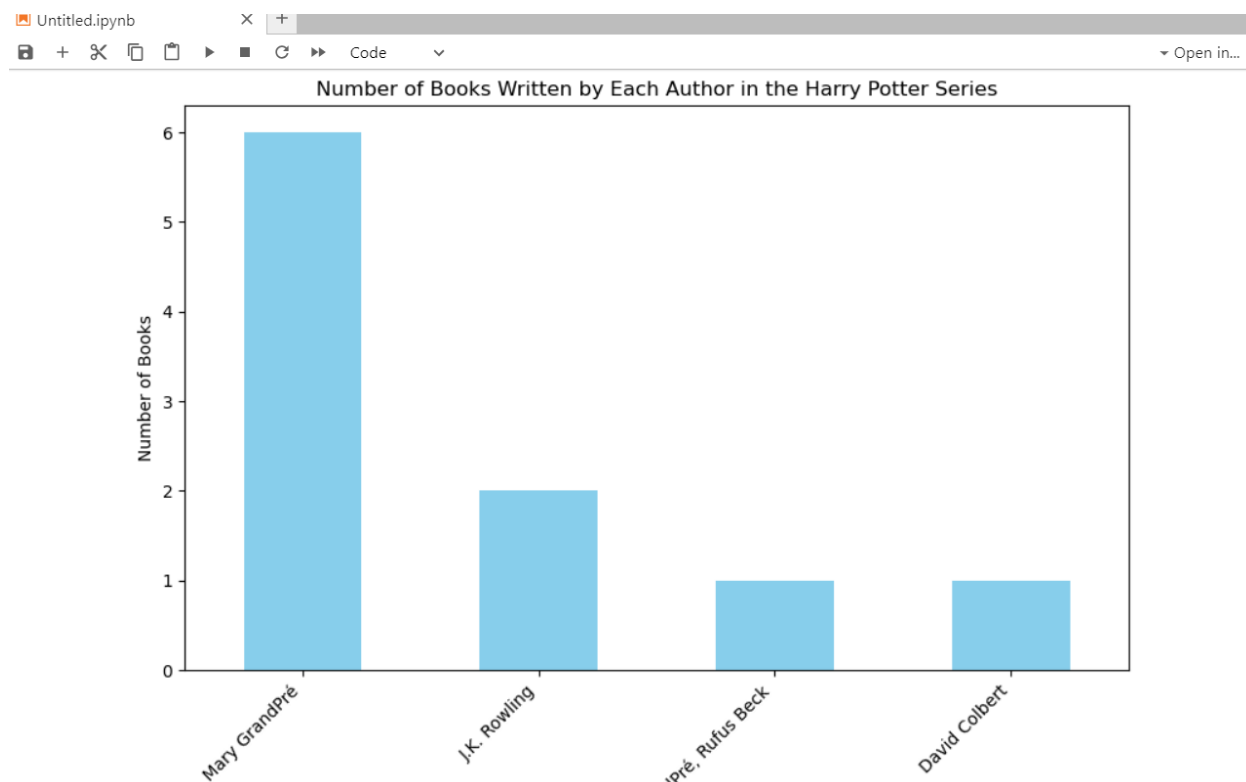
```
: print(harry_potter_books) #some details about harry potter book series
```

	book_id	goodreads_book_id	best_book_id	work_id	books_count \
1	2	3	3	4640799	491
6	18	5	5	2402163	376
8	21	2	2	2809203	307
9	23	15881	15881	6231171	398
10	24	6	6	3046572	332
11	25	136251	136251	2963218	263
12	27	1	1	41335427	275
96	422	862041	862041	2962492	76
613	3753	10	10	21457570	6
1036	7018	483445	483445	471792	42

	isbn	isbn13	authors \
1	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré
6	043965548X	9.780440e+12	J.K. Rowling, Mary GrandPré, Rufus Beck
8	439358078	9.780439e+12	J.K. Rowling, Mary GrandPré
9	439064864	9.780439e+12	J.K. Rowling, Mary GrandPré
10	439139600	9.780439e+12	J.K. Rowling, Mary GrandPré
11	545010225	9.780545e+12	J.K. Rowling, Mary GrandPré

**Analysis on authors :** I wanted to know how much books is written by each author so I did bar blot



**Analysis on ratings** the thing that the ratings were in form of like give rate 1,2,3,4,5.

	work_ratings_count	work_text_reviews_count	ratings_1	ratings_2	\
1	4800065	75867	75504	101676	
6	1969375	36099	6716	20413	
8	1840548	28685	9528	31577	
9	1906199	34172	8253	42251	
10	1868642	31084	6676	20210	
11	1847395	51942	9363	22245	
12	1785676	27520	7308	21516	
96	204125	6508	1105	1285	
613	26274	882	203	186	
1036	15145	267	329	1125	

As you see you will see that the first book has the most ratings so this means that it is the most selling :

```
# Sort the Harry Potter books by total ratings count in descending order
most_selling_books = harry_potter_books.sort_values(by='work_ratings_count', ascending=False)

# Print the most selling book within the Harry Potter series
most_selling_book = most_selling_books.iloc[0] # Assuming the first book in the sorted DataFrame has the highest ratings count
print("The most selling book within the Harry Potter series is:")
print(most_selling_book[['title', 'work_ratings_count']])

The most selling book within the Harry Potter series is:
title                Harry Potter and the Sorcerer's Stone (Harry P...
work_ratings_count              4800065
Name: 1, dtype: object
```

Now we want to know the average rating for each book but as I told you there is five rating columns so will create a new column called average rating that have average rating for each type rating then I will get the average of the overall averages:

```
# Calculate the average rating for each book
harry_potter_books['average_rating'] = (harry_potter_books['ratings_1'] * 1 +
                                         harry_potter_books['ratings_2'] * 2 +
                                         harry_potter_books['ratings_3'] * 3 +
                                         harry_potter_books['ratings_4'] * 4 +
                                         harry_potter_books['ratings_5'] * 5) / harry_potter_books['work_ratings_count']

# Calculate the overall average rating of the Harry Potter series
overall_average_rating = harry_potter_books['average_rating'].mean()
print("The average rating of the Harry Potter books series is:", overall_average_rating)

The average rating of the Harry Potter books series is: 4.490890609929265
```