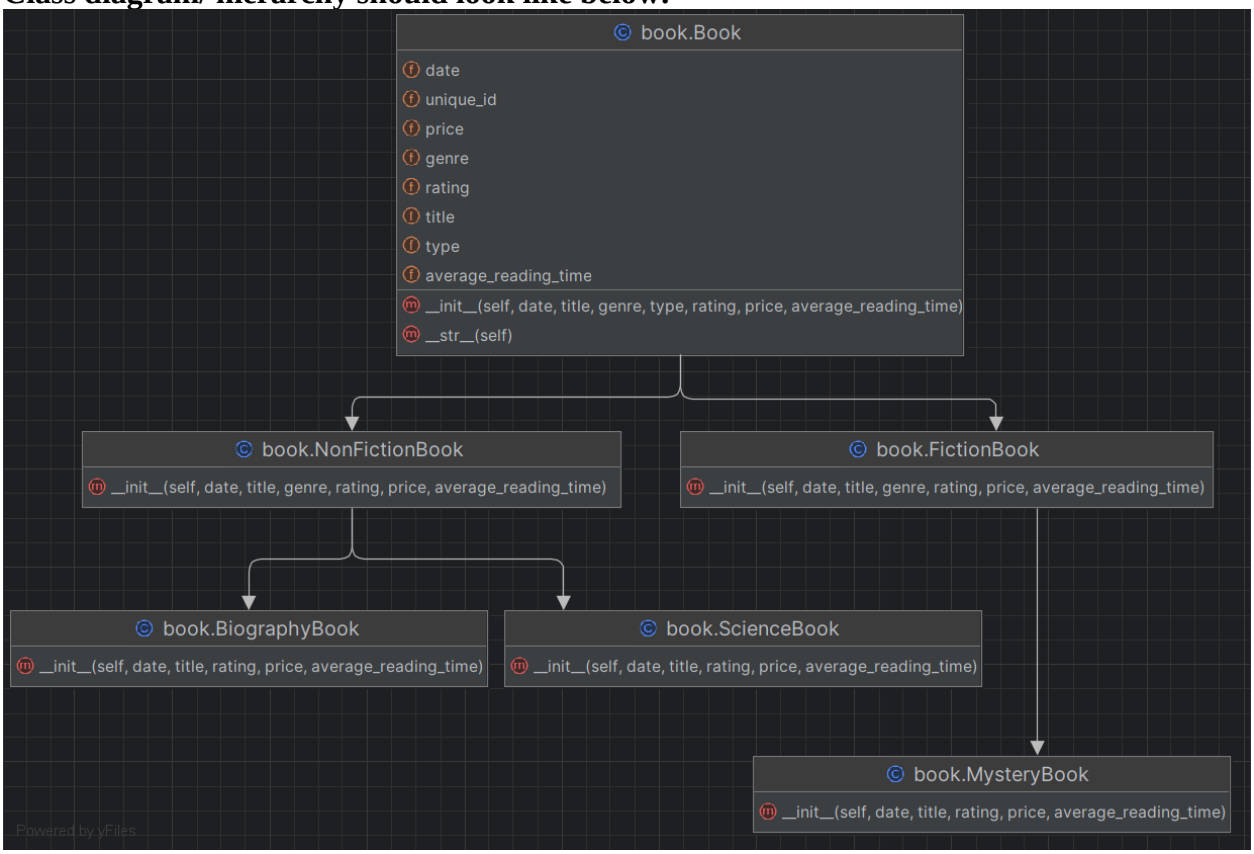
Appendix:
Follow the instructions carefully to make sure the dataset can be loaded properly:

Scenario #7: Data about Books in a library from their catalog.

1. Create a class called Book with the following attributes in its constructor (__init__ method) **in this order**:
    a. date: Date on which a particular book is added to the catalog.
    b. title: Title of the book.
    c. genre: Book's genre.
    d. type: Type of the book.
    e. rating: Customers's rating for the book.
    f. price: Price of the book (in USD).
    g. average_reading_time: Average time required to read the book (in hours).
2. In the constructor, add another attribute **unique_id** (Unique Identifier) and set the attribute's value to **id(self)** – the memory id of the instance of the class.
3. Implement the __str__ method within the Book class to return a formatted string containing the **values** of all the attributes in this format: "unique_id,date,title,genre,type,rating,price,average_reading_time"(no spaces)
4. Create two subclasses: FictionBook and NonFictionBook, both inheriting from the Book class
5. The constructors in the FictionBook and NonFictionBook classes should:
    a. Take input **in this order**: date,title,genre,rating,price,average_reading_time
    b. Use super() to call the constructor of the parent Book class, passing the type attribute as "Fiction" for FictionBook and "Non-Fiction " for NonFictionBook with rest of the input.
6. Create three additional subclasses: MysteryBook, ScienceBook, BiographyBook, inheriting from FictionBook and NonFictionBook, NonFictionBook respectively.
7. The constructors in the MysteryBook, ScienceBook, and BiographyBook classes should:
    a. Take input **in this order**: date,title,rating,price, average_reading_time
    b. Set the genre as class name (i.e. genre = " ScienceBook " if class is ScienceBook).
    c. Use super() to call the constructor of the parent class with the rest of the input.
8. Once the classes are ready, test your code with the examples shown below:
    a. ScienceBookBook= ScienceBook ("2024-10-16","The Selfish Gene", 3.0, 69.19, 38.00)
    b. print(str(ScienceBookBook))
    c. should return "**[unique_id],2024-10-16, The Selfish Gene,ScienceBook,Non-Fiction,3.0,69.19,38.00**"
9. Once that is verified, load the file provided in this link.
10. Create a csv file with the steps below:
    a. Write a first line that denotes the column headers as "unique_id,date,title,genre,type,rating,price,average_reading_time"(no spaces)
    b. Loop through all the objects retrieved from the above pickle file and use str method to print formatted string as mentioned in step 3. Refer to snippet of code provided below.
11. Use the above generated csv file to create visualizations in Python.

Class diagram/ hierarchy should look like below:



Snippet for step 10:

```
with open('data.csv', 'w') as f:
    f.write("unique_id,date,title,genre,type,rating,price,average_reading_time\n")
    for obj in objects:
        f.write(str(obj)+'\n')
```