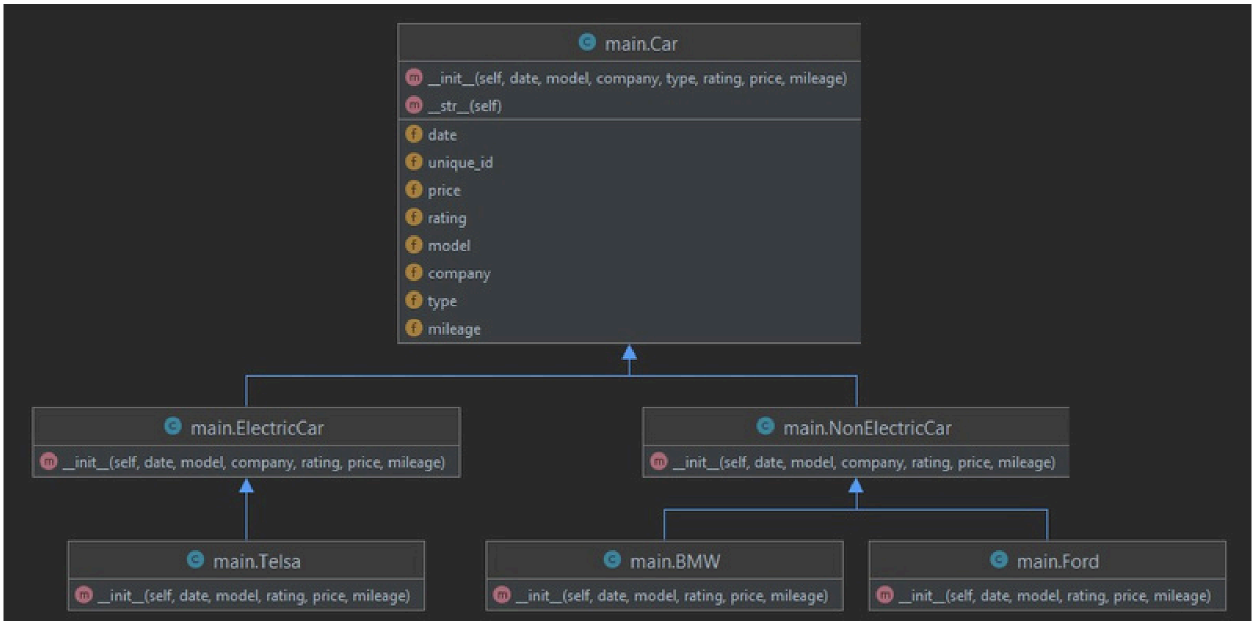
Project Instruction

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

Scenario #1: Data about Cars from an auction website

1. Create a class called Car with the following attributes in its constructor (__init__ method) **in this order:**
   a. date: Manufactured date.
   b. model: Car's model name.
   c. company: Company/Manufacturer name.
   d. type: Car's type.
   e. rating: Car's rating on the auction website.
   f. price: Car's price (in USD).
   g. mileage: Car's mileage (in miles).
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 Car class to return a formatted string containing the **values** of all the attributes in this format: "unique_id,date,model,company,type,rating,price,mileage" (no spaces)
4. Create two subclasses: ElectricCar and NonElectricCar, both inheriting from the Car class
5. The constructors in the ElectricCar and NonElectricCar classes should:
   a. Take input **in this order:** date,model,company,rating,price,mileage
   b. Use super() to call the constructor of the parent Car class, passing the type attribute as "Electric" for ElectricCar and "non-Electric" for NonElectricCar with rest of the input.
6. Create three additional subclasses: Telsa, Ford, BMW, inheriting from ElectricCar and NonElectricCar, NonElectricCar respectively.
7. The constructors in the Telsa, Ford, and BMW classes should:
   a. Take input **in this order:** date,model,rating,price,mileage
   b. Set the company as class name (i.e. company = "Ford" if class is Ford).
   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. FordCar = Ford("2022-11-15","modelA", 2, 641, 86.06)
   b. print(str(FordCar))
   c. should return "**[unique_id],2022-11-15, modelA,Ford,NonElectric,2,641,86.06**"
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,model,company,type,rating,price,mileage" (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:


main.Car
__init__(self, date, model, company, type, rating, price, mileage)
__str__(self)
date
unique_id
price
rating
model
company
type
mileage
main.ElectricCar
__init__(self, date, model, company, rating, price, mileage)
main.NonElectricCar
__init__(self, date, model, company, rating, price, mileage)
main.Telsa
__init__(self, date, model, rating, price, mileage)
main.BMW
__init__(self, date, model, rating, price, mileage)
main.Ford
__init__(self, date, model, rating, price, mileage)

Snippet for step 10:

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