Challenging Task-2

T.Akash Reddy

21MIS1128

Question-4

Aim:

To analyze the mercedes.csv dataset to identify the following:

- Cars from the year 2023 to 2024 with the highest ratings.
- Unique cars with a rating greater than 4.
- Mercedes Benz model cars with the highest price in the year 2023.

Procedure:

1. Data Loading and Preprocessing:

    - Load the mercedes.csv dataset into a pandas DataFrame.
    - Extract the year from the 'Name' column using regular expressions and convert it to an integer type.
    - Clean the 'Price' column by removing dollar signs and commas, then convert it to a numeric type, handling "Not Priced" values by replacing them with NaN.

2. Identifying Highest Rated Cars (2023-2024):

    - Filter the DataFrame to select cars from the years 2023 to 2024.
    - Find the highest rating among these cars.
    - Filter the DataFrame again to select only the cars with the highest rating.
    - Extract and display the names of these cars.

3. Listing Unique Cars with Rating > 4:

    - Filter the DataFrame to select cars with a rating greater than 4.
    - Extract the unique car names from this filtered DataFrame.

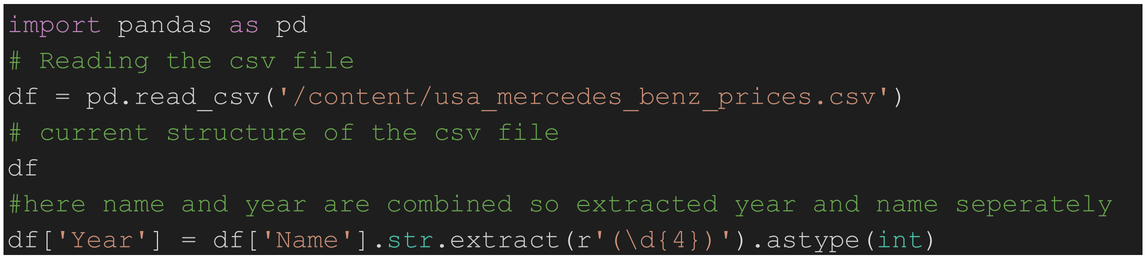
- Display the list of unique cars.

4. Finding Highest Priced Cars in 2023:

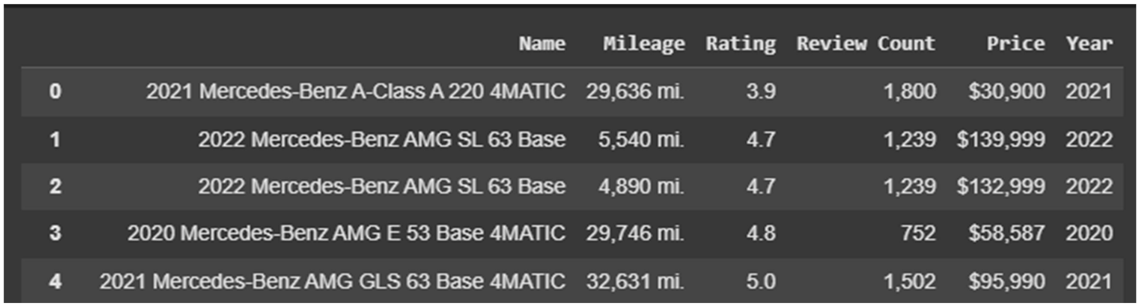
   - Filter the DataFrame to select cars from the year 2023 and with non-missing 'Price' values.
   - Find the highest price among these cars.
   - Filter the DataFrame again to select only the cars with the highest price.
   - Extract and display the names of these cars.

Find out the cars in the year 2023 to 2024 having highest ratings

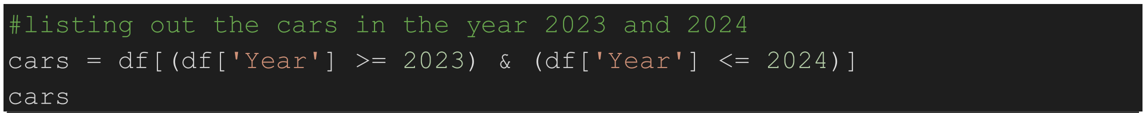
Code:



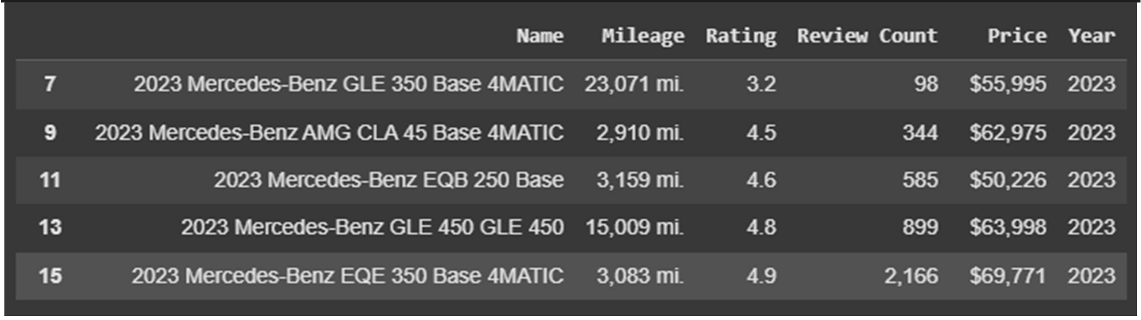
```python
import pandas as pd
# Reading the csv file
df = pd.read_csv('/content/usa_mercedes_benz_prices.csv')
# current structure of the csv file
df
#here name and year are combined so extracted year and name seperately
df['Year'] = df['Name'].str.extract(r'(\d{4})').astype(int)
```



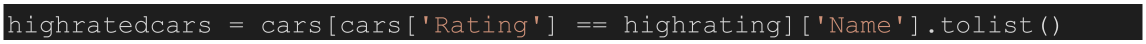
	Name	Mileage	Rating	Review Count	Price	Year
0	2021 Mercedes-Benz A-Class A 220 4MATIC	29,636 mi.	3.9	1,800	$30,900	2021
1	2022 Mercedes-Benz AMG SL 63 Base	5,540 mi.	4.7	1,239	$139,999	2022
2	2022 Mercedes-Benz AMG SL 63 Base	4,890 mi.	4.7	1,239	$132,999	2022
3	2020 Mercedes-Benz AMG E 53 Base 4MATIC	29,746 mi.	4.8	752	$58,587	2020
4	2021 Mercedes-Benz AMG GLS 63 Base 4MATIC	32,631 mi.	5.0	1,502	$95,990	2021



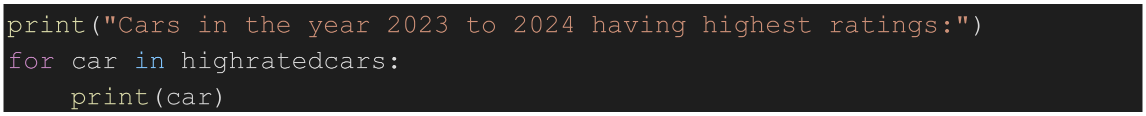
```python
#listing out the cars in the year 2023 and 2024
cars = df[(df['Year'] >= 2023) & (df['Year'] <= 2024)]
cars
```



	Name	Mileage	Rating	Review Count	Price	Year
7	2023 Mercedes-Benz GLE 350 Base 4MATIC	23,071 mi.	3.2	98	$55,995	2023
9	2023 Mercedes-Benz AMG CLA 45 Base 4MATIC	2,910 mi.	4.5	344	$62,975	2023
11	2023 Mercedes-Benz EQB 250 Base	3,159 mi.	4.6	585	$50,226	2023
13	2023 Mercedes-Benz GLE 450 GLE 450	15,009 mi.	4.8	899	$63,998	2023
15	2023 Mercedes-Benz EQE 350 Base 4MATIC	3,083 mi.	4.9	2,166	$69,771	2023

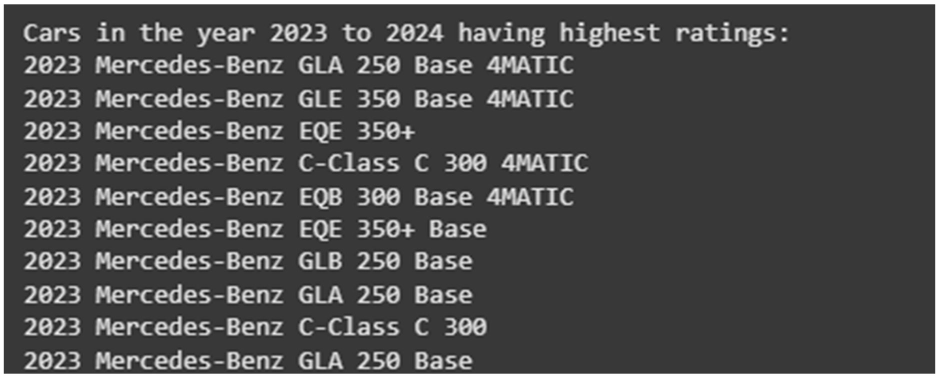


```python
highratedcars = cars[cars['Rating'] == highrating]['Name'].tolist()
```



```
print("Cars in the year 2023 to 2024 having highest ratings:")
for car in highratedcars:
    print(car)
```

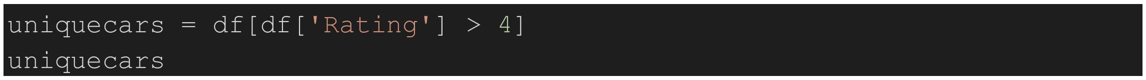
output:



```
Cars in the year 2023 to 2024 having highest ratings:
2023 Mercedes-Benz GLA 250 Base 4MATIC
2023 Mercedes-Benz GLE 350 Base 4MATIC
2023 Mercedes-Benz EQE 350+
2023 Mercedes-Benz C-Class C 300 4MATIC
2023 Mercedes-Benz EQB 300 Base 4MATIC
2023 Mercedes-Benz EQE 350+ Base
2023 Mercedes-Benz GLB 250 Base
2023 Mercedes-Benz GLA 250 Base
2023 Mercedes-Benz C-Class C 300
2023 Mercedes-Benz GLA 250 Base
```

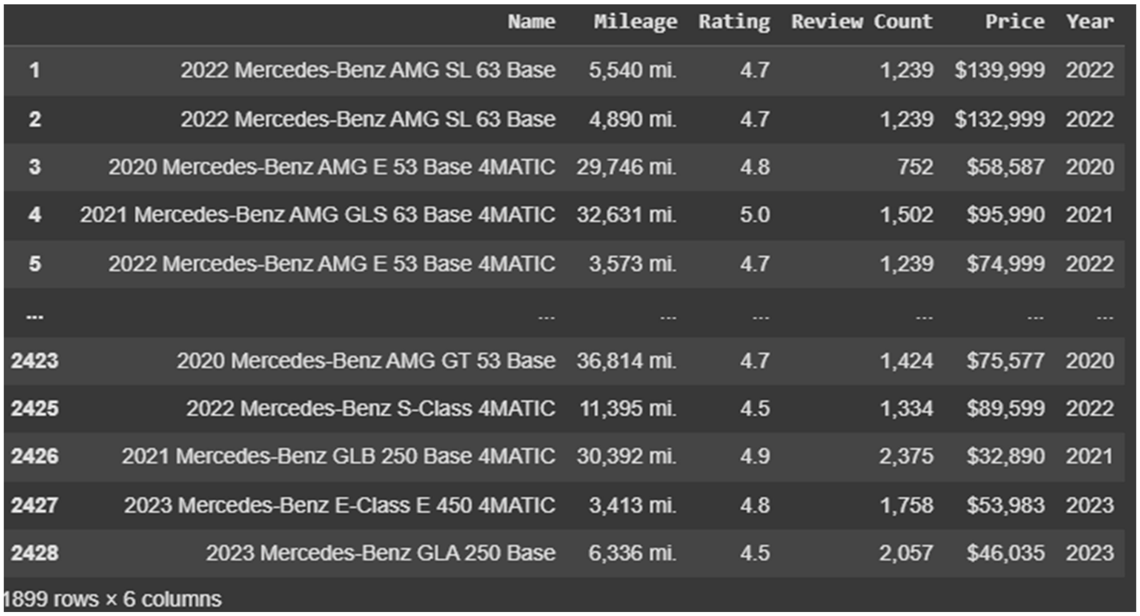
List out the unique cars whose rate is more than 4

<u>Code:</u>



```
uniquecars = df[df['Rating'] > 4]
uniquecars
```

output:

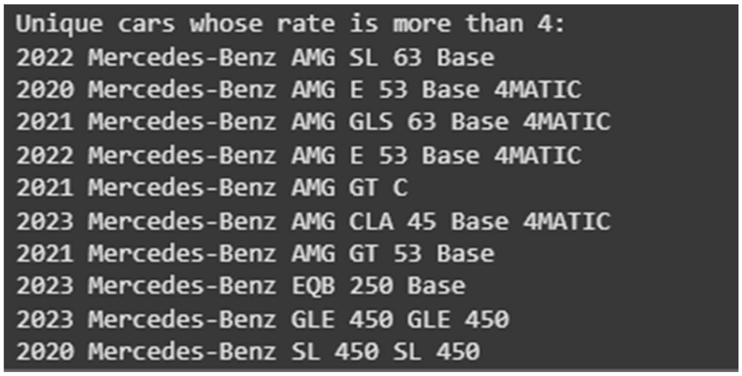


	Name	Mileage	Rating	Review Count	Price	Year
1	2022 Mercedes-Benz AMG SL 63 Base	5,540 mi.	4.7	1,239	$139,999	2022
2	2022 Mercedes-Benz AMG SL 63 Base	4,890 mi.	4.7	1,239	$132,999	2022
3	2020 Mercedes-Benz AMG E 53 Base 4MATIC	29,746 mi.	4.8	752	$58,587	2020
4	2021 Mercedes-Benz AMG GLS 63 Base 4MATIC	32,631 mi.	5.0	1,502	$95,990	2021
5	2022 Mercedes-Benz AMG E 53 Base 4MATIC	3,573 mi.	4.7	1,239	$74,999	2022
...	...	...	...	...	...	...
2423	2020 Mercedes-Benz AMG GT 53 Base	36,814 mi.	4.7	1,424	$75,577	2020
2425	2022 Mercedes-Benz S-Class 4MATIC	11,395 mi.	4.5	1,334	$89,599	2022
2426	2021 Mercedes-Benz GLB 250 Base 4MATIC	30,392 mi.	4.9	2,375	$32,890	2021
2427	2023 Mercedes-Benz E-Class E 450 4MATIC	3,413 mi.	4.8	1,758	$53,983	2023
2428	2023 Mercedes-Benz GLA 250 Base	6,336 mi.	4.5	2,057	$46,035	2023

1899 rows × 6 columns

```
unique_cars = uniquecars['Name'].unique()
print("Unique cars whose rate is more than 4:")
for car in unique_cars:
    print(car)
```

output:

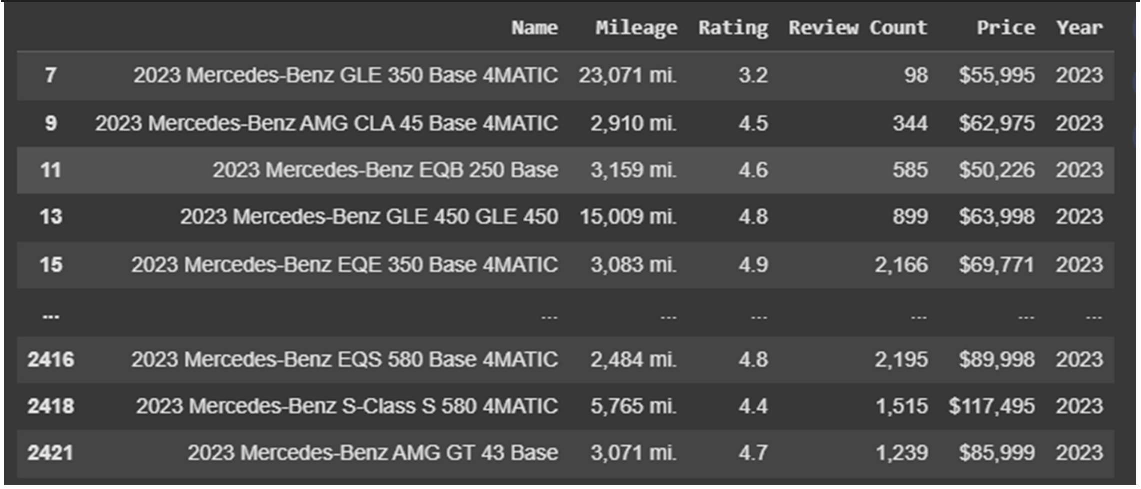


```
Unique cars whose rate is more than 4:
2022 Mercedes-Benz AMG SL 63 Base
2020 Mercedes-Benz AMG E 53 Base 4MATIC
2021 Mercedes-Benz AMG GLS 63 Base 4MATIC
2022 Mercedes-Benz AMG E 53 Base 4MATIC
2021 Mercedes-Benz AMG GT C
2023 Mercedes-Benz AMG CLA 45 Base 4MATIC
2021 Mercedes-Benz AMG GT 53 Base
2023 Mercedes-Benz EQB 250 Base
2023 Mercedes-Benz GLE 450 GLE 450
2020 Mercedes-Benz SL 450 SL 450
```

List out the Mercedes Benz model cars whose price is highest in the year 2023

<u>Code:</u>

```
#creating a data frame of the year 2023
df2023 = df[df['Year'] == 2023]
df2023
```



	Name	Mileage	Rating	Review Count	Price	Year
7	2023 Mercedes-Benz GLE 350 Base 4MATIC	23,071 mi.	3.2	98	$55,995	2023
9	2023 Mercedes-Benz AMG CLA 45 Base 4MATIC	2,910 mi.	4.5	344	$62,975	2023
11	2023 Mercedes-Benz EQB 250 Base	3,159 mi.	4.6	585	$50,226	2023
13	2023 Mercedes-Benz GLE 450 GLE 450	15,009 mi.	4.8	899	$63,998	2023
15	2023 Mercedes-Benz EQE 350 Base 4MATIC	3,083 mi.	4.9	2,166	$69,771	2023
...	...	...	...	...	...	...
2416	2023 Mercedes-Benz EQS 580 Base 4MATIC	2,484 mi.	4.8	2,195	$89,998	2023
2418	2023 Mercedes-Benz S-Class S 580 4MATIC	5,765 mi.	4.4	1,515	$117,495	2023
2421	2023 Mercedes-Benz AMG GT 43 Base	3,071 mi.	4.7	1,239	$85,999	2023

```
#there is a dollar symbol and comma and also a string "not priced in
price column replaced them"
df2023['Price'] = pd.to_numeric(df2023['Price'].str.replace('[\$,]',
'', regex=True), errors='coerce')
```

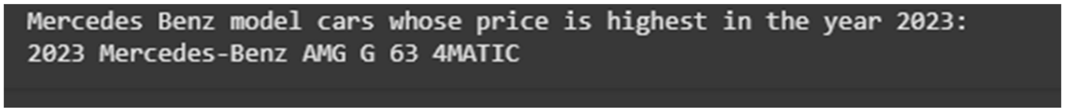
```
df_2023 = df2023[(df2023['Year'] == 2023) & (df2023['Price'].notna())]

hprice = df_2023['Price'].max()

highpricecars = df_2023[df_2023['Price'] == hprice]['Name'].tolist()

print("Mercedes Benz model cars whose price is highest in the year 2023:")
for car in highpricecars:
    print(car)
```

output:



```
Mercedes Benz model cars whose price is highest in the year 2023:
2023 Mercedes-Benz AMG G 63 4MATIC
```