IoT Challenging Task 2

Suryakumar P 21MIS1146

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**QN. 2:**

**Aim**

To analyze the correlation between various parameters (Price, Mileage, Rating, Review Count) affecting the price of used Mercedes-Benz cars, identify the most influential factors, and determine the best-selling car models between 2020 and 2023.

**Procedure**

**1. Data Preprocessing:**

- Load the dataset of used Mercedes-Benz car prices.

- Clean the data:

- Remove currency symbols ('$'), commas, and unit suffixes ('mi.') from relevant columns.

- Handle missing or invalid values (like 'Not Priced') in the 'Price' column by replacing with NaN (Not a Number).

- Convert 'Price' and 'Mileage' to numeric data types. Invalid entries that cannot be converted to numbers will become NaN.

- Extract the year of manufacture from the car name.

- Convert the extracted 'Year' column to numeric format, handling any non-numeric values with NaN.

**2. Correlation Analysis:**

- Calculate the correlation matrix among 'Price', 'Mileage', 'Rating', and 'Review Count'

- Visualize the correlation matrix as a heatmap using Seaborn.

- Analyze the correlation coefficients to determine the relationships between these variables. Positive correlation implies that the variables tend to move in the same direction (increase or decrease together). Negative correlation indicates the variables tend to move in opposite directions. Coefficients closer to +1 or -1 indicate stronger relationships.

**3. Identify Important Parameters:**

- Based on the correlation analysis, identify which parameters have the strongest correlation with 'Price'.

- Provide explanations for why these parameters might influence the car price. For example, mileage is expected to have a negative correlation with price, while the rating and number of reviews might have a positive correlation.

**4. Best-Selling Cars (2020-2023):**

- Filter the dataset to include cars sold between 2020 and 2023.

- Group the filtered data by the car model ('Name').

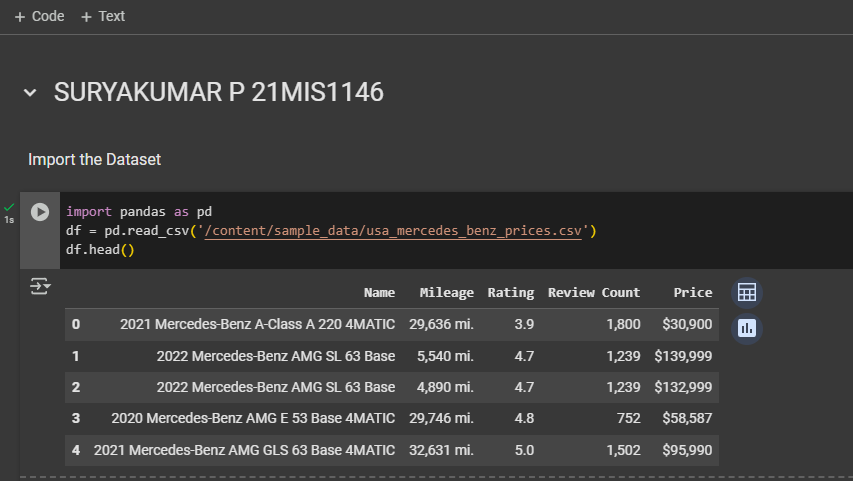
- Count the number of sales for each car model within the specified period. This effectively sums the number of instances each car appears in the dataframe.

- Sort the car models by the number of sales in descending order to identify the top-selling cars.

- Display the top-selling car models.

**CODE AND OUTPUT:**

1. **Import the Dataset:**



1. **Preprocess the data:**

import pandas as pd

Remove commas in float values

for col in ['Price', 'Mileage', 'Review Count']:

df[col] = df[col].astype(str).str.replace(',', '')

Remove 'mi.' from Mileage column

df['Mileage'] = df['Mileage'].astype(str).str.replace(' mi.', '')

Remove '$' from Price column

df['Price'] = df['Price'].astype(str).str.replace('$', '')

Handle 'Not Priced' values in Price column

df['Price'] = df['Price'].replace('Not Priced', pd.NA)

Convert Price and Mileage to numeric, coercing errors to NaN

df['Price'] = pd.to\_numeric(df['Price'], errors='coerce')

df['Mileage'] = pd.to\_numeric(df['Mileage'], errors='coerce')

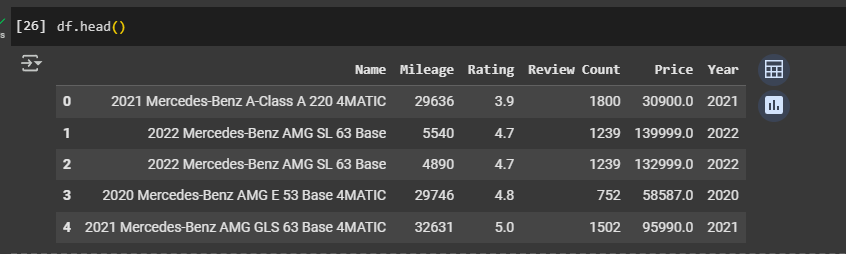
Extract 'Year' from the 'Name' column

df['Year'] = df['Name'].str.split().str[0]

Convert 'Year' to numeric, coercing errors to NaN

df['Year'] = pd.to\_numeric(df['Year'], errors='coerce')

Output:



**QN1: Perform the correlation among the parameters**

import matplotlib.pyplot as plt

import seaborn as sns

Calculate the correlation matrix

corr\_matrix = df[['Price', 'Mileage', 'Rating', 'Review Count']].corr()

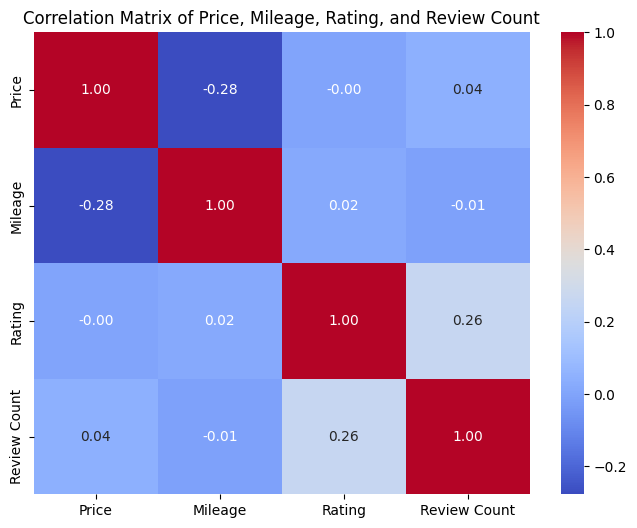
Visualize the correlation matrix using a heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix of Price, Mileage, Rating, and Review Count')

plt.show()

Output:  


**QN2: Identify the important parameters with explanation**

1. **Price**: This is the target variable, the dependent variable we're trying to understand or predict. Its correlation with other features is crucial.

2. **Mileage**: Represents the car's mileage. We expect it to be negatively correlated with price (higher mileage, lower price). The code specifically handles preprocessing of the mileage data to convert it into a numeric value for analysis.

3. **Rating**: Likely represents the car's rating (e.g. customer reviews). A positive correlation with price would be expected (higher rating, higher price).

4. **Review Count**: Number of reviews the car has received. May or may not have a correlation with the price. A large number of reviews could indicate popularity, but doesn't guarantee a high price.

The code calculates and visualizes the correlation matrix. The heatmap shows how strongly these parameters are related. The annotations show the correlation coefficients which help to numerically determine the strength of the relationship between the variables. A correlation coefficient of +1 indicates a perfect positive correlation, 0 indicates no correlation and -1 indicates a perfect negative correlation. The focus is on analyzing the impact of mileage, rating, and review count on the car's price.

**QN3: List out the cars which sold more between 2020 to 2023.**

**import pandas as pd**

**# Filter data for the years between 2020 and 2023 (inclusive)**

**df\_filtered = df[(df['Year'] >= 2020) & (df['Year'] <= 2023)]**

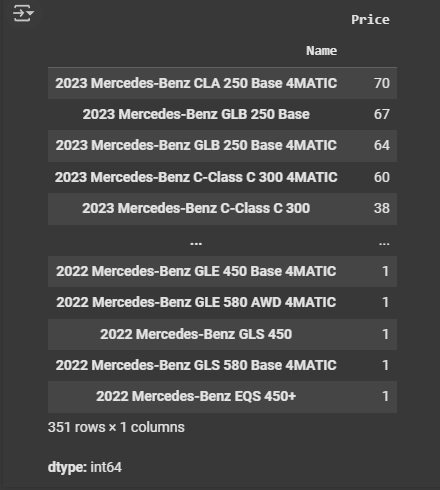
**# Group by car name and sum the sales (assuming each row represents one sale)**

**car\_sales = df\_filtered.groupby('Name')['Price'].count().sort\_values(ascending=False)**

**# Display the top cars**

**car\_sales**

Output:



COLAB NOTEBOOK LINK: [IoT-ChallengingTask2\_21MIS1146.ipynb](https://colab.research.google.com/drive/1CWcawgu6Rgck9rVyRQA42pgVVwCk0mKt?usp=sharing)