**Name** : Satyaprakash Gupta **Email id** : satyaprakashgupta9975@gmail.com

```
1 !pip install q kaggle
```

```
Collecting q
  Downloading q-2.7-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages
Installing collected packages: q
Successfully installed q-2.7
```

```
1 from google.colab import files
2 import pandas as pd
3 import numpy as np
4 from google.colab import autoviz
5 import seaborn as sns
```

```
1 files.upload()
```

```
Choose Files   No file chosen         Upload widget is only available when the cell has been
executed in the current browser session. Please rerun this cell to enable.
Saving kaggle.json to kaggle.json
{'kaggle.json':
```

```
1 !mkdir ~/.kaggle1
```

```
1 !cp kaggle.json ~/.kaggle/
```

```
1 ! chmod 600 ~/.kaggle/kaggle.json
```

```
1 ! kaggle datasets list
```

```
ref                                              title
-----------------------------------------------  --------------------------
iamsouravbanerjee/customer-shopping-trends-dataset   Customer Shopping Trends D
nelgiriyewithana/top-spotify-songs-2023              Most Streamed Spotify Song
bkcoban/customer-transactions                        Customer Transactions
yakhyojon/tiktok                                     TikTok User Engagement Dat
nelgiriyewithana/billionaires-statistics-dataset     Billionaires Statistics Da
nelgiriyewithana/credit-card-fraud-detection-dataset-2023   Credit Card Fraud Detectic
muhammadawaistayyab/amazon-movies-and-films          Amazon- Movies and Films
anshtanwar/monthly-food-price-estimates              Global Food Price Inflation            44MB   2023-10-08 05:22
redpen12/mobile-phone-market-in-ghana                Mobile Phone Market in Ghana           49KB   2023-10-18 10:33
teamincribo/cyber-security-attacks                   Cyber Security Attacks                 5MB    2023-10-14 14:12
iamsouravbanerjee/heart-attack-prediction-dataset    Heart Attack Risk Prediction Dataset   519KB  2023-09-27 07:07
rajatsurana979/hotel-reservation-data-repository     Hotel Reservation Data Repository      4MB    2023-10-17 04:02
willianoliveiragibin/electric-vehicle-population     Electric Vehicle Population            5MB    2023-10-13 20:08
iamsouravbanerjee/cars-dataset                       Cars Dataset                          571KB   2023-10-17 05:58
sujaykapadnis/horror-movies-profits-dataset          Horror Movies - the best Hollywood Investments?  102KB  2023-10-14 04:09
willianoliveiragibin/10000-data-about-movies-1915-2023   10,000 Data  about movies (1915 - 2023)  1MB   2023-10-11 19:31
sujaykapadnis/spotify-songs                          spotify songs                          3MB    2023-10-17 09:17
willianoliveiragibin/fatalities-in-the-israeli-palestinian   Fatalities in the Israeli-Palestinian  463KB  2023-10-10 19:02
asaniczka/wages-by-education-in-the-usa-1973-2022    Wages by Education in the USA (1973-2022)  7KB   2023-10-09 09:43
uom190346a/health-and-demographics-dataset           Health and Demographics Dataset        75KB   2023-10-18 07:53
```

```
1 !kaggle datasets download -d hhs/health-insurance-marketplace
```

```
Downloading health-insurance-marketplace.zip to /content
 99% 817M/829M [00:11<00:00, 105MB/s]
100% 829M/829M [00:11<00:00, 76.9MB/s]
```

```
1 !unzip health-insurance-marketplace.zip -d data/
```

```
Archive:  health-insurance-marketplace.zip
  inflating: data/BenefitsCostSharing.csv
  inflating: data/BusinessRules.csv
  inflating: data/Crosswalk2015.csv
  inflating: data/Crosswalk2016.csv
```

```
      inflating: data/Network.csv
      inflating: data/PlanAttributes.csv
      inflating: data/Rate.csv
      inflating: data/ServiceArea.csv
      inflating: data/database.sqlite
      inflating: data/hashes.txt
      inflating: data/raw/._.sentinel
      inflating: data/raw/._2014
      inflating: data/raw/._2015
      inflating: data/raw/._2016
      inflating: data/raw/._Benefits_Cost_Sharing_PUF.csv
      inflating: data/raw/._Benefits_Cost_Sharing_PUF_2015-12-08.csv
      inflating: data/raw/._Business_Rules_PUF.csv
      inflating: data/raw/._Business_Rules_PUF_2015-12-08.csv
      inflating: data/raw/._Business_Rules_PUF_Reformat.csv
      inflating: data/raw/._Machine_Readable_PUF_2015-12-21.xlsx
      inflating: data/raw/._Network_PUF.csv
      inflating: data/raw/._Network_PUF_2015-12-08.csv
      inflating: data/raw/._Plan_Attributes_PUF.csv
      inflating: data/raw/._Plan_Attributes_PUF_2014_2015-03-09.csv
      inflating: data/raw/._Plan_Attributes_PUF_2015-12-08.csv
      inflating: data/raw/._Plan_Crosswalk_PUF_2014-12-22.csv
      inflating: data/raw/._Plan_ID_Crosswalk_PUF_2015-12-07.CSV
      inflating: data/raw/._Rate_PUF.csv
      inflating: data/raw/._Rate_PUF_2015-12-08.csv
      inflating: data/raw/._ServiceArea_PUF_2015-12-08.csv
      inflating: data/raw/._Service_Area_PUF.csv
      inflating: data/raw/.sentinel
      inflating: data/raw/2014/.sentinel
      inflating: data/raw/2014/Benefits_Cost_Sharing_PUF.csv
      inflating: data/raw/2014/Business_Rules_PUF.csv
      inflating: data/raw/2014/Network_PUF.csv
      inflating: data/raw/2014/Plan_Attributes_PUF_2014_2015-03-09.csv
      inflating: data/raw/2014/Rate_PUF.csv
      inflating: data/raw/2014/Service_Area_PUF.csv
      inflating: data/raw/2015/.sentinel
      inflating: data/raw/2015/Benefits_Cost_Sharing_PUF.csv
      inflating: data/raw/2015/Business_Rules_PUF_Reformat.csv
      inflating: data/raw/2015/Network_PUF.csv
      inflating: data/raw/2015/Plan_Attributes_PUF.csv
      inflating: data/raw/2015/Plan_Crosswalk_PUF_2014-12-22.csv
      inflating: data/raw/2015/Rate_PUF.csv
      inflating: data/raw/2015/Service_Area_PUF.csv
      inflating: data/raw/2016/.sentinel
      inflating: data/raw/2016/Benefits_Cost_Sharing_PUF_2015-12-08.csv
      inflating: data/raw/2016/Business_Rules_PUF_2015-12-08.csv
      inflating: data/raw/2016/Machine_Readable_PUF_2015-12-21.xlsx
      inflating: data/raw/2016/Network_PUF_2015-12-08.csv
      inflating: data/raw/2016/Plan_Attributes_PUF_2015-12-08.csv
      inflating: data/raw/2016/Plan_ID_Crosswalk_PUF_2015-12-07.CSV
      inflating: data/raw/2016/Rate_PUF_2015-12-08.csv
      inflating: data/raw/2016/ServiceArea_PUF_2015-12-08.csv
      inflating: data/raw/Benefits_Cost_Sharing_PUF.csv
```

```python
1 display(dbutils.fs.ls('file:/databricks/driver/data/'))
```

```python
1 source = 'file:/databricks/driver/data/'
2 target = '/mnt/data/'
3
4 dbutils.fs.cp(source, target, recurse=True)
```

```python
1 dbutils.fs.refreshMounts()
```

```python
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import when, col, mean, round, expr
3 spark = SparkSession.builder.appName("health").getOrCreate()
```

```python
1 data = spark.read.csv("/mnt/data/Rate.csv", header=True, inferSchema=True)
2 data = data.select("Year", "InsurancePlanID", "PolicyAge", "PolicyRate")
3
4 data_filtered = data.filter(col('PolicyAge') != 'Family Option')
5
6 condition_child = (data_filtered['PolicyAge'] == '0-20')
7 condition_adult = (data_filtered['PolicyAge'] != '0-20')
8
9 data_filtered = data_filtered.withColumn('AgeGroup', when(condition_child, 'Young').when(condition_adult, 'Older').otherwise(None))
10
11 result = data_filtered.groupBy('Year', 'InsurancePlanID', 'AgeGroup') \
12                 .agg(round(mean('PolicyRate'), 2).alias('AverageRate')) \
13                 .fillna({'AgeGroup': 'All'}).fillna({'AverageRate': 0})
14
15 result.show()
```

```python
1 import re
2 from pyspark.sql.types import StringType
```

```python
1 def categorize_age_group(text_data):
2     child_match_found = re.search(r'Child', text_data, re.IGNORECASE)
3     adult_match_found = re.search(r'Adult', text_data, re.IGNORECASE)
4
5     if child_match_found and not adult_match_found:
6         return 'AgeGroup: Child'
7     elif adult_match_found and not child_match_found:
8         return 'AgeGroup: Adult'
9     else:
10         return 'AgeGroup: Unknown'
11
12 insurance_records = spark.read.csv('/mnt/data/BenefitsCostSharing.csv', header=True, in
13 insurance_records = insurance_records.select("CoverageBenefitName", "PolicyYear", "Poli
14
15 insurance_records = insurance_records.withColumn("PolicyIdentifier", expr("substring(Po
16
17 categorize_age_group_udf = udf(categorize_age_group, StringType())
18 insurance_records = insurance_records.withColumn("AgeGroupCategory", categorize_age_grou
19 insurance_records = insurance_records.na.drop()
20
21 merged_insurance_data = insurance_records.join(rate_pivot, on=['PolicyYear', 'Poli
22 merged_insurance_data = merged_insurance_data.withColumnRenamed("AverageRate", "Po
23 merged_insurance_data = merged_insurance_data.fillna(0, subset=['PolicyAverageRate
24
25 merged_insurance_data.show()
```

```python
1 !mkdir transformed
```

```python
1 final_data = merged_data.repartition(1)
2 final_data.write.csv('file:/databricks/driver/transformed/final.csv', header=True, mode
```

```python
1 display(dbutils.fs.ls('file:/databricks/driver/transformed/final.csv/'))
```

```python
1 source = 'file:/databricks/driver/transformed/'
2 target = '/mnt/transformed/'
3
4 dbutils.fs.cp(source, target, recurse=True)
```

```python
1 dbutils.fs.refreshMounts()
```

```python
1 display(dbutils.fs.ls('/mnt/transformed/final.csv/'))
```

```python
1 from pyspark.sql import SparkSession
2 spark = SparkSession.builder.appName("health").getOrCreate()
3
4 directory_path = '/mnt/transformed/final.csv/'
5
6 files = dbutils.fs.ls(directory_path)
7
8 csv_files = [file for file in files if file.name.endswith('.csv')]
9
10 file_path = directory_path + csv_files[0].name
11
12 print(file_path)
```

Set

Hi! Here some our recommendations to get the best out
BLACKBOX:

Be as clear as possible

End the question in what language you want
answer to be, e.g: 'connect to mongodb in py

or you can just          Go to Blackbo

Here are some suggestion (choos

Write a function that reads data from a jso

How to delete docs from mongodb in ph

Connect to mongodb in no

BLACKBOX AI

Blackbox AI Chat is in beta and Blackbox is not liable for the content generated. By using Blackb
acknowledge that you agree to agree to Blackbox's Terms and Privacy Policy