**2. Analysis on Physician MIPS Score parameters**

**Retrieving the datasets from the source data.medicare.com**

1. Downloaded the following datasets -

* Physician\_Compare\_2017\_Individual\_EC\_Public\_Reporting\_Overall\_MIPS\_Performance.csv
* Physician Compare National Downloadable File.csv

1. Uploaded the dataset on to the S3 bucket in the folder (/gt-data-analytics-2020/MedicareDataAnalysis/PhysicianData)
2. On main.py, Import the required modules for establishing connection with AWS S3 bucket along with config file containing awskey and secret key. Created a class DataLoader and method LoadFile() that can download a file from S3 bucket and create a file handle object.
3. Using the DataLoader.LoadFile() method imported(inherited) from main.py, created an object
4. physiciangeneraldata=DataLoader.LoadFile(FilePhysicianNationalData)

physicianMISPdata=DataLoader.LoadFile(FilePhysicianIndivialMIPSData)

**Cleaning of the dataset**

1. Considering the parameters used to measure the Physician’s general data , extracted the following columns into another dataset (clean\_physiciangeneraldata) and columns renamed

NPI

Individual Pac ID

Last Name

First Name

Gender

Medical School

Graduation Yr

Primary Speciality

Organization Name

Organization Pac ID

No. Org Memebers

Address ln 1

Address ln 2

City

State

Zip

Phone No

1. Physician\_Compare\_2017\_Individual\_EC\_Public\_Reporting\_Overall\_MIPS\_Performance.csv data was merged into the clean\_physiciangeneraldata

merge\_physiciandata\_MIPS = pd.merge(clean\_physiciangeneraldata,physicianMISPdata, on="NPI", how ="inner")

1. Redundant columns were dropped, and the final columns were renamed to match the data.
2. Dropped the rows with MIPS score of 0.
3. Final clean dataset was saved as csv file (final\_physician\_data.csv) and uploaded to S3 for analysis

**Physician MIPS Score dataset**

1. Created another dataframe to get the dataset with unique NPIs
2. Grouped the dataset by gender and calculated the average MIPS score for each gender.

grouped\_gender=unique\_pnpi.groupby(['Gender'])

avg\_gendermips=grouped\_gender["Final MIPS Score"].mean()

1. Grouped the dataset by state and calculated the average MIPS score for each state, displaying the list from largest to smallest by average MIPS score.

grouped\_state=unique\_pnpi.groupby(['State'])

avg\_statemips=grouped\_state["Final MIPS Score"].mean()

avg\_statemips.nlargest(50)

**Plotting the summarized data**

1. Pie Plots displaying the distribution by gender

[Pie Chart Gender Distribution](https://github.com/ishanku/MedicareDataAnalysisProject/blob/master/output/gender_dist_MIPS.png)

1. Stacked bar plots displaying the states by average MIPS score of Physicians.

[MIPS Score State Compare](https://github.com/ishanku/MedicareDataAnalysisProject/blob/master/output/State_AvgMIPS.png)

1. Bar plot displaying MIPS score by gender.

[MIPS Score by gender](https://github.com/ishanku/MedicareDataAnalysisProject/blob/master/output/gender_AvgMIPS.png)

# 1. Analysis on Hospital ranking parameters

## Retrieving the dataset from the source data.medicare.com

1. Downloaded the latest(2019) Hospital General Information.csv from data.medicare.gov

2. Uploaded the dataset on to the S3 bucket in the folder (/gt-data-analytics-2020/MedicareDataAnalysis/HospitalData)

3. On main.py, Import the required modules for establishing connection with AWS S3 bucket along with config file containing awskey and secret key. Created a class DataLoader and method LoadFile() that can download a file from S3 bucket and create a file handle object.

4. Using the DataLoader.LoadFile() method imported(inherited) from main.py, created an object

hospitalgeneraldata=DataLoader.LoadFile(FileHospitalGeneralData)

## Cleaning of the dataset

1. Considering the parameters used to measure the Hospital overall rating, extracted the following columns into another dataset

1. Effectiveness of care national compare

2. Timeliness of care national compare

3. Efficient use of medical imaging national compare

4. Patient experience national compare

5. Readmission national compare

2. The columns in the consolidated dataset had "Not Available" values and the rows containing these values were deleted.

3. Column names were shortened to exclude the string "national compare" which is redundant among all columns

## Hospital Overall Ratings dataset

1. Created another dataframe extracting the column "Hospital Overall Rating" from the file object.

2. Cleaned the dataset to exclude "Not available" values

3. Extracted the count of number of hospitals under each Rank category using seriesRatings.str.count()

4. Created a Ranking\_df with the counts of each rank.

5. Extracted the count of number of hospitals scored for each of the five measure parameter under the below categories:

1. Above the national average

2. Below the national average

3. Same as the national average

6. Create a 5x3 dataframe (Count\_summary\_df) with the count values under each category for all five parameters.

## Plotting the summarized data

1. Pie Plots displaying the summarized information on hospital ranking parameters

[Pie Charts Percentage Image](https://github.com/ishanku/MedicareDataAnalysisProject/blob/Roopa-branch/output/Pie-Charts-Percentage.png)

2. Stacked bar plots displaying the number of hospitals categorized as above, below and same as national average.

[Measures National Compare](https://github.com/ishanku/MedicareDataAnalysisProject/blob/Roopa-branch/output/Measures-National-Compare.png)

3. Bar plot displaying under the number of hospitals ranked overall.

[Hospital Overall Ranking](https://github.com/ishanku/MedicareDataAnalysisProject/blob/Roopa-branch/output/Hospital-Overall-Ranking.png)