**Import Data**

**Databricks Link:**

<https://dbc-4b25e3a7-c263.cloud.databricks.com/editor/notebooks/430884827852682?o=2161976920830377>

Transfer Dataset

A screenshot of a computer program

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

OMR dataset

A screenshot of a computer code

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

Admission dataset

A screenshot of a computer code

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

Patient dataset

A screenshot of a computer code

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

Due to the dataset being too big, I selected to use the parquet function to compress the file. Using the parquet format allows row-based files, such as CSV, to transition to columnar storage, which can help users reduce their data storage requirements by around one-third on large datasets (Databricks, n.d.). Therefore, this can significantly improve the scan and retrieve time, while reducing the overall cost and minimizing the latency of accessing data. Figure 1 indicates the usage of computer resources of the CSV file and the Parquet file. Besides that, I also input the code to let the computer show the total rows and total columns, so I can easily understand the details of every dataset.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1 Comparison table of converting data into Parquet from CSV

Therefore, from the above 4 datasets, I can get 2 insights, which are:

1. Admission table: Determine the average duration of hospital stay for patients based on each admission type.
2. OMR and Patient table: The average height and weight of male and female patients.

Apart from that, there also have a MapReduce analysis objective:

1. Transfers table: Average process time within the care unit with different event type

**Insight 1: Determine the average duration of hospital stay for patients based on each admission type.**

**Databrick Link:**

<https://dbc-4b25e3a7-c263.cloud.databricks.com/editor/notebooks/3973729708646714?o=2161976920830377>

A screen shot of a computer code

AI-generated content may be incorrect.

Firstly, let the computer read the admission parquet file, then select the columns that are related to the insight, which are ‘admittime’, ‘dischtime’, and ‘admission\_type’. At the same time, using the printSchema() function to let the computer show the datatype of each column, this can help me to identify which column needs to be transformed for subsequent calculations.

Output:

A screenshot of a computer code

AI-generated content may be incorrect.

During the output of the Schema, there was a nullable=True, which means that the column allows the occurrence of null. If the nullable=False, which means the column does not allow null included in the column.

A screenshot of a computer code

AI-generated content may be incorrect.The above code is to check whether the remaining column has a missing value or not. The method I am using is col(c).isNull() to check whether every row has null or not; if that row has null, it will return a boolean value which is ‘True’ or ‘False’. After that, using the cast.(‘int’) to change the boolean value to an integer number (1=True, 0=False), which makes it easy to use the sum to count the number of nulls. Lastly, using the sum function to sum the entire column and find out how many 1's there are, which is the total number of null values in the column. Then let the computer show the total number of null values for each column and show the column that I selected to ensure the selected column is the correct one.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

The next step is to check the typo of the admission type column, according to the table, Admission (2020). I create valid types of arrays to define the correct admission type names. After that, create another array called ‘typo\_df’ to store the invalid names of the admission type. The function of upper() is to change the data in the dataset to uppercase, like ‘emergency’ to ‘EMERGENCY’, and using the trim() to delete the leading and trailing spaces to ensure the data has the same condition as the valid array. Then, the computer will start checking and filtering the data, meanwhile storing the data not meet the premise of the valid array in the typo\_df. Lastly, count how many misspellings in the typo\_df to know the number of typos in the admission type column.

Output:



A computer code with many colorful text

AI-generated content may be incorrect.

The above code starts to calculate the patient's stay time in the hospital. I added a column called ‘stay\_hours’ in the dataset first, then converted the time string into a UNIX timestamp and used the second to become the unit. For example, the admittime is ‘2169-10-12T16:30:00.000+00:00’, it will convert to 1641105000 by using the unix\_timestamp function. Although we get the staying time, the unit is seconds; thus, we divided 3600 to let it become hours as a unit. Besides that, using the groupBy function to put each admission\_type as a group. Therefore, the computer can calculate the total number of records and total hospitalization hours for this admission type, so this can easily to get the average staying time in the hospital.

A computer screen shot of a computer code

AI-generated content may be incorrect.

This step is to convert the average length of stay in hospital (in hours) into a human-readable time string in the format HH H MM M SS S and store it in the column Average\_stay\_time. Firstly, change the stay hours to second, while ensuring the datatype is ‘long’ so that it can facilitate subsequent floor operations. Using the floor function to get the integer divided result, and using ‘% 3600’ is to extract the ‘remaining seconds’. After that, using the concact\_ws to let the computer automatically put the spaces in the concatenation, with clear labels ‘H’, ‘M’, ‘S’. Therefore, the result will show very clearly and be readable like ‘45H 45M 45 S’. Lastly, selected the column that I want in the result\_df and sorted it by the number of records in ascending order.

A computer screen shot of a computer code

AI-generated content may be incorrect.

This code is to show the result and the datatype of those columns in the result table.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

The above analysis result shows that the differences in average staying time for different types of admissions. This can help the stakeholders to understand which admission type is taking the longest time to process, so they can optimize the bed allocation, surgical arrangements, and discharge processes. Especially the staying time for the urgent admission type is obviously high, so it is recommended to review the process to improve the operational efficiency and control the costs to prevent unnecessary wasting like human resources.

**Insight 2: The average height and weight of male and female patients**.

**Databricks Link:**

<https://dbc-4b25e3a7-c263.cloud.databricks.com/editor/notebooks/384593714155195?o=2161976920830377>

A screenshot of a computer program

AI-generated content may be incorrect.

For this objective, I will need to retrieve the data from the OMR table and the Patient table. Therefore, I need to join these two tables together with the column that I want, which is the gender and subject ID from the patient table, and result name, result value and subject ID from the OMR table. At the same time, I will only keep the records where subject\_id appears in both tables; the purpose of this is to analyse the patients who have both gender information and physical examination records. Then, show the number of rows that meet the criteria.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Then check the result name got typos, or not, because my objective is to calculate the average height and weight of male and female patients, so I need to make sure the result name is spelled correctly and no capitalization or space difference. Create a valid array according to the table, OMR (2023) to store the legal data, then let the computer check if the data in the result name column meets the valid array or not. If it gets misspelled, it will add to the typo\_df, then show how many typos.

Output:



A screenshot of a computer program

AI-generated content may be incorrect.

After ensuring the result name does not have any misspelling issues, I need to filter out the records related to height and weight from the merged dataset for subsequent gender classification and average calculation. Using the filter function to preserve the records with result names ("WEIGHT (LBS)", "HEIGHT (INCHES)", "WEIGHT", "HEIGHT"). After that, let the computer count and print the number of weight and height data remaining after filtering.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

This step is the crucial step in the data preprocessing, which is checking whether the remaining column has a missing value or not.

Output:

A white background with black text

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

As the dataSchema shows, the datatype of the result value is string, so it cannot do any calculation in the next step. So, I need to change the datatype from string to double to do calculations for finding the average height and weight. First, the computer will check the data in the result\_value column to check whether the data is in the legal digital format. If yes, convert to double and store in the new column value\_num; otherwise, set it as null and remove it. For example, if the dataset got ‘-167.5’, ‘1720 mm’, they will be seen as invalid types and get dropped.

Due to the result name, there are two types of Weight records, which are ‘WEIGHT (LBS)’ and ‘WEIGHT’, and Height records also have two types, which are ‘HEIGHT (INCHES)’ and ‘HEIGHT’

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a weight chart

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

After reviewing the data in the result\_name called ‘Weight’, the result\_value is too high for the unit kg; therefore, I considered the unit as LBS. Same reason for Height, so I considered the unit for ‘Height’ as Inches.

A screenshot of a computer program

AI-generated content may be incorrect.

So, need to standardize the units of Weight and Height, which are setting them as kg and m. At the same time, rename the different expressions like ‘HEIGHT’ and ‘HEIGHT (INCHES)’ to ‘Height (m) and ‘WEIGHT (LBS)’ and ‘WEIGHT’ to ‘Weight (kg)’.

A screenshot of a computer code

AI-generated content may be incorrect.

Let the conversion result only show two decimal numbers and check the number of conversions.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

The above code is doing the outlier removal using the classic method --- IQR (interquartile range). Calculating the Q1 and Q3 by gender, height, and weight, then removing the outliers, can improve the reliability of the result. Split the dataset into 4 groups, which are ‘F, Height (m)’, ‘F, Weight (kg)’, ‘M, Height (m)’, and ‘M, Weight (kg)’. Based on each group's result value, find the outlier then using the subtract() function to remove the outliers from the original table. After that, print the result that for each group how many outliers they have removed and show the total outliers has been removed from the dataset. In this stage, the computer will also need to show how many patients in the dataset, so it can help to verify that the result\_value entries are consistently recorded for the same individuals without duplication or not.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Due to the above results shows that the number of patients is less than the record value, it shows that there have multiple records for the same person. To ensure the accuracy of statistic and do not count the same person twice, I need to first calculate the individual average for each patient to prevent that particular patient from affecting the overall average due to too many records. Lastly, calculating the averaging the individual averages of all patients to get the group average. Then show the result.

Output:

A white text with black letters

AI-generated content may be incorrect.

These results can be used for developing more precise medical strategies, like developing a personalized treatment pan or dosage drug formulations based on the average height and weight of different genders. Besides that, the doctor can used this result as a guideline to determine whether the patient’s weight and height deviates form the normal range or not. So, they can start to conduct an early nutritional interventions or disease screening.

**Map Reducing: Average process time within the care unit with different event type**

**Databricks Link:**

<https://dbc-4b25e3a7-c263.cloud.databricks.com/editor/notebooks/4387384411271456?o=2161976920830377>

A screen shot of a computer program

AI-generated content may be incorrect.

Due to the transfer table has 7 column and I only using the 4 columns which are ‘eventtype’, ‘careunit’, ‘intime’ and ‘outtime’. Therefore, I just selected and extracted these 4 columns data and show the datatype of them.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

According to the details in the transfers table (2020), I have known the valid types of eventtype. So, I created an array to check the data in the eventtype row is same with the valid array or not. If got typo, the computer will show the number of misspellings. I also using the trim() to remove the spaces and upper() function to change the data to uppercase. The final objective of this code is to filter out all misspelled or illegal format eventtype.

Output:



A screenshot of a computer code

AI-generated content may be incorrect.

Since the ‘DISCHARGE’ indicates the time when the patient is discharged from the hospital. Therefore, the processing process in each care unit during hospitalization, ‘DISCHARGE’ cannot be used to represent the patient’s processing or stay time in the hospital. If the discharge is kept, the average processing time of each care unit will be distorted, so it must be removed first. This can ensure that the analysis is of the in-hospital process and avoids irrelevant events to interrupt the calculation of average time.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

This code is to check the remaining dataset got missing value or not. Based on the column that we have ‘eventtype’ ,’careunit’, ‘intime’, and ‘outtime’, check each column got how many missing values. After that, print the number of missing values that each column has; if any one of these 4 columns is null, the entire row will be deleted. Then, show the before and after dropping missing value results of total rows.

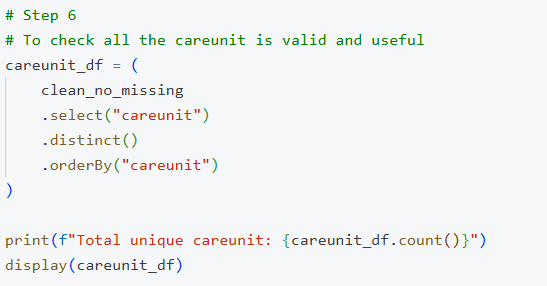
Output:

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



In the ‘careunit’ column, there has a lot of different care unit. Therefore, using this code can find the unique care unit and know there has how many care units in the dataset.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A white background with blue lines

AI-generated content may be incorrect.

After filtering the care unit, I realized that there has two care units call ‘UNKNOWN’ and ‘Unknown’ which cannot bring any meaning to the dataset. So, the next step is to remove the row with the care unit name with not meaningful (‘UNKNOWN’ and ‘Unknown’).

A screenshot of a computer code

AI-generated content may be incorrect.

The code above is to filter the care unit is unknown. After that, it will show the result of before and after removing the ‘unknown’ care unit.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

This step starts to do the map reduce. Before starting the map reduce, I need to define the key and the value. The value will be the staying time which is ‘outtime - intime’, so that we can get the staying time. For the key value which is ‘careunit’ and the ‘eventtype’. Therefore, the final output will like (Cardiac Surgery, admit), (value: 4567,1).

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

After mapping stage, it will move the reduce stage. In the reduce stage, using the groupBy function to group the same key meanwhile using the agg() function to aggregate all the staying time together with the same keys. Then, using the count(“\*”) function to get the instances of the staying time in the same key.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a computer code

AI-generated content may be incorrect.

During the reduce stage, we need to do the calculation of finding the average processing time in the different care unit with their event type. The algorithm of its is to sum all the staying seconds divided by the count in the same key. After calculation, the result will be represented like key: (Observation, transfer), value: (768990.09).

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

Although we already get the result in the reduce stage, the answer is shown as seconds which is a not intuitive format. After converting the seconds into the ‘hour: minute: second’ format, the medical staff and decision maker can easily understand the average stay time of each care unit and their event type, which can help them for easily manage and do the decision-making.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

This result can be used to optimize the treatment time of patients. Based on the average stay time in each department and their event type, the hospital can justify and optimize the treatment plan. For example, the average staying time of patients in certain departments is longer than in other departments, so the hospital needs to evaluate the procedures in the treatment process. This can help them discover the issues and find a solution to optimize the treatment plan to improve the quality of medical services.

**MapReduce VS Other Methods**

For the above scenario, which is to identify the average process time within the care unit with different event types, MapReduce and other methods, such as SQL and Spark, both have their pros and cons. But in this situation, MapReduce is more suitable than other methods for the following reasons:

1. Data Processing Speed

MapReduce is suitable for processing large datasets like Petabytes or Terabytes because these datasets are too large for in-memory operations, will means the computer cannot allocate the computer resources to handle these datasets (Tobin, 2023). MapReduce has strong capabilities to perform basic and complex analyses on large datasets like summarization, filtering, and joining large datasets. This can make the computer become more efficient because this can let the computer use disk-based storage rather than in-memory to process these large volume datasets.

2. Performance of handling big datasets

Spark processes the data in the random-access memory (RAM), while MapReduce stores the data on the disk after a map or reduce action (Tobin, 2023). In addition, Spark requires a lot of memory to store data. Besides that, when the datasets are loaded into Spark without enough space to fit, the performance of Spark may be directly degraded, and the computer may lag or crash. On the other hand, MapReduce will terminate the process when the job is done, which can release the resources to the computer to run with other services. Therefore, MapReduce is suitable for the data that does not fit in memory, like large datasets that come to the ETL (Extract, Transform, Load), and enables the computer to run well alongside other services.

In conclusion, due to the large amount of data in the transfer dataset (2,413,581 rows and 7 columns), MapReduce is more suitable for processing this dataset because it can better handle the processing demands of large datasets and can effectively utilize the computer’s disk storage resources. Apart from that, MapReduce can provide a clear visualization of key and value, which can let stakeholders immediately understand the processing results and insights. This structure of breakdown and presentation of data in a simple format allows for quick decision-making and more efficient analysis of large datasets.