# Assignment – Day 17

-Sarthak Niranjan Kulkarni (Maverick)

- <u>sarthakkul2311@gmail.com</u> - (+91) 93256 02791

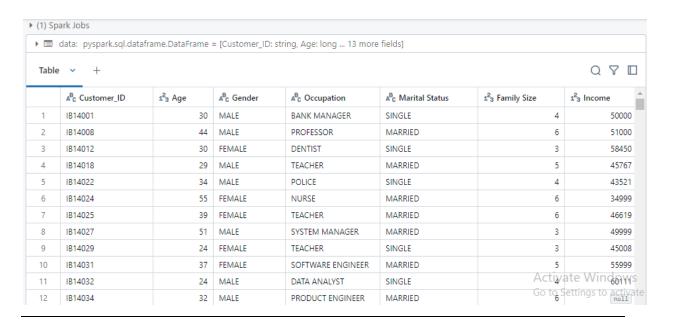
## 28/11/2024 (Thursday)

## **Practice EDA Analysis: -**

#### 1. Reading and Displaying Data from the Loan Table in Databricks

→ data = spark.read.table("hive\_metastore.default.loan")

display(data)



#### 2. Getting Row Count and Schema Information of the Data

→ # Total row count

data.count()

# Schema information

data.printSchema()

#### ▶ (2) Spark Jobs

```
root
|-- Customer_ID: string (nullable = true)
|-- Age: long (nullable = true)
|-- Gender: string (nullable = true)
|-- Occupation: string (nullable = true)
|-- Marital Status: string (nullable = true)
|-- Family Size: long (nullable = true)
|-- Income: long (nullable = true)
|-- Expenditure: long (nullable = true)
|-- Use Frequency: long (nullable = true)
|-- Loan Category: string (nullable = true)
|-- Loan Amount: string (nullable = true)
|-- Overdue: long (nullable = true)
|-- Debt Record: string (nullable = true)
|-- Returned Cheque: long (nullable = true)
|-- Dishonour of Bill: long (nullable = true)
```

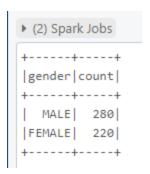
#### 3. Displaying Summary Statistics for 'Income' Column

→ # Summary statistics for 'Income'

data.describe(['Income']).show()

#### 4. Counting Rows Grouped by Gender

→ data.groupBy('gender').count().show()



## 5. Displaying Top 5 Highest Incomes

→ # Top 5 Highest Incomes

data.orderBy(data.Income.desc()).limit(5).show()

```
|Customer_ID|Age|Gender| Occupation|Marital Status|Family Size|Income|Expenditure|Use Frequency| Loan Category|Loan Amount
|Overdue | Debt Record | Returned Cheque | Dishonour of Bill |
  | IB14157| 35| MALE| BANK MANAGER| MARRIED| | 5| 5| 5| | 1B14107| 44|FEMALE|ACCOUNT MANAGER| MARRIED|
                                                                      35680 | 6 |
                                                        4 930000
                                                                                               HOUSING 6,79,040
                                                                      15632
                                                         4 800000
                                                                                             AUTOMOBILE 23,65,478
   4 | 800000 |
                                                                      15632
                                                                                      8 COMPUTER SOFTWARES | 23,65,478
           20,145| 3|
   IB14256 | 44 | FEMALE | ACCOUNT MANAGER | MARRIED |
                                                         4 | 800000 |
                                                                      15632
                                                                                      8 COMPUTER SOFTWARES | 23,65,478
           20,145| 3|
    IB14128 | 46 | FEMALE | CLERK | 4 | 16,324 | 3 |
                                        MARRIED
                                                         3 | 750000 |
                                                                      25641
                                                                                               GOLD LOAN 2,14,569
```

#### 6. Grouping Employees by Salary Buckets and Counting

→ # Salary Distribution

from pyspark.sql.functions import ceil, col

# Add salary buckets

data\_with\_buckets = data.withColumn('salary\_bucket', ceil(col('Income') / 20000) \* 20000)

# Count employees in each bucket

data\_with\_buckets.groupBy('salary\_bucket').count().orderBy('salary\_bucket').show()

```
▶ (2) Spark Jobs
  data_with_buckets: pyspark.sql.dataframe.DataFrame = [Customer_ID: string, Age: long ... 14 more fields]
|salary bucket|count|
   -----+
         NULL
                 32
         40000
                 70
                200
        60000
         80000
                136
       100000
       440000
       700000
                  1
       760000
       800000
       940000
                   1
```

# **Summary of EDA Analysis: -**

I worked on a dataset from the hive\_metastore.default.loan table using PySpark in Databricks. First, I loaded the data into a Spark DataFrame and displayed it to get a view of the records. I calculated the total number of rows in the dataset with the count() function, which shows how many entries there are. Then, I examined the schema of the data to understand the structure of the table, such as the column names and data types.

I also performed summary statistics for the Income column, which gave me basic measures like the count, mean, and standard deviation. I grouped the data by gender to count how many records fall into each gender category. To further explore the data, I identified the top 5 highest incomes by sorting the data in descending order based on the Income column. Finally, I created salary buckets by dividing the Income into ranges and counted how many employees fall into each bucket, helping me understand the distribution of income within the dataset.