Pyspark CaseStudy

Online Banking Analysis

-Bavatharani S

CSV files:

Loan.csv

Txn.csv

Credit card.csv

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
# Start SparkSession
spark = SparkSession.builder \
    .appName("Online Banking Analysis") \
    .getOrCreate()
# Load loan.csv
loan_df = spark.read.csv("/content/loan.csv", header=True, inferSchema=True)
# Load credit card.csv
credit_df = spark.read.csv("/content/credit card.csv", header=True, inferSchema=True)
# Load txn.csv
txn_df = spark.read.csv("/content/txn.csv", header=True, inferSchema=True)
# Display schema
loan_df.printSchema()
credit_df.printSchema()
txn_df.printSchema()
# Show first few records
loan_df.show(5)
credit_df.show(5)
txn_df.show(5)
```

```
root
 |-- RowNumber: integer (nullable = true)
 |-- CustomerId: integer (nullable = true)
 |-- Surname: string (nullable = true)
 |-- CreditScore: integer (nullable = true)
 |-- Geography: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Tenure: integer (nullable = true)
 |-- Balance: double (nullable = true)
 |-- NumOfProducts: integer (nullable = true)
 |-- IsActiveMember: integer (nullable = true)
 |-- EstimatedSalary: double (nullable = true)
 |-- Exited: integer (nullable = true)

root
 |-- Account No: string (nullable = true)
 |-- TRANSACTION DETAILS: string (nullable = true)
 |-- VALUE DATE: string (nullable = true)
 |--  WITHDRAWAL AMT : string (nullable = true)
 |--  DEPOSIT AMT : string (nullable = true)
 |-- BALANCE AMT: double (nullable = true)

+-----------+---+------+------------+--------------+-----------+------+-----------+---------------+---------------+-----------+-------+-----------+-----------------+-----------------+
|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|
+-----------+---+------+------------+--------------+-----------+------+-----------+---------------+---------------+-----------+-------+-----------+-----------------+-----------------+
|    IB14001| 30|  MALE|BANK MANAGER|        SINGLE|          4| 50000|      22199|            6|      HOUSING| 10,00,000 |      5|     42,898|                6|                9|
|    IB14008| 44|  MALE|   PROFESSOR|       MARRIED|          6| 51000|      19999|            4|     SHOPPING|     50,000|      3|     33,999|                1|                5|
|    IB14012| 30|FEMALE|     DENTIST|        SINGLE|          3| 58450|      27675|            5|   TRAVELLING|     75,000|      6|     20,876|                3|                1|
|    IB14018| 29|  MALE|     TEACHER|       MARRIED|          5| 45767|      12787|            3|    GOLD LOAN|  6,00,000 |      7|     11,000|                0|                4|
|    IB14022| 34|  MALE|      POLICE|        SINGLE|          4| 43521|      11999|            3|   AUTOMOBILE|  2,00,000 |      2|     43,898|                1|                2|
+-----------+---+------+------------+--------------+-----------+------+-----------+---------------+---------------+-----------+-------+-----------+-----------------+-----------------+
only showing top 5 rows

+---------+----------+-------+-----------+---------+------+---+------+---------+-------------+---------------+----------+------+
|RowNumber|CustomerId| Surname|CreditScore|Geography|Gender|Age|Tenure|  Balance|NumOfProducts|IsActiveMember|EstimatedSalary|Exited|
+---------+----------+-------+-----------+---------+------+---+------+---------+-------------+---------------+----------+------+
|        1|  15634602|Hargrave|        619|   France|Female| 42|     2|      0.0|            1|              1|  101348.88|     1|
|        2|  15647311|    Hill|        608|    Spain|Female| 41|     1| 83807.86|            1|              1|  112542.58|     0|
|        3|  15619304|    Onio|        502|   France|Female| 42|     8| 159660.8|            3|              0|  113931.57|     1|
|        4|  15701354|    Boni|        699|   France|Female| 39|     1|      0.0|            2|              0|   93826.63|     0|
|        5|  15737888|Mitchell|        850|    Spain|Female| 43|     2|125510.82|            1|              1|    79084.1|     0|
+---------+----------+-------+-----------+---------+------+---+------+---------+-------------+---------------+----------+------+
only showing top 5 rows

+-------------+--------------------+----------+----------------+-------------+-------------+
|   Account No| TRANSACTION DETAILS|VALUE DATE| WITHDRAWAL AMT | DEPOSIT AMT |BALANCE AMT|
+-------------+--------------------+----------+----------------+-------------+-----------+
|409000611074'|TRF FROM  Indiafo...| 29-Jun-17|            NULL|      1000000|  1000000.0|
|409000611074'|TRF FROM  Indiafo...|  5-Jul-17|            NULL|      1000000|  2000000.0|
|409000611074'|FDRL/INTERNAL FUN...| 18-Jul-17|            NULL|       500000|  2500000.0|
|409000611074'|TRF FRM  Indiafor...|  1-Aug-17|            NULL|      3000000|  5500000.0|
|409000611074'|FDRL/INTERNAL FUN...| 16-Aug-17|            NULL|       500000|  6000000.0|
+-------------+--------------------+----------+----------------+-------------+-----------+
only showing top 5 rows
```

In loandata.csv file

1. Number of loans in each category

loan_df.groupBy("Loan Category").count().show()

```
[3]  loan_df.groupBy("Loan Category").count().show()
```

```
+-------------------+-----+
|      Loan Category|count|
+-------------------+-----+
|            HOUSING|   67|
|          TRAVELLING|   53|
|        BOOK STORES|    7|
|        AGRICULTURE|   12|
|           GOLD LOAN|   77|
|   EDUCATIONAL LOAN|   20|
|          AUTOMOBILE|   60|
|            BUSINESS|   24|
|COMPUTER SOFTWARES|   35|
|            DINNING|   14|
|            SHOPPING|   35|
|        RESTAURANTS|   41|
|        ELECTRONICS|   14|
|            BUILDING|    7|
|          RESTAURANT|   20|
|    HOME APPLIANCES|   14|
+-------------------+-----+
```

2. Number of people who have taken more than 1 lack loan

Filtering customers who took a loan greater than 1 lakh

loan_df.filter(loan_df["Loan_Amount"] > 100000).count()

```
[6]
    # Filtering customers who took a loan greater than 1 lakh
    loan_df.filter(loan_df["Loan_Amount"] > 100000).count()
```

```
0
```

3. Number of people with income greater than 60000 rupees

Filtering rows where income is greater than 60,000

loan_df.filter(col("Income") > 60000).count()

```
    # Filtering rows where income is greater than 60,000
    loan_df.filter(col("Income") > 60000).count()
```

```
198
```

4. Number of people with 2 or more returned cheques and income less than 50000

Applying multiple conditions using logical AND (&)

loan_df.filter((col("Returned_Cheque") >= 2) & (col("Income") < 50000)).count()

```
[10] # Applying multiple conditions using logical AND (&)
     loan_df.filter((col("Returned_Cheque") >= 2) & (col("Income") < 50000)).count()
```

```
137
```

5. Number of people with 2 or more returned cheques and are single

Filtering by returned cheques and marital status

loan_df.filter((col("Returned_Cheque") >= 2) & (col("Marital_Status") == "Single")).count()

```
# Filtering by returned cheques and marital status
loan_df.filter((col("Returned_Cheque") >= 2) & (col("Marital_Status") == "Single")).count()
```

```
0
```

6. Number of people with expenditure over 50000 a month

Filtering high monthly spenders

loan_df.filter(col("Expenditure") > 50000).count()

```
[12] # Filtering high monthly spenders
     loan_df.filter(col("Expenditure") > 50000).count()
```

```
6
```

7. Number of members who are elgible for credit card

Example rule: Eligible if debt record is good and they use banking services frequently

loan_df.filter((col("Use_Frequency") > 10) & (col("Debt_Record") == "Good")).count()

```
# Example rule: Eligible if debt record is good and they use banking services frequently
loan_df.filter((col("Use_Frequency") > 10) & (col("Debt_Record") == "Good")).count()
```

```
0
```

In credit.csv file

1. Credit card users in Spain

credit_df.filter(col("Geography") == "Spain").count()

```
# 1. Show number of credit card users located in Spain

credit_df.filter(col("Geography") == "Spain").count()
```

```
2477
```

2. Number of members who are elgible and active in the bank

Example threshold: Credit Score >= 650 is eligible

credit_df.filter((col("CreditScore") >= 650) & (col("IsActiveMember") == 1)).count()

credit_df.filter((col("CreditScore") >= 650) & (col("IsActiveMember") == 1)).show()

```
[17] # Example threshold: Credit Score >= 650 is eligible
     credit_df.filter((col("CreditScore") >= 650) & (col("IsActiveMember") == 1)).count()
```

```
2672
```

```
credit_df.filter((col("CreditScore") >= 650) & (col("IsActiveMember") == 1)).show()
```

```
+---------+----------+---------+-----------+---------+------+---+------+---------+-------------+--------------+---------------+------+
|RowNumber|CustomerId|  Surname|CreditScore|Geography|Gender|Age|Tenure|  Balance|NumOfProducts|IsActiveMember|EstimatedSalary|Exited|
+---------+----------+---------+-----------+---------+------+---+------+---------+-------------+--------------+---------------+------+
|        5| 15737888| Mitchell|        850|    Spain|Female| 43|     2|125510.82|            1|             1|        79084.1|     0|
|        7| 15592531| Bartlett|        822|   France|  Male| 50|     7|      0.0|            2|             1|        10062.8|     0|
|       10| 15592389|       H?|        684|   France|  Male| 27|     2|134603.88|            1|             1|       71725.73|     0|
|       20| 15568982|      Hao|        726|   France|Female| 24|     6|      0.0|            2|             1|       54724.03|     0|
|       21| 15577657| McDonald|        732|   France|  Male| 41|     8|      0.0|            2|             1|      170886.17|     0|
|       24| 15725737|   Mosman|        669|   France|  Male| 46|     3|      0.0|            2|             1|        8487.75|     0|
|       25| 15625047|      Yen|        846|   France|Female| 38|     5|      0.0|            1|             1|      187616.16|     0|
|       27| 15736816|    Young|        756|  Germany|  Male| 36|     2|136815.64|            1|             1|      170041.95|     0|
|       35| 15732963| Clements|        722|    Spain|Female| 29|     9|      0.0|            2|             1|      142033.07|     0|
|       38| 15729599|  Lorenzo|        804|    Spain|  Male| 33|     7|  76548.6|            1|             1|       98453.45|     0|
|       39| 15717426|Armstrong|        850|   France|  Male| 36|     7|      0.0|            1|             1|        40812.9|     0|
|       45| 15684171|  Bianchi|        660|    Spain|Female| 61|     5|155931.11|            1|             1|      158338.39|     0|
|       46| 15754849|    Tyler|        776|  Germany|Female| 32|     4|109421.13|            2|             1|      126517.46|     0|
|       47| 15602280|   Martin|        829|  Germany|Female| 27|     9|112045.67|            1|             1|      119708.21|     1|
|       64| 15751208|  Pirozzi|        684|    Spain|  Male| 56|     8| 78707.16|            1|             1|       99398.36|     0|
|       66| 15789484|  Hammond|        751|  Germany|Female| 36|     6|169831.46|            2|             1|       27758.36|     0|
|       68| 15641582|  Chibugo|        735|  Germany|  Male| 43|    10|123180.01|            2|             1|      196673.28|     0|
|       69| 15638424|  Glauert|        661|  Germany|Female| 35|     5|150725.53|            2|             1|      113656.85|     0|
|       73| 15812518|  Palermo|        657|    Spain|Female| 37|     0|163607.18|            1|             1|       44203.55|     0|
|       77| 15614049|       Hu|        664|   France|  Male| 55|     8|      0.0|            2|             1|      139161.64|     0|
+---------+----------+---------+-----------+---------+------+---+------+---------+-------------+--------------+---------------+------+
only showing top 20 rows
```

In Transactions file

1. Maximum withdrawal amount in transactions

from pyspark.sql.functions import max, col

txn_df.select(

max(col("WITHDRAWAL_AMT").cast("double")).alias("Max_Withdrawal")

).show()

```
from pyspark.sql.functions import max, col

txn_df.select(
    max(col("WITHDRAWAL_AMT").cast("double")).alias("Max_Withdrawal")
).show()
```

```
+--------------+
|Max_Withdrawal|
+--------------+
|         4.0E8|
+--------------+
```

2. MINIMUM WITHDRAWAL AMOUNT OF AN ACCOUNT in txn.csv

#Minimum Withdrawal Amount

txn_df.select(min("WITHDRAWAL_AMT").alias("Min_Withdrawal_Amount")).show()

```
#Minimum Withdrawal Amount
txn_df.select(min("WITHDRAWAL_AMT").alias("Min_Withdrawal_Amount")).show()
```

```
+---------------------+
|Min_Withdrawal_Amount|
+---------------------+
|                 0.01|
+---------------------+
```

3. MAXIMUM DEPOSIT AMOUNT OF AN ACCOUNT

#Maximum Deposit Amount

txn_df.select(max("DEPOSIT_AMT").alias("Max_Deposit_Amount")).show()

```
[26] #Maximum Deposit Amount
     txn_df.select(max("DEPOSIT_AMT").alias("Max_Deposit_Amount")).show()
```

```
+------------------+
|Max_Deposit_Amount|
+------------------+
|           9999999|
+------------------+
```

4. MINIMUM DEPOSIT AMOUNT OF AN ACCOUNT

Minimum Deposit Amount

txn_df.select(min("DEPOSIT_AMT").alias("Min_Deposit_Amount")).show()

```
# Minimum Deposit Amount
txn_df.select(min("DEPOSIT_AMT").alias("Min_Deposit_Amount")).show()
```

```
+------------------+
|Min_Deposit_Amount|
+------------------+
|       -1667026170|
+------------------+
```

5. sum of balance in every bank account

#Sum of Balance Amount in Each Bank Account

```
txn_df.groupBy("Account_No") \
    .sum("BALANCE_AMT") \
    .withColumnRenamed("sum(BALANCE_AMT)", "Total_Balance") \
    .show()
```

```
#Sum of Balance Amount in Each Bank Account

txn_df.groupBy("Account_No") \
      .sum("BALANCE_AMT") \
      .withColumnRenamed("sum(BALANCE_AMT)", "Total_Balance") \
      .show()
```

```
+-------------+--------------------+
|   Account_No|       Total_Balance|
+-------------+--------------------+
|409000438611'|-2.49486577068339...|
|     1196711'|-1.37810146616539...|
|     1196428'| -2.8115133425562E13|
|409000493210'|-3.27584952132095...|
|409000611074'|       1.615533622E9|
|409000425051'|-3.77211841164998...|
|409000405747'|-2.43108047067000...|
|409000493201'|1.0420831829499985E9|
|409000438620'|-7.12237285703308...|
+-------------+--------------------+
```

6. Number of transaction on each date

Number of Transactions Happening on Each Date

```
txn_df.groupBy("VALUE_DATE") \
    .agg(count("*").alias("Transaction_Count")) \
    .orderBy("VALUE_DATE") \
    .show()
```

```
# Number of Transactions Happening on Each Date

txn_df.groupBy("VALUE_DATE") \
      .agg(count("*").alias("Transaction_Count")) \
      .orderBy("VALUE_DATE") \
      .show()
```

```
+----------+-----------------+
|VALUE_DATE|Transaction_Count|
+----------+-----------------+
|      NULL|                1|
|  1-Aug-15|               27|
|  1-Aug-16|               46|
|  1-Aug-17|               13|
|  1-Aug-18|               65|
|  1-Dec-16|               73|
|  1-Dec-17|                2|
|  1-Dec-18|               38|
|  1-Feb-16|               51|
|  1-Feb-17|               44|
|  1-Feb-18|               17|
|  1-Feb-19|               37|
|  1-Jan-15|                1|
|  1-Jan-18|               22|
|  1-Jan-19|               33|
|  1-Jul-15|                4|
|  1-Jul-16|               65|
|  1-Jul-17|              191|
|  1-Jun-16|               72|
|  1-Jun-17|               14|
+----------+-----------------+
only showing top 20 rows
```

7. List of customers with withdrawal amount more than 1 lakh

Customers With Withdrawal Amount Greater Than ₹1,00,000

```
txn_df.filter(col("WITHDRAWAL_AMT") > 100000) \
    .select("Account_No", "WITHDRAWAL_AMT", "VALUE_DATE") \
    .orderBy(col("WITHDRAWAL_AMT").desc()) \
    .show()
```

```
# Customers With Withdrawal Amount Greater Than ₹1,00,000

txn_df.filter(col("WITHDRAWAL_AMT") > 100000) \
      .select("Account_No", "WITHDRAWAL_AMT", "VALUE_DATE") \
      .orderBy(col("WITHDRAWAL_AMT").desc()) \
      .show()
```

```
+--------------+---------------+-----------+
|    Account_No|WITHDRAWAL_AMT|VALUE_DATE|
+--------------+---------------+-----------+
|409000438620'|        9947700| 15-May-17|
|409000438620'|         994755| 30-Jan-17|
|409000438620'|        9939786| 23-May-17|
|409000438620'|         993499| 14-Mar-17|
|409000438611'|        9900000| 15-Jan-18|
|409000438611'|        9900000| 31-Jul-18|
|409000438620'|        9900000|  2-May-17|
|409000438620'|        9900000|  2-May-18|
|409000438620'|         983537| 11-Jan-17|
|409000438611'|        9800000| 14-Dec-18|
|409000438620'|         977095| 27-Jan-17|
|409000438620'|        9710101| 17-Jan-19|
|409000438611'|        9700000| 27-Aug-18|
|409000438611'|        9700000| 25-Sep-18|
|409000438611'|        9700000| 27-Nov-18|
|409000438620'|        9700000| 26-Oct-17|
|409000438620'|        9600000| 16-May-17|
|409000438620'|        9600000| 30-Nov-17|
|409000438620'|        9600000| 23-Feb-18|
|409000438620'|        9600000|  7-Mar-18|
+--------------+---------------+-----------+
only showing top 20 rows
```