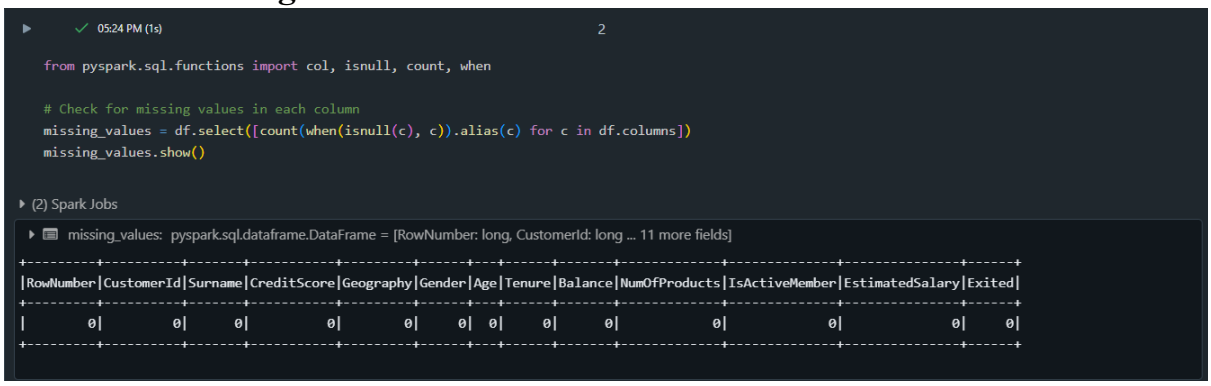
Azure Databricks Assignment 2

Name: **Harish Er**

1. Load the Dataset:

2. Check for Missing Values:



3. Basic Statistics for Numerical Columns:

```
05:24 PM (2s)                                                    3                                                    Python

# Basic statistics for numerical columns
df.describe().show()

(2) Spark Jobs
```

```
+-------+-----------------+-----------------+-------+-----------------+---------+------+-----------------+-----------------+-----------------+--------
---------+------------------+-----------------+------------------+
|summary|        RowNumber|       CustomerId|Surname|      CreditScore|Geography|Gender|              Age|           Tenure|          Balance|     Num
OfProducts|     IsActiveMember|  EstimatedSalary|            Exited|
+-------+-----------------+-----------------+-------+-----------------+---------+------+-----------------+-----------------+-----------------+--------
---------+------------------+-----------------+------------------+
|  count|            10000|            10000|  10000|            10000|    10000| 10000|            10000|            10000|            10000|
10000|             10000|            10000|             10000|
|   mean|           5000.5|  1.56909405694E7|   NULL|         650.5288|     NULL|  NULL|          38.9218|           5.0128|76485.88928799961|
1.5302|            0.5151|100090.2398809998|            0.2037|
| stddev|2886.8956799071675|71936.18612274907|   NULL|96.65329873613035|     NULL|  NULL|10.487806451704587|2.8921743770496837|62397.40520238599|0.581654
3579989917|0.4997969284589181|57510.49281769821|0.40276858399486065|
|    min|                1|         15565701|  Abazu|              350|   France|Female|               18|                0|              0.0|
1|                 0|            11.58|                 0|
|    max|            10000|         15815690| Zuyeva|              850|    Spain|  Male|               92|               10|        250898.09|
4|                 1|         199992.48|                 1|
+-------+-----------------+-----------------+-------+-----------------+---------+------+-----------------+-----------------+-----------------+--------
---------+------------------+-----------------+------------------+
```

4. Handle Missing Values:

```
05:24 PM (1s)                                                    4                                                    Python

# Fill missing values with median (numerical columns only)
numerical_columns = [col_name for col_name, dtype in df.dtypes if dtype in ('int', 'double')]

for column in numerical_columns:
    median_value = df.approxQuantile(column, [0.5], 0.01)[0]
    df = df.fillna({column: median_value})

# Show updated DataFrame
df.show(5)

(3) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [RowNumber: long, CustomerId: long ... 11 more fields]
```

```
+---------+----------+--------+-----------+---------+------+---+------+---------+-------------+--------------+---------------+------+
|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
```

5. Drop Duplicate Rows:

```
05:25 PM (1s)                                                    5                                                    Python

# Drop duplicate rows
df = df.dropDuplicates()

# Show updated DataFrame
df.show(5)
```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [RowNumber: long, CustomerId: long ... 11 more fields]

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	IsActiveMember	EstimatedSalary	Exited
30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	53483.21	0
220	15774854	Fuller	592	France	Male	54	8	0.0	1	1	28737.71	1
423	15674551	Fitch	535	Germany	Male	40	7	111756.5	1	0	8128.32	1
443	15672145	Swift	534	France	Female	34	7	121551.58	2	1	70179.0	0
806	15756026	Hooper	790	Spain	Female	46	9	0.0	1	0	14679.81	1

only showing top 5 rows

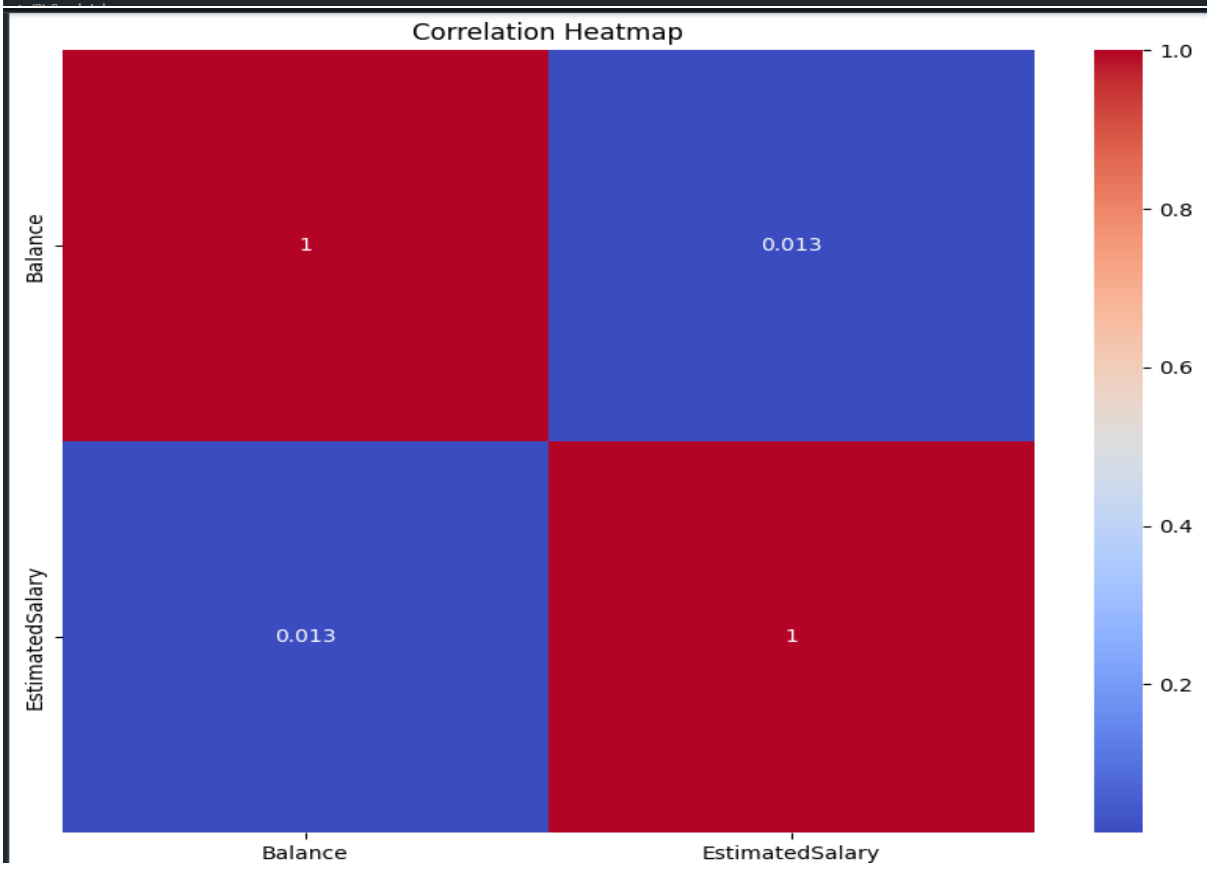
6. Correlation Heatmap for Numerical Columns:

```
05:25 PM (1s)                                                    6                                                    Python

import matplotlib.pyplot as plt
import seaborn as sns

# Correlation heatmap for numerical columns
numerical_columns = [col_name for col_name, dtype in df.dtypes if dtype in ('int', 'double')]
pandas_df = df.select(numerical_columns).toPandas()  # Convert to Pandas for visualization
correlation_matrix = pandas_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

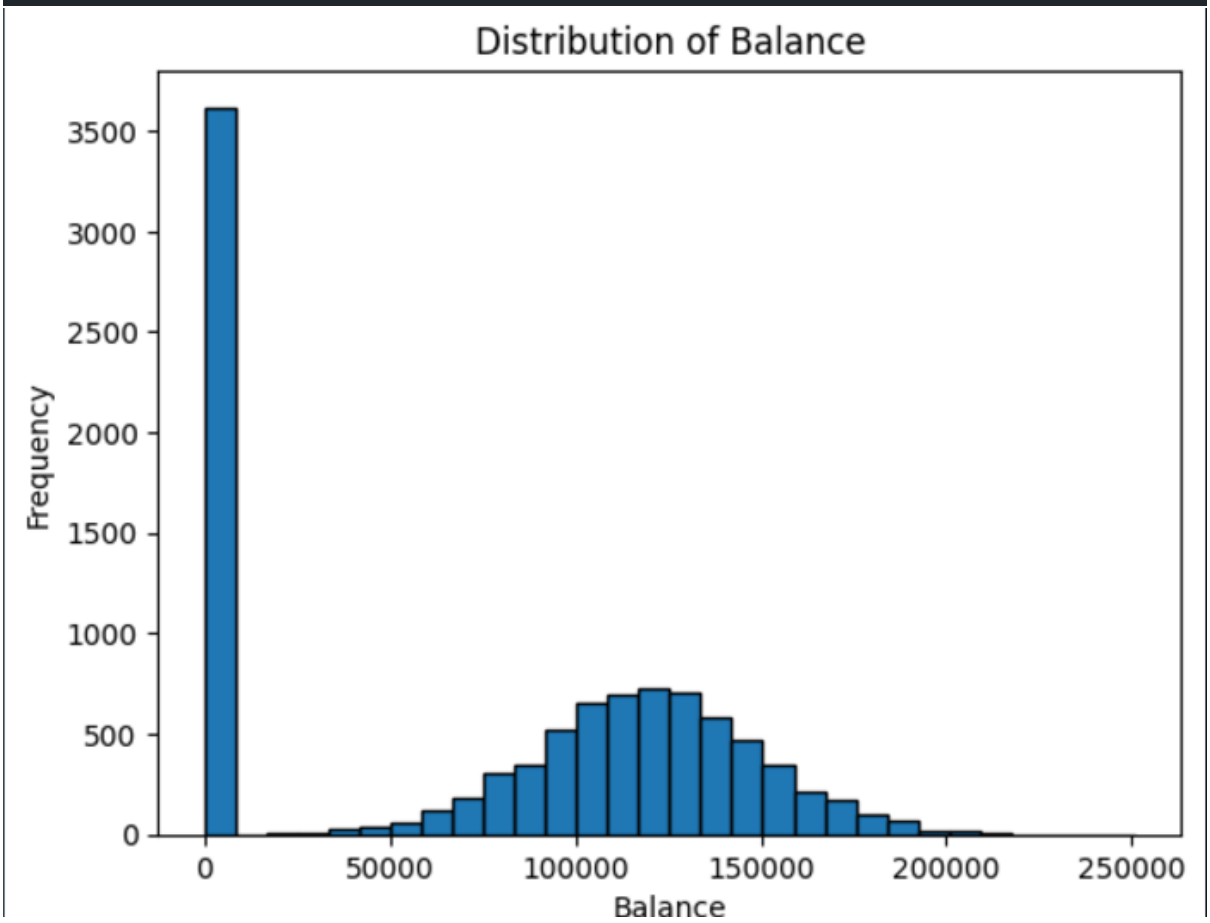


7. Distribution Plot for a Specific Column:

```
05:25 PM (<1s)                                    7                                    Python

# Distribution plot for one column (replace with your column name)
column_to_plot = numerical_columns[0]  # Replace with your choice

pandas_df[column_to_plot].plot(kind='hist', bins=30, edgecolor='black')
plt.title(f"Distribution of {column_to_plot}")
plt.xlabel(column_to_plot)
plt.ylabel("Frequency")
plt.show()
```



8. Value Counts for Categorical Columns:

```
05:26 PM (3s)                                     8                                    Python

# Value counts for categorical columns
categorical_columns = [col_name for col_name, dtype in df.dtypes if dtype == 'string']

for column in categorical_columns:
    df.groupBy(column).count().show()
```

(9) Spark Jobs

```
+------------+-----+
|     Surname|count|
+------------+-----+
|      Clunie|    1|
|      Piccio|   13|
|      Lavrov|    2|
|   Bezrukova|    2|
|  Kambinachi|    5|
|      Ohearn|    1|
|      Avdeev|    1|
|     Palermo|   12|
|    Sokolova|    2|
|      Azarov|    1|
|       Tyler|    4|
|     Wofford|    1|
|      Duigan|    2|
|    Rawlings|    1|
|       Rubeo|    1|
| Baryshnikov|    2|
|    Lazareva|    3|
|      Arbour|    1|
```

9. Boxplot for Outlier Detection:

05:26 PM (<1s) 9 Python

```
# Boxplot for outlier detection (example for a numerical column)
sns.boxplot(x=pandas_df[column_to_plot])
plt.title(f"Boxplot for {column_to_plot}")
plt.show()
```