**Azure Databricks Coding Challenge**

**Name : Tejaswini Gokanakonda**

**Roll No : DE142**

**Date : 10-12-2024**

**Creating a cluster :**

### Step 1: Access the Clusters Page

* On the Databricks workspace interface, find the left-hand navigation menu.
* Click on "Compute" or "Clusters", depending on your version of Databricks.

### Step 2: Create a New Cluster

Click the "Create Cluster" button at the top of the Clusters page.

### Step 3: Configure the Cluster

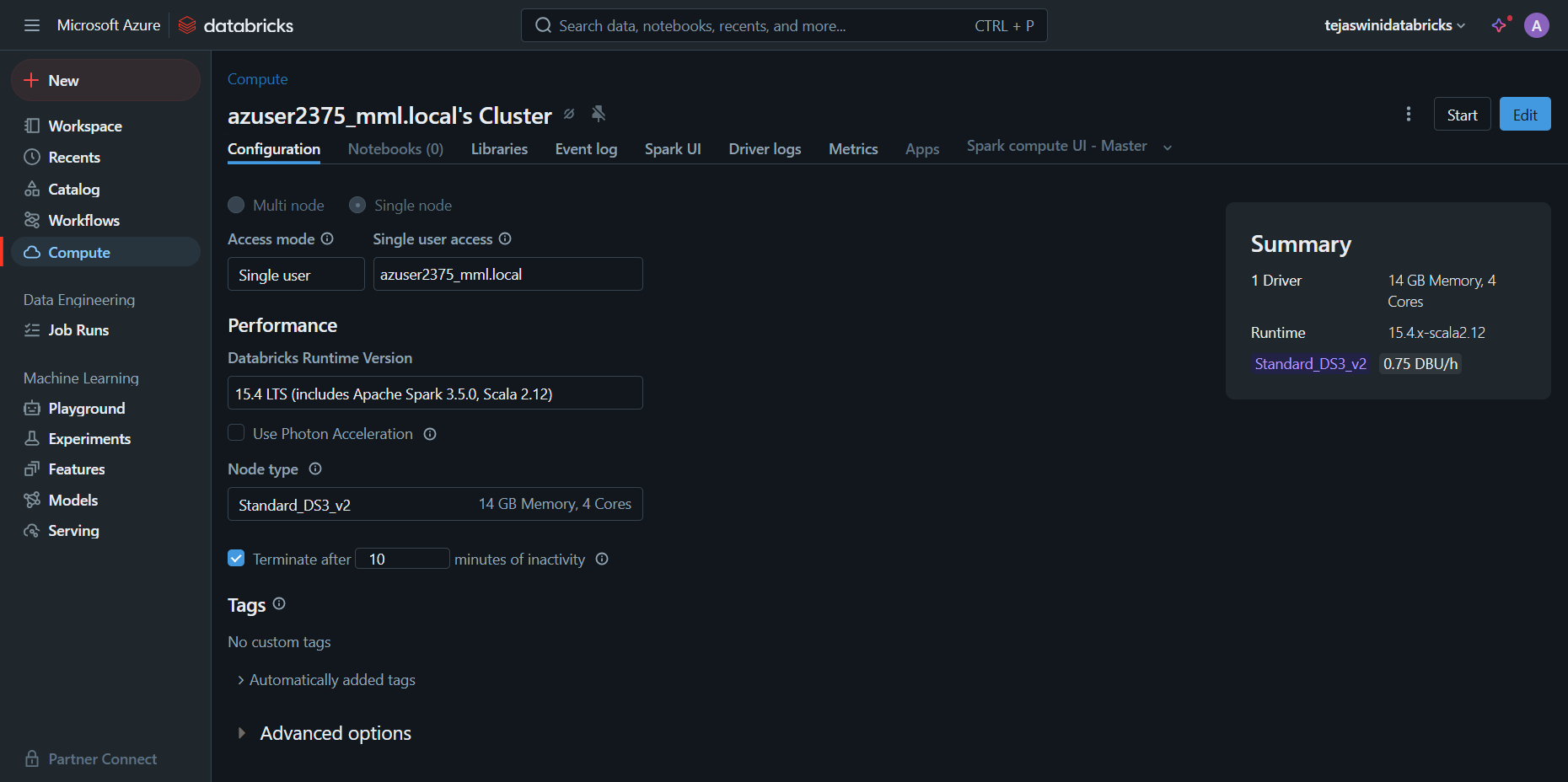
A form will appear where you can configure your cluster. Below are the key settings:

#### **Basic Settings:**

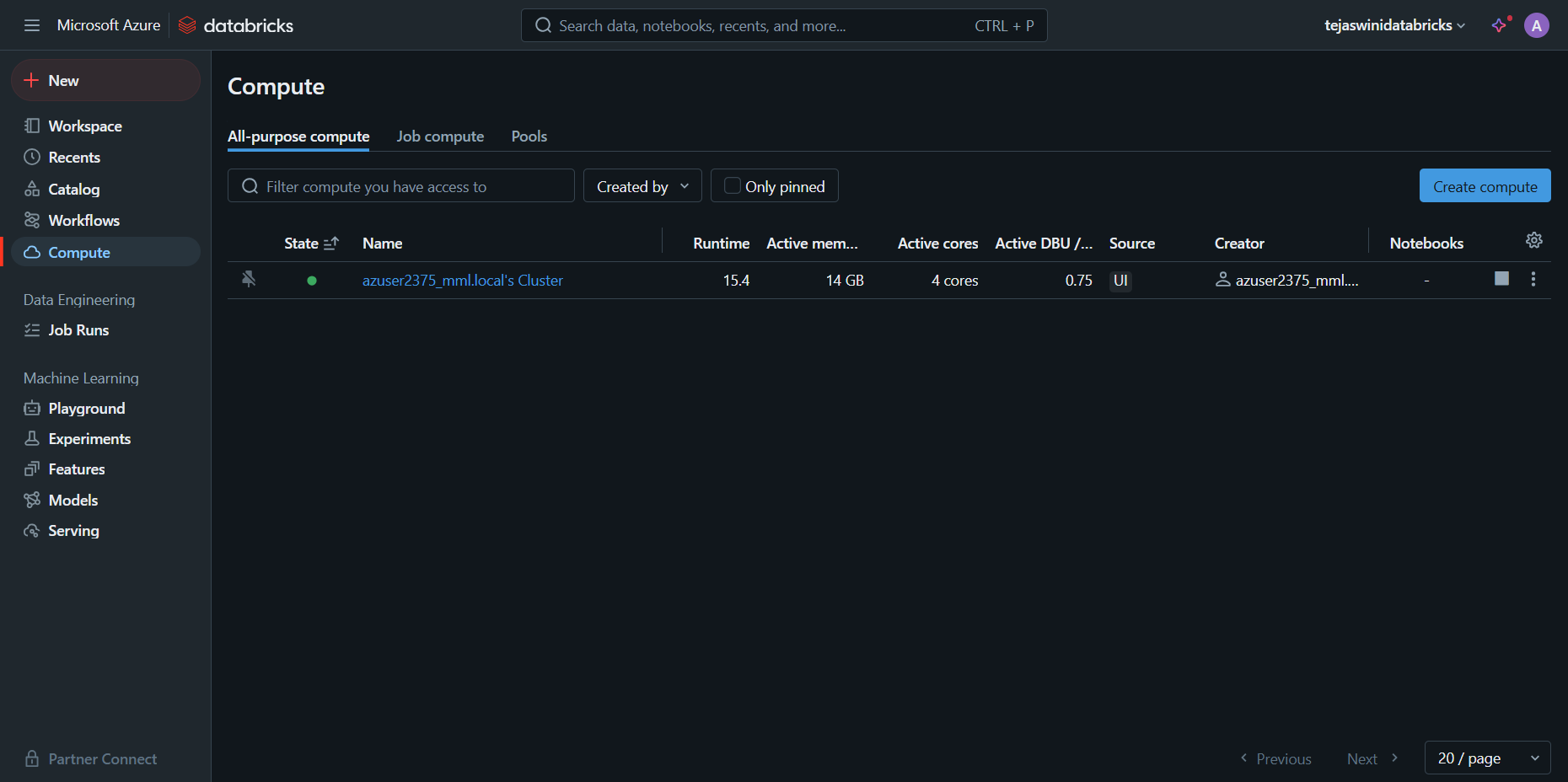
* **ClusterName:**  
  Enter a descriptive name for your cluster (e.g., "MyFirstCluster").
* **Cluster Mode:**
  + Standard: Select this for most workloads.
  + High Concurrency: Suitable for collaborative use cases with shared resources.
  + Single Node: For lightweight tasks or development.

#### **Databricks Runtime Version:**

* Select the appropriate Databricks runtime version based on your workload. For example:
  + Databricks Runtime for ML: For machine learning tasks.
  + Databricks Runtime for Genomics: For genomic analysis.
  + Default Runtime: For general-purpose tasks.



**Cluster Creation successful :**



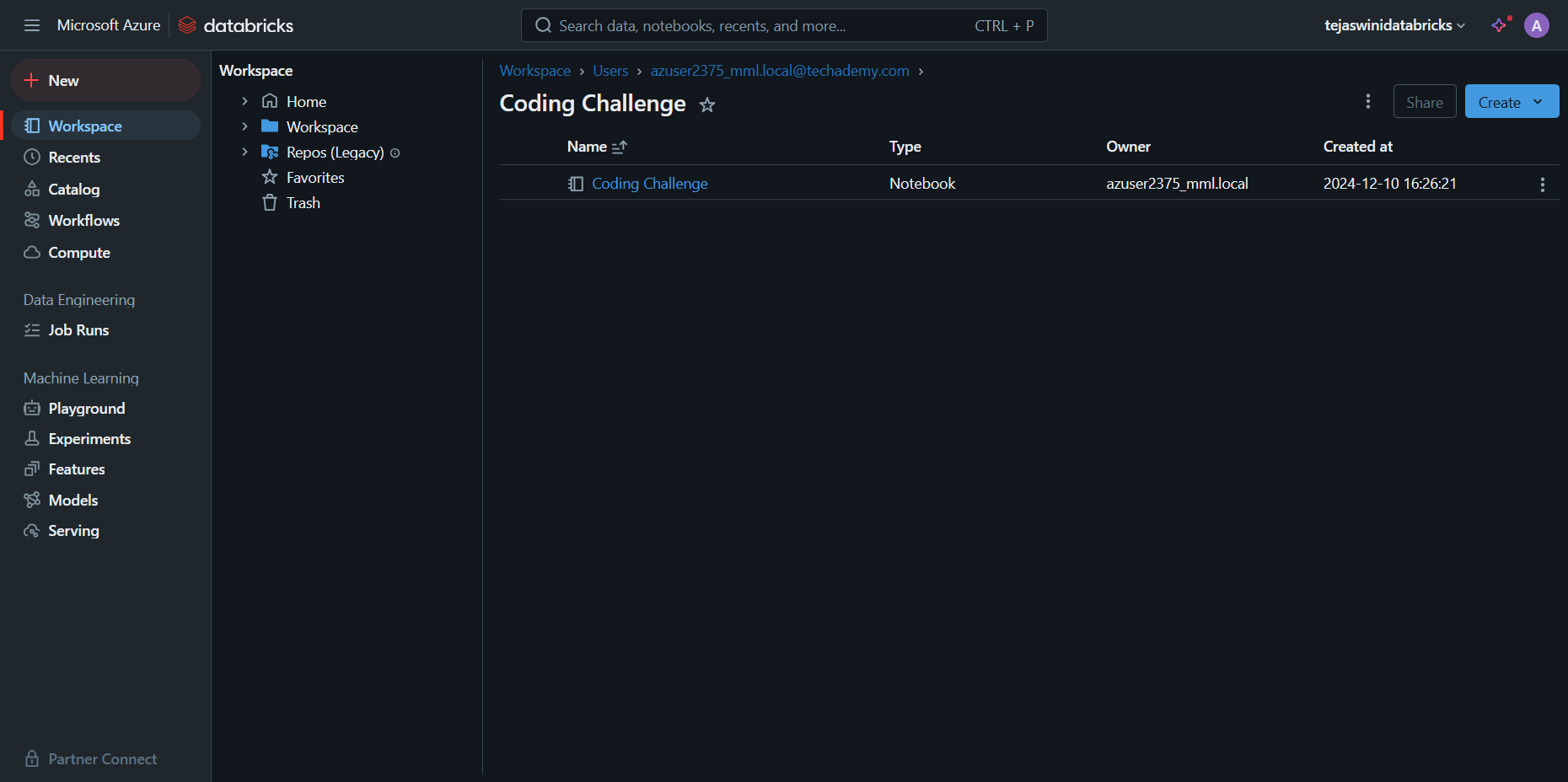
Step 4: Launch the Cluster

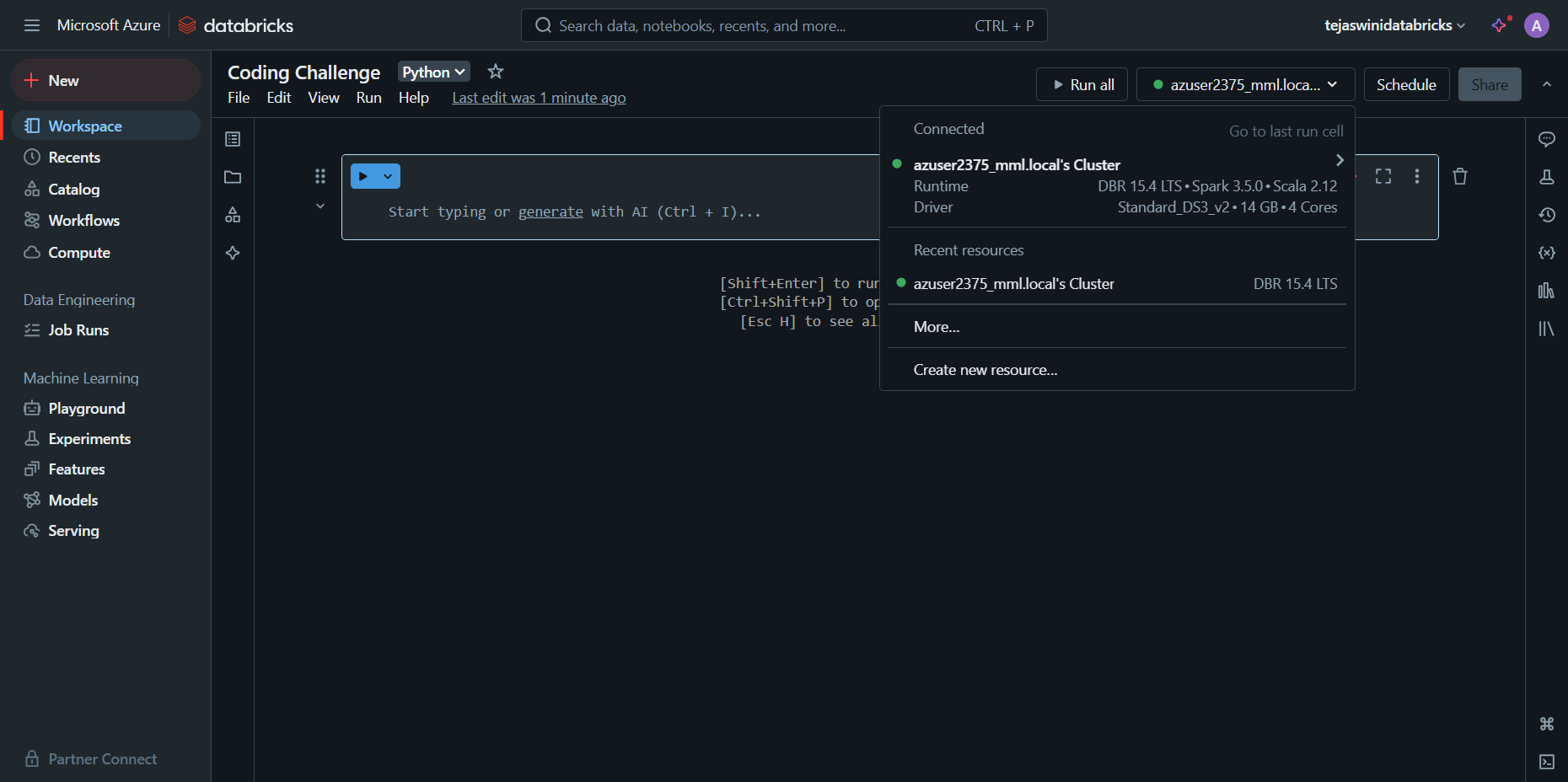
* After configuring the settings, click the "Create Cluster" button.
* The cluster will move into a "Pending" state and start provisioning. This may take a few minutes.

Step 5: Attach Your Notebook to the Cluster

* Open your Databricks notebook.
* Click the "Detached" dropdown in the top-right corner of the notebook.
* Select your newly created cluster from the list. The notebook is now attached to the cluster.

**Attaching notebooks :**

****

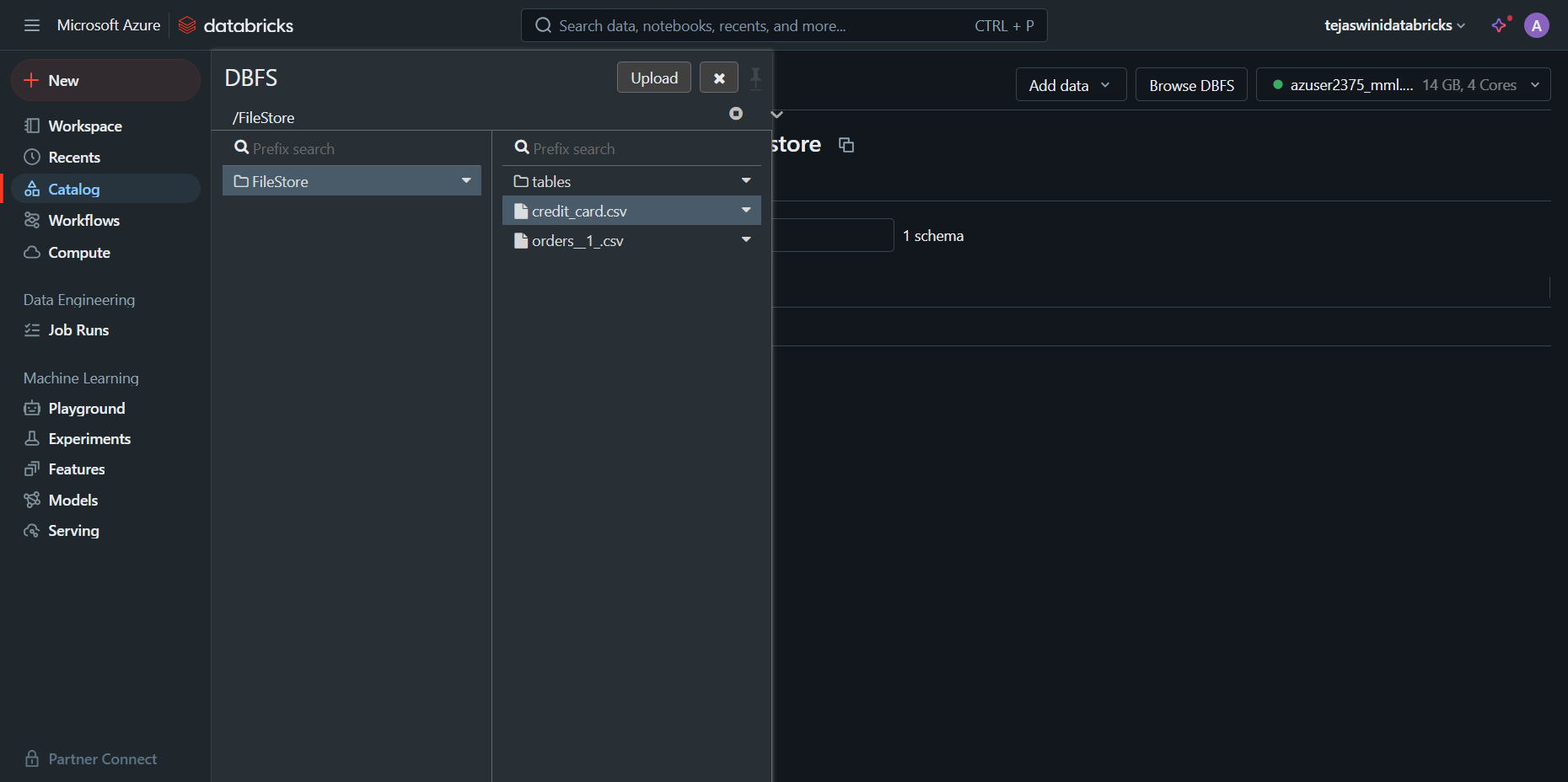


Step 6: Verify Cluster Status

* Go back to the Clusters page.
* Ensure your cluster shows a "Running" status. If it doesn’t, check the logs for any errors.

### **Steps to Attach a CSV File in Databricks**

1. **Upload the File**
   * Go to **Data** in the left menu.
   * Click **Add Data** > **Upload File**.
   * Upload your CSV and note the path (e.g., /FileStore/tables/your\_file.csv).

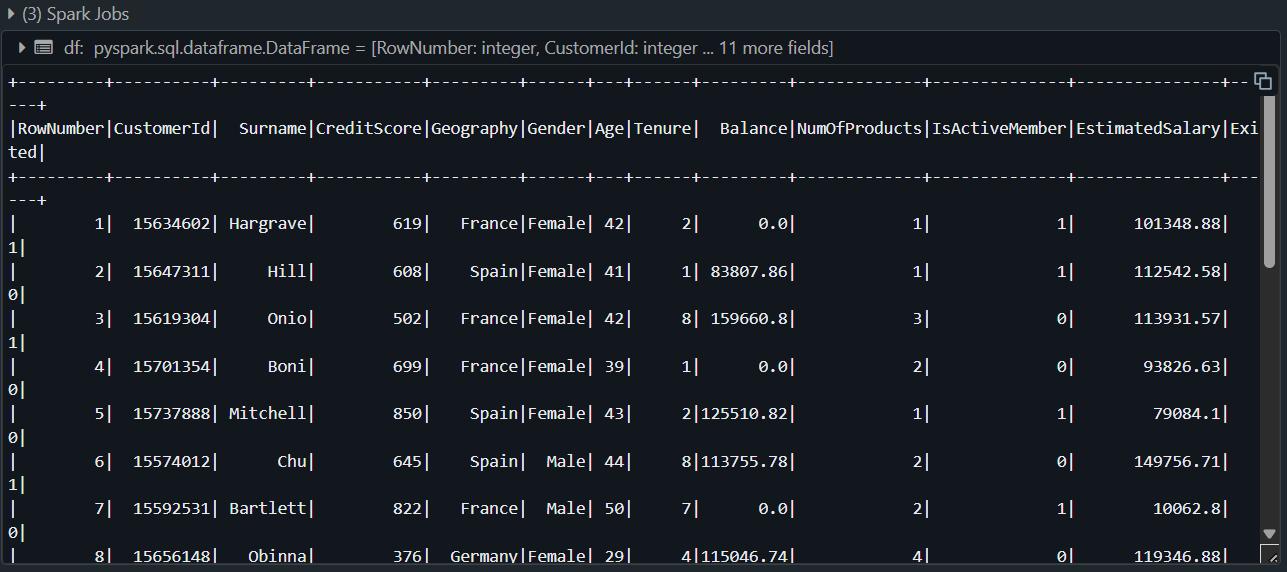


**2. Reading the file and creating a Dataframe**

|  |
| --- |
| df = spark.read.csv("dbfs:/FileStore/credit\_card.csv", header=True, inferSchema=True)  df.show() |

**3. Inspect Data**

Use df.printSchema() or df.describe().show() to check your data.

****

**Commands :**

**1. Customer Churn Analysis**

Customer churn is a critical metric for businesses to monitor, as it reflects the percentage of customers who stop using a company's services over a given period. This analysis focuses on identifying trends and patterns among churned and retained customers by examining key attributes such as credit score, age, and balance.

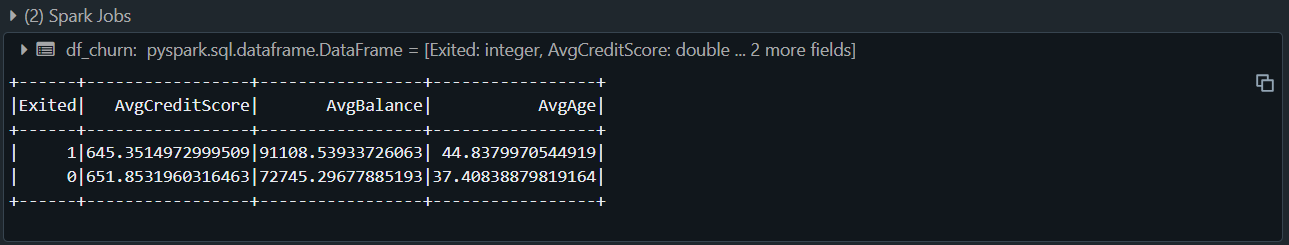
### **Objective**

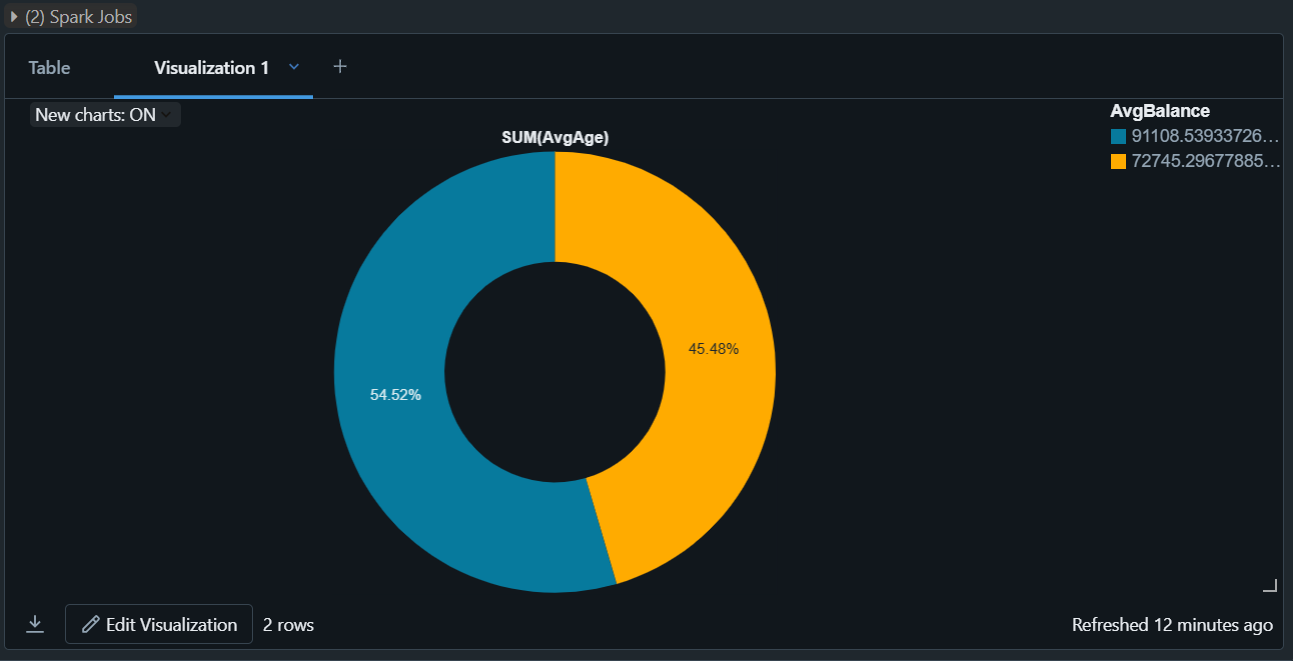
The analysis aims to compare the average values of these attributes for customers who:

* Exited (Exited = 1): Representing those who left the service.
* Retained (Exited = 0): Representing those who remained.

|  |
| --- |
| df\_churn = df.groupBy("Exited").agg(  {"CreditScore": "avg", "Age": "avg", "Balance": "avg"}  ).withColumnRenamed("avg(CreditScore)", "AvgCreditScore") \  .withColumnRenamed("avg(Age)", "AvgAge") \  .withColumnRenamed("avg(Balance)", "AvgBalance")  df\_churn.show() |

This aggregated data serves as a foundation for deeper analysis and decision-making, offering actionable insights to reduce churn and boost customer loyalty.

****

****

**2. Geographical Distribution of Customers**

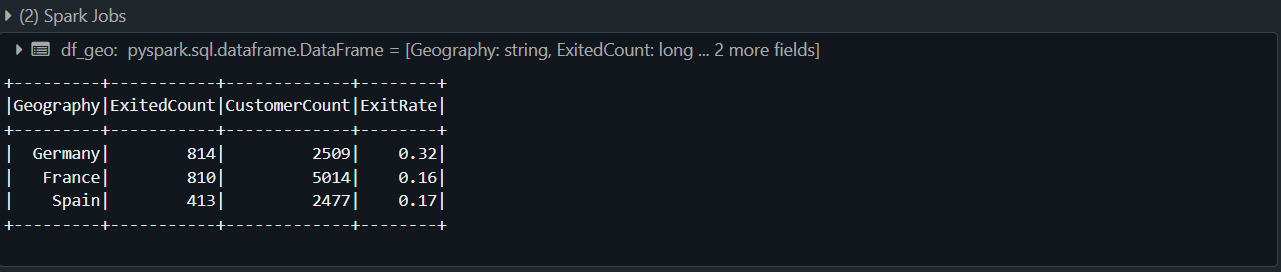
This analysis explores the geographical distribution of customers across various regions, providing insights into customer behavior and churn rates specific to each geography. By summarizing customer data by location, businesses can identify regional trends and tailor strategies to meet localized needs.

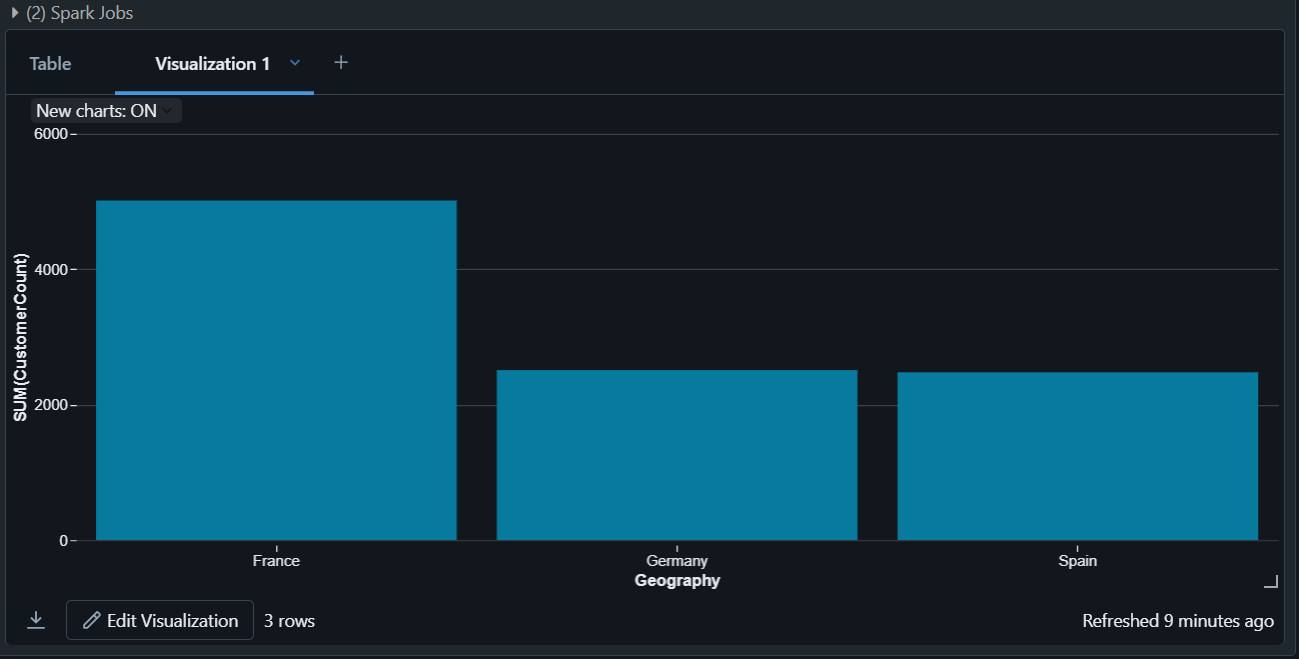
**Objective**

The aim is to understand how customer count and churn rate vary by geography. This analysis calculates:

1. **Customer Count:** The total number of customers in each region.
2. **Exited Count:** The number of customers who have churned (left the service) in each region.
3. **Exit Rate:** The proportion of churned customers relative to the total customers, expressed as a percentage for better comparability.

|  |
| --- |
| from pyspark.sql.functions import col, round  df\_geo = df.groupBy("Geography").agg(  {"CustomerId": "count", "Exited": "sum"}  ).withColumnRenamed("count(CustomerId)", "CustomerCount") \  .withColumnRenamed("sum(Exited)", "ExitedCount") \  .withColumn("ExitRate", round(col("ExitedCount") / col("CustomerCount"), 2))  df\_geo.show() |





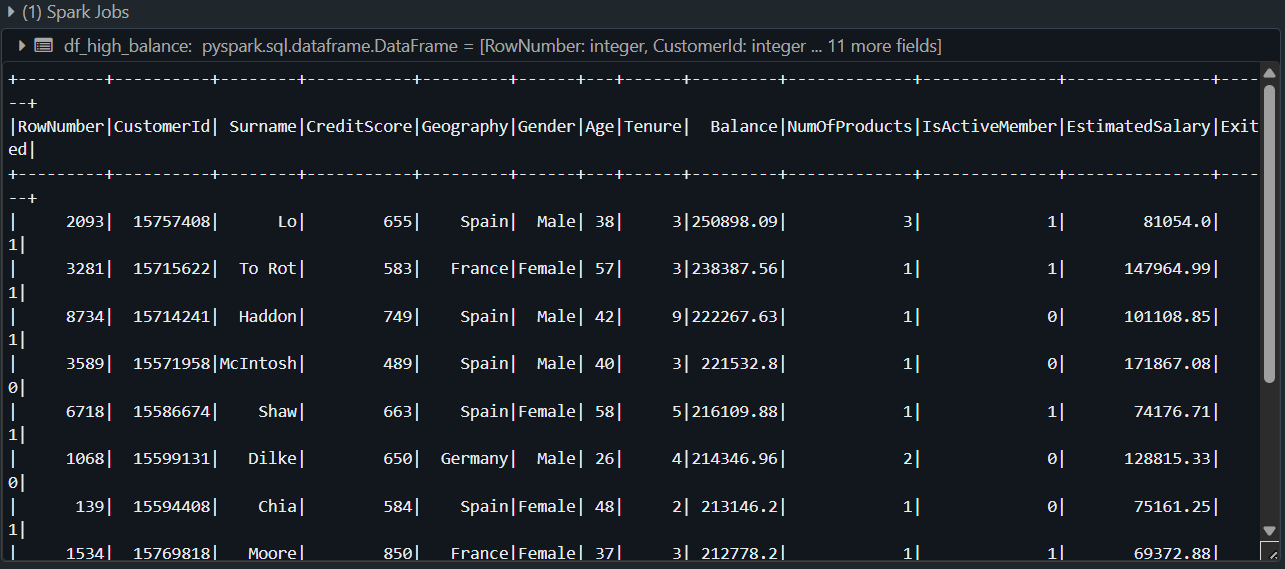
**3. High Balance Customers**

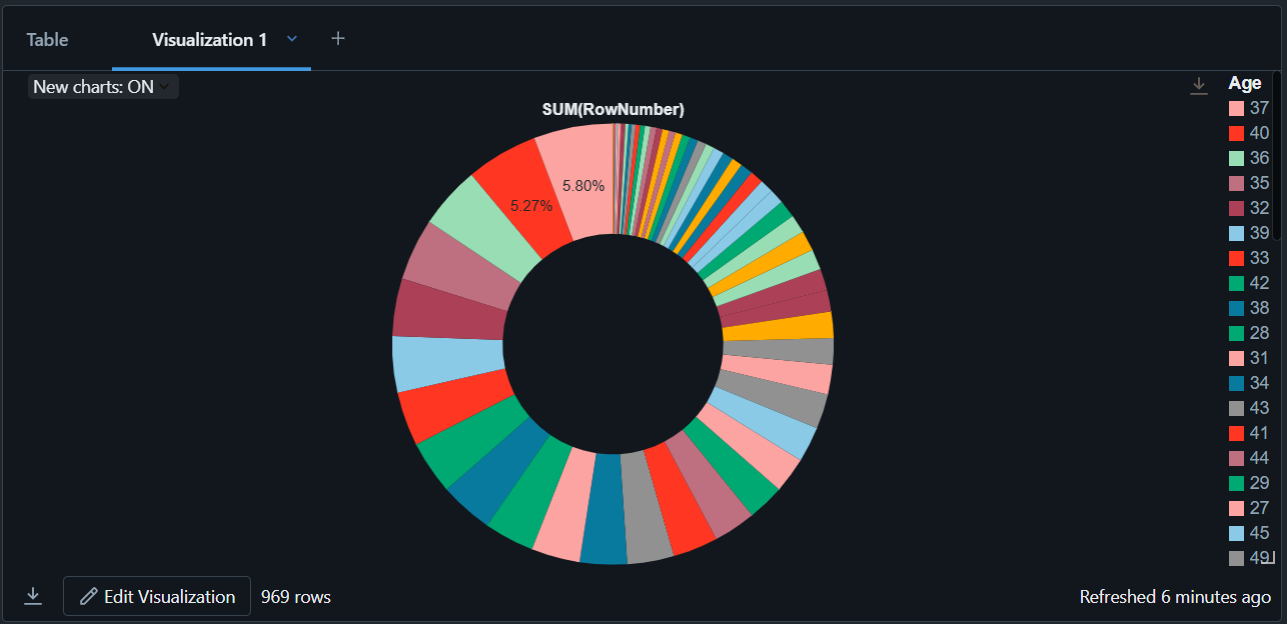
* This analysis identifies customers with exceptionally high account balances, focusing on those with balances exceeding a threshold of ₹150,000.
* These high-balance customers are often considered high-value clients, making them a critical segment for businesses aiming to enhance customer satisfaction and maximize profitability.

### **Objective**

* The goal is to pinpoint customers with substantial account balances and analyze their profiles to uncover patterns, preferences, or behaviors that could influence business strategies.
* By filtering and prioritizing customers based on account balances, this analysis enables businesses to focus on their most valuable clientele. It also provides insights into how to better serve and retain these customers, ensuring long-term profitability and enhanced customer satisfaction.

|  |
| --- |
| df\_high\_balance = df.filter(df['Balance'] > 150000).orderBy(df['Balance'].desc())  df\_high\_balance.show(10) |





**4. Customer Activity Analysis :**

This analysis examines the relationship between customer activity levels and churn rates, grouping customers based on their activity status. Understanding the activity patterns of customers provides critical insights into their engagement and potential satisfaction with the service.

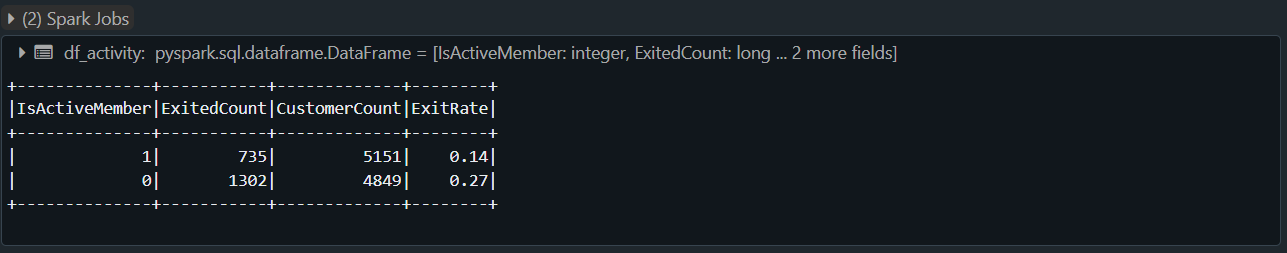
### **Objective**

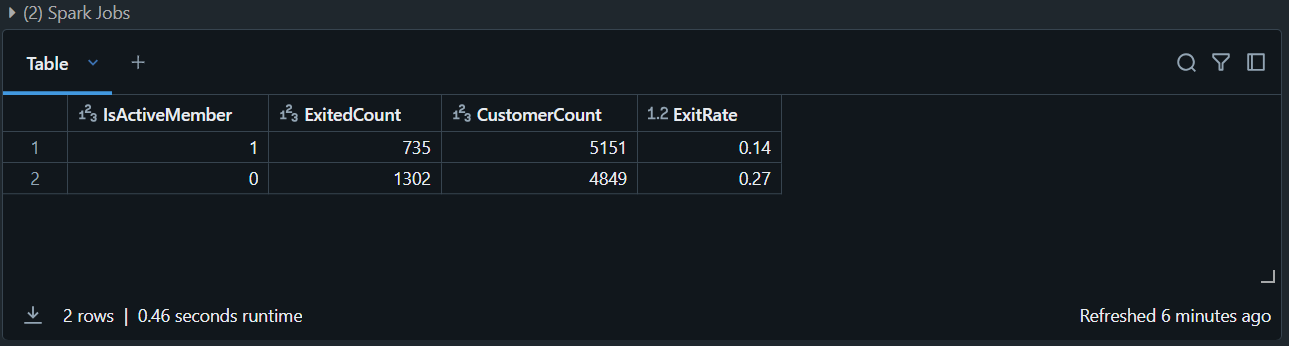
The goal is to compare churn behavior between active and inactive customers by calculating:

1. Customer Count: Total number of active and inactive customers.
2. Exited Count: Number of customers who churned within each group.
3. Exit Rate: Proportion of churned customers in each activity group.

This analysis enables businesses to differentiate between active and inactive customer behaviors, helping to build strategies for improving engagement and retention. It highlights the critical role that customer activity plays in predicting and preventing churn.

|  |
| --- |
| df\_activity = df.groupBy("IsActiveMember").agg(  {"Exited": "sum", "CustomerId": "count"}  ).withColumnRenamed("sum(Exited)", "ExitedCount") \  .withColumnRenamed("count(CustomerId)", "CustomerCount") \  .withColumn("ExitRate", round(col("ExitedCount") / col("CustomerCount"), 2))  df\_activity.show() |





**5. Age Distribution of Customers :**

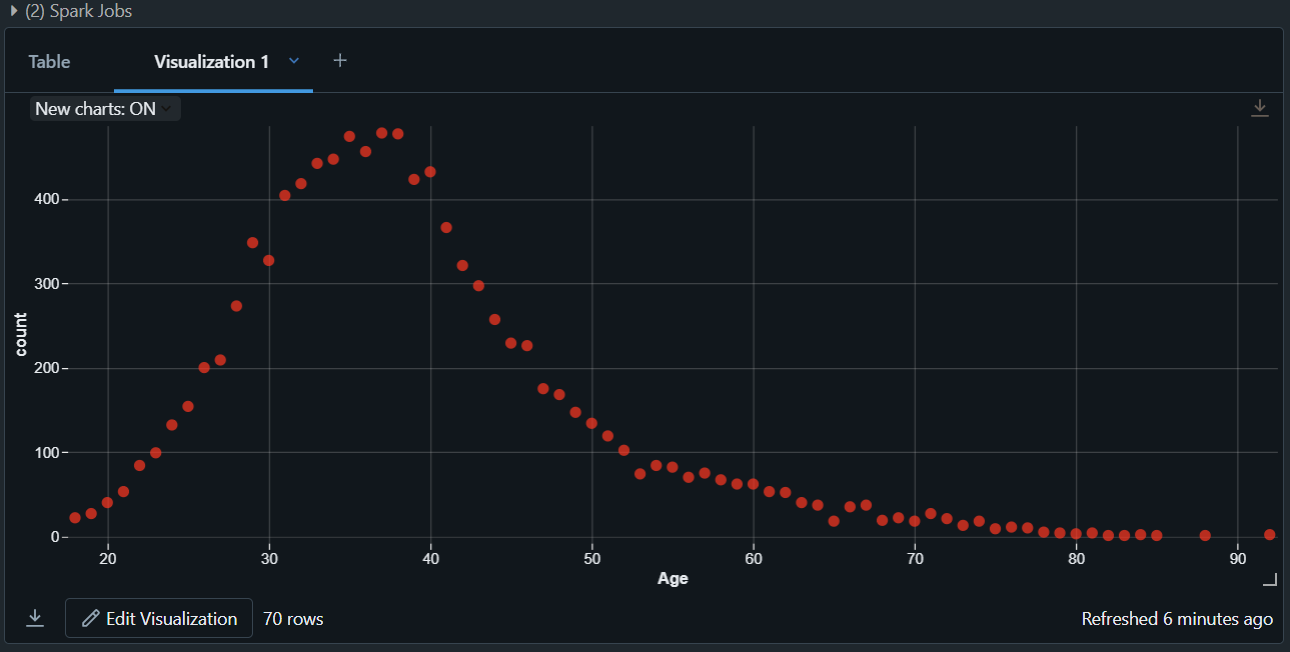
* This analysis focuses on the age distribution of customers, providing insights into the demographic spread across different age groups.
* Understanding customer age distribution helps businesses tailor their marketing efforts, product offerings, and customer engagement strategies to specific age segments.

**Objective**

* The goal is to examine the number of customers in each age group, allowing businesses to identify which age demographics are most represented and whether certain age groups require more attention.
* This age distribution analysis helps businesses understand their customer base in terms of age demographics. By segmenting customers based on age, businesses can develop more targeted marketing, services, and retention strategies, ultimately enhancing customer satisfaction and loyalty.

|  |
| --- |
| df\_age\_dist = df.groupBy("Age").count().orderBy("Age")  df\_age\_dist.show() |





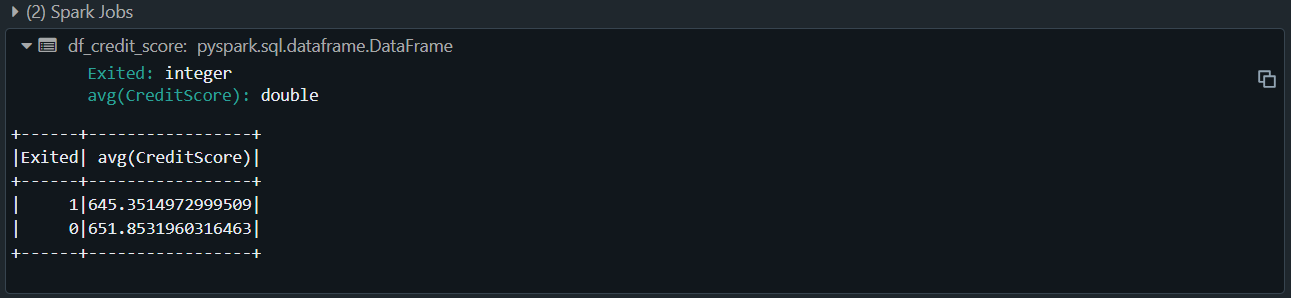
**6. Credit Score Analysis :**

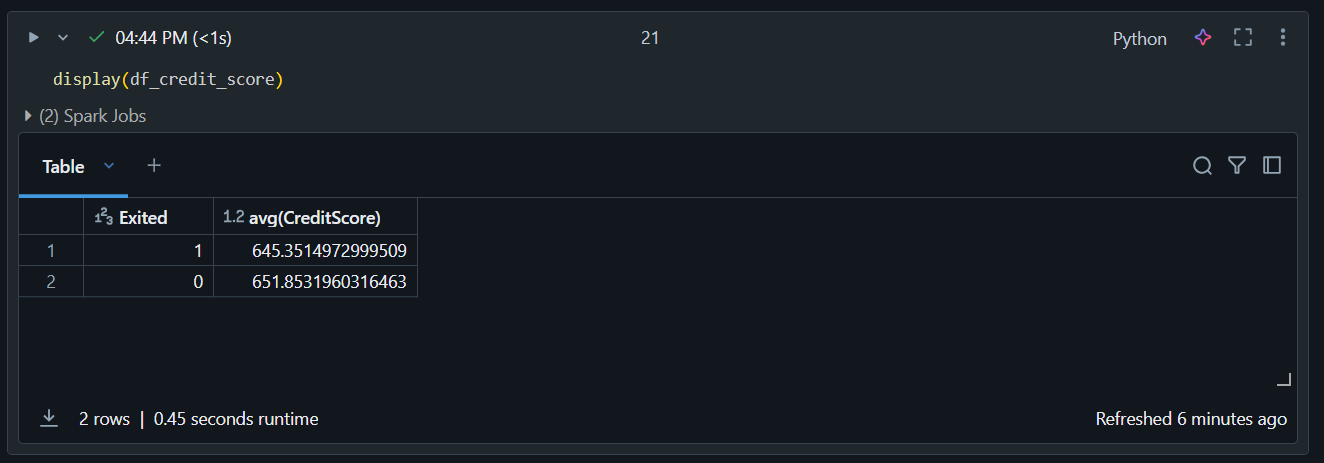
This analysis explores the relationship between customer credit scores and churn behavior. By comparing the average credit score of customers who have exited (churned) versus those who have stayed, businesses can identify whether credit score plays a significant role in customer retention.

**Objective**

* The aim is to calculate the average **Credit Score** for churned (Exited = 1) and retained (Exited = 0) customers. This comparison helps businesses assess if credit score correlates with a customer's likelihood to leave the service.
* This credit score analysis offers valuable insights into how customer credit scores may influence their decision to stay or leave. By identifying trends in credit scores, businesses can take proactive steps to enhance retention efforts, tailor services, and minimize churn.

|  |
| --- |
| df\_credit\_score = df.groupBy("Exited").agg({"CreditScore": "avg"})  df\_credit\_score.show() |





**7. Tenure Impact on Churn :**

This analysis examines how the length of a customer's tenure (the time they have been with the service) influences their likelihood to churn. By analyzing tenure in relation to churn rates, businesses can identify patterns and trends that help in developing strategies to retain customers for longer periods.

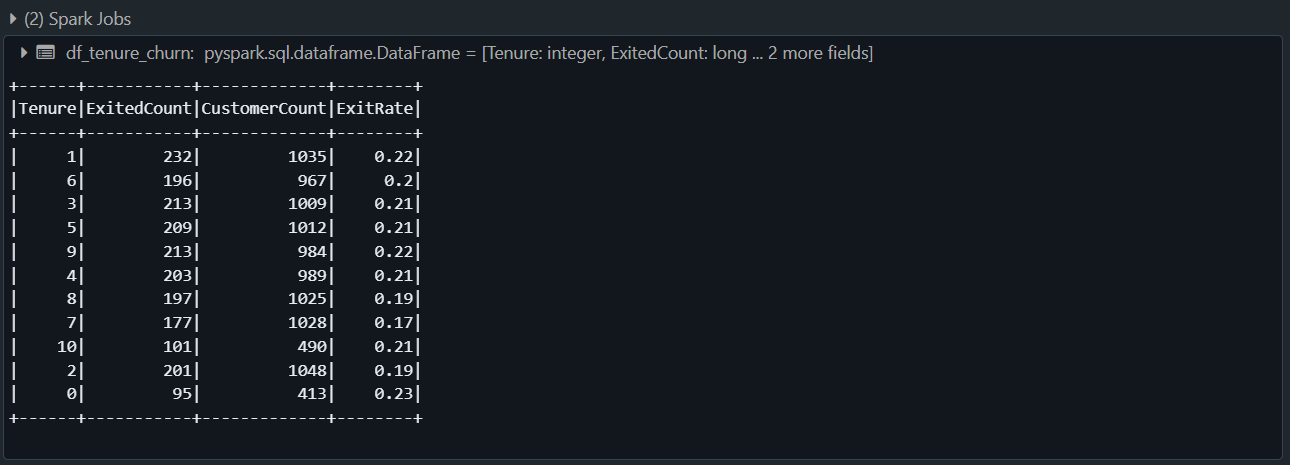
**Objective**

The goal is to assess whether customers with longer or shorter tenures have different churn behaviors by calculating:

1. **Customer Count**: The total number of customers at each tenure level.
2. **Exited Count**: The number of customers who have churned within each tenure group.
3. **Exit Rate**: The proportion of churned customers relative to the total customers for each tenure.

By analyzing how tenure impacts churn, businesses can identify which stages of customer life cycles are most vulnerable to churn. This allows them to implement proactive measures to enhance retention, optimize customer engagement, and maximize long-term customer loyalty.

|  |
| --- |
| df\_tenure\_churn = df.groupBy("Tenure").agg(  {"Exited": "sum", "CustomerId": "count"}  ).withColumnRenamed("sum(Exited)", "ExitedCount") \  .withColumnRenamed("count(CustomerId)", "CustomerCount") \  .withColumn("ExitRate", round(col("ExitedCount") / col("CustomerCount"), 2))  df\_tenure\_churn.show() |





**8. Product Ownership and Churn :**

This analysis explores the relationship between the number of products a customer owns and their likelihood to churn. By examining how product ownership correlates with churn behavior, businesses can identify whether customers with more or fewer products are more likely to stay or leave.

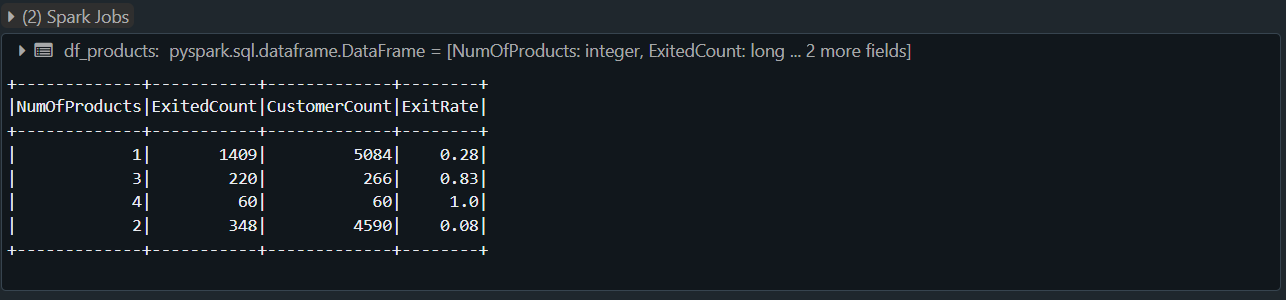
**Objective**

The aim is to analyze the impact of the number of products a customer owns on their likelihood to churn by calculating:

1. **Customer Count:** The total number of customers who own a specific number of products.
2. **Exited Count:** The number of customers who have churned in each product ownership category.
3. **Exit Rate:** The proportion of churned customers relative to the total customers in each product ownership group.

This analysis helps businesses understand the link between product ownership and churn, enabling them to identify opportunities for customer retention, cross-selling, and product bundling. By targeting customers based on their product ownership, businesses can create more effective strategies to enhance loyalty and reduce churn.

|  |
| --- |
| df\_products = df.groupBy("NumOfProducts").agg(  {"Exited": "sum", "CustomerId": "count"}  ).withColumnRenamed("sum(Exited)", "ExitedCount") \  .withColumnRenamed("count(CustomerId)", "CustomerCount") \  .withColumn("ExitRate", round(col("ExitedCount") / col("CustomerCount"), 2))  df\_products.show() |





**9. Salary vs. Exit**

This analysis explores the relationship between customer salaries and their likelihood of exiting (churning) the service. By comparing the average salary of customers who have churned (Exited = 1) versus those who have stayed (Exited = 0), businesses can determine if salary plays a role in the likelihood of customer churn.

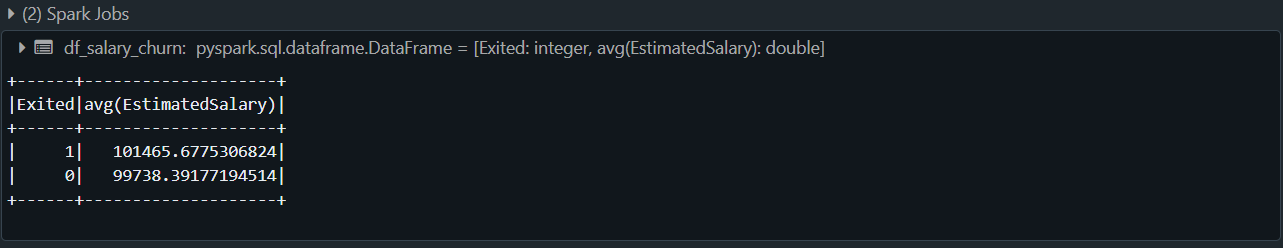
### **Objective**

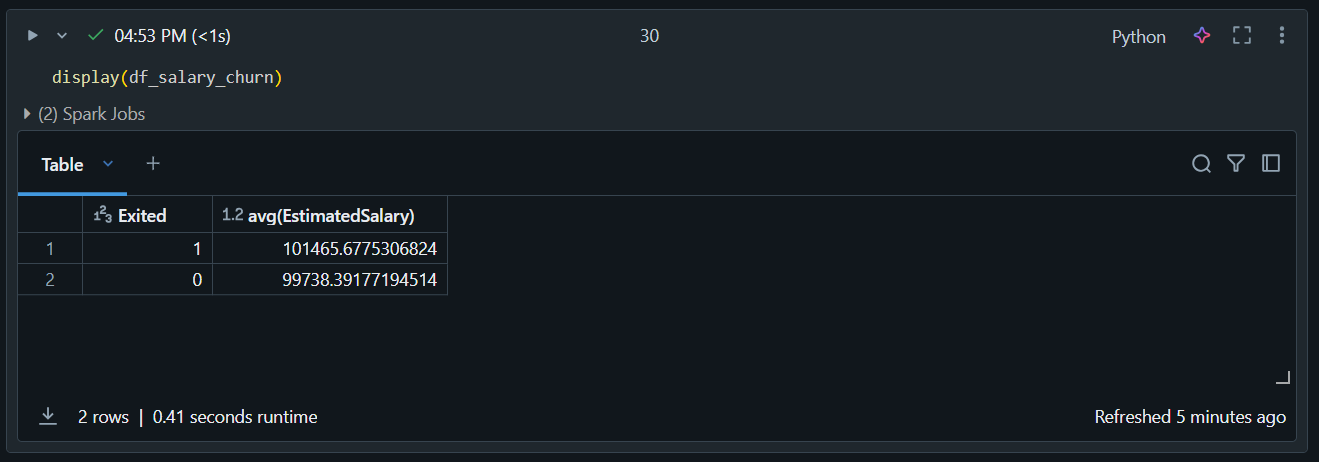
The goal is to assess whether salary levels correlate with customer retention or churn by calculating:

1. **Exited Group Average Salary:** The average salary of customers who have exited (churned).
2. **Retained Group Average Salary:** The average salary of customers who have stayed.

This analysis provides valuable insights into how salary might influence customer churn. By understanding salary patterns in relation to churn behavior, businesses can design more effective retention strategies, financial assistance programs, and personalized offers to improve customer loyalty and reduce churn rates.

|  |
| --- |
| df\_salary\_churn = df.groupBy("Exited").agg({"EstimatedSalary": "avg"})  df\_salary\_churn.show() |





**10. Gender-wise Churn Analysis**

This analysis investigates whether there are any gender-based differences in customer churn rates. By comparing the churn rates for male and female customers, businesses can identify if gender influences the likelihood of customers leaving the service and take action accordingly.

**Objective**

The goal is to analyze churn behavior based on gender by calculating:

1. **Customer Count:** The total number of male and female customers.
2. **Exited Count:** The number of male and female customers who have churned.
3. **Exit Rate:** The proportion of churned customers within each gender group.

This gender-based churn analysis helps businesses understand if gender influences customer retention. By identifying whether male or female customers are more likely to churn, businesses can develop targeted strategies, improve customer satisfaction, and reduce churn rates for specific gender groups.

|  |
| --- |
| df\_gender\_churn = df.groupBy("Gender").agg(  {"Exited": "sum", "CustomerId": "count"}  ).withColumnRenamed("sum(Exited)", "ExitedCount") \  .withColumnRenamed("count(CustomerId)", "CustomerCount") \  .withColumn("ExitRate", round(col("ExitedCount") / col("CustomerCount"), 2))  df\_gender\_churn.show() |

