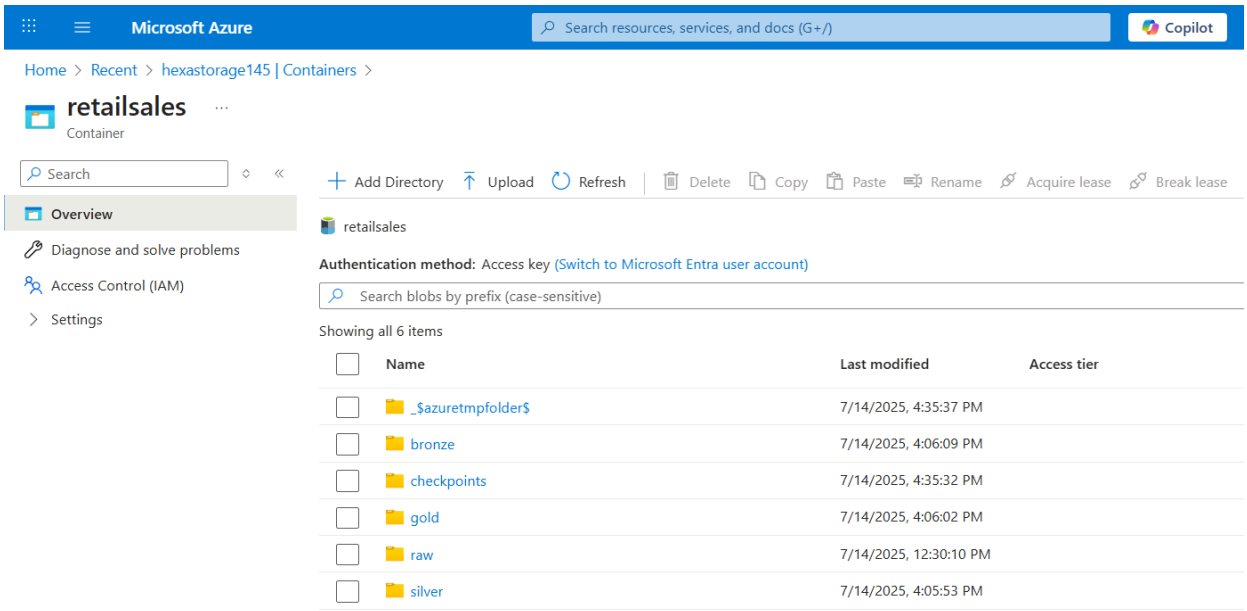
Retail Sales Management - Documentation

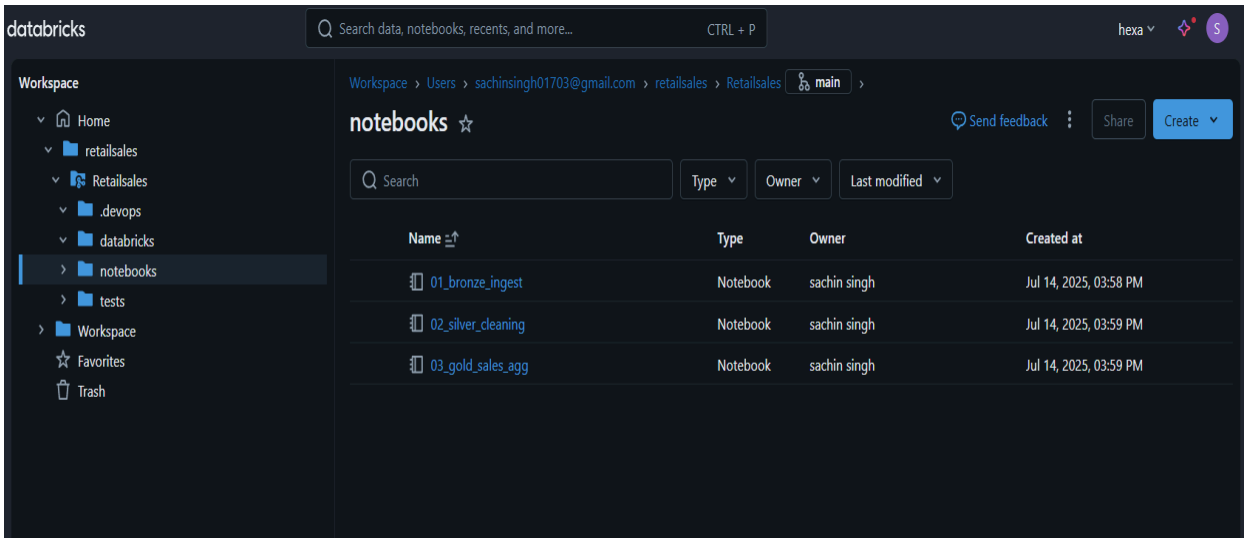
This document outlines the steps and components of a **Retail Sales ETL and ML Pipeline** implemented using **Azure Databricks**, **Delta Lake**, **MLflow**, and **Azure DevOps**. Screenshots are included to visualize each stage of the process.

1. 📦 Storage Setup



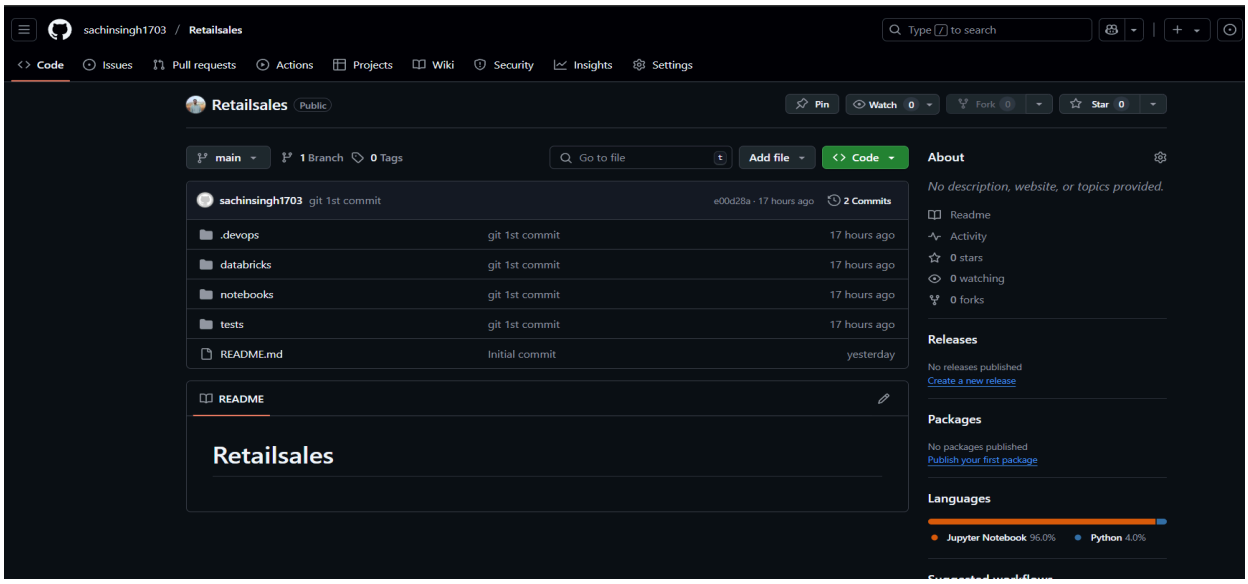
The foundational step is setting up storage for raw and processed data. ADLS Gen2 is used with directories like `/bronze`, `/silver`, and `/gold`.

2. 📓 Notebooks Overview



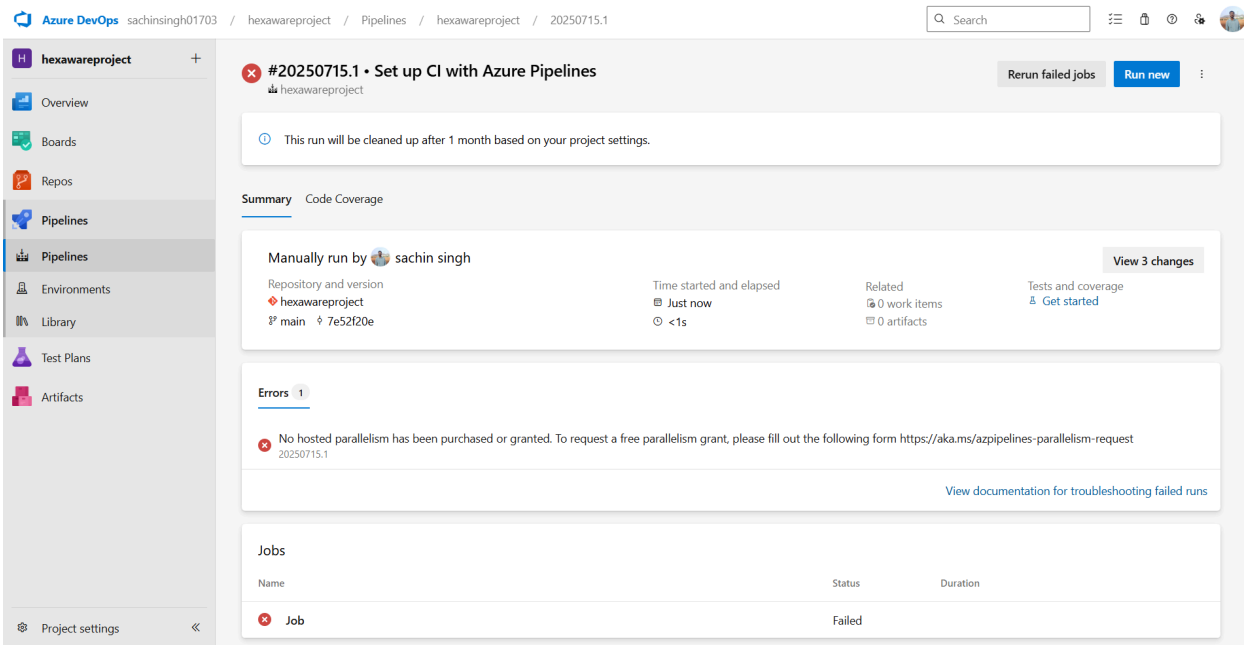
Azure Databricks notebooks were used for ETL and ML tasks. Each notebook handles a specific stage—Bronze, Silver, and Gold layers.

3. 🗂️ Files Uploaded to GitHub



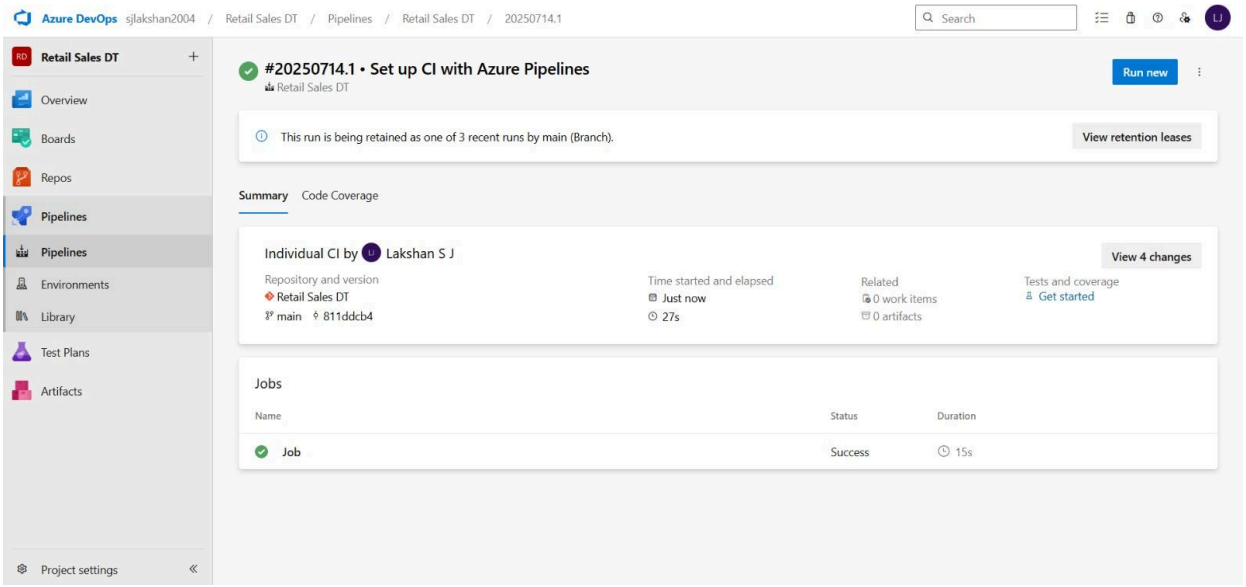
All notebooks and scripts were version-controlled and stored in GitHub for CI/CD via Azure DevOps.

4. ❌ Failed DevOps Pipeline



Early pipeline runs failed due to config or code issues, which were later resolved.

5. ✅ Successful DevOps Pipeline



After resolving errors, pipelines were successfully triggered to deploy notebooks into Databricks.

6. 🟫 Bronze Ingestion Code

19 hours ago 2

```python
from pyspark.sql.types import *
from pyspark.sql.functions import input_file_name

# Define schema
schema = StructType([
    StructField("Store", IntegerType()),
    StructField("DayOfWeek", IntegerType()),
    StructField("Date", StringType()),
    StructField("Sales", DoubleType()),
    StructField("Customers", IntegerType()),
    StructField("Open", IntegerType()),
    StructField("Promo", IntegerType()),
    StructField("StateHoliday", StringType()),
    StructField("SchoolHoliday", IntegerType())
])

input_path = "/mnt/retailsales/raw/"
bronze_path = "/mnt/retailsales/bronze/train/"
checkpoint_path = "/mnt/retailsales/checkpoints/bronze/"

df = (spark.readStream
    .format("cloudFiles")
    .option("cloudFiles.format", "csv")
    .option("header", "true")
    .option("cloudFiles.useNotifications", "false")
    .schema(schema)
    .load(input_path)
    .withColumn("source_file", input_file_name())
)

(df.writeStream
    .format("delta")
    .outputMode("append")
    .option("checkpointLocation", checkpoint_path)
    .start(bronze_path)
)
```

(1) Spark Jobs

0ea039d7-0a77-4b79-9b6c-70b1657ec267 *Last updated: 19 hours ago*

df: pyspark.sql.dataframe.DataFrame = [Store: integer, DayOfWeek: integer ... 8 more fields]

```
<pyspark.sql.streaming.query.StreamingQuery at 0x7f4ca8942ae0>
```

Initial ingestion of raw CSVs from storage into Delta Lake using Databricks Autoloader.

7. ⚙️ Silver Cleaning Code

```
19 hours ago (55s)                                                    1
from pyspark.sql.functions import col, to_date

bronze_df = spark.read.format("delta").load("/mnt/retailsales/bronze/train/")

silver_df = (bronze_df
    .filter(col("Open") == 1)
    .withColumn("Date", to_date("Date", "yyyy-MM-dd"))
    .withColumn("StateHoliday", col("StateHoliday").cast("string"))
)

silver_df.write.format("delta").mode("overwrite").save("/mnt/retailsales/silver/train/")

(2) Spark Jobs
bronze_df:  pyspark.sql.dataframe.DataFrame = [Store: integer, DayOfWeek: integer ... 8 more fields]
silver_df:  pyspark.sql.dataframe.DataFrame = [Store: integer, DayOfWeek: integer ... 8 more fields]
```

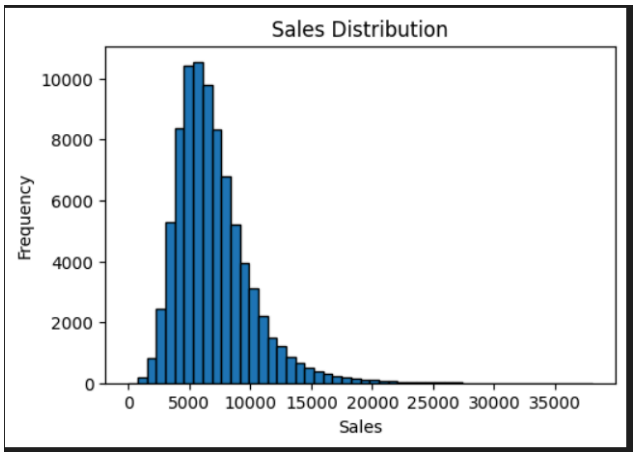
Data cleaning and transformations were applied in this stage, including null handling and formatting.

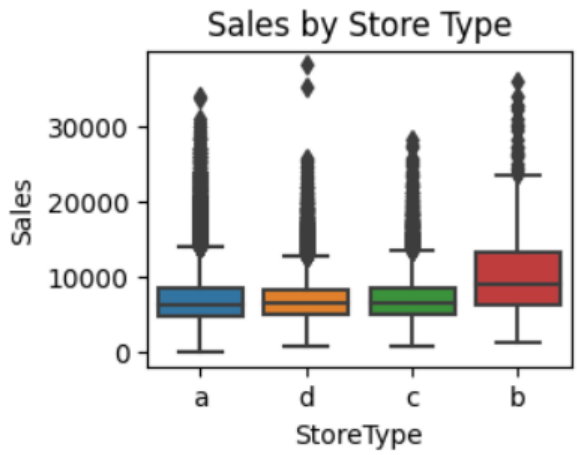
8. 🟨 Gold Sales Code

```
19 hours ago (42s)                                                    1
from pyspark.sql.functions import sum, avg, count

silver_df = spark.read.format("delta").load("/mnt/retailsales/silver/train/")

gold_df = (silver_df
    .groupBy("Store")
    .agg(
        sum("Sales").alias("TotalSales"),
        avg("Customers").alias("AvgCustomers"),
        count("Date").alias("DaysOpen")
    )
)

gold_df.write.format("delta").mode("overwrite").save("/mnt/retailsales/gold/store_sales_summary/")

(3) Spark Jobs
gold_df:  pyspark.sql.dataframe.DataFrame = [Store: integer, TotalSales: double ... 2 more fields]
silver_df:  pyspark.sql.dataframe.DataFrame = [Store: integer, DayOfWeek: integer ... 8 more fields]
```

Final aggregations and metrics calculated and saved in gold tables for analytics and ML.

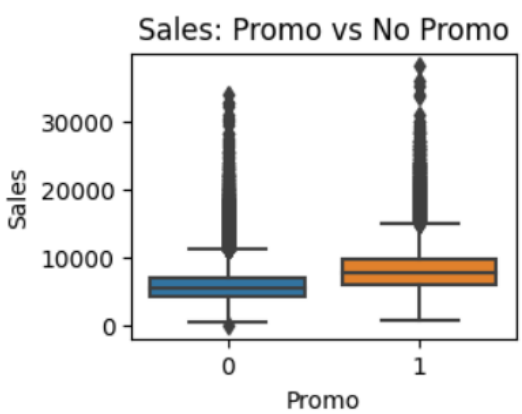
9. 📊 Sales Distribution

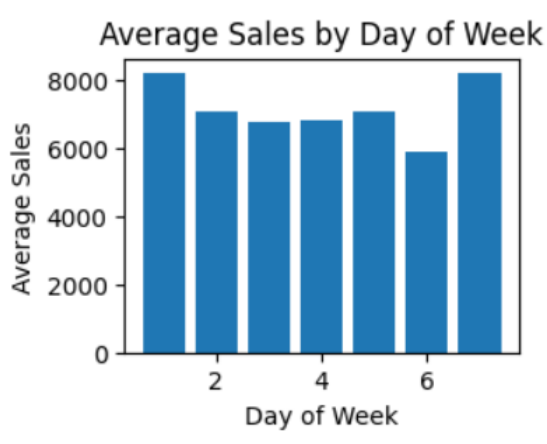




Exploratory data analysis (EDA) was conducted on gold data, visualizing trends and distributions.

10. 🤖 Model Training in Progress





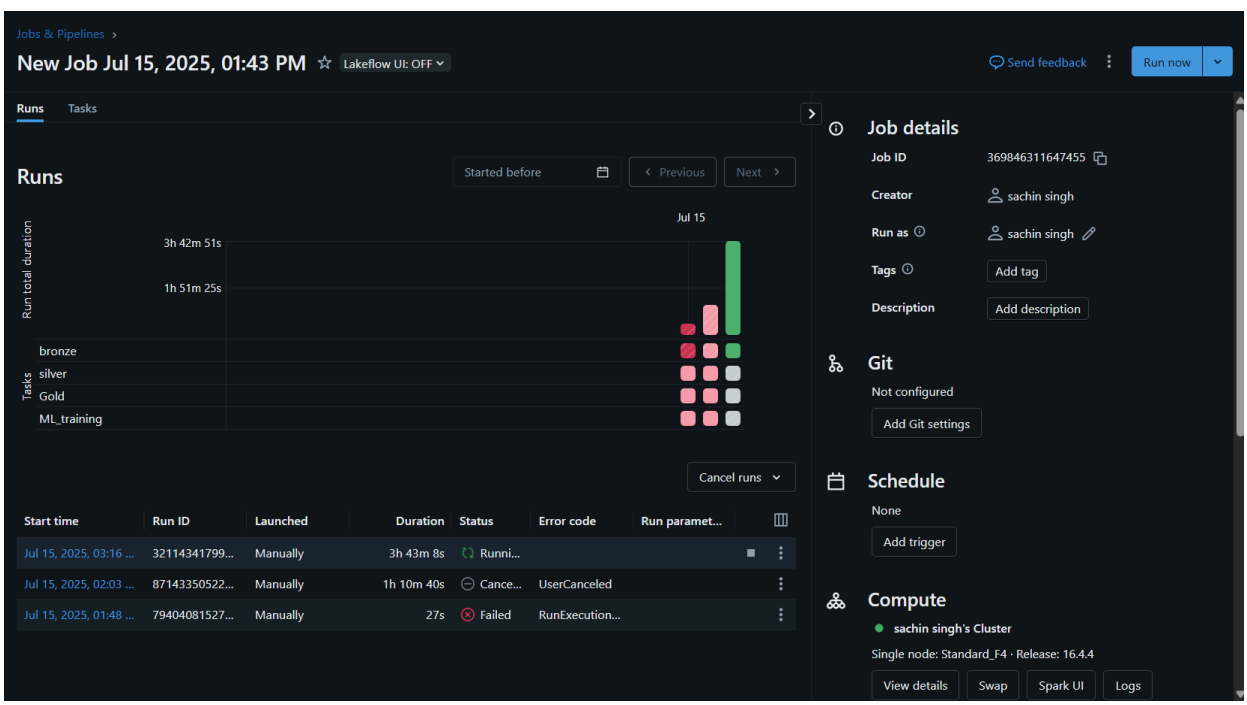
Initial model training phase where regression models are trained to predict sales.

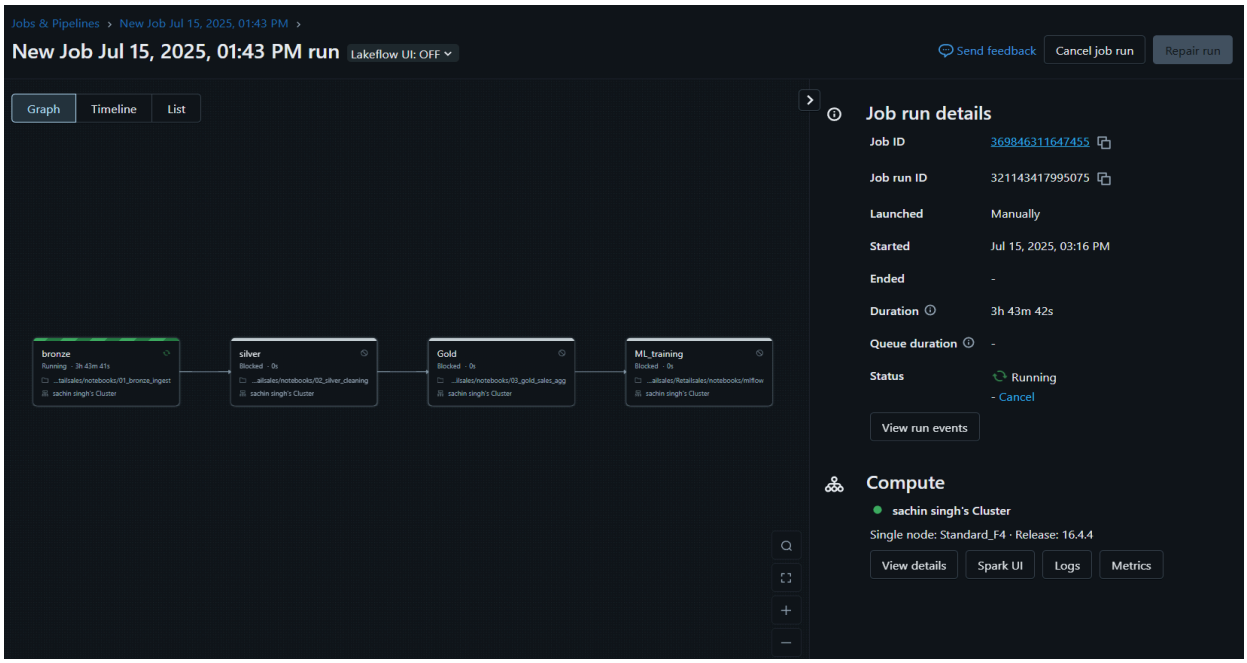
11. ✅ Model Training Completed

```
Model Summary:
    - Algorithm: Linear Regression
    - Accuracy: 91.9%
    - RMSE: 885
    - MAE: 545
    - Features: 20
    - Training Records: 675,212
    - Test Records: 169,180
```

Successfully trained model is logged and registered using MLflow.

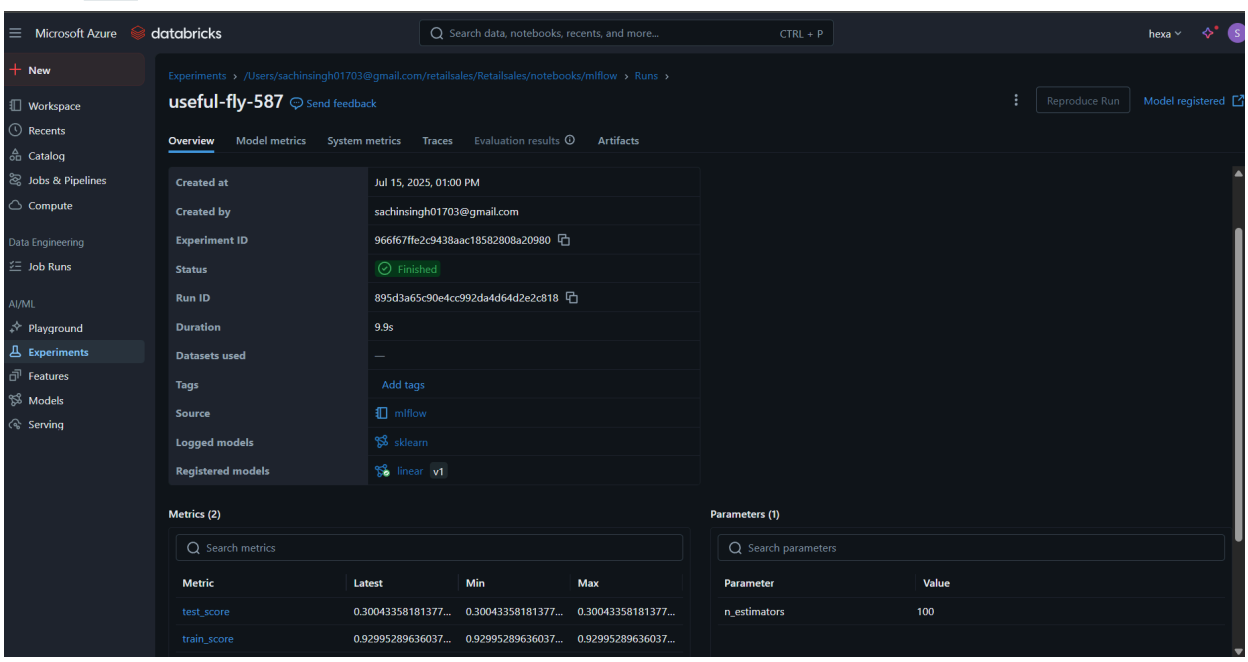
12. 🛠️ Pipeline Creation in Databricks





Workflows and jobs were orchestrated in Databricks for automation.

13. 📈 MLflow Tracking



All training runs, parameters, and metrics were tracked using MLflow.