**Data Lake Analytics**

**Team g : Tejaswini Gokanakonda - DE142 & Podutur Lahari - DE126**

**Date : 14-12-2024**

**Project Overview:**

The project involves building a scalable data pipeline using **Azure Data Factory (ADF), Azure Data Lake Storage (ADLS),** and **Azure Databricks.** ADF facilitates data extraction from various sources and moves it to ADLS, where data is organized into raw, curated, and processed zones with logical folder structures and partitioning. Azure Databricks is used for advanced data processing and analytics, including data cleansing, transformations, and machine learning model development. The pipeline ensures secure data handling with **Role-Based Access Control (RBAC)** and optimized workflows for querying and analytics. This solution supports real-time and batch processing, enabling actionable insights and business intelligence.

**Data Lake Analytics Overview :**

Azure Data Lake Analytics is an on-demand analytics service that simplifies big data processing by allowing users to write and execute massively parallelized queries over petabytes of data. It works seamlessly with Azure Data Lake Storage, enabling data to be processed directly without the need to move or transform it. Using a U-SQL-based query language, it combines the benefits of SQL with the extensibility of C#. This serverless service eliminates the need to provision hardware, scaling resources dynamically based on workload. Ideal for log analysis, ETL processes, machine learning, and real-time analytics, Data Lake Analytics is cost-effective, as users only pay for the processing power they consume.

**Dataset Overview :**

**Dataset Source :** Kaggle  
**Dataset Size :** 500 rows and 15 columns

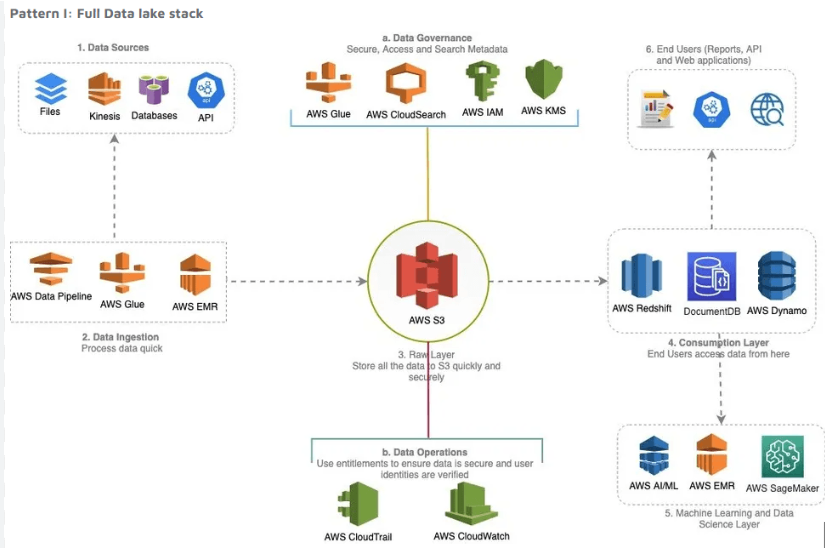
#### **Description:**

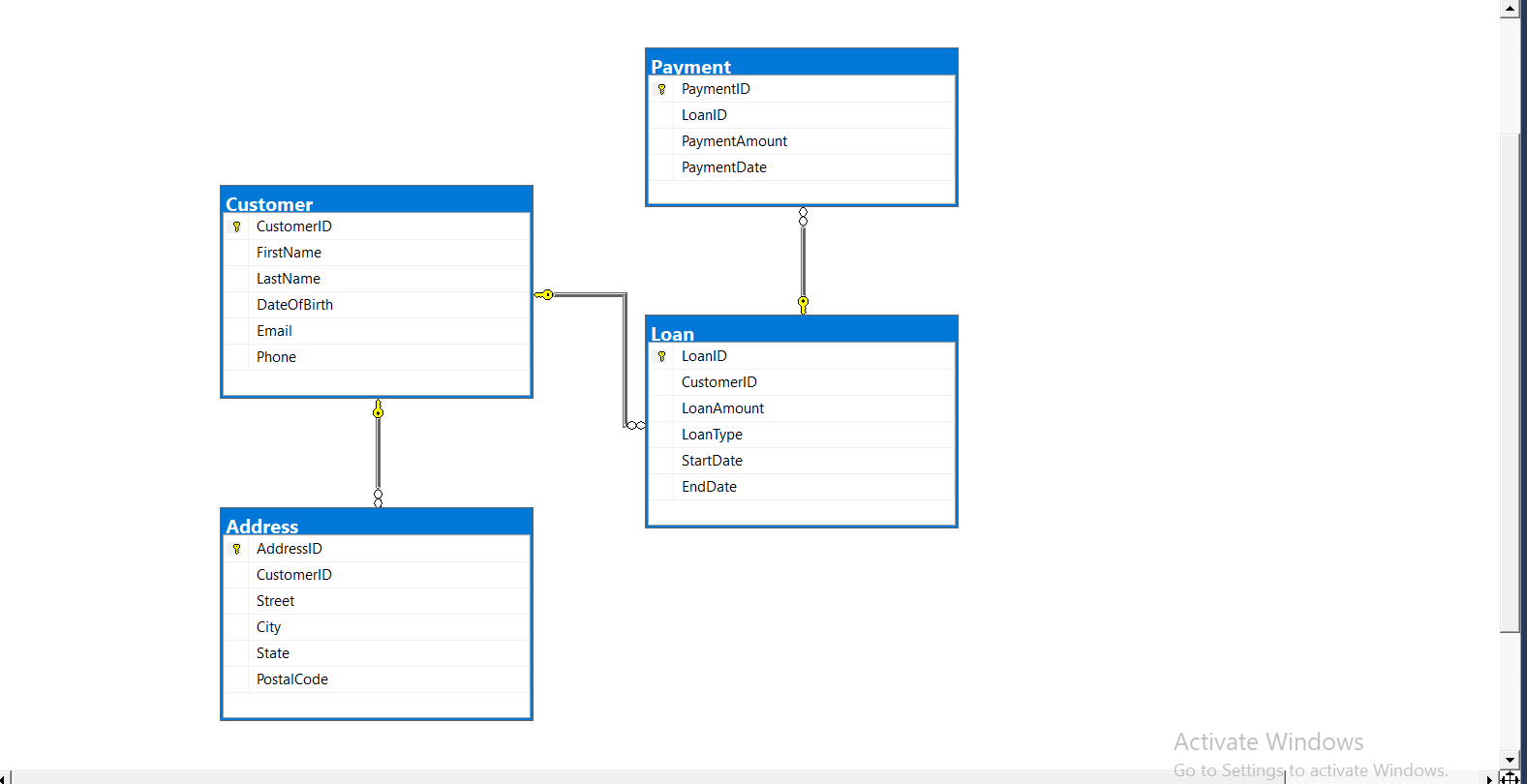
The dataset contains customer data related to loans, financial behavior, and repayment trends. It includes demographic information, financial metrics, and loan-specific insights, making it ideal for analyzing creditworthiness, financial activity, and loan default risks.

#### **Columns:**

* **Customer\_ID** : Unique identifier for each customer.
* **Age :** Customer's age.
* **Gender :** Customer's gender.
* **Occupation :** Customer's profession.
* **Marital Status :** Marital status of the customer.
* **Family Size :** Number of family members.
* **Income :** Customer's monthly income.
* **Expenditure :** Monthly expenditure of the customer.
* **Use Frequency :** Frequency of using financial services.
* **Loan Category :** Type of loan taken (e.g., Housing, Gold Loan).
* **Loan Amount :** Amount of loan sanctioned.
* **Overdue :** Number of overdue loan repayments.
* **Debt Record :** Outstanding debt of the customer.
* **Returned Cheque :** Number of returned cheques.
* **Dishonour of Bill :** Instances of bill dishonoring

**Architecture Diagram :**



**ER - Diagram :** 

**How it works :**

### **Source Data Files :**

The dataset provided is structured customer data related to loan activities and financial behavior. The data is organized as follows:

| **File Name** | **File Type** | **Description** |
| --- | --- | --- |
| CustomerDetails | CSV | Contains demographic details such as Customer ID, Age, Gender, Occupation, and Marital Status. |
| FinancialMetrics | CSV | Provides financial data including Income, Expenditure, Loan Amount, and Debt Record. |
| LoanCategories | CSV | Specifies the type of loan taken (e.g., Housing, Gold Loan, Automobile). |
| OverdueRecords | Single Line JSON | Tracks loan repayment issues such as Overdue payments, Returned Cheques, and Dishonored Bills. |
| ActivityFrequency | Multi Line JSON | Details usage patterns, including the frequency of service use and loan-related activities. |

This structured dataset supports ingestion, preprocessing, and analysis for developing insights into customer behavior, loan default risks, and financial trends.

**Execution Overview :**

* The Loan Data Analytics project processes customer loan data by ingesting demographic, financial, and loan-specific information into Azure Databricks. Data is sourced from CSV and JSON files stored in Azure Data Lake Storage or Blob Storage.
* Using PySpark, the data is preprocessed and transformed to clean inconsistencies, handle missing values, and calculate metrics like overdue rates, debt-to-income ratios, and customer loan usage trends. The processed data is stored in Delta Lake for efficient querying and analytics.
* Insights are visualized using Databricks Notebooks or integrated visualization tools like Power BI, enabling stakeholders to monitor customer behavior, predict loan default risks, and track financial activity trends. The solution is optimized for scalability using Databricks clusters and supports monitoring and alerting features for operational efficiency.

**ETL Pipeline :**

**1. Ingestion Pipeline (Bronze Zone):**

**Data Sources:** Raw data files (CSV, JSON) are ingested, including customer details (demographics), financial data (income, loan balance), and activity logs (overdue records, debt status).

**Transformations:** Minimal transformations such as dropping unnecessary columns, renaming headers, applying schema, and adding audit columns (ingestion\_date, file\_source, and file\_date).

**Output:** The processed raw data is stored in Delta Lake within the Bronze Zone.

**2. Transformation Pipeline (Silver to Gold Zone):**

**Input:** Preprocessed data from the Bronze Zone.

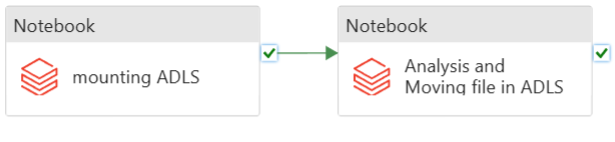
**Transformations:** Data is cleansed through deduplication, enriched with external data (e.g., credit score), joined with relevant tables, and aggregated to create insights such as overdue trends, financial health metrics, and customer behavior models.

**Output:** The transformed data is saved in Delta Lake in the Gold Zone for easy querying and analytics.

**3. Automation and Scheduling:**

The pipeline is scheduled to run every Monday at 9 PM using Azure Data Factory (ADF). If no new data is available, the pipeline skips execution.

A Tumbling Window Trigger is used to execute both the ingestion and transformation pipelines, with file\_date as the parameter.

This ETL pipeline ensures smooth, automated data flow from raw ingestion to enriched data, ready for real-time analytics.

**Project Requirements :**

**Azure Databricks**:

* **Apache Spark** is used for data processing, cleaning, and transformations.
* **Databricks SQL** is utilized for querying and transforming the data into the final dimensional model.
* **Notebooks** are employed for real-time data visualization and analytics.

**Azure Data Factory (ADF)**:

* **ADF** orchestrates and schedules the ETL pipelines (ingestion and transformation).
* **Tumbling Window Trigger** is configured for scheduling data processing.
* **Pipeline Scheduling**: Runs every Monday at 9 PM.

**Azure Data Lake Storage / Azure Blob Storage**:

* **Azure Data Lake Storage (ADLS)** and **Blob Storage** are used to store raw and processed data in formats like CSV, JSON, and Delta.
* Data is ingested into **Blob Storage** and stored in **Delta Lake** for efficient processing and querying.

**Azure Event Hubs / Apache Kafka**:

* Used for real-time data streaming to ingest data from various sources (e.g., loan activity logs, customer transactions).
* **Event Hubs** or **Apache Kafka** captures streaming data and feeds it into the **Azure Databricks** environment for processing.

**Azure SQL Database / Delta Lake**:

* Stores processed data in a structured format, such as tables for dimensional models and insights.
* **Delta Lake** ensures scalable storage with ACID transactions for data consistency and integrity.

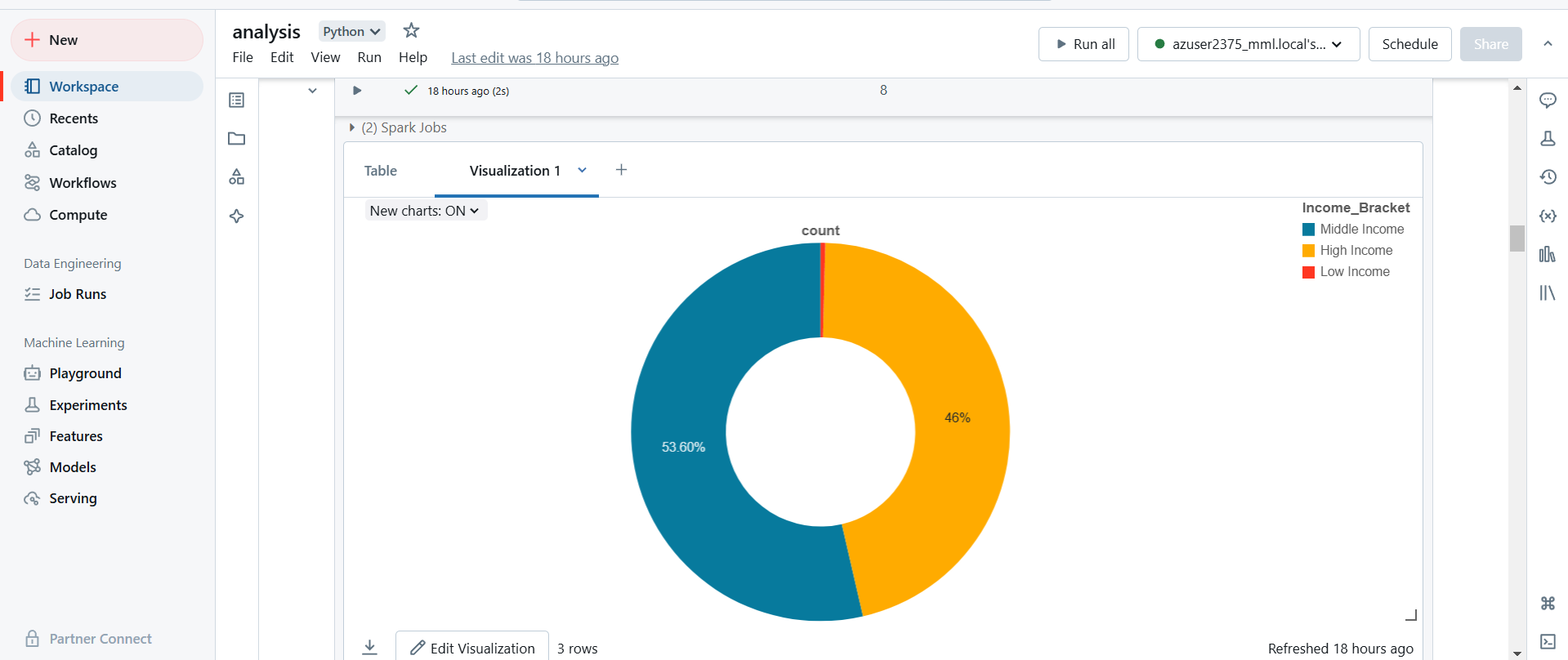
**Power BI / Tableau**:

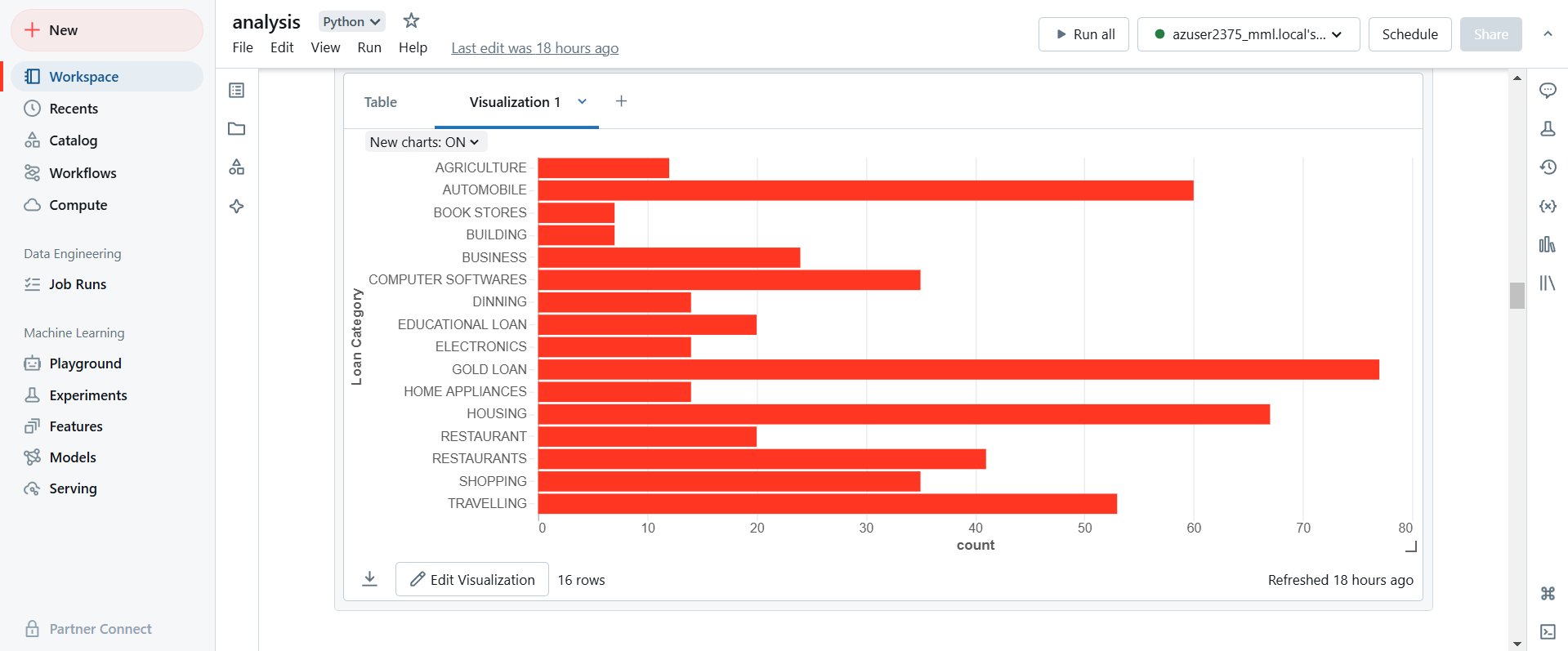
* Used for data visualization and building interactive dashboards that provide real-time insights from the processed data.

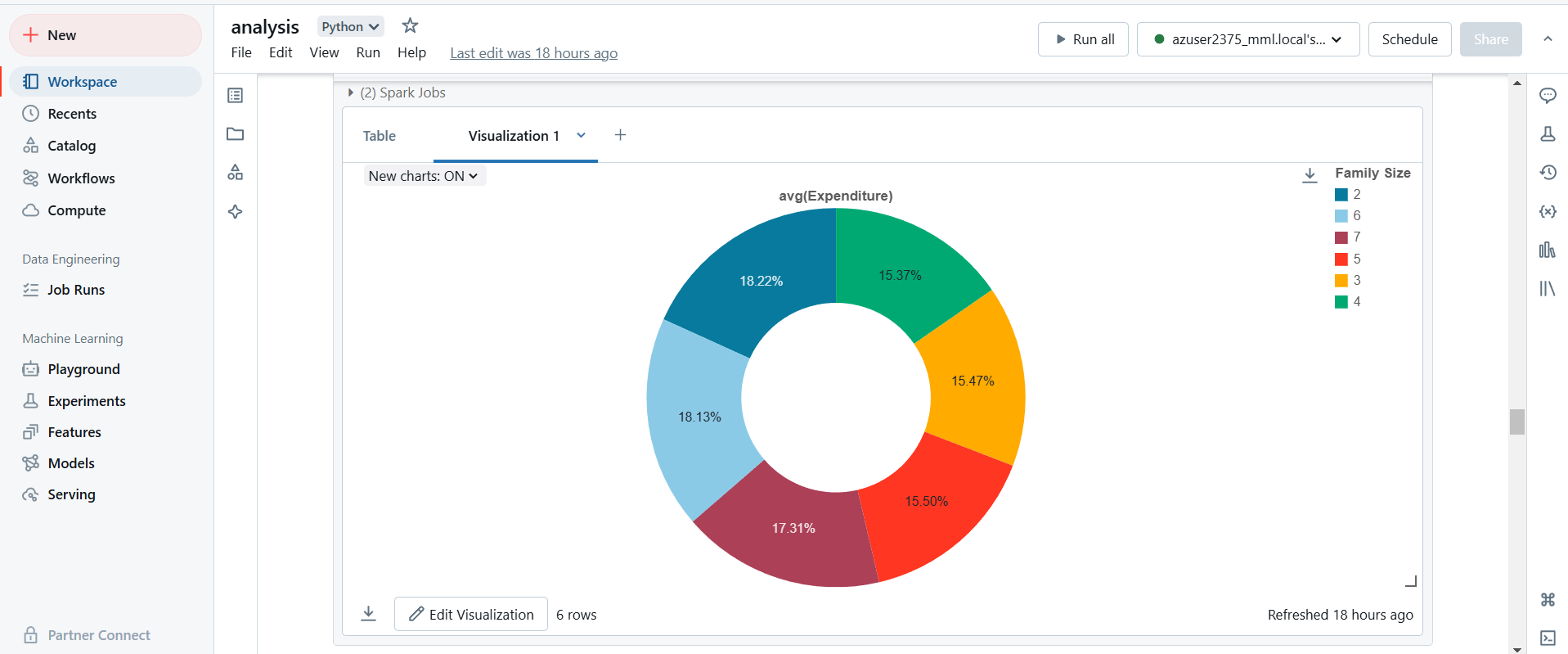
**Azure Monitor**:

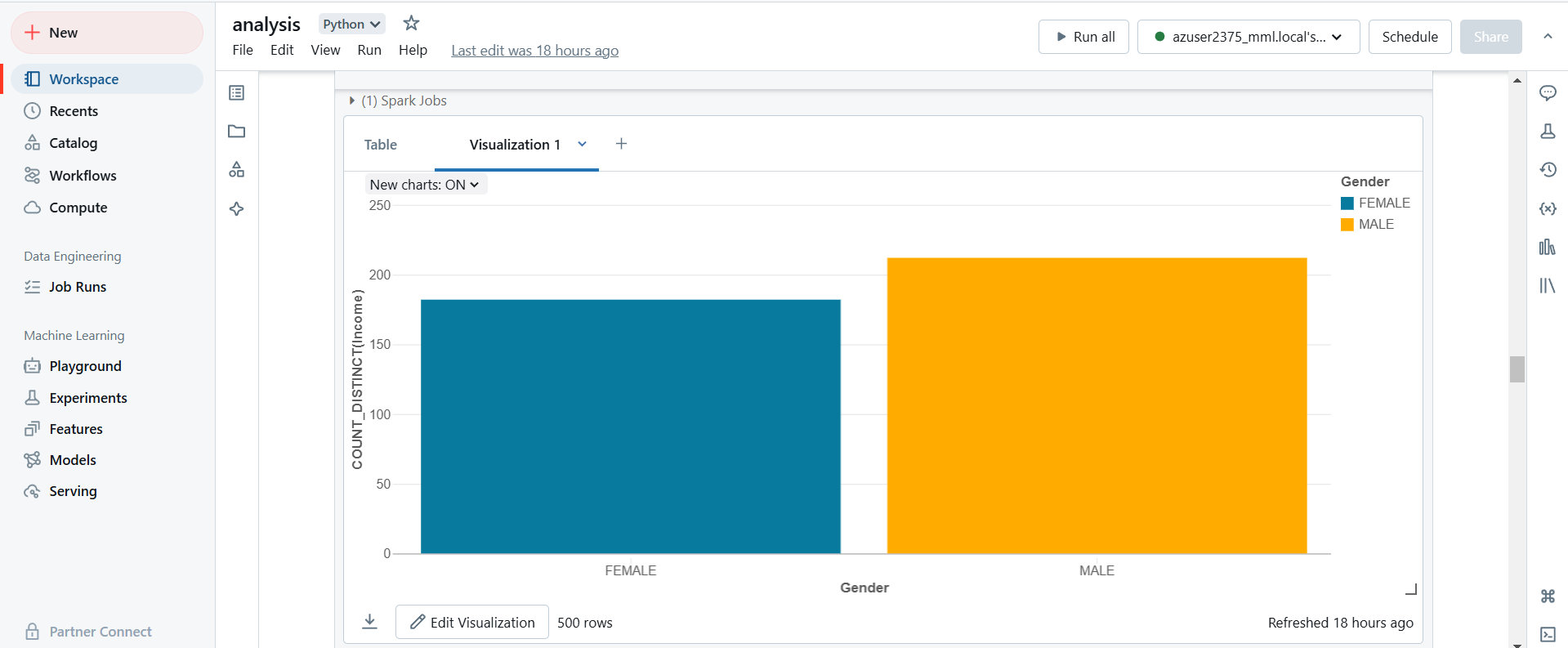
* Monitors the pipeline execution, logs, and triggers real-time alerts for job failures or processing anomalies, ensuring smooth pipeline operation and quick issue resolution.

**Analysis Results:**









**Tasks Performed :**

* **Solution Architecture :** Designed a solution architecture for loan data analytics using Azure Databricks, Azure Data Lake Gen2, Azure Data Factory, and Power BI for real-time data processing and visualization.
* **Azure Databricks Setup :** Created and managed the Azure Databricks service, establishing a seamless architecture within Azure to ensure smooth integration for processing customer loan data.
* **Databricks Notebooks :** Worked with Databricks notebooks, utilizing Databricks utilities, magic commands, and various data transformation techniques to process loan-related data in real-time.
* **Notebook Workflows :** Passed parameters between notebooks and created notebook workflows to automate data processing tasks for continuous data transformation.
* **Databricks Cluster Management :** Configured, monitored, and optimized Databricks clusters, cluster pools, and jobs for scalable data processing, ensuring real-time analytics capabilities.
* **Secure Storage Access:** Mounted Azure Storage in Databricks using secrets stored in Azure Key Vault for secure access to raw and processed data.
* **Databricks Tables and DBFS:** Worked with Databricks Tables and Databricks File System (DBFS) for efficient data storage and retrieval during processing tasks.
* **Delta Lake Implementation:** Used Delta Lake to implement a Lakehouse architecture for storing and processing loan data in real-time, enabling efficient querying and reliable data updates.
* **Power BI Dashboards:** Developed interactive dashboards in Power BI to visualize processed loan data from Databricks tables, providing actionable insights to business stakeholders.
* **Power BI Integration :** Connected Power BI to Azure Databricks tables for seamless data visualization, ensuring real-time reporting capabilities.

#### **Spark (PySpark and SQL):**

* **Spark Architecture:** Utilized Spark architecture, Data Sources API, and Dataframe API for processing both structured and unstructured loan data.
* **PySpark Data Ingestion:** Ingested CSV and JSON files into Azure Data Lake and transformed them into Parquet tables for optimized querying.
* **Data Transformation in PySpark:** Applied various transformations such as Filter, Join, Aggregations, GroupBy, and Window functions to prepare the loan data for in-depth analytics.
* **SQL for Data Processing:** Used Spark SQL to create databases, tables, and views for structured data processing and querying.
* **Efficient Data Processing:** Implemented full refresh and incremental load patterns using partitions for processing large volumes of data efficiently.

#### **Delta Lake:**

* **Data Operations:** Performed Read, Write, Update, Delete, and Merge operations in Delta Lake using both PySpark and SQL, ensuring data consistency and accurate tracking of loan data.
* **Delta Lake Features:** Leveraged Delta Lake features like History, Time Travel, and Vacuum for managing historical loan data and optimizing storage.
* **Parquet to Delta Conversion:** Converted Parquet files to Delta files for enhanced real-time data processing and performance improvements.
* **Incremental Data Loading:** Implemented incremental load patterns using Delta Lake to handle large loan data volumes efficiently.

#### 

#### **Azure Data Factory:**

* **Pipeline Creation:** Created ETL pipelines in Azure Data Factory to execute Databricks notebooks, automating the data processing and transformation tasks for loan data.
* **Pipeline Design:** Designed robust pipelines to handle unexpected scenarios, such as missing files or invalid data formats, ensuring smooth data flow through the system.
* **Activity Dependencies:** Created dependencies between activities and pipelines to ensure the correct execution sequence, enhancing pipeline reliability.
* **Pipeline Scheduling:** Scheduled pipelines using ADF triggers to execute at regular intervals, ensuring timely data processing and real-time analytics.
* **Pipeline Monitoring:** Monitored ADF triggers and pipelines to ensure successful execution, with error handling and alerts for prompt issue resolution.

**Technologies/Tools Used:**

* Pyspark
* Spark SQL
* Delta Lake
* Azure Databricks
* Azure Data Factory
* Azure Date Lake Storage Gen2
* Azure Key vault
* Power BI (Optional)