**Team 10: Instacart Product Recommendation System and Analysis using Azure Cosmos DB**

**Introduction:**

Instacart is a popular grocery delivery service that offers customers a convenient way to shop for groceries from the comfort of their own homes. With the growing demand for online shopping, Instacart is continually looking for ways to enhance their customer experience. In this project, we propose the development of an Instacart Product Recommendation System using Azure Cosmos DB, a globally distributed, multi-model database service.

This system will analyze customer purchase history and recommend products based on their preferences. We will also perform an analysis of the recommendation system to evaluate its effectiveness and identify potential areas for improvement. With this project, we aim to provide a more personalized and enjoyable shopping experience for Instacart customers while improving business efficiency and increasing revenue.

**Objectives:**

* Representing orders and prior orders as a document database to improve recommendation accuracy.
* Using a document-graph hybrid model to analyze and visualize relationships between customers and the products they ordered, visualize the location of products that are more likely to be reordered.
* Implementing a graph database model to visualize the relationships between customers, products, and product availability locations.
* Analyzing the details, frequency, and likelihood of the most frequently ordered products.

**Entity Relationship Diagram:**

Diagram

Description automatically generated

**Graph Data Model:**

Diagram

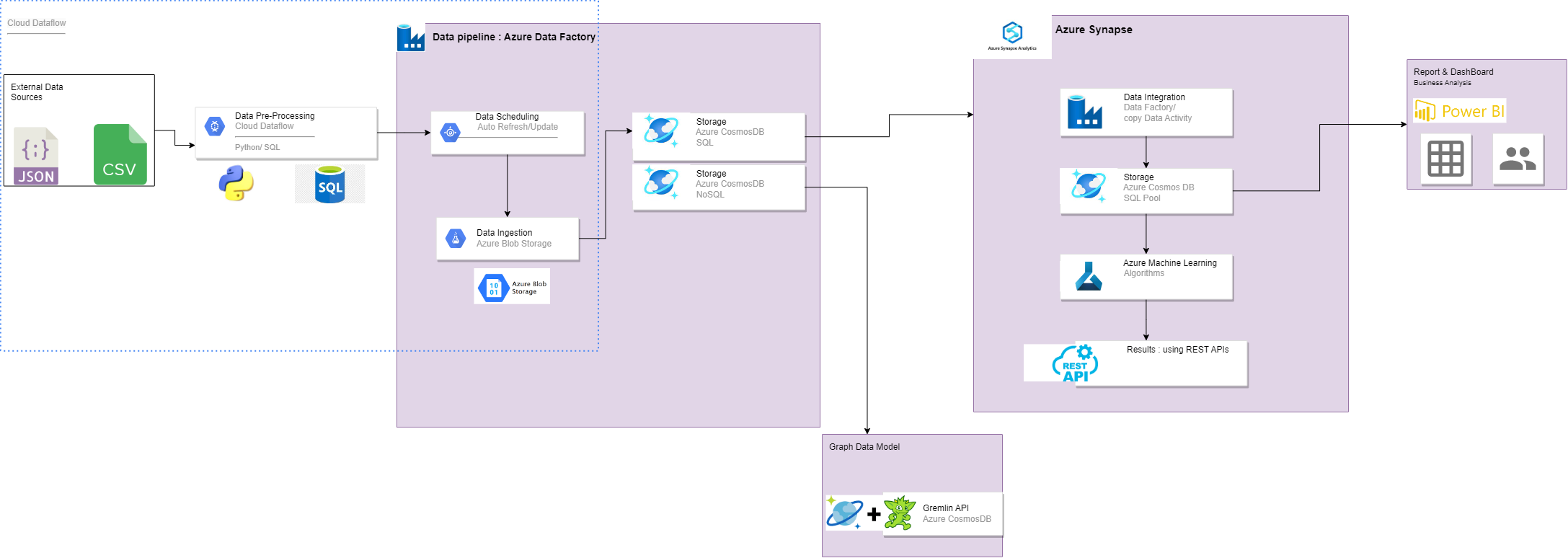
Description automatically generated

**Entity Relationship Diagram and Graph Data Model Explanation:**

Our ERD comprises of two sections, one is for document data model and the other one is for graph data model. Firstly, the document data model comprises of two tables namely order and order\_prod\_prior. In orders table, we have assigned order\_id to be a primary key and passing the same as foreign key to order\_prod\_prior table. In order\_prod\_prior table we have set the product\_id to be a primary key.

Secondly, the graph data model section is further classified into two sections - Graph model 1 and Graph model 2. Graph model 1 involves the product table, department table and aisle table. In the product table, we get the product\_id as a primary key and we are passing it as a foreign key to the order\_prod\_prior table of the document data model section. Also, we are getting dept\_id and aisle\_id as a primary key from department and aisle tables respectively and are passing the same as a foreign key to the products table. Moving forward, Graph model 2 involves the product table, cart, and user table. Starting from the users table, we are initially setting the user\_id as a primary key and we are passing the same as a foreign key to the cart table. Now in the cart table, cart\_id is set as a primary key and gets the foreign key, product\_id from the products table which is supposedly the primary key of that table.

**Architecture Diagram:**

****

**Technologies used:**

* Microsoft SQL Server as the database management system.
* Python for data cleaning and pre-processing
* Azure Data Factory for scheduling, data pipeline, and other ETL tasks.
* Azure Blob Storage for object storage solution for the cloud.
* Azure Cosmos DB for multi-model (relational and NoSQL) database service on the cloud.
* Azure Synapse for cloud data warehouse, storage, analytics, and machine learning.
* Azure Machine Learning for managing the machine learning project lifecycle.
* Apache Gremlin is the graph traversal language of Apache TinkerPop for Graph modeling.
* Power BI as an interactive data visualization software for reporting and dashboarding.
* Microsoft Excel and Outlook for project operation/execution.
* Draw.io for data modeling and data architecture design.

**Data Workflow:**

* Source of data is online from the Kaggle, an online community of data scientists and data repositories.
* Data cleaning and Preprocessing is to be done using Pandas library of Python Programming language.
* Data is then exported to SQL Server in a flat file format (e.g., CSV or JSON).
* Initial Data modeling is to be performed on the dataset locally.
* Data is then exported to Azure Blob Storage using Azure Portal, Azure Storage Explorer, or Azure CLI.
* Data is moved from Blob storage to Azure Cosmos DB by creating a data Pipeline in Azure Data Factory.
* Scheduler is run to regularly ingest data into Azure Cosmos DB.
* Non-relational data is then pushed to Apache TinkerPop Gremlin API to be used for graph data model implementation.
* Another Data pipeline is created to push the relational data from Azure Cosmos DB to Azure Synapse for training and evaluation of the recommendation system Machine Learning model.
* Data is split between training and test data sets and is Synapse SQL Pool to be trained for the model.
* Azure Machine Learning is to be used for model training and evaluation.
* Model is deployed in Azure Machine Learning as a web service which can be called using REST APIs.
* Simultaneously data from SQL Pool is to be used via direct query into Power BI for visualization, reporting, and dashboarding purposes.

**Future Scope:**

* The recommendation model can be implemented and integrated on a web application or mobile application for more realistic and application interface-based product recommendations. Example: Instacart mobile application.
* The recommendation model can be combined with more user data which could be further improved by integrating a hybrid recommendation system which involves collaborative filtering and content-based filtering.
* Since the model is scalable and available, the data pipeline and Azure data workflows could be more optimized as the number of users grow to millions of millions of users.

**Team Members:**

* Name - Shashank Dongre NUID - 002747740 Email - [dongre.s@northeastern.edu](mailto:dongre.s@northeastern.edu)
* Name - Ashwini Khedkar NUID - 002738717 Email – khedkar.as@northeastern.edu
* Name - Keerthanaa Kannan NUID - 001593845 Email - kannan.ke@northeastern.edu
* Name - Samradni Pathari NUID – 002655149 Email - pathari.s@northeastern.edu