# Advanced Analytics & Machine Learning over Regulated & Sensitive Data

# Business Needs & Requirements

There is a business need for customers operating in regulated industries to perform advanced analytics and machine learning in the cloud over tokenized data.

# Microsoft Solution

The solution below addresses the business needs to perform advanced analytics and machine learning in the cloud within the scope of data regulated industries.

With this sample solution, the data tokenization is requested from Power Apps as a user-friendly interface. We have developed a solution that addresses customer needs with the objectives below.

Reminder

This is a demonstration of a solution. The Microsoft solutions (Power Apps, Azure Synapse Analytics, Azure Machine Learning) that are used in this scenario may change open subject to customer need. This is a reference solution.

Microsoft can provide a sample data tokenization stored procedure. On the other hand, it is expected from customers to develop their own data tokenization stored procedure.

The machine learning prediction code developed in python is a sample in this demonstration. Microsoft does not provide a final solution. Customers must develop their own machine learning code.

# Reference Solution Architecture

# 

# Objectives:

* Data is masked with tokenization logic at on-premises servers/customer environment. In order to perform this task, there is a stored procedure at customer environment that performs data tokenization. Data tokenization is requested from an application (Data Tokenization App) that is developed on no code/low code app development platform - Microsoft Power Apps.
* In Power Apps application (Data Tokenization App), you select the source environment, tables and fields that need to be tokenized and also the target environment where tokenized data will be sent. A request is created after you select the required fields. An email is sent to admin group(s) (approvers). Once the admin group(s) approves the request, data tokenization stored procedure at customer environment is triggered for data tokenization. After data is tokenized, tokenized data is sent to Azure Synapse Analytics.

Data tokenization stored procedure at customer environment performs the data tokenization request and sends the tokenized/masked data to Azure Synapse Analytics for advanced analytics.

Azure Synapse Analytics Pipelines orchestrates below cycles from beginning to end:

* + Collect metadata information (which tables and columns can move to Azure , also which columns needs to be tokenized)
  + Trigger Token Generation Procedures on On-Premise Server (Token generation needs to be done at on-premise environment)
  + Copy tokenized data to Azure Synapse Analytics Storage Environment as a target. (Target can be changed if business needs change)
  + If any Tokenized results needs to move on-Premise
    - Copy Tokenized data to On-Premise Environment
    - Trigger Detokenize Procedure on On-Premise Server (this procedure replaces all tokens with original Value)

Azure Machine Learning Notebooks performs:

Experiment Works with Tokenized Data

Training Machine Learning models with Tokenized Data

Prediction with Tokenized Data, and write results into Synapse Analytics Workspace Storage

# Requirements for Demo platform

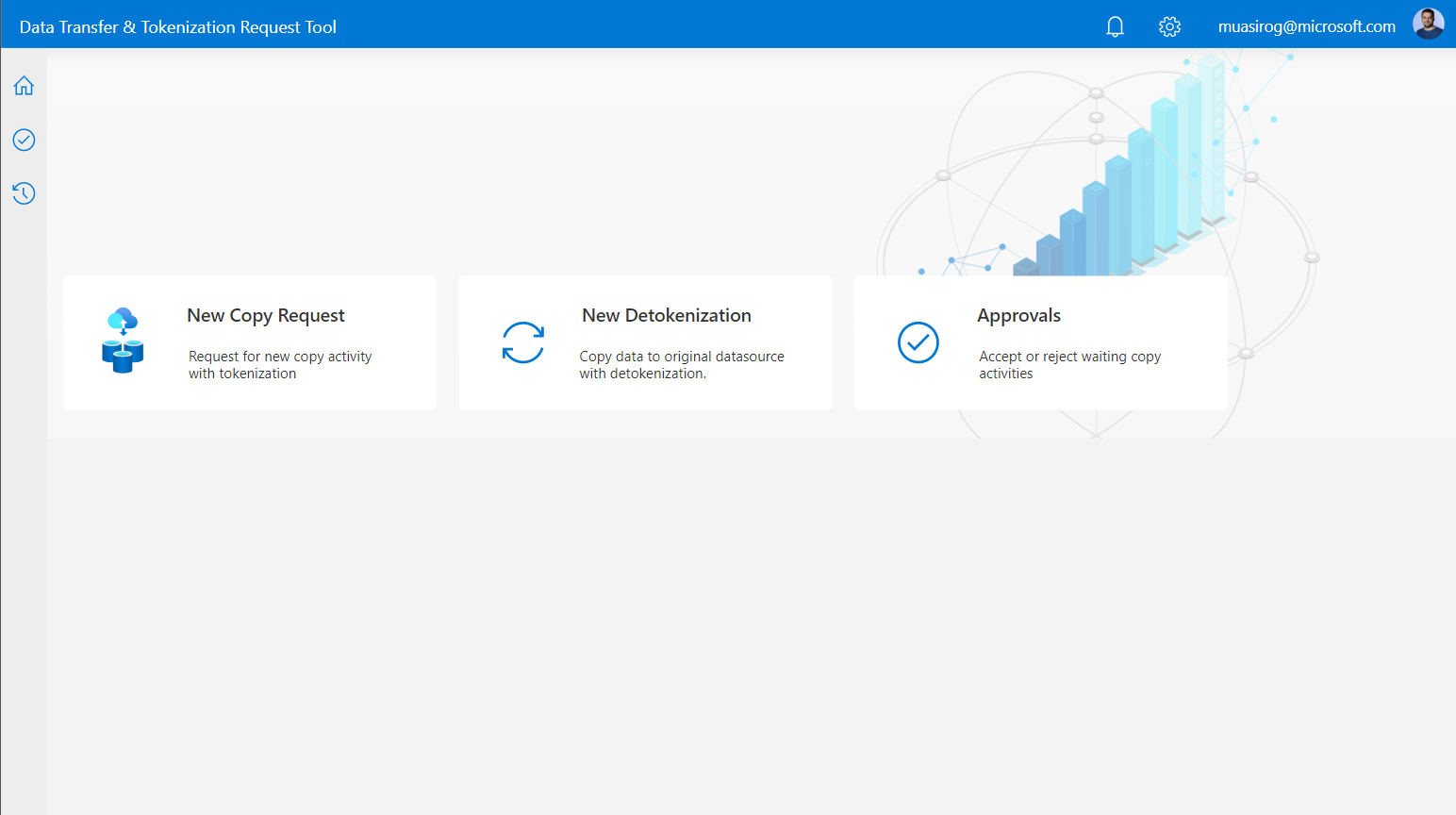
* + All On-Premises Data Sources is MSSQL Server
  + Metadata Parts can be hold separated SQL Server
  + Token Generator, Detokenized procedures and Tokens table must be hold on Source Database (it uses joins with original table, token table and/or result table)

# Technical Details

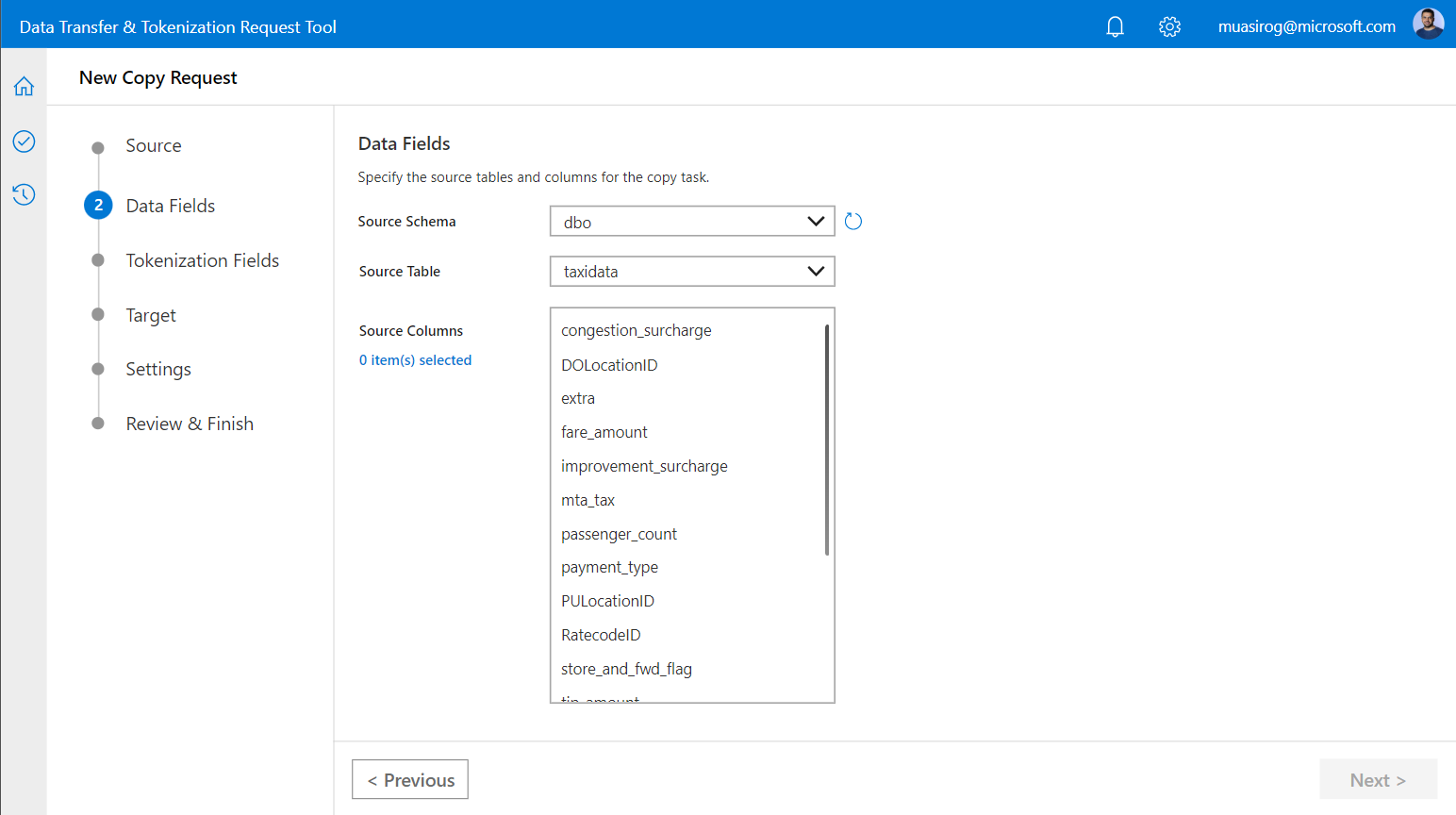
## Power Apps

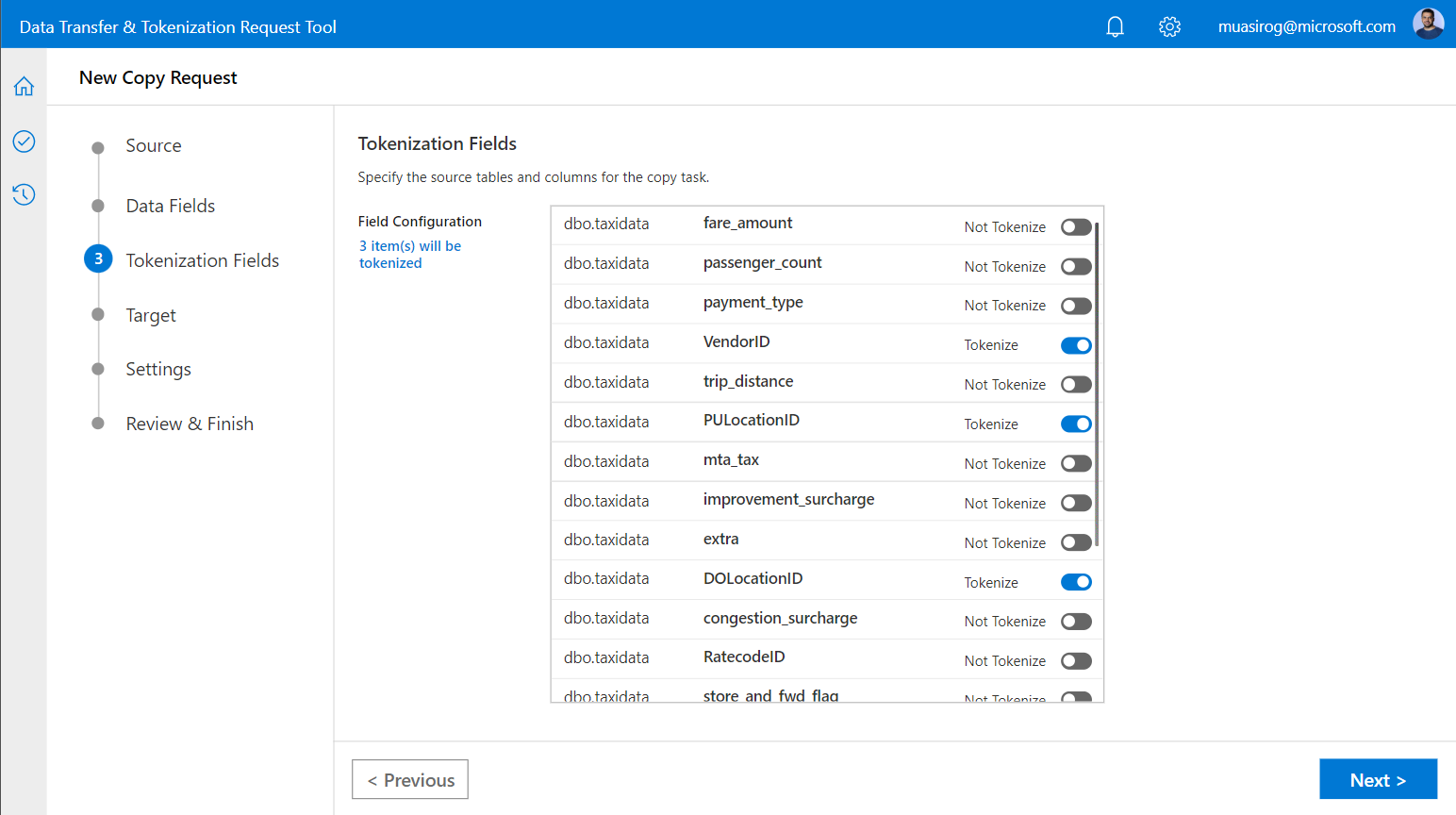
We created a simple app utilizing Microsoft Power Apps for management and approval process for the reference solution. This app works on metadata SQL database which includes table names, column names, selected columns for tokenization and approval process data.

Splash screen of the app:

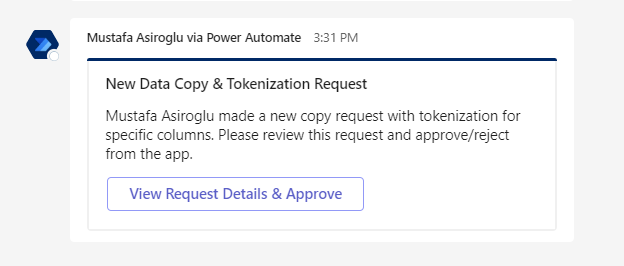


New Copy request can be created using simple UX as shown below. User selects data source, source table & columns and selects fields to be tokenized.

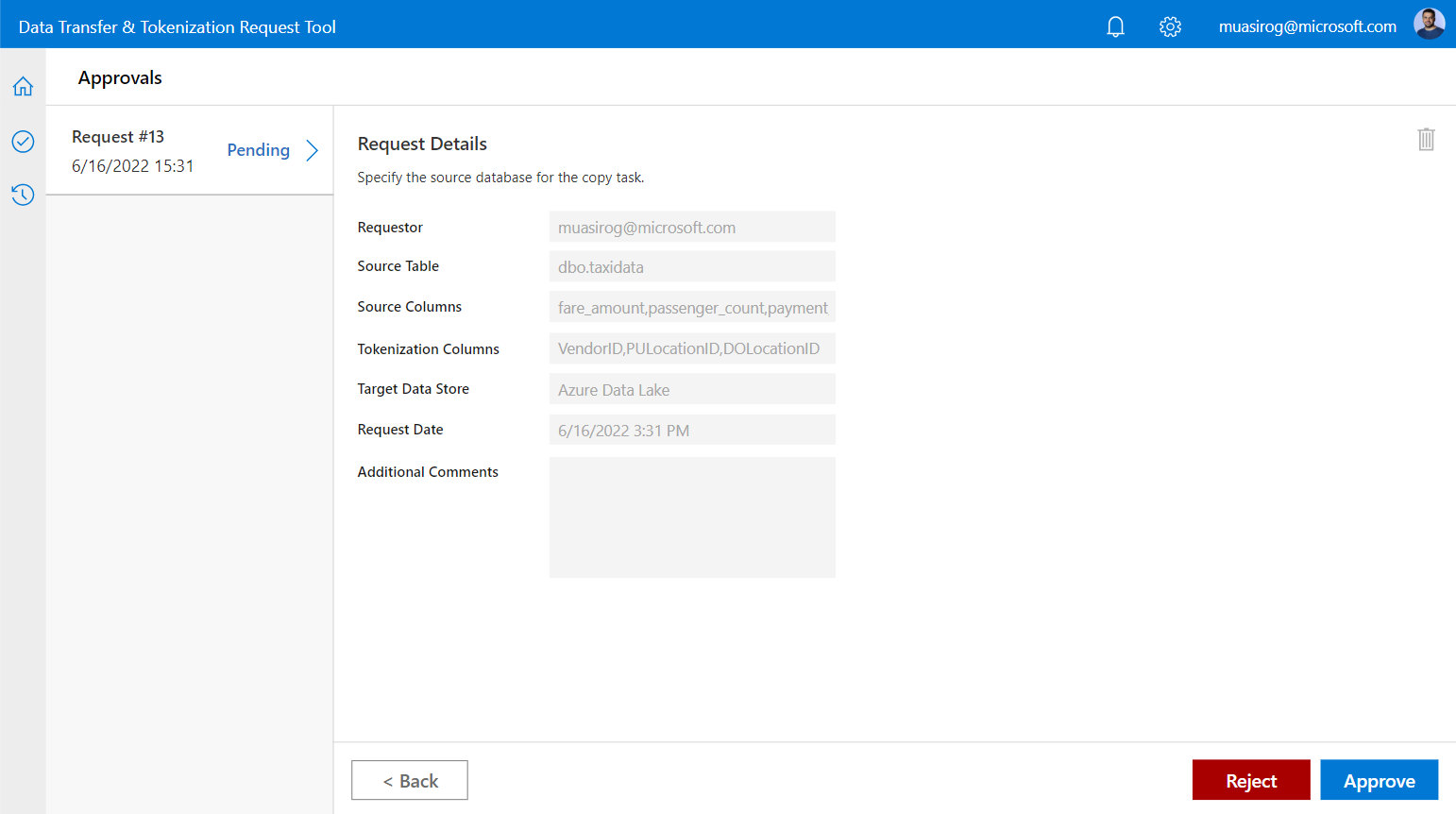




When request is submitted by requestor, the app sends a notification to default approver via Teams or email. This notification and approval mechanism is created using Power Automate and can be easily customized based on business needs.



Then approver opens the app and approves/rejects the request after review.



After approver approves the request, it updates metadata database with new copy activity record. Then, Azure Synapse Data Pipeline checks this database for pending actions and triggers tokenization and data transfer job.

## Azure Synapse Analytics

Azure Synapse Analytics has two pipelines:

**Pipeline 1: Move Data To Azure**

This general structure of the pipeline is reflected in Figure 1:

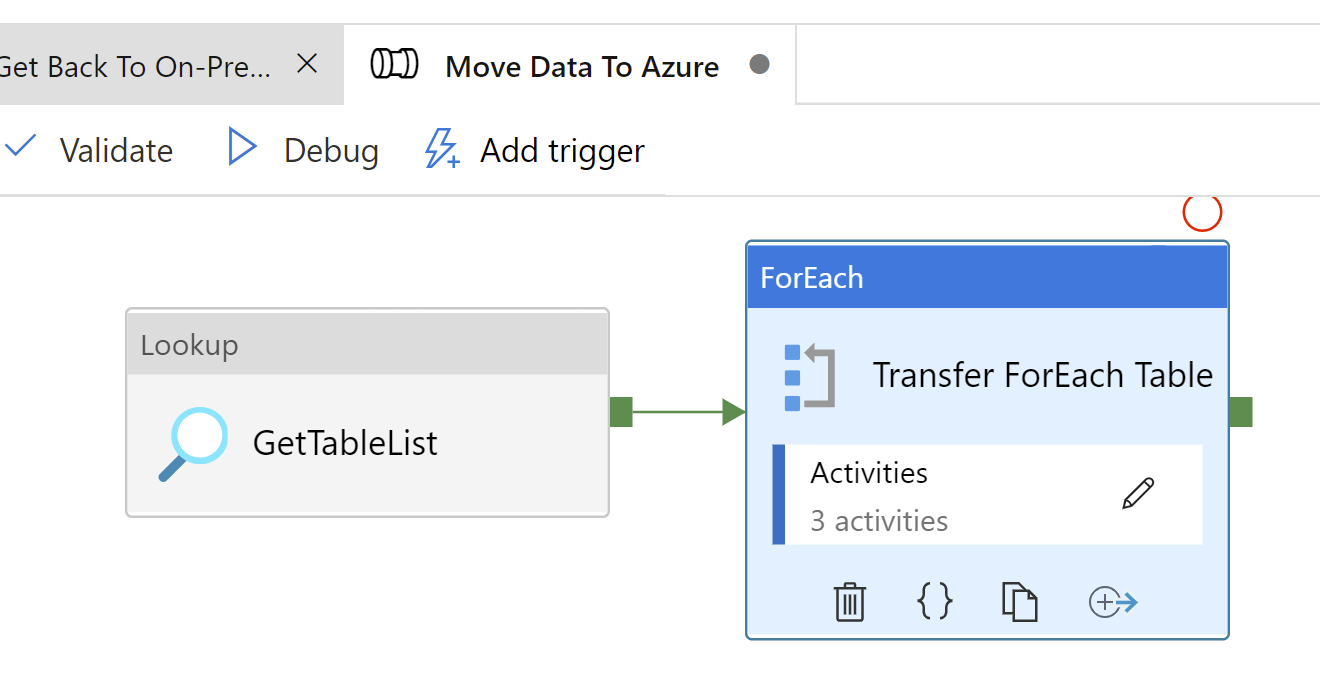
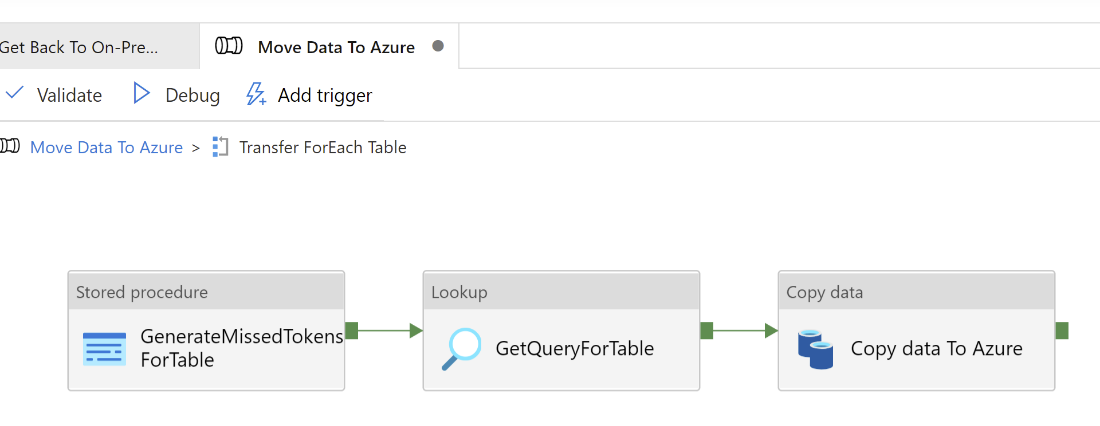


Figure 1: Move Data to Azure Pipeline

**Get Table List** : Get List of table to marked as permitted to move Azure with Query

**Transfer ForEach Table** : Loop for all tokenizable tables

In ForEach Loop

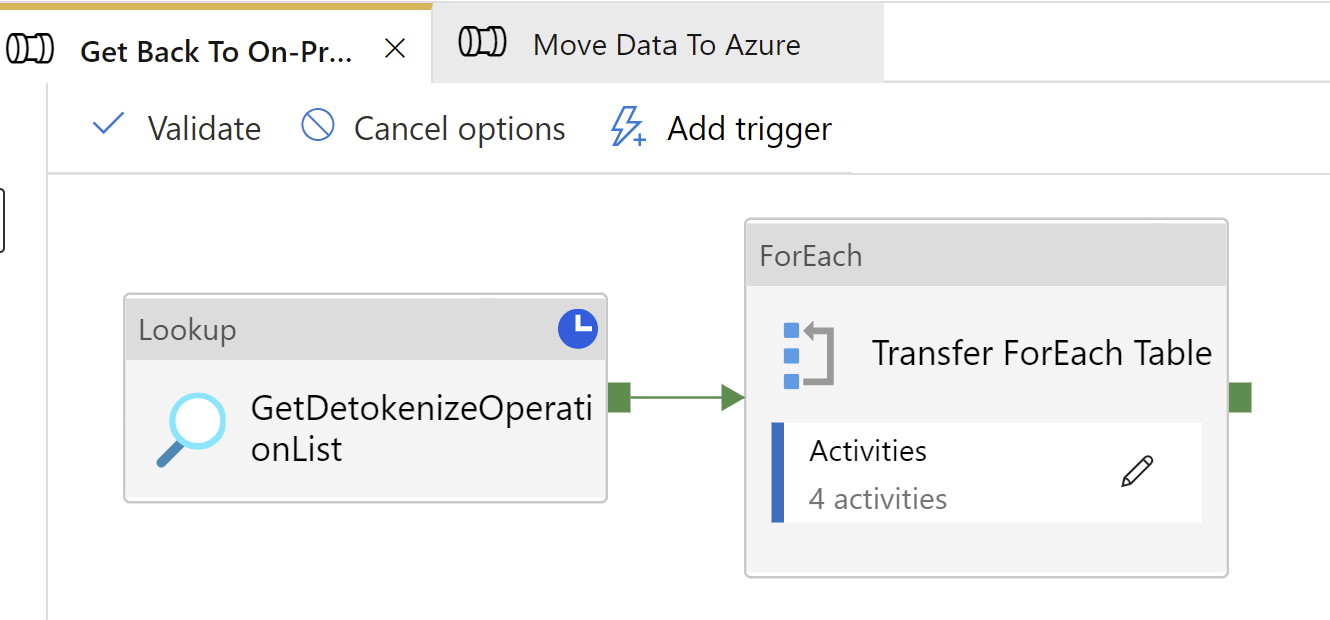


**GenerateMissedTokensForTable:** Trigger an SP to Populate tbTokens Table with original Data and Tokens. in next executions if tokens exist for a specific original value, procedure did not generate new token again.

**GetQueryForTable:** This query getter procedure generate a query to return tokens for tokenized columns also real values for nontokenized columns

**Copy data To Azure:** Copy query data into a file. File name generated by schema name and table name

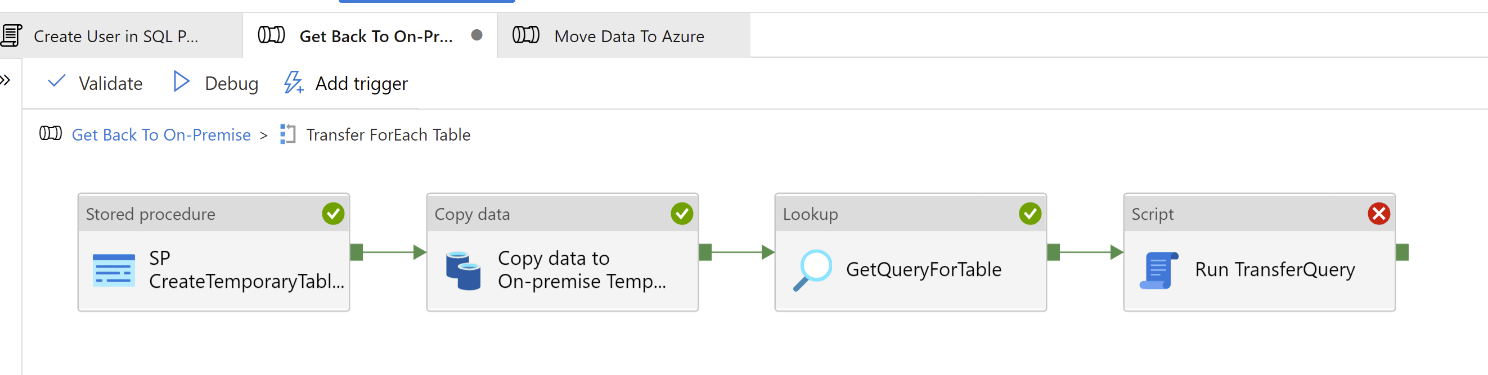
**Pipeline 2: Get Back To On-Premise**



**GetDetokenizeOperationList** : Get List of Data Detokenize operations from MetaData Tables

**Transfer ForEach Table** : Loop for all tables

In ForEach Loop



**SP CreateTemporaryTableOnDeTokenizeTarget :** Create a Temporary Table on DeTokenize Target for join tokens to get original values

**Copy data to On-premise Temp Table :** Copy Data From Azure Side to On-Premise Temporary Table (Just Created before)

**GetQueryForTable :** This query getter procedure generate a query to return original Values for tokenized columns also real values for nontokenized columns

**Run TransferQuery :** Trigger previously generated Query to populate Target table with Original values (For tokenized columns) and real values (for Non tokenized columns) over On-Premise Environment

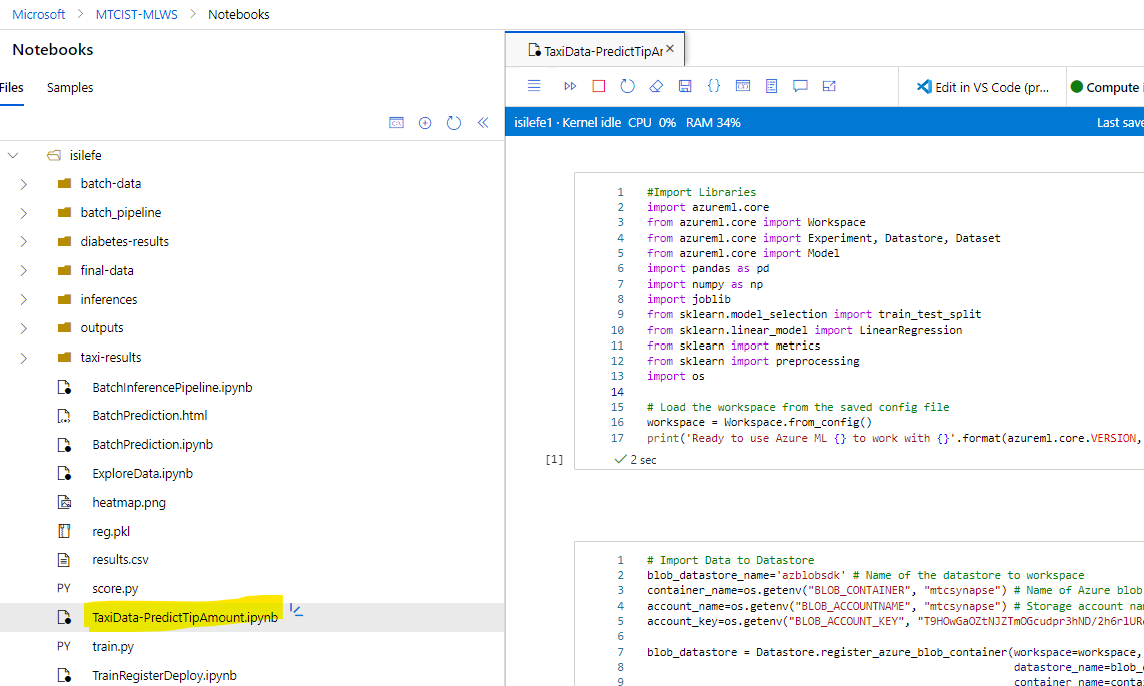
## Azure Machine Learning

Please run all codes in Python3.

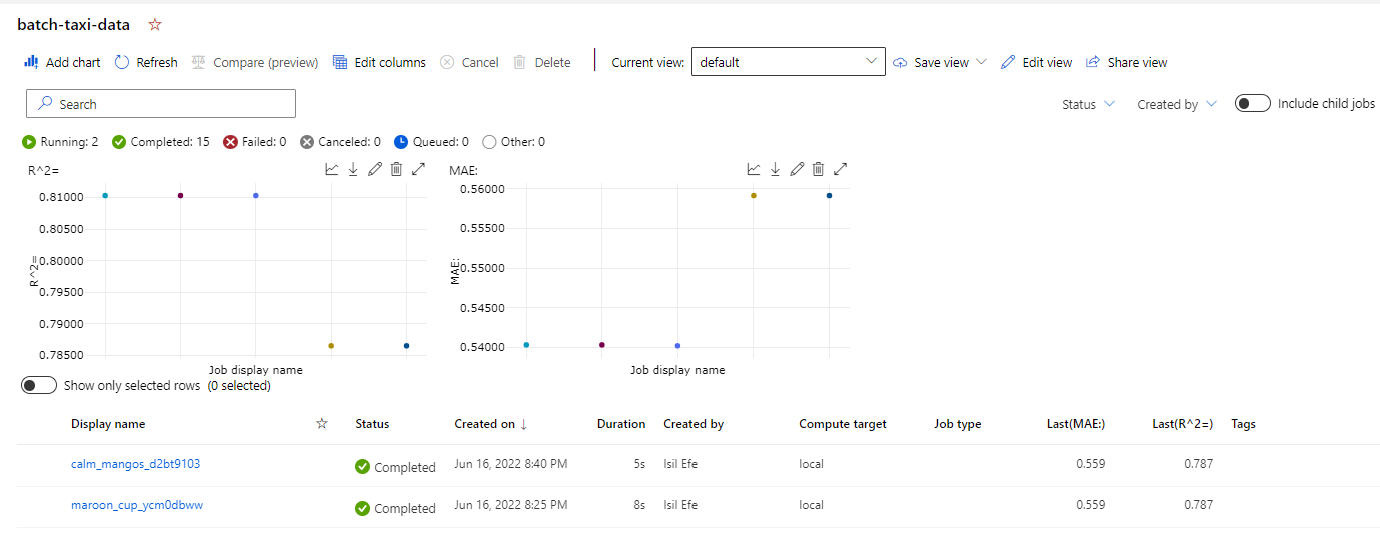
**Notebooks from Azure ML studio used.**

## 

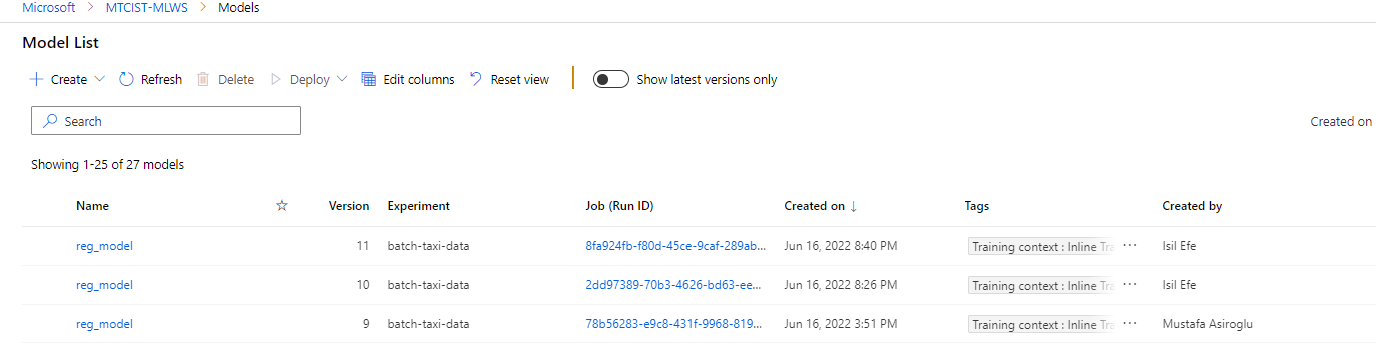
## 



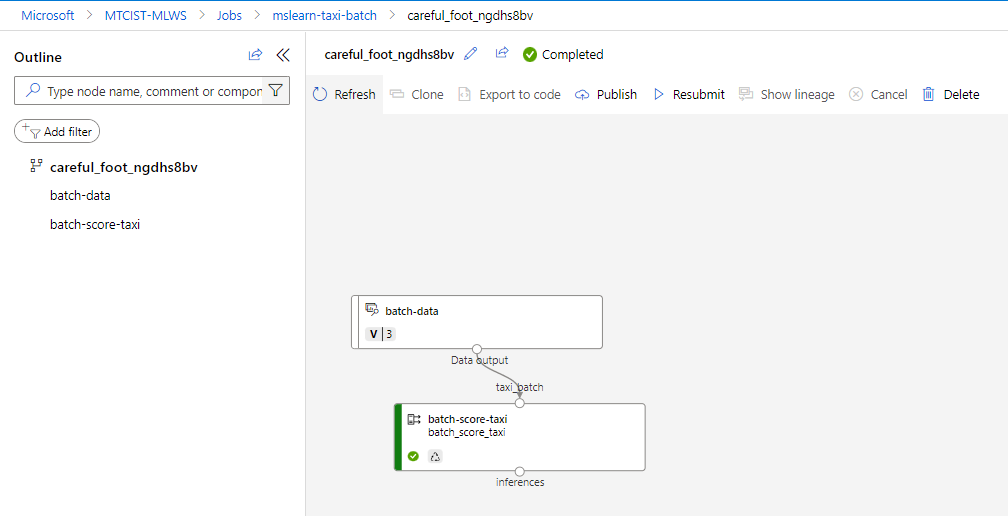
* Notebook name: Files> taxi-results > TaxiData-PredictTipAmount.ipynb
* Tip\_amount is predicted in this python code with linear regression model.
* Training runs are logged in batch-taxi-data experiment with MAE and R2 metrics.



* Trained model is registered to keep versions.



* Inference pipeline is created for prediction.



**Input Data with tokenized sensitive columns**

Tokenized fields: VendorID, PULocationID, DOLocationID

# 

**End to End data science solution from getting data to prediction in Python notebook:**

* Get Tokenized Data
* Prepare – Cleanse – Transform Data
* Split Data as Train and Test
* Train & Register Model
* Make Predictions

**Output Data with Predictions (sensitive data- tokenized)**

# 

# Azure Solutions Used

* Business Applications: Power Platform – Power Apps
* Data & AI: Azure Synapse Analytics
* Data & AI: Azure Machine Learning

# Links to access demo environment

Contact with MTC Istanbul Team

# Team

* Rahmi Atılgener, CSA
* Işıl Efe, CE
* Mustafa Asiroglu, MTC TA
* Arzu Turkeli, MTC TA