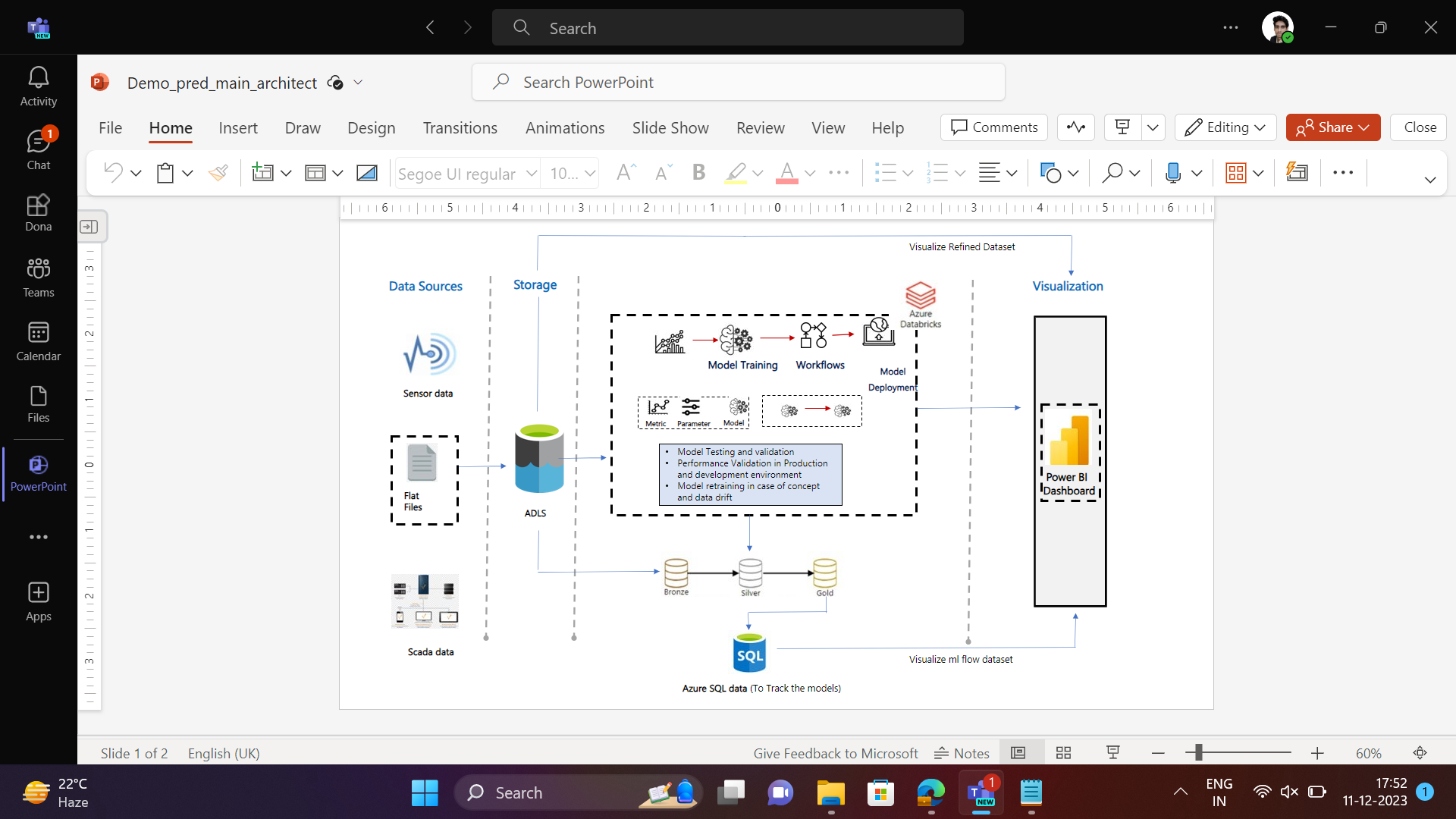
**Predictive Maintenance Demo**

Predictive maintenance is commonly applied in various industries, including manufacturing, energy, transportation, and healthcare, among others. Access to high-quality and diverse predictive maintenance datasets is crucial for developing accurate and effective predictive maintenance models. Researchers and data scientists use these datasets to explore new algorithms, validate models, and contribute to advancements in predictive maintenance technologies.

**Work Flow -**



* **Data Source** Here We have the Flat (CSV) Files from the Scada Environment which is the Sensor Dataset.
* **Storage** We use the Azure Data lake Storage Gen 2 To store the Dataset.
* **Azure Databricks**  Here with the Databricks workspace we Perform the EDA on the dataset and Perform Modelling .Then after we use the ml flow to track the model.
* Using ml flow client API to get the ml flow data and creating a dataset of it
* Pushing the ml flow Dataset to Azure SQL.
* **Power BI,** Using the Power BI to visualize the Refined Dataset and the SQL ml flow dataset.

**Overview -**

1) Reading Data from Azure storage gen 2 and creating a mount point of dataset

2) Performing EDA on the dataset

3) Storing Refined dataset to the Gen 2

4) Used Random Forest to modelling

5) Performed MLflow and logging essential params, metrics etc.

6) Fetched data from mlflow and created a dataset of it

7) Pushed this updated Dataset to Azure Sql and updating the records and doing these tasks in a scripted way

8) Integrating the Databricks workspace with Power BI

9) worked on Power BI to integrate the Azure SQL , Storage account

**1) Reading Data from Azure storage gen 2 and creating a mount point of dataset**

* Create a azure storage account and creating a container in that storage account
* After that upload a dataset in azure storage container that we have created.
* In data bricks workspace we can use this code snippet to fetch the uploaded dataset from the Azure Storage data lake Storage gen 2
* Creating a Mount Point in databricks and access the dataset.
* **DBFS (Databricks File System):** DBFS is a distributed file system that is built on top of cloud object stores like ADLS or S3. It allows you to store and access data in a distributed and scalable manner.
* **Mount Point:** A mount point in Databricks is a pointer to a directory in an external storage system (e.g., ADLS or S3) that you want to make available in DBFS. When you mount a directory, you essentially create a link between that external directory and a directory in DBFS.

dbutils.fs.mount(

source = "[wasbs://container\_name@Storage\_account\_name.blob.core.windows.net/](about:blank)",

mount\_point = "/mnt/azurestorage/pred/", #This is a mount point You can change as you want.

extra\_configs = {"fs.azure.account.key.deletenahi.blob.core.windows.net": "paste Your Azure storage account access key here"})

- Read the dataset

path='dbfs:/mnt/azurestorage/pred/train.csv' #Mount Point Path

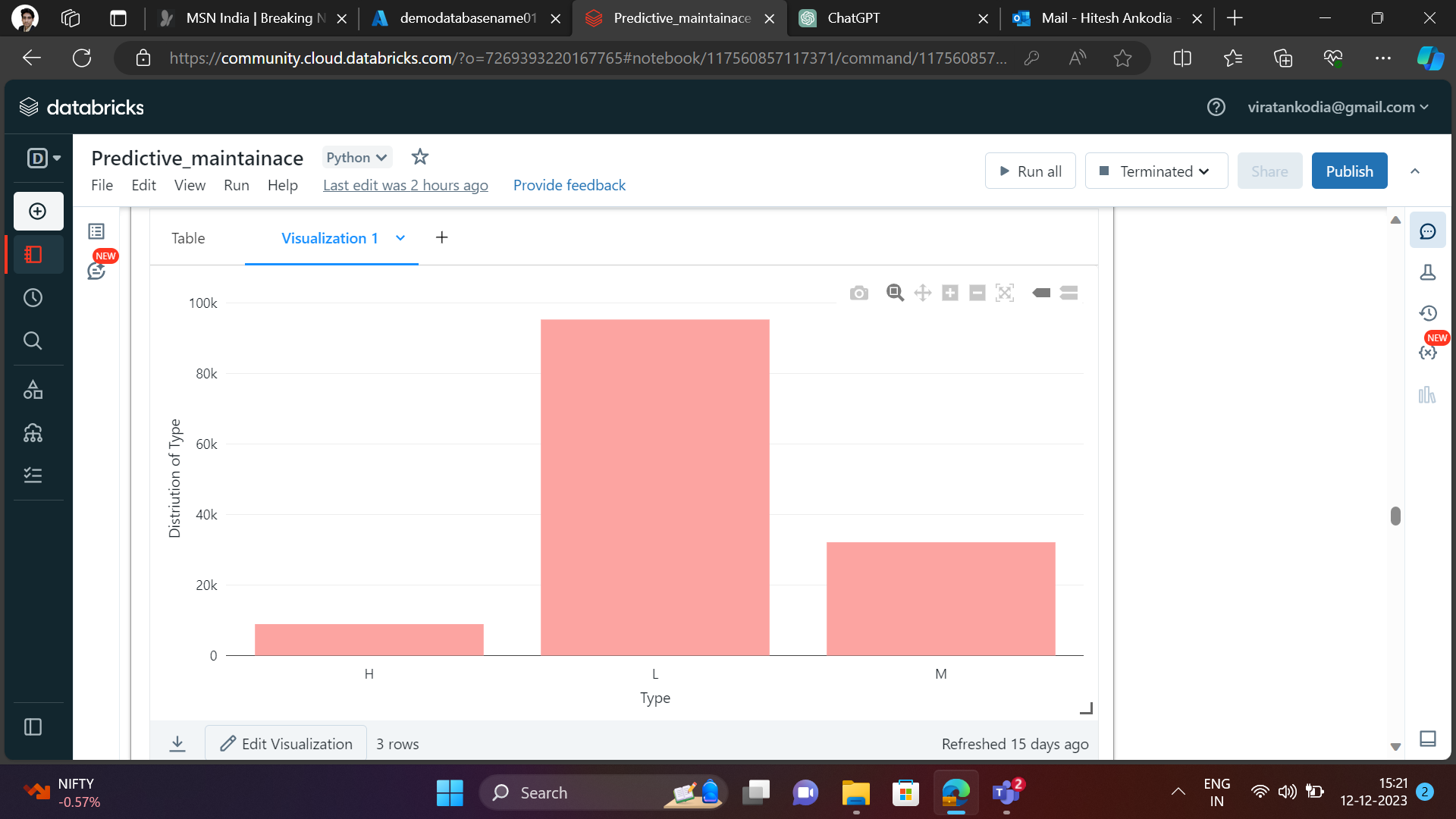
df = spark.read.format('csv').load(path, header = True, inferSchema = True)

display(df)

* Once mounted, you can access the data in the external storage system through the **mount\_point** directory in DBFS. This allows you to seamlessly read and write data between Databricks and the external storage
* Mounting external storage is useful when you want to leverage the benefits of distributed computing in Databricks while keeping your data in a scalable and cost-effective storage system like ADLS or S3.

**2) Performing EDA and Transformation on the dataset -**

"Exploratory Data Analysis." Exploratory Data Analysis is an approach to analyzing datasets to summarize their main characteristics, often with the help of statistical graphics and other data visualization methods.





**3) Storing Refined dataset to the Gen 2**

- Here we are storing the Refined dataset after EDA and Transformation which has “Time\_stamp” column append in the dataset and has few columns in the dataset.

# Azure Storage Account information

storage\_account\_name = "storage\_name"

storage\_account\_key = "key" # or use SAS token

container\_name = "container\_name"

# Define the storage options

storage\_options = {

"fs.azure.account.key.{0}.blob.core.windows.net".format(storage\_account\_name): storage\_account\_key

# If using SAS token instead of key:

# "fs.azure.sas.{0}.{1}.blob.core.windows.net".format(container\_name, storage\_account\_name): "your\_sas\_token"

}

# Write the DataFrame to CSV format in Azure Storage

df\_with\_timestamp.write.format("csv").mode("overwrite").option("header", "true").save("wasbs://{container\_name}@{storage\_account\_name}.blob.core.windows.net/", storage\_options=storage\_options)

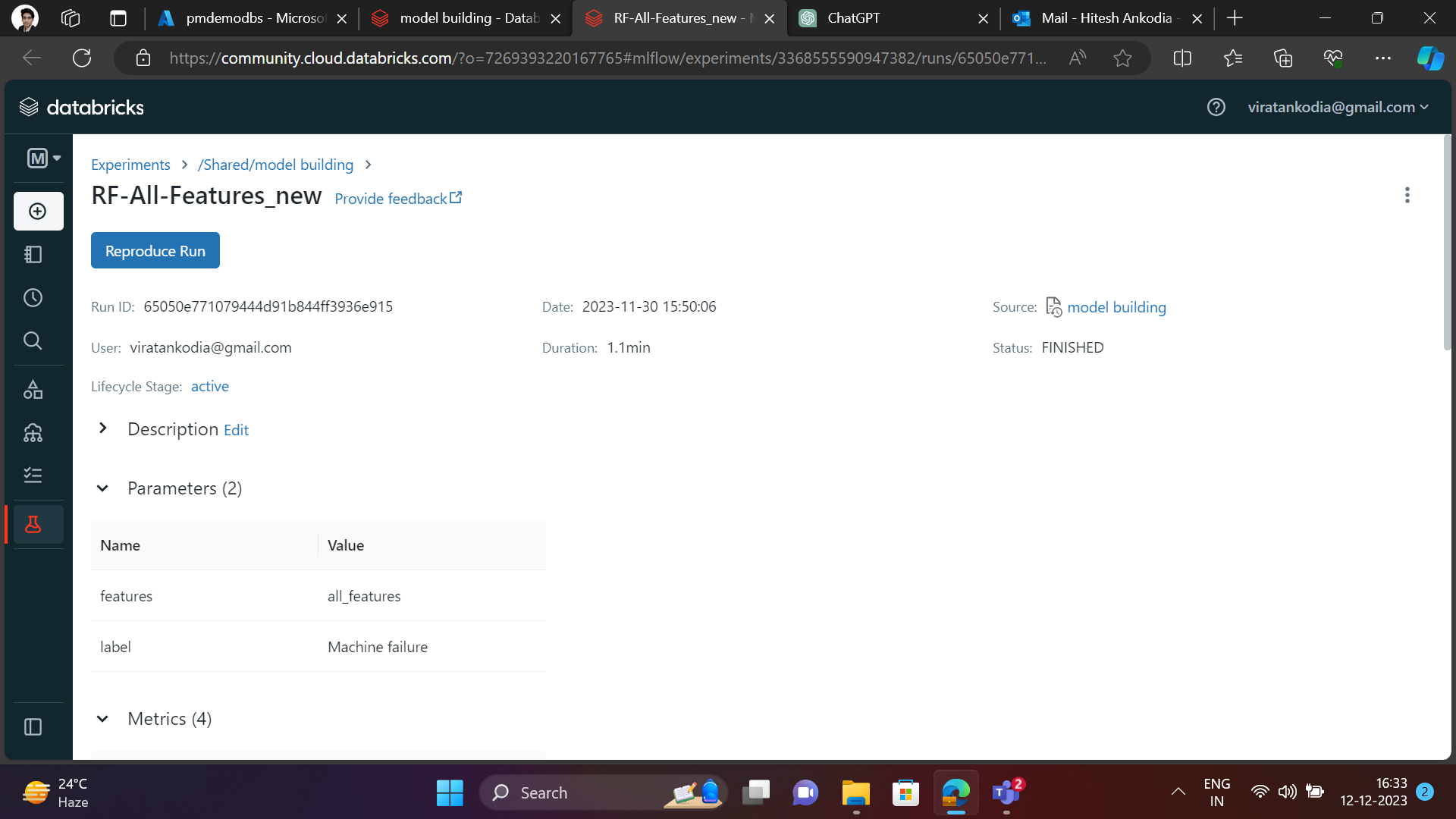
**4) Used Random Forest to modelling**

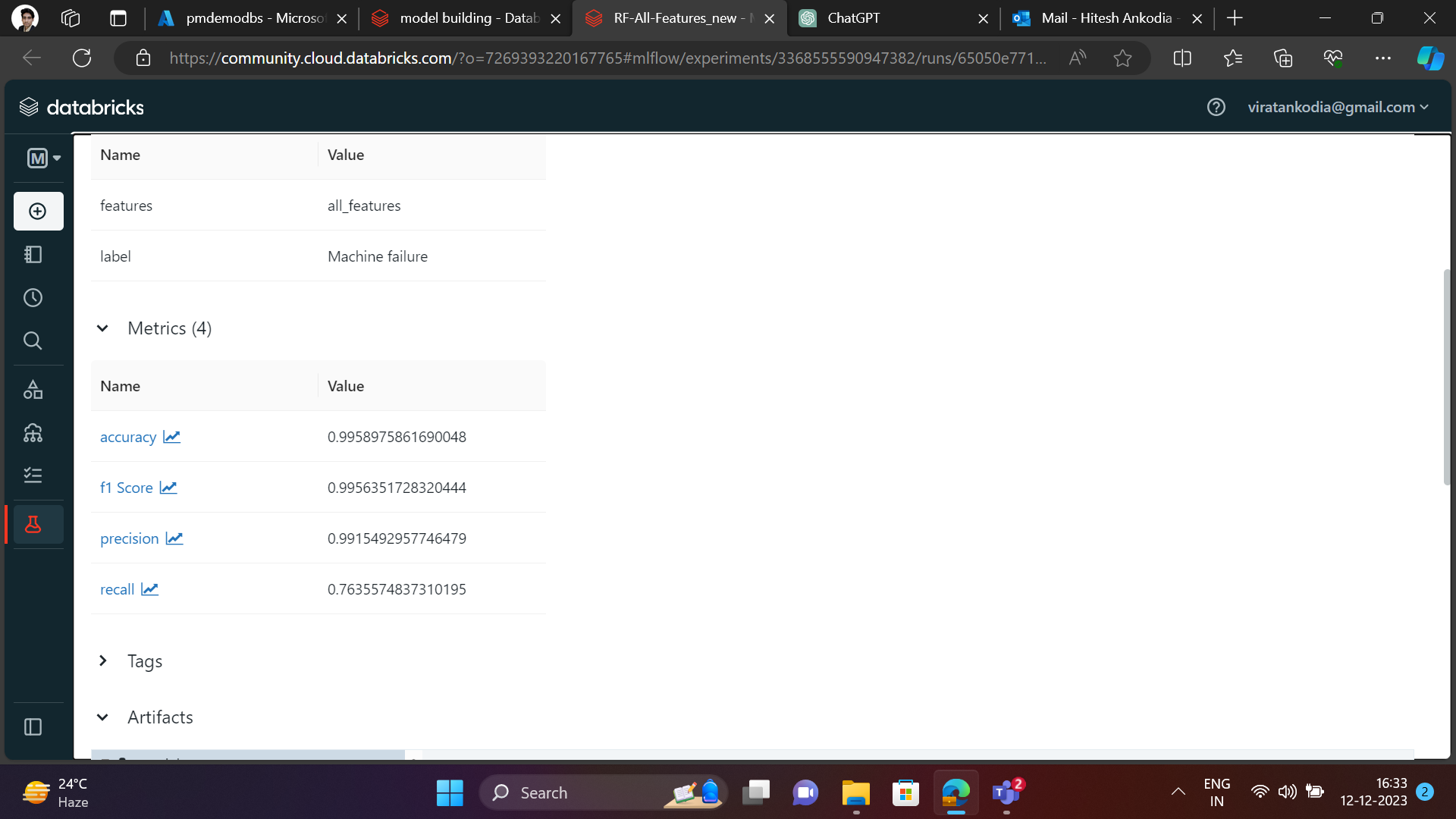
Random Forest is an ensemble learning method used for both classification and regression tasks in machine learning. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees

* First we use the data bricks mount point that we created while accessing the dataset from azure storage account.
* Then we use the random forest model to train the model
* After training the model we get the metrics as follow : - Accuracy: 0.9958975861690048, Precision: 0.9915492957746479, Recall: 0.7635574837310195

**5) Performed ML flow and logging essential params, metrics etc.**

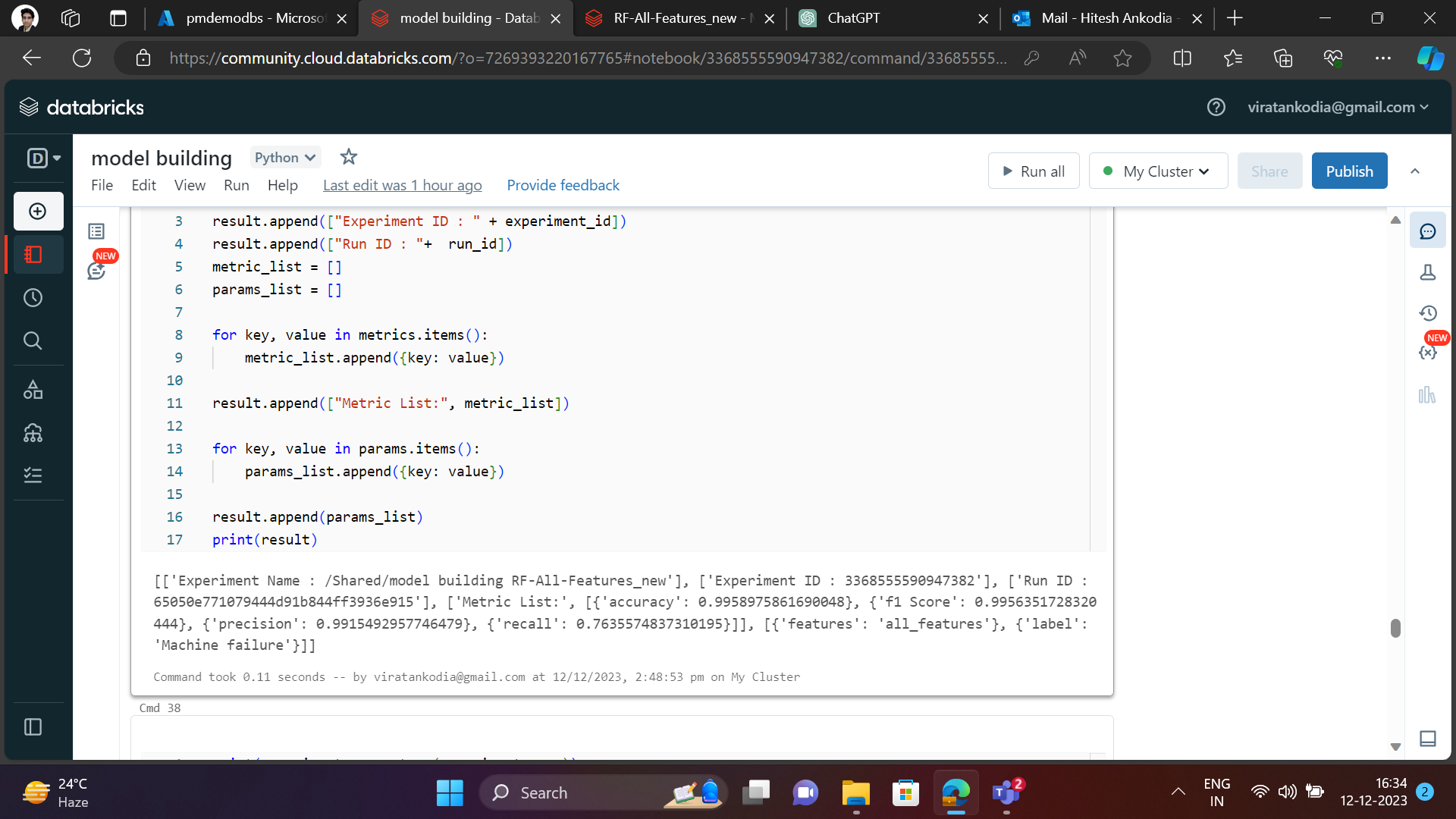
MLflow is an open-source platform for managing the end-to-end machine learning lifecycle. It supports experiment tracking, reproducibility, and deployment. When using MLflow in Databricks, you can take advantage of integrated capabilities and features

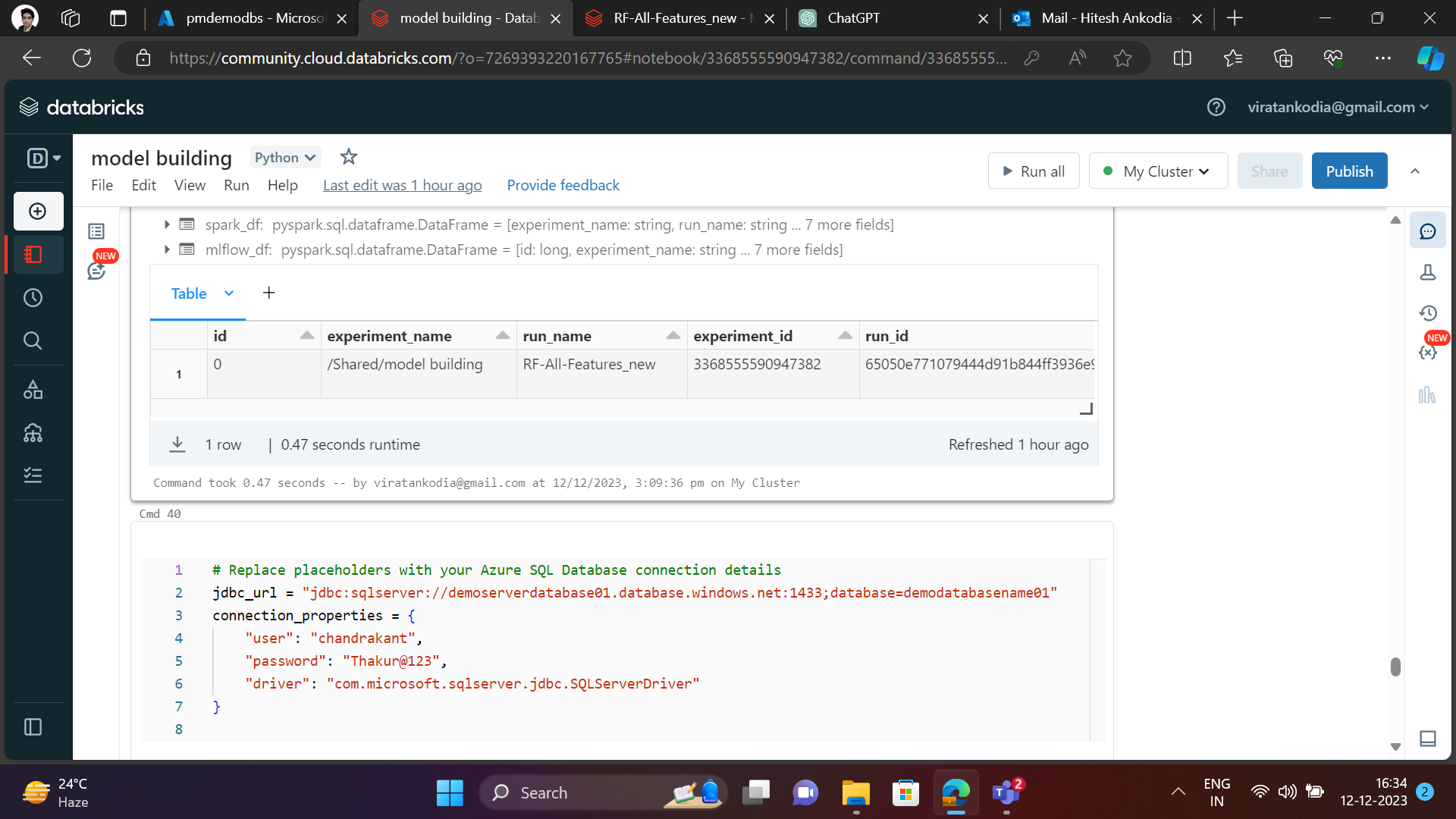




**6) Fetched data from ml flow and created a dataset of it.**

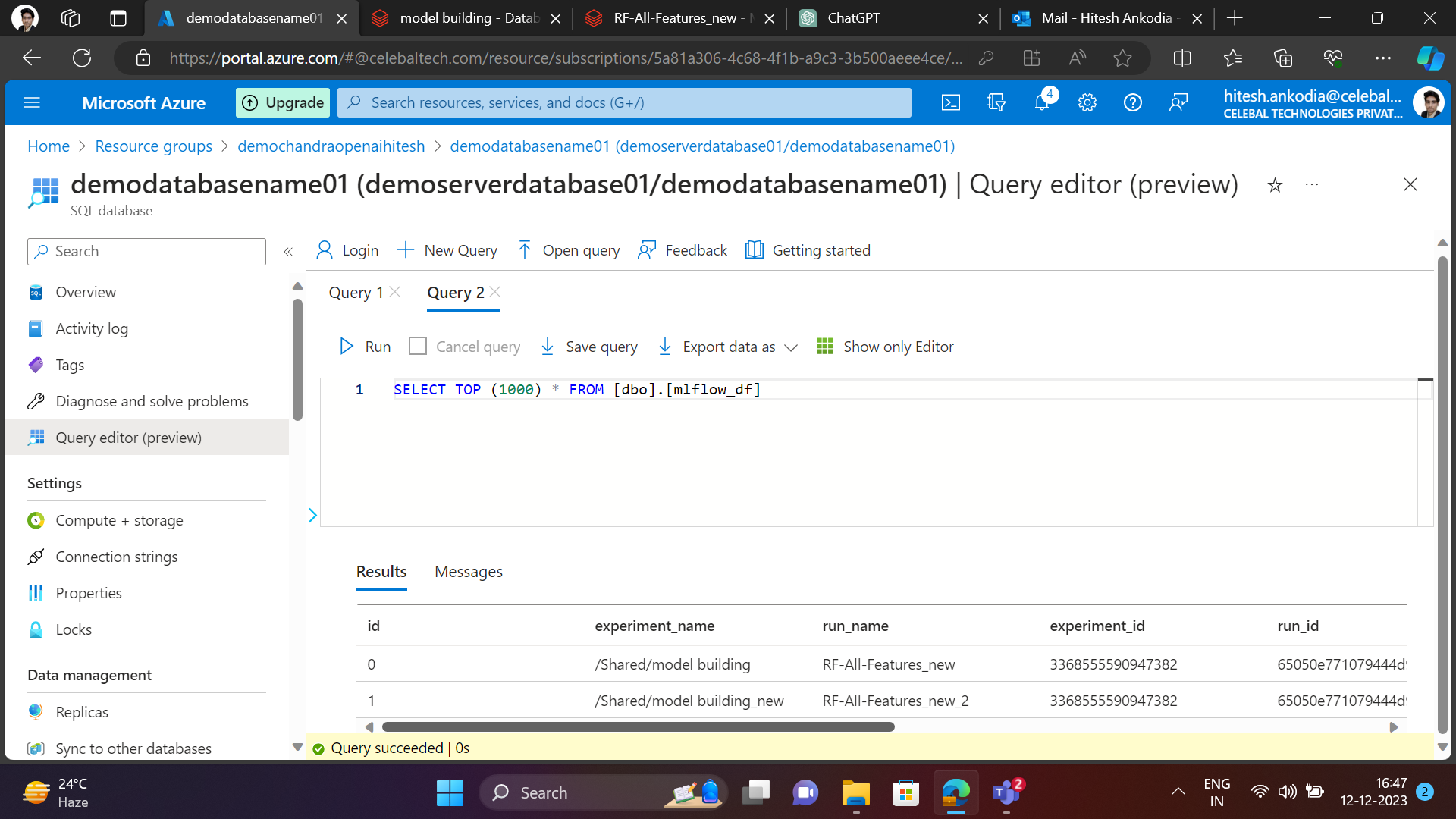
To fetch data from the MLflow repository, you can use the MLflow REST API. The API provides endpoints for various operations, including listing experiments, getting run details, downloading artifacts, and more

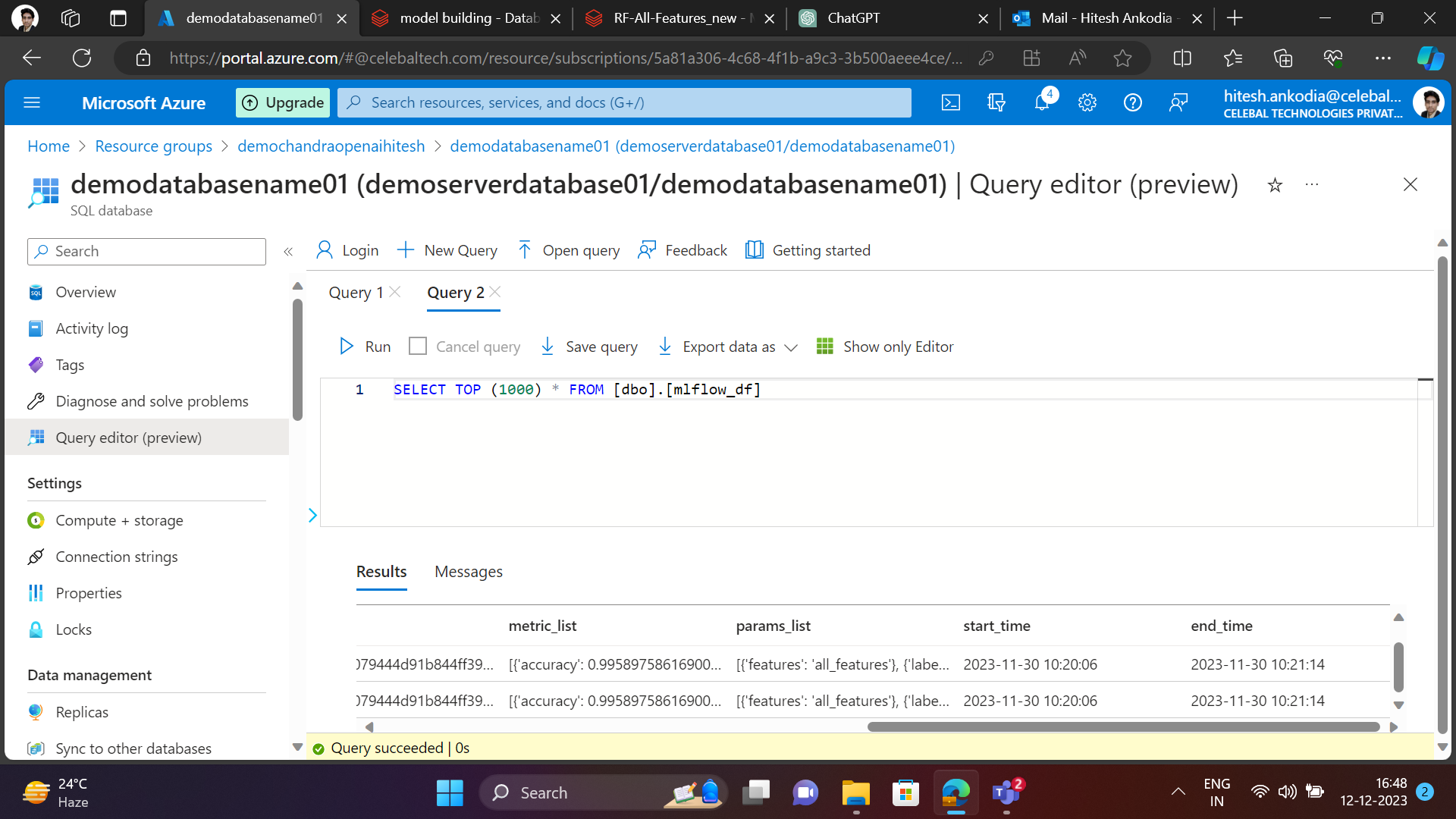




**7) Pushed this updated Dataset to Azure Sql and updating the records.**

Azure SQL refers to the family of fully managed, secure, and intelligent relational database services provided by Microsoft on the Azure cloud platform. It includes different offerings to meet various database needs. The primary Azure SQL services are Azure SQL Database and Azure SQL Managed Instance.





# Replace placeholders with your Azure SQL Database connection details

jdbc\_url = "jdbc:sqlserver://sever\_name.database.windows.net:1433;database=database\_name"

connection\_properties = {

"user": "user\_name",

"password": "password",

"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"

}

#To push the dataset to azure SQL

mlflow\_df.write.jdbc(url=jdbc\_url, table="table\_name", mode="overwrite", properties=connection\_properties)

#To update the Azure SQl database with new records.

df\_updated.write.jdbc(url=jdbc\_url, table="mlflow\_df", mode="append", properties=connection\_properties)

**8) Integrating the Databricks workspace with Power BI**

**1) Create a compute**

**2) In cluster Configuration**

**3) Copy the JDBC/ODBC details -** JDBC is specifically designed for Java applications to interact with databases, while ODBC is a more general API that can be used by applications written in various languages. Both provide a standardized way to access relational databases, making it easier for developers to work with different database systems.

**4) Go to Partner Connect option from Left bottom**

**5) Click the Microsoft Power BI file you will download this - ‘download connection file’**

**6) Go to Power BI Desktop Application** -

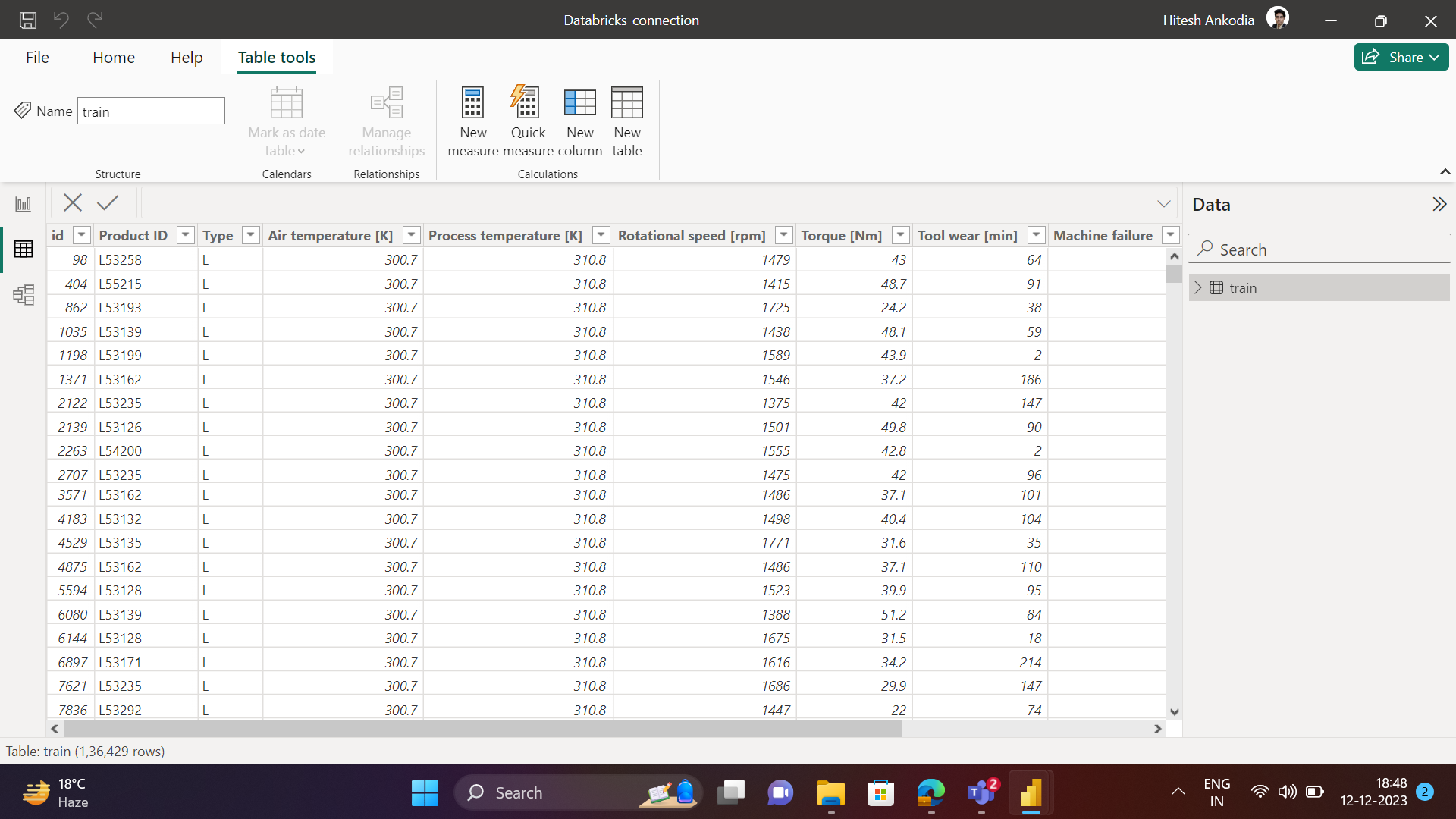
- Go to ‘Get data’ from home option , select Azure Databricks ,

- Enter the server hostname and HTTP Path from ODBC/JDBC details.

- Login to azure Active directory and sign in then connect.

- Provide the access token for data bricks connection.

- After that you will get connected to Databricks workspace and You can access the databricks dbfs table in the PowerBI workspace.



**9) Connect Azure platform(Azure SQL, Azure Blob Storage)**

- You can connect the Power BI with Azure SQl, Azure Blob storage with opening – Get data from top left of the Power BI home page.

* Now You have to provide the essential credentials details from Azure.