

# Analyzing Gold & Silver Price Data with Azure Services: A Step-by-Step Guide for Data Engineering Project

S

Syed Muhammad Raqim Ali Shah

8 min read · Just now

Listen

Share

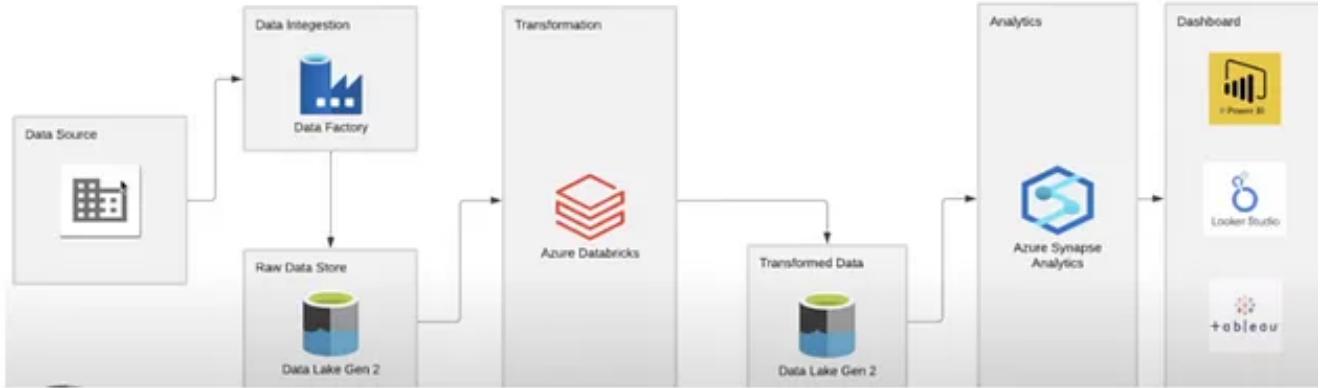
More



## Introduction

Analyzing gold price data using Azure services can provide valuable insights for informed decision-making. In this article, we will outline the key steps involved in

this process, from data ingestion to visualization, using Azure Data Factory, Data Lake Gen 2, Synapse Analytics, and Azure Databricks. Here is our work flow for this project:



## Step 1: Data Ingestion

To kickstart the analysis, the first step is to gather historical and real-time gold price data from various sources. Azure Data Factory, a robust data integration service on Azure, plays a crucial role here. It allows us to automate the ingestion process, making data collection efficient and reliable.

Firstly we will extract the data from the Kaggle using Azure data Factory that is kind of like the data pipeline tool available on Azure it will build a flow like this and load our data onto the Azure data Lake storage so first we will load the raw-data then using Azure databricks we will write our spark code and transform our data and load our data back to our transformed-data Lake storage once that is done we will use synapse analytics to run the SQL queries on top of the transform data so that we can find the insights and get the visualization on top of it.

In this project, we will take the gold-silver price data that is available on Kaggle.

Here is the link of dataset:

[https://github.com/muhammadrajqimshah/gold\\_silver\\_price\\_EDA/tree/main/Database](https://github.com/muhammadrajqimshah/gold_silver_price_EDA/tree/main/Database)

First We have to create Storage account and also making resource group during the process:

Microsoft Azure Upgrade Search resources, services, and docs (S+T)

Home > Storage accounts > Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Important: Azure Storage includes Azure Blob (object), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts.

**Project details**

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription: Azure subscription 1

Resource group: goldprice [Create new](#)

**Instance details**

Storage account name: testaccount199

Region: East US [Deploy to an edge zone](#)

Performance: Standard (Recommended for most scenarios (general-purpose v2 account))  Premium (Recommended for scenarios that require low latency)

Redundancy: Geo-redundant storage (GRS)  Make read access to data available in the event of regional unavailability.

[Review](#) [Previous](#) [Next: Advanced](#) [Give Feedback](#)

Microsoft Azure Upgrade Search resources, services, and docs (S+T)

Home > Storage accounts > Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Common options have been disabled by default due to the combination of storage account performance, redundancy, and region.

**Security**

Configure security settings that impact your storage account.

Require secure transfer for REST API operations

Allow enabling anonymous access on individual containers

Enable storage account key access

Default to Azure Active Directory authorization in the Azure portal

Minimum TLS version: Version 1.2

Permitted scope for copy operations (preview): From any storage account

**Hierarchical Namespace**

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs). Learn more.

Enable hierarchical namespace

[Review](#) [Previous](#) [Next: Networking](#) [Give Feedback](#)

Microsoft Azure Upgrade Search resources, services, and docs (S+)

Home > Storage accounts > Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

**Network connectivity**

You can connect to your storage account either publicly via public IP addresses or service endpoints, or privately using a private endpoint.

Network access \*

Enable public access from all networks

Enable public access from selected virtual networks and IP addresses

Disable public access and use private access

Enabling public access from all networks might make this resource available publicly unless public access is required, we recommend using a more restricted access type. [Learn more](#)

**Network routing**

Determine how to route your traffic as it travels from the source to its Azure endpoint. Microsoft network routing is recommended for most customers.

Routing preference \*

Microsoft network routing

Internet routing

[Review](#) [Previous](#) [Next : Data protection](#) [Give Feedback](#)

Microsoft Azure Upgrade Search resources, services, and docs (S+)

Home > Storage accounts > Create a storage account

Basics Advanced Networking **Data protection** Encryption Tags Review

**Recovery**

Protect your data from accidental or erroneous deletion or modification.

Enable point-in-time restore for containers

Use point-in-time restore to restore one or more containers to an earlier state. If point-in-time restore is enabled, then versioning, change feed, and blob soft delete must also be enabled. [Learn more](#)

Point-in-time restore and hierarchical namespace cannot be enabled simultaneously

Enable soft delete for blobs

Soft delete enables you to recover blobs and directories that were previously marked for deletion. [Learn more](#)

Days to retain deleted blobs:

Enable soft delete for containers

Soft delete enables you to recover containers that were previously marked for deletion. [Learn more](#)

Days to retain deleted containers:

Enable soft delete for file shares

Soft delete enables you to recover file shares that were previously marked for deletion. [Learn more](#)

Days to retain deleted file shares:

**Tracking**

Manage versions and keep track of changes made to your blob data.

Enable versioning for blobs

Use versioning to automatically maintain previous versions of your blobs. Learn more

Consider your workloads, their impact on the number of versions created, and the resulting costs. Optimize costs by automatically managing the data lifecycle. [Learn more](#)

[Review](#) [Previous](#) [Next : Encryption](#) [Give Feedback](#)

Microsoft Azure Upgrade Search resources, services, and docs (Get it!)

Home > Storage accounts > Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Encryption type \*  Microsoft-managed keys (ARM)  Customer-managed keys (CMK)

Enable support for customer-managed keys  Blobs and files only  All service types (blobs, files, tables, and queues)

⚠ This option cannot be changed after this storage account is created.

Enable infrastructure encryption

Review < Previous Next > Tags Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (Get it!)

Home > Storage accounts

Storage accounts

+ Create ⚡ Recycle Manage view Refresh Export to CSV Open query Assign tags Delete

Filter for any field: Subscription equals all Resource group equals all Location equals all Add filter

No grouping List view

Showing 1 to 1 of 1 records

| Name          | Type            | Kind      | Resource group | Location | Subscription       |
|---------------|-----------------|-----------|----------------|----------|--------------------|
| gridpricedata | Storage account | StorageV2 | gridsource     | East US  | Azure subscription |

< Previous Page 1 of 1 Next > Give feedback

Now it's time to create a Container in Storage:

The screenshot shows the Microsoft Azure Storage account interface for 'goldpricedata'. The left sidebar contains navigation links for Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Data storage (with Containers selected), File shares, Queues, and Tables. The main content area displays the 'Containers' blade with a search bar and a table. The table has columns for Name, Last modified, Anonymous access level, and Lease state. It lists one item: 'Logs' (Last modified: 8/29/2023, 3:57:50 PM, Private, Available).

This screenshot is identical to the first one, but it shows a second container named 'gold-price-data' added to the list. The table now includes two rows: 'Logs' and 'gold-price-data'. Both entries have the same properties: Last modified: 8/29/2023, 3:57:50 PM, Private, Available.

And make raw-data and transformed-data folder in the container:

The screenshot shows the Microsoft Azure Storage Explorer interface. In the center, there's a list of blobs in a container named 'gold-price-data'. The blobs are 'raw-data' and 'transformed-data', both marked with a yellow folder icon. To the right of the blob list, there's a message box with a green checkmark that says 'Successfully added directory' and 'Successfully added directory 'transformed-data''. On the left side, there's a sidebar with sections like 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', 'Settings', and 'Metadata'. At the top, there are navigation links for 'Home', 'Storage accounts', and 'goldprice-data (Containers)'. The top bar also includes standard browser controls like 'Search resources, services, and docs (Ctrl+F)' and 'Upgrade'.

Now setting up the Data Factory for the data ingestion:

Now Creating Pipeline to load the data into Data Factory:

The screenshot shows the 'Create Data Factory' wizard in the Microsoft Azure portal. The 'Basics' step is currently selected. It contains the following configuration details:

- Subscription:** Azure subscription 1
- Resource group:** goldprice (with a 'Create new' option)
- Name:** price-ing
- Region:** East US
- Version:** V2

At the bottom of the screen, there are navigation buttons: 'Previous', 'Next', and 'Review + create' (which is highlighted in blue).

Microsoft Azure Upgrade Search resources, services, and docs (GA)

Home > Data factories > Create Data Factory

Basics Git configuration Networking Advanced Tags Review + create

Azure Data Factory allows you to configure a Git repository with either Azure DevOps or GitHub. Git is a version control system that allows for easier change tracking and collaboration.

Learn more about Git integration in Azure Data Factory.

Configure Git later

Previous Next Review + create Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (GA)

Home > Data factories > Create Data Factory

Basics Git configuration Networking Advanced Tags Review + create

Managed virtual network

Choose whether you want the default AutoResolveIntegrationRuntime to be provisioned on demand inside an ADF-managed virtual network. If this setting is disabled, after the data factory is created, you can still choose whether to provision explicitly created Azure integration runtime inside an ADF-managed virtual network.

Learn more

Enable Managed Virtual Network on the  default AutoResolveIntegrationRuntime

Self-hosted integration runtime inbound connectivity to Azure Data Factory service

Choose whether to connect your self-hosted integration runtime to Azure Data Factory via public endpoint or private endpoint. This applies to self-hosted integration runtime running either on-premises or inside customer managed Azure virtual networks.

Learn more

Connect via \*  Public endpoint  Private endpoint

💡 You can change this or configure another connectivity method after this resource is created. [Learn more](#)

Previous Next Review + create Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (GA)

Home > Data factories > Create Data Factory

Basics Get configuration Networking Advanced Tags Review + create

**Datafactory encryption**

By default, data is encrypted with Microsoft-managed keys. For additional control over encryption keys, you can supply customer-managed keys to use for encryption of blob and file data. Customer-managed keys must be stored in an Azure Key Vault. You can either create your own keys and store them in a key vault, or you can use the Azure Key Vault APIs to generate keys. The storage account and the key vault must be in the same region, but they can be in different subscriptions.

Enable encryption using a Customer Managed Key

Previous Next Review + create Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (GA)

Home > MicrosoftDataFactory-20230829162906 | Overview >

**price-edata** Data factory (V2)

Delete  JSON View

**Overview**

Activity log Access control (IAM) Tags Diagnosis and solve problems

Resource group:  **infrastructure** Status: **Succeeded** Location: East US Subscription:  **Azure subscription 1** Subscription ID: **50919864-7e8f-4286-bae3-1c1d665280d**

Type: Data factory (V2) Getting started: [Quick start](#)

**Settings**

Networking Managed identities Properties Locks

**Getting started**

Quick start Monitoring Automation Support + troubleshooting

**Azure Data Factory Studio**

Launch studio

Quick Starts Tutorials Template Gallery Training Modules

Monitoring PipelineRuns ActivityRuns TriggerRuns

Microsoft Azure | Data Factory | price-edu | Search factory and documentation | Set up code repository

Data factory  
price-edu

Ingest  
Copy data at scale once or on a schedule

Orchestrate  
Code-free data pipelines

Transform data  
Transform your data using data flows

Configure SSIS  
Manage & run your SSIS packages in the cloud

Recent resources

No items to show

Your recently opened resources will show up here.

Discover more

Browse patterns (preview)

Pipeline template

SAP pipeline template

It's time to create a database by uploading the gold and silver datasets onto Data Factory :

Microsoft Azure | Data Factory | price-edu | Search factory and documentation | Preview experience

Validate all | Publish all

Factory Resources

Activities

Copy data

gold

Properties

Name: data-ingestion

Description:

Annotations

Microsoft Azure | Data Factory | price-edta

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric: Data Factory.

Home Author Monitor Manage Learning Center

Data Factory > Validate all Publish all

Factory Resources

Pipelines data-ingestion

Activities

Copy data

Properties

General

Name: data-ingestion

Description:

Annotations

Add trigger

Preview experience: Off

Copy data

gold

General

Name: gold

Description:

Activity state (preview): Active

Timeout: 0:12:00:00

Retry: 0

Retry interval (sec): 30

Secure output:

Secure input:

Microsoft Azure | Data Factory | price-edta

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric: Data Factory.

Home Author Monitor Manage Learning Center

Data Factory > Validate all Publish all

Factory Resources

Pipelines data-ingestion

Activities

Copy data

New dataset

Select a data store: http

All Azure Database File Generic protocol NoSQL Services and apps

HTTP

Source dataset: Select...

Continue Cancel

Microsoft Azure | Data Factory > price-data

Search factory and documentation

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

Data Factory > Validate all Publish all

Factory Resources

Activities

Copy data

Source dataset

Select...

General Source Sink Mapping Settings User properties

Choose the format type of your data

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |

Copy data

gold

Add trigger

Cancel Back Create

Microsoft Azure | Data Factory > price-data

Search factory and documentation

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

Data Factory > Validate all Publish all

Factory Resources

Activities

Copy data

Source dataset

Select...

General Source Sink Mapping Settings User properties

New linked service

HTTP

Name: gold-HTTP

Description:

Connect via integration runtime: AutoDiscoverIntegrationRuntime

Base URL: https://raw.githubusercontent.com/fahimmuhammedRasimShan230308040DG/gold-silver-price

Server Certificate Validation: Enable

Authentication type: Anonymous

Auth headers:

Annotations:

Parameters:

Advanced

Create Cancel Test connection

Microsoft Azure | Data Factory | price-mets

Search factory and documentation

Microsoft Recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric: Data Factory.

Preview experience: Off

Data Factory - Validate all Publish all

Factory Resources

Activities

Pipelines

data-ingestion

Validate

Validate copy runtime

Debug

Add trigger

Copy data

gold

Properties

General

Name: data-ingestion

Description:

Annotations

General

Source

Sink

Mapping

Settings

User properties

Source dataset: gold

Request method: GET

Additional headers:

Request body:

Request timeout:

Microsoft Azure | Data Factory | price-mets

Search factory and documentation

Microsoft Recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric: Data Factory.

Preview experience: Off

Data Factory - Validate all Publish all

Factory Resources

Activities

Pipelines

data-ingestion

Validate

Validate copy runtime

Debug

Add trigger

Copy data

gold

Properties

General

Name: data-ingestion

Description:

Annotations

General

Source

Sink

Mapping

Settings

User properties

Source dataset: gold

Request method: GET

Additional headers:

Request body:

Request timeout:

Microsoft Azure | Data Factory | price-edu

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

Data Factory > Validate all > Publish all

Factory Resources

Pipelines

data-ingestion

Activities

Copy data

Sink dataset:

General Source Sink Mapping Settings User properties

New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store

All Azure Database File Generic protocol NoSQL Services and apps

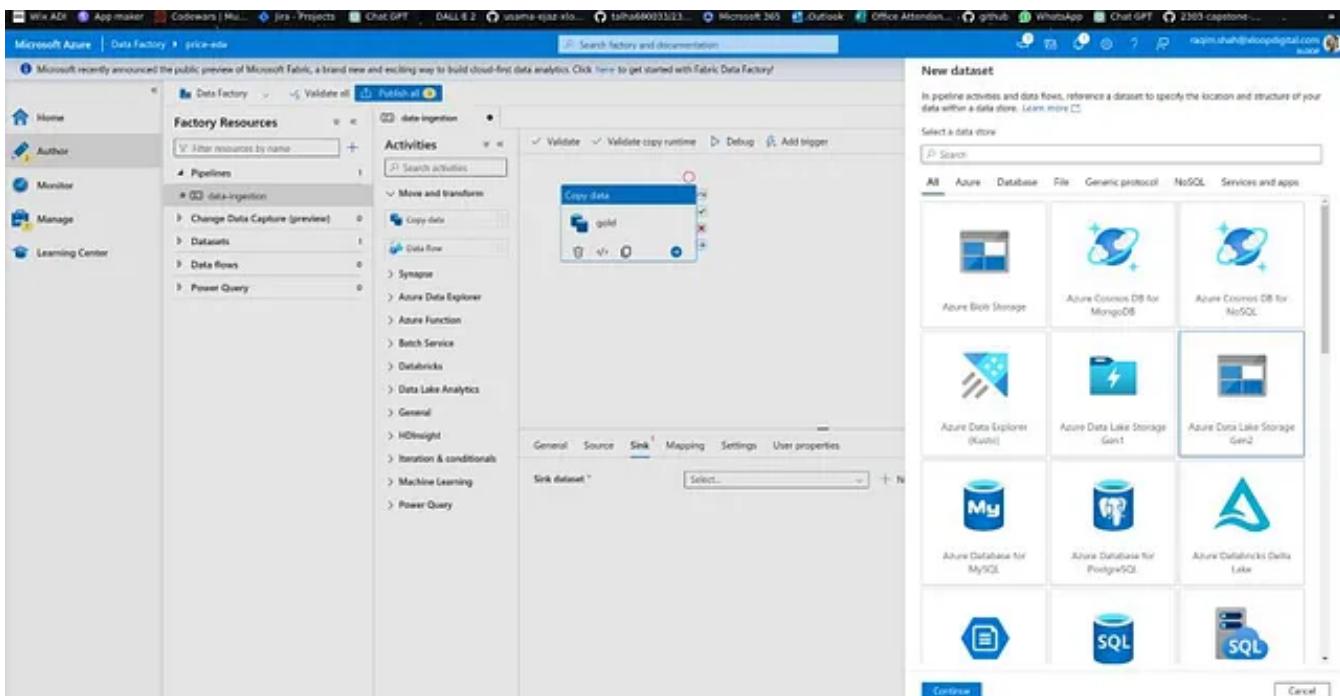
Azure Blob Storage Azure Cosmos DB for MongoDB Azure Document DB for NoSQL

Azure Data Explorer (Kusto) Azure Data Lake Storage Gen1 Azure Data Lake Storage Gen2

Azure Database for MySQL Azure Database for PostgreSQL Azure Databricks Delta Lake

Azure SQL Database SQL

Continue Cancel



Microsoft Azure | Data Factory | price-edu

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

Data Factory > Validate all > Publish all

Factory Resources

Pipelines

data-ingestion

Activities

Copy data

Sink dataset:

General Source Sink Mapping Settings User properties

New linked service

Name: AzureDataLakeStorage1

Description:

Connect via Integration runtime: AutoDiscoverIntegrationRuntime

Authentication type: Account key

Account selection method: From Azure subscription

Azure subscription: Azure subscription 1 (60919054-1e8f-42f6-ba43-7c7d8fa298d)

Storage account name: goldpricedata

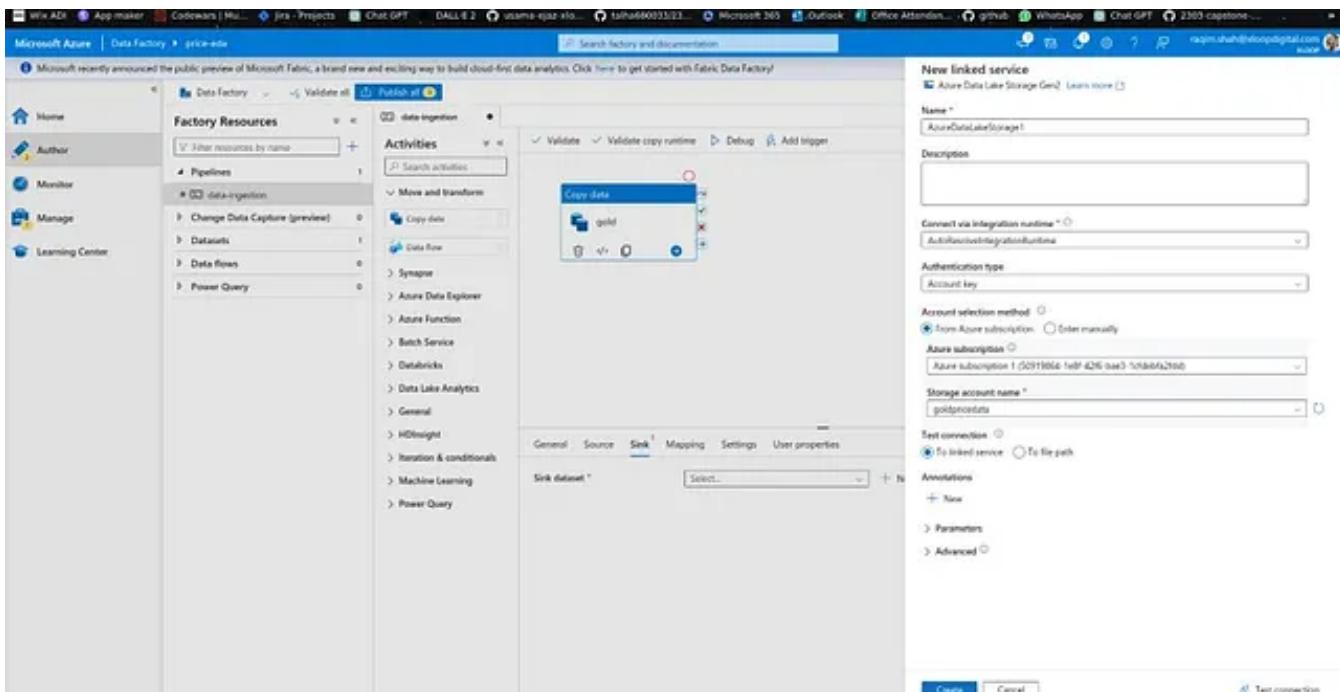
Test connection: To linked service

Annotations: + New

Parameters: >

Advanced: >

Create Cancel Test connection



The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar is open, showing 'Activities' like Pipelines, Data Flows, and Power Query. A 'Copy data' activity is selected. The main pane displays the 'Copy data' configuration dialog. The 'Sink' tab is active, showing a dropdown for 'Sink dataset'. To the right, a 'New linked service' dialog is open, titled 'Azure Data Lake Storage Gen2'. It contains fields for 'Name' (AzureDataLakeStorage1), 'Description', 'Connect via integration runtime' (AutoResolveIntegrationRuntime), 'Authentication type' (Account key), 'Account selection method' (From Azure subscription), 'Azure subscription' (Azure subscription 1), 'Storage account name' (goldpricenext), 'Test connection' (To linked service), and 'Annotations'.

This screenshot shows the same Azure Data Factory interface and pipeline creation process. The 'Copy data' activity is now configured with specific properties. In the 'Sink' tab, the 'File path' is set to 'gold-prices-data / raw-data / gold.csv'. The 'Import schema' section is set to 'From sample file' (selected). The 'Set properties' dialog is open on the right, showing the 'Name' field set to 'ADLS'. Other fields in the 'Set properties' dialog include 'Linked service' (AzureDataLakeStorage1), 'File path' (gold-prices-data / raw-data / gold.csv), 'First row as header' (selected), and 'Import schema' (From sample file selected). The 'OK' button is visible at the bottom of the dialog.

Now for silver-price dataset:

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the navigation menu includes Home, Author, Monitor, Manage, and Learning Center. Under the Author section, the 'Pipelines' tab is selected, showing a pipeline named 'data-ingestion'. This pipeline contains two activities: 'Copy data' from 'gold' to 'silver' (both succeeded) and another 'Copy data' from 'gold' to 'silver' (also succeeded). The pipeline run ID is 82928042-4e4f-4361-8686-53c25a362395, and the status is 'Succeeded'. The interface also includes tabs for Parameters, Variables, Settings, and Output.

Once collected, the raw data is stored in Azure Data Lake Gen 2. Azure Data Lake Storage is an excellent choice for this purpose, as it can handle large volumes of structured and unstructured data, ensuring scalability and flexibility.

To build a connection between Databricks and ADLS(Azure Data Lake Storage), we have to make a connection using app registration, this is step involved that we have to create some apps and get some credentials to create the connection from Azure Databricks to the ADLS(Azure Data Lake Storage)

The screenshot shows the Microsoft Azure portal's 'Access Control (IAM)' blade for a resource named 'gold-price-data'. The 'Access Control (IAM)' tab is selected. The page includes sections for 'My access' (with a 'View my access' button) and 'Check access' (with a 'Check access' button). Below these are three cards: 'Grant access to this resource' (with a 'Add role assignment' button), 'View access to this resource' (with a 'View' button), and 'View deny assignments' (with a 'View' button).

Microsoft Azure Upgrade Search resources, services, and docs (Ctrl+F)

Home > goldprice-data | Containers > gold-price-data

## gold-price-data | Access Control (IAM) ...

Containers

Add role assignment Add role assignments Edit columns Refresh × Report Feedback

Add role assignment Add co-administrator Roles Deny assignments Classic administrators

My access View my level of access to this resource View my access

Check access Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more Check access

Grant access to this resource Grant access to resources by assigning a role. Learn more Add role assignment

View access to this resource View the role assignments that grant access to this and other resources. Learn more View

View deny assignments View the role assignments that have been denied access to specific actions at this scope. Learn more View

Settings Shared access tokens Manage ACL Access policy Properties Metadata

Microsoft Azure Upgrade Search resources, services, and docs (Ctrl+F)

Home > goldprice-data | Containers > gold-price-data | Access Control (IAM) >

## Add role assignment ...

Role Members Conditions (optional) Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more

Assignment type:

Job Function roles Privileged administrator roles

Grant access to Azure resources based on job functions such as the ability to create virtual machines.

storage blob

| Name                              | Description   | Type        | Category | Details |
|-----------------------------------|---|-------------|----------|---------|
| Defender for Storage Data Scanner | Grants access to read blobs and update index tags. This role is used by the data scanner of Defender for Storage. | BuiltInRole | None     | View    |
| Storage Blob Data Contributor     | Allows for read, write and delete access to Azure Storage Blob containers and data.                               | BuiltInRole | Storage  | View    |
| Storage Blob Data Owner           | Allows for full access to Azure Storage Blob containers and data, including assigning POSIX access control.       | BuiltInRole | Storage  | View    |
| Storage Blob Data Reader          | Allows for read access to Azure Storage Blob containers and data.   | BuiltInRole | Storage  | View    |
| Storage Blob Data Writer          | Allows for generation of a user delegation key which can be used to sign SAS tokens.                              | BuiltInRole | Storage  | View    |

Showing 1 - 5 of 5 results.

Review + assign Previous Next Feedback

**Add role assignment**

**Role** Members **Conditions (optional)** Review + assign

**Selected rule** Storage Blob Data Contributor

**Assign access to**  User, group, or service principal  Managed identity

**Members**  Select members

| Name                | Object ID | Type |
|---------------------|-----------|------|
| No members selected |           |      |

**Description**

**Select members**

Selected members: No members selected. Search for and add one or more members you want to assign the role for this resource.

Learn more about RBAC

**Review + assign** **Previous** **Next** **Select** **Close**

**gold-price-data | Access Control (IAM)**

**Check access** **Role assignments** **Roles** **Deny assignments** **Classic administrators**

**My access** View my level of access to this resource. **View my access**

**Check access** Review the level of access a user, group, service principal, or managed identity has to this resource. **Learn more** **Check access**

**Grant access to this resource** Grant access to resources by assigning a role. **Learn more** **Add role assignment**

**View access to this resource** View the role assignments that grant access to this and other resources. **Learn more** **View**

**View deny assignments** View the role assignments that have been denied access to specific actions at this scope. **Learn more** **View**

**goldprice** Resource group

**Overview** **Create** **Manage view** **Delete resource group** **Refresh** **Export to CSV** **Open query** **Assign tags** **Move** **Delete** **Export template** **JSON View**

**Resources** **Recommendations**

Filter for any field... Type equals all Location equals all Add filter

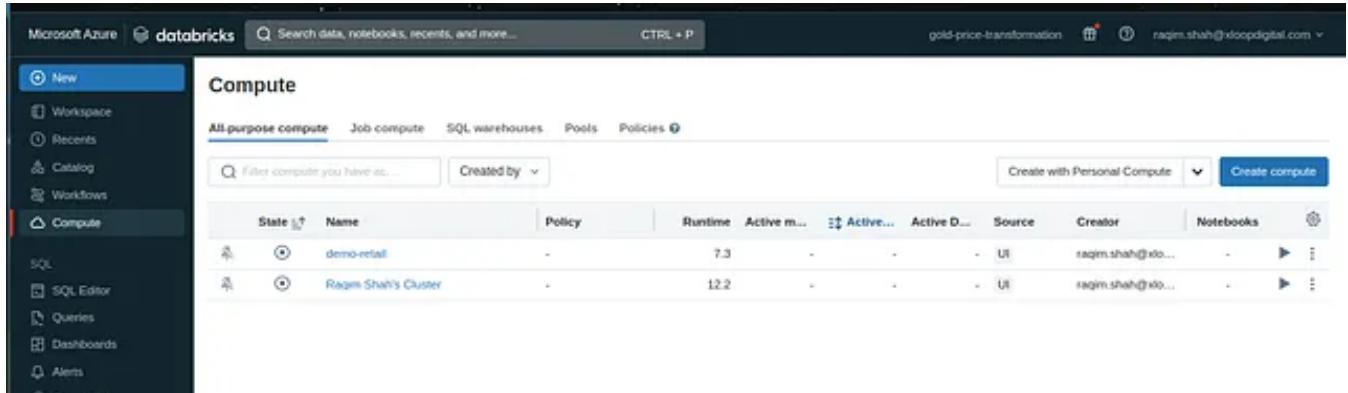
Showing 1 to 5 of 5 records.  Show hidden types

| Name                      | Type                     | Location |
|---------------------------|--------------------------|----------|
| analysis-gold-price       | Data factory (V2)        | East US  |
| analysis-sa               | Synapse workspace        | East US  |
| gold-price-transformation | Azure Databricks Service | East US  |
| goldpricedata             | Storage account          | East US  |
| price-edw                 | Data factory (V2)        | East US  |

## Step 2: Data Preparation

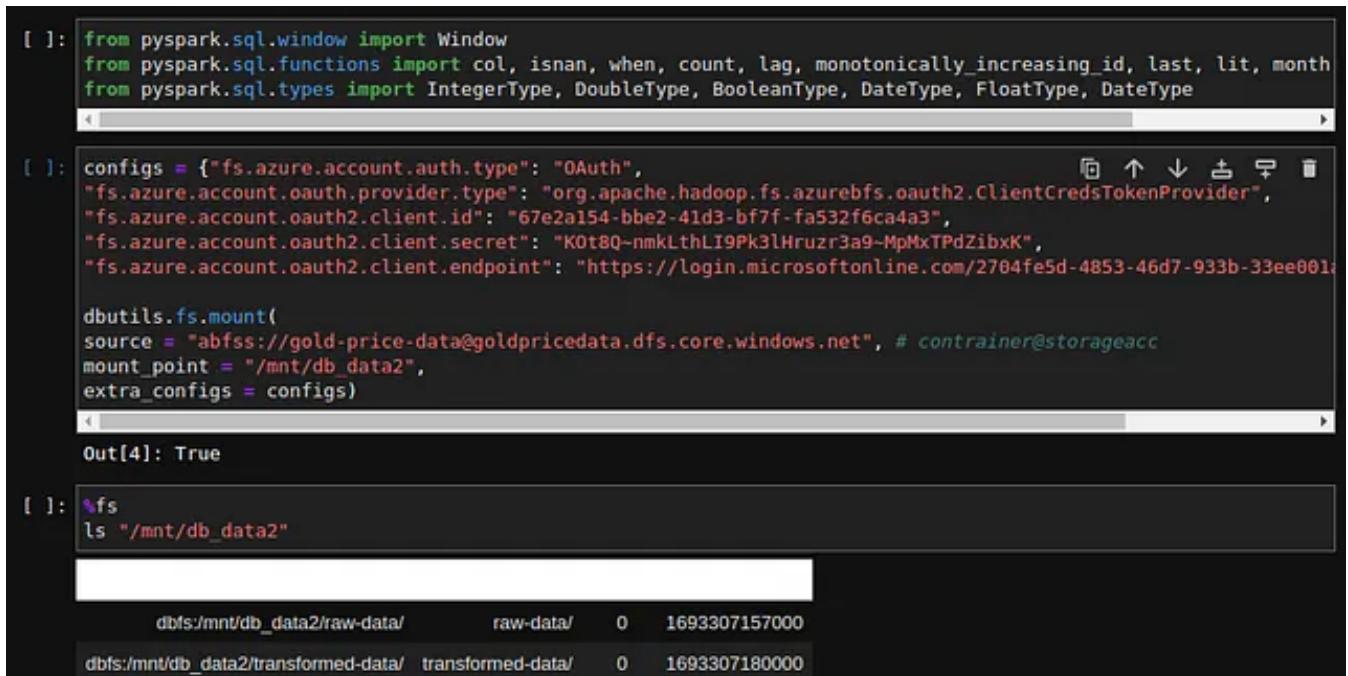
Before diving into analysis, it's essential to clean and transform the data as needed. Azure Databricks would be a valuable tool in this step, assisting in tasks such as handling missing values, data normalization, and feature engineering. Clean data is the foundation of meaningful analysis using Spark.

We created new compute cluster having name Raqim Shah's Cluster



The screenshot shows the Microsoft Azure Databricks Compute page. On the left, there is a sidebar with options like New, Workspace, Recents, Catalog, Workflows, Compute (which is selected), SQL, SQL Editor, Queries, Dashboards, and Alerts. The main area is titled 'Compute' and shows a table of existing clusters. The columns include State, Name, Policy, Runtime, Active m..., Active ..., Active D..., Source, Creator, Notebooks, and three more icons. Two clusters are listed: 'demo-retail' and 'Raqim Shah's Cluster'. The 'Raqim Shah's Cluster' row shows a runtime of 12.2 minutes, UI as the source, and Raqim Shah as the creator.

Making connections in Databricks and import required libraries and data from storage account.



```
[1]: from pyspark.sql.window import Window
from pyspark.sql.functions import col, isnan, when, count, lag, monotonically_increasing_id, last, lit, month
from pyspark.sql.types import IntegerType, DoubleType, BooleanType, DateType, FloatType, DataType

[2]: configs = {"fs.azure.account.auth.type": "OAuth",
"fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
"fs.azure.account.oauth2.client.id": "67e2a154-bbe2-41d3-bf7f-fa532f6ca4a3",
"fs.azure.account.oauth2.client.secret": "K0t8Q-nmkLthLI9Pk3lHruzr3a9-MpMxTPdZibxK",
"fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/2704fe5d-4853-46d7-933b-33ee001"}
```

```
[3]: dbutils.fs.mount(
source = "abfss://gold-price-data@goldpricedata.dfs.core.windows.net", # contrainer@storageacc
mount_point = "/mnt/db_data2",
extra_configs = configs)
```

```
[4]: Out[4]: True
```

```
[5]: !fs
ls "/mnt/db_data2"
```

| Path                                 | Type              | Size | Last Modified |
|--------------------------------------|-------------------|------|---------------|
| dbfs:/mnt/db_data2/raw-data/         | raw-data/         | 0    | 1693307157000 |
| dbfs:/mnt/db_data2/transformed-data/ | transformed-data/ | 0    | 1693307180000 |

```
[ ]: spark
      SparkSession - hive
      SparkContext
      Spark UI
      Version          v3.3.2
      Master           local[*, 4]
      AppName         Databricks Shell

      Loading Data from raw-data folder from account-storage

[ ]: gold = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/db_data2/raw-data/gold.csv")
silver = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/db_data2/raw-data/silver.csv")
```

## Transformation of Data

```
[ ]: gold.show()
+-----+-----+-----+-----+
|     Date|Close/Last|Volume|  Open|  High|  Low|
+-----+-----+-----+-----+
|2023-08-17|    1915.2|146770|1922.4|1933.5|1914.2|
|2023-08-16|    1928.3|124766|1933.1|1938.2|1922.0|
|2023-08-15|    1935.2|161512|1939.4|1944.3|1927.5|
|2023-08-14|    1944.0|117514|1945.6|1948.2|1934.2|
|2023-08-11|    1946.0|117514|1945.6|1948.2|1934.2|
```

```
[ ]: gold.printSchema()
root
 |-- Date: date (nullable = true)
 |-- Close/Last: double (nullable = true)
 |-- Volume: string (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)

[ ]: gold = gold.withColumn("Date",col("Date").cast(DateType()))\
    .withColumn("Close/Last",col("Close/Last").cast(FloatType()))\
    .withColumn("Volume",col("Volume").cast(DoubleType()))\
    .withColumn("Open",col("Open").cast(FloatType()))\
    .withColumn("High",col("High").cast(FloatType()))\
    .withColumn("Low",col("Low").cast(FloatType()))

[ ]: gold.printSchema()
root
 |-- Date: date (nullable = true)
 |-- Close/Last: float (nullable = true)
 |-- Volume: double (nullable = true)
 |-- Open: float (nullable = true)
 |-- High: float (nullable = true)
 |-- Low: float (nullable = true)
```

```
[ ]: gold_Columns=["Date", "Close/Last", "Volume", "Open", "High", "Low"]
missing_counts_expressions = []

# Loop through columns and create expressions to count missing values based on data type
for c in gold_Columns:
    if c == "Date":
        expr = count(when(col(c).isNull(), c)).alias(c)
    elif c in ["Close/Last", "Open", "High", "Low"]:
        expr = count(when(col(c).isNull() | isnan(col(c)), c)).alias(c)
    else:
        expr = count(when(col(c).isNull(), c)).alias(c)
    missing_counts_expressions.append(expr)

missing_counts = gold.select(missing_counts_expressions)
missing_counts.show()

+---+-----+-----+-----+---+
|Date|Close/Last|Volume|Open|High|Low|
+---+-----+-----+-----+---+
| 0|     0|    28|   0|   0|  0|
+---+-----+-----+-----+---+
```

## Removing null values from Volumn Column

```
[ ]: gold.count()
Out[29]: 2539
```

```
[ ]: # Specify the column name
column_name = "Volume"

# Create a Window specification to order by the DataFrame's index or any other suitable column
window_spec = Window.orderBy(monotonically_increasing_id())

# Replace null values with the previous non-null value using the last() window function
gold = gold.withColumn(column_name, last(column_name, True).over(window_spec))
gold.show()

+---+-----+-----+-----+---+
| Date|Close/Last| Volume| Open| High| Low|
+---+-----+-----+-----+---+
```

[Open in app ↗](#)



```
[ ]: gold_Columns=["Date", "Close/Last", "Volume", "Open", "High", "Low"]
missing_counts_expressions = []

# Loop through columns and create expressions to count missing values based on data type
for c in gold_Columns:
    if c == "Date":
        expr = count(when(col(c).isNull(), c)).alias(c)
    elif c in ["Close/Last", "Open", "High", "Low"]:
        expr = count(when(col(c).isNull() | isnan(col(c)), c)).alias(c)
    else:
        expr = count(when(col(c).isNull(), c)).alias(c)
    missing_counts_expressions.append(expr)

missing_counts = gold.select(missing_counts_expressions)
missing_counts.show()

+---+-----+-----+-----+---+
|Date|Close/Last|Volume|Open|High|Low|
+---+-----+-----+-----+---+
| 0|     0|    0|   0|   0|  0|
+---+-----+-----+-----+---+
```

```
[ ]: silver.printSchema()
root
 |-- Date: date (nullable = true)
 |-- Close/Last: double (nullable = true)
 |-- Volume: string (nullable = true)
 |-- Open: double (nullable = true)
```

```

|-- Open: double (nullable = true)
|-- High: double (nullable = true)
|-- Low: double (nullable = true)

[ ]: silver = silver.withColumn("Date", col("Date").cast(DateType()))\
    .withColumn("Close/Last", col("Close/Last").cast(FloatType()))\
    .withColumn("Volume", col("Volume").cast(DoubleType()))\
    .withColumn("Open", col("Open").cast(FloatType()))\
    .withColumn("High", col("High").cast(FloatType()))\
    .withColumn("Low", col("Low").cast(FloatType()))

[ ]: silver.printSchema()

root
|-- Date: date (nullable = true)
|-- Close/Last: float (nullable = true)
|-- Volume: double (nullable = true)
|-- Open: float (nullable = true)
|-- High: float (nullable = true)
|-- Low: float (nullable = true)

```

```

[ ]: silver_Columns=["Date", "Close/Last", "Volume", "Open", "High", "Low"]
missing_counts_expressions = []

# Loop through columns and create expressions to count missing values based on data type
for c in silver_Columns:
    if c == "Date":
        expr = count(when(col(c).isNull(), c)).alias(c)
    elif c in ["Close/Last", "Open", "High", "Low"]:
        expr = count(when((col(c).isNull() | isnan(col(c))), c)).alias(c))
    else:
        expr = count(when(col(c).isNull(), c)).alias(c)
    missing_counts_expressions.append(expr)

missing_counts = silver.select(missing_counts_expressions)
missing_counts.show()

+----+-----+----+----+----+
|Date|Close/Last|Volume|Open|High|Low|
+----+-----+----+----+----+
|   0|     0|   63|   0|   0|   0|
+----+-----+----+----+----+

```

```

[ ]: # Specify the column name
column_name = "Volume"

# Create a Window specification to order by the DataFrame's index or any other suitable column
window_spec = Window.orderBy(monotonically_increasing_id())

# Replace null values with the previous non-null value using the last() window function
silver = silver.withColumn(column_name, last(column_name, True).over(window_spec))
silver.show()

+-----+-----+-----+-----+-----+
|      Date|Close/Last| Volume|  Open|  High|  Low|
+-----+-----+-----+-----+-----+
|2023-08-17|  23.042|11441.0| 22.8|23.385|22.725|
|2023-08-16|  22.856|10802.0|22.905| 23.17| 22.77|
|2023-08-15|  22.656|60396.0| 22.65| 22.77|22.265|
|2023-08-14|  22.708|51908.0| 22.74| 22.82| 22.41|
|2023-08-11|  22.743|48043.0| 22.76| 22.91| 22.61|
|2023-08-10|  22.821|71226.0| 22.73| 23.06|22.665|
|2023-08-09|  22.731|60561.0| 22.82| 22.99| 22.68|
|2023-08-08|  22.807|73338.0| 23.2|23.255| 22.72|
|2023-08-07|  23.232|55345.0| 23.72|23.775|23.145|
|2023-08-04|  22.716|556747.0| 22.71|22.805|22.325|

```

```
[ ]: silver_Columns=["Date", "Close/Last", "Volume", "Open", "High", "Low"]
missing_counts_expressions = []

# Loop through columns and create expressions to count missing values based on data type
for c in silver_Columns:
    if c == "Date":
        expr = count(when(col(c).isNull(), c)).alias(c)
    elif c in ["Close/Last", "Open", "High", "Low"]:
        expr = count(when(col(c).isNull() | isnan(col(c)), c)).alias(c)
    else:
        expr = count(when(col(c).isNull(), c)).alias(c)
    missing_counts_expressions.append(expr)

missing_counts = silver.select(missing_counts_expressions)
missing_counts.show()

+---+-----+-----+-----+---+
|Date|Close/Last|Volume|Open|High|Low|
+---+-----+-----+-----+---+
|  0|       0|     0|   0|   0|   0|
+---+-----+-----+-----+---+
```

```
[ ]: gold = gold.withColumn("Category", lit("gold")).withColumn("Currency", lit("USD"))
```

```
[ ]: silver = silver.withColumn("Category", lit("silver")).withColumn("Currency", lit("USD"))
```

```
[ ]: gold = gold.withColumn("Month", month(col("Date"))).withColumn("Year", year(col("Date")))
gold = gold.withColumn("DayOfWeek", date_format("Date", "EEEE"))
```

```
[ ]: silver = silver.withColumn("Month", month(col("Date"))).withColumn("Year", year(col("Date")))
silver = silver.withColumn("DayOfWeek", date_format("Date", "EEEE"))
```

```
[ ]: silver.show()
```

| Date       | Close/Last | Volume  | Open   | High   | Low    | Category | Currency | Month | Year | DayOfWeek |
|------------|------------|---------|--------|--------|--------|----------|----------|-------|------|-----------|
| 2023-08-17 | 23.042     | 11441.0 | 22.8   | 23.385 | 22.725 | silver   | USD      | 8     | 2023 | Thursday  |
| 2023-08-16 | 22.856     | 10802.0 | 22.985 | 23.17  | 22.77  | silver   | USD      | 8     | 2023 | Wednesday |
| 2023-08-15 | 22.656     | 60396.0 | 22.65  | 22.77  | 22.265 | silver   | USD      | 8     | 2023 | Tuesday   |
| 2023-08-14 | 22.708     | 51908.0 | 22.74  | 22.82  | 22.41  | silver   | USD      | 8     | 2023 | Monday    |
| 2023-08-11 | 22.743     | 48043.0 | 22.76  | 22.91  | 22.61  | silver   | USD      | 8     | 2023 | Friday    |
| 2023-08-10 | 22.821     | 71226.0 | 22.73  | 23.06  | 22.665 | silver   | USD      | 8     | 2023 | Thursday  |
| 2023-08-09 | 22.731     | 60561.0 | 22.82  | 22.99  | 22.68  | silver   | USD      | 8     | 2023 | Wednesday |
| 2023-08-08 | 22.807     | 73338.0 | 23.2   | 23.255 | 22.72  | silver   | USD      | 8     | 2023 | Tuesday   |
| 2023-08-07 | 23.232     | 55345.0 | 23.72  | 23.775 | 23.145 | silver   | USD      | 8     | 2023 | Monday    |
| 2023-08-04 | 23.716     | 56747.0 | 23.71  | 23.895 | 23.275 | silver   | USD      | 8     | 2023 | Friday    |
| 2023-08-03 | 23.697     | 58992.0 | 23.835 | 23.94  | 23.41  | silver   | USD      | 8     | 2023 | Thursday  |
| 2023-08-02 | 23.872     | 71917.0 | 24.48  | 24.63  | 23.76  | silver   | USD      | 8     | 2023 | Wednesday |
| 2023-08-01 | 24.326     | 51464.0 | 24.9   | 24.905 | 24.255 | silver   | USD      | 8     | 2023 | Tuesday   |
| 2023-07-31 | 24.972     | 44254.0 | 24.475 | 24.985 | 24.32  | silver   | USD      | 7     | 2023 | Monday    |
| 2023-07-28 | 24.495     | 38405.0 | 24.265 | 24.545 | 24.24  | silver   | USD      | 7     | 2023 | Friday    |
| 2023-07-27 | 24.367     | 81985.0 | 25.075 | 25.325 | 24.18  | silver   | USD      | 7     | 2023 | Thursday  |

## Saving Transformed data into transformed-data folder in account-storage

```
[ ]: gold.repartition(1).write.mode("overwrite").option("header","true").csv("/mnt/db_data2/transformed-data/gold"
silver.repartition(1).write.mode("overwrite").option("header","true").csv("/mnt/db_data2/transformed-data/sil
[ ]]

[ ]: gold.printSchema()

root
 |-- Date: date (nullable = true)
 |-- Close/Last: float (nullable = true)
 |-- Volume: double (nullable = true)
 |-- Open: float (nullable = true)
 |-- High: float (nullable = true)
 |-- Low: float (nullable = true)
 |-- Category: string (nullable = false)
 |-- Currency: string (nullable = false)
 |-- Month: integer (nullable = true)
 |-- Year: integer (nullable = true)
 |-- DayOfWeek: string (nullable = true)

[ ]:
```

## Step 3: Data Warehousing

Now, the cleaned data is loaded into Azure Synapse Analytics, previously known as SQL Data Warehouse. Synapse Analytics is a powerful data warehousing solution that provides analytical capabilities necessary for in-depth examination of the gold price data. It acts as a central repository for structured data, facilitating efficient querying.

In Azure Synapse Analytics, we can perform a wide range of analyses on the gold price data. This includes running SQL queries for aggregations, descriptive statistics, conducting time-series analysis, and identifying trends and patterns. The platform's analytical capabilities empower us to derive meaningful insights from our data.

Microsoft Azure Upgrade Search resources, services, and docs (beta)

Home > Azure Synapse Analytics > Create Synapse workspace

**Basics** **Security** Networking Tags Review + create

Create a Synapse workspace to develop an enterprise analytics solution in just a few clicks.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

Subscription \*  Resource group \*  Create new Managed resource group \*

**Workspace details**

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

Workspace name \*  Region \*  Select Data Lake Storage Gen2 \*   Account name \*  Create new File system name \*  Create new  Assign myself the Storage Blob Data Contributor role on the Data Lake Storage Gen2 account to interactively query it in the workspace.

**Review + create** **< Previous** **Next: Security >**

Microsoft Azure Upgrade Search resources, services, and docs (beta)

Home > Azure Synapse Analytics > Create Synapse workspace

**Basics** **Security** **Networking** Tags Review + create

Configure security options for your workspace.

**Authentication**

Choose the authentication method for access to workspace resources such as SQL pools. The authentication method can be changed later on. [Learn more](#)

Authentication method \*

SQL Server admin login \*  SQL Password \*  Confirm password \*

**System assigned managed identity permission**

Select to grant the workspace identity access to the Data Lake Storage Gen2 account using the workspace system identity. [Learn more](#)

Allow network access to Data Lake Storage Gen2 account. The selected Data Lake Storage Gen2 account does not restrict network access using any network access rules, or you selected a storage account manually via URL under Basics tab. [Learn more](#)

**Workspace encryption**

Double encryption configuration cannot be changed after opting into using a customer-managed key at the time of workspace creation.

Choose to encrypt all data at rest in the workspace with a key managed by you (customer-managed key). This will provide double encryption with encryption at the infrastructure layer that uses platform-managed keys. [Learn more](#)

Double encryption using a customer-managed key \*

**Review + create** **< Previous** **Next: Networking >**

Microsoft Azure | Synapse Analytics > analysis-1a

Search

Synapse live Validate all Publish all 1

Data Workspace Linked

Filter resources by name

Lake database Database1

Tables gold silver

SQL script 1

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Run Undo Publish Query plan Connect to Built-in Use database Database1

```
1 SELECT gold.[Date], gold.[Close_Last] AS gold_price, silver.[Close_Last] AS silver_price
2 FROM gold
3 JOIN silver ON silver.[Date] = gold.[Date]
4 GROUP BY gold.[Date], gold.[Close_Last], silver.[Close_Last]
5 ORDER BY gold.[Date];
```

Properties

General Related (0)

Name \* SQL script 1

Description

Type sql script

Size 0 bytes

Results settings per query

First 5000 rows (default)

All rows

Results Messages

View Table Chart Save as image

Chart type Line

Category column Date

Legend (series) columns gold\_price, silver\_price

Legend position bottom - center

Legend (series) label

00:00:06 Query executed successfully.

| Date       | gold_price | silver_price |
|------------|------------|--------------|
| 2013-01-01 | 1800       | 1500         |
| 2014-01-01 | 1900       | 1600         |
| 2015-01-01 | 2000       | 1700         |
| 2016-01-01 | 2100       | 1800         |
| 2017-01-01 | 2200       | 1900         |
| 2018-01-01 | 2300       | 2000         |
| 2019-01-01 | 2400       | 2100         |
| 2020-01-01 | 2500       | 2200         |
| 2021-01-01 | 2600       | 2300         |
| 2022-01-01 | 2700       | 2400         |
| 2023-01-01 | 2800       | 2500         |

Microsoft Azure | Synapse Analytics > analysis-1a

Search

Synapse live Validate all Publish all 1

Data Workspace Linked

Filter resources by name

Lake database Database1

Tables gold silver

SQL script 1

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Run Undo Publish Query plan Connect to Built-in Use database Database1

```
1 SELECT gold.[Date], gold.[Close_Last] AS gold_price, silver.[Close_Last] AS silver_price
2 FROM gold
3 JOIN silver ON silver.[Date] = gold.[Date]
4 GROUP BY gold.[Date], gold.[Close_Last], silver.[Close_Last]
5 ORDER BY gold.[Date];
```

Properties

General Related (0)

Name \* SQL script 1

Description

Type sql script

Size 0 bytes

Results settings per query

First 5000 rows (default)

All rows

Results Messages

View Table Chart Save as image

Chart type Line

Category column Date

Legend (series) columns gold\_price, silver\_price

Legend position bottom - center

Legend (series) label

00:00:06 Query executed successfully.

| Date       | gold_price | silver_price |
|------------|------------|--------------|
| 2013-01-01 | 1800       | 1500         |
| 2014-01-01 | 1900       | 1600         |
| 2015-01-01 | 2000       | 1700         |
| 2016-01-01 | 2100       | 1800         |
| 2017-01-01 | 2200       | 1900         |
| 2018-01-01 | 2300       | 2000         |
| 2019-01-01 | 2400       | 2100         |
| 2020-01-01 | 2500       | 2200         |
| 2021-01-01 | 2600       | 2300         |
| 2022-01-01 | 2700       | 2400         |
| 2023-01-01 | 2800       | 2500         |

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, the Data pane displays a 'Lake database' with 'Database1' containing 'Tables' like 'gold' and 'silver'. The main area shows a SQL script titled 'SQL script 1' with the following code:

```

1 SELECT gold.[Date], gold.[Close_Last] AS gold_price, silver.[Close_Last] AS silver_price
2 FROM gold
3 JOIN silver ON silver.[Date] = gold.[Date]
4 GROUP BY gold.[Date], gold.[Close_Last], silver.[Close_Last]
5 ORDER BY gold.[Date];

```

The results pane shows a line chart with two series: 'gold\_price' (blue line) and 'silver\_price' (orange line). The chart displays price fluctuations over time from 2013-08-19 to 2021-04-17. The properties pane on the right indicates the script is a 'sql script' with 0 bytes size.

This screenshot shows the same workspace environment. The Data pane now displays 'Database1' with tables 'gold' and 'silver' under their respective columns. The results pane shows the same SQL script as before, but the output is a table with the following data:

| Date                        | gold_price | silver_price |
|-----------------------------|------------|--------------|
| 2013-08-19T00:00:00.0000000 | 1965.7     | 23.21        |
| 2013-08-20T00:00:00.0000000 | 1971.6     | 23.11        |
| 2013-08-21T00:00:00.0000000 | 1970.1     | 23.09        |
| 2013-08-22T00:00:00.0000000 | 1970.8     | 23.08        |
| 2013-08-23T00:00:00.0000000 | 1995.8     | 23.78        |

The properties pane shows the script is a 'sql script' with 153 bytes size.

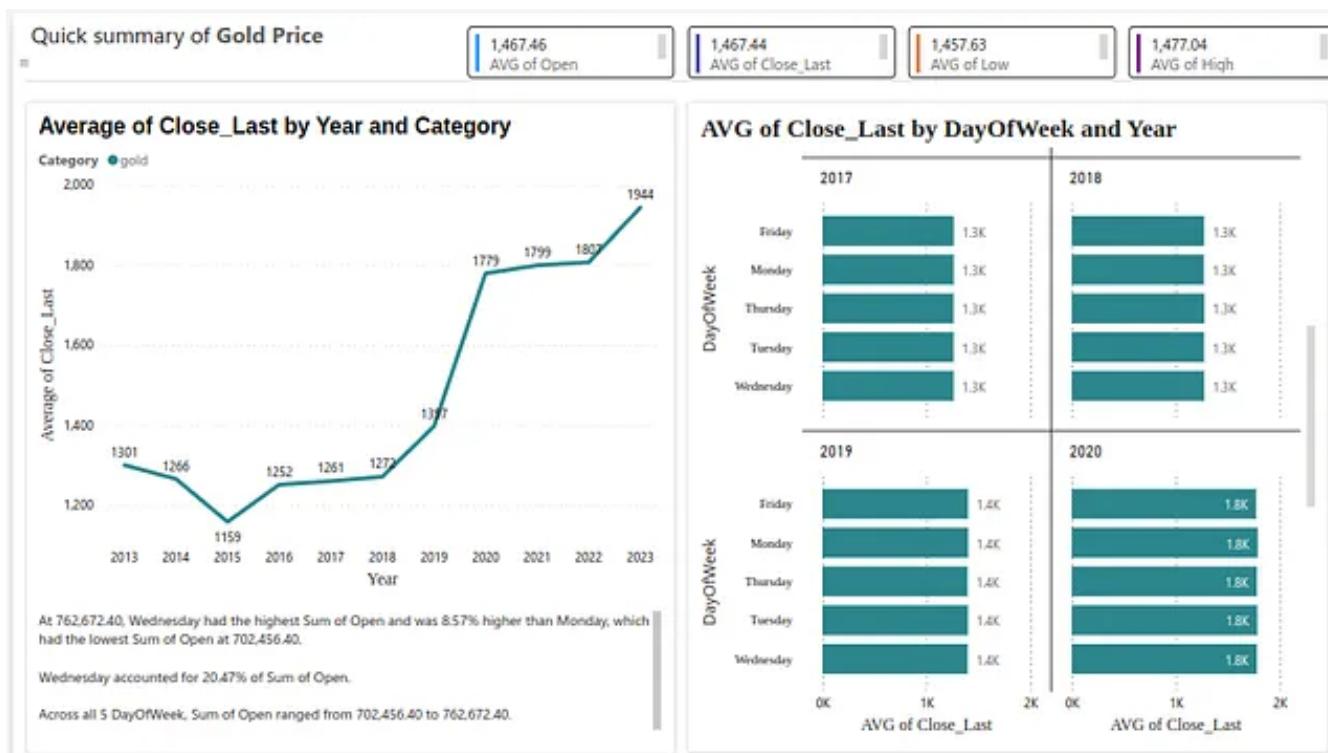
## Step 4: Data Visualization

To effectively communicate our findings, we create compelling visualizations using tools like Power BI. These visualization tools enable us to build interactive dashboards and reports, making it easier to convey the insights generated during the analysis. Visualization enhances data-driven decision-making by presenting information in a digestible format.

Now it's time for visualization using the Power Bi plug-in and there are most easy steps to create Bi report using automatic report generation. Here is the Bi report:

- On the first page of the report, we are visualizing the average of close\_last\_price of gold by year using a line graph.

- An average of close\_last\_price of gold by weekdays of each year.
- Also, we have small boxes representing average of low, high, open, and close of gold price.



## Quick summary of Gold Price

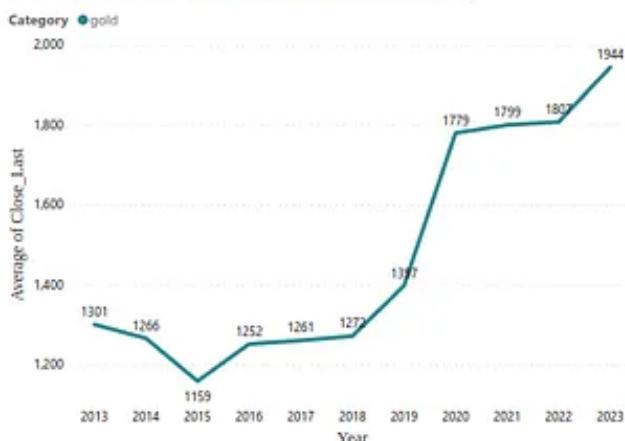
1,467.46  
AVG of Open

1,467.44  
AVG of Close\_Last

1,457.63  
AVG of Low

1,477.04  
AVG of High

### Average of Close\_Last by Year and Category

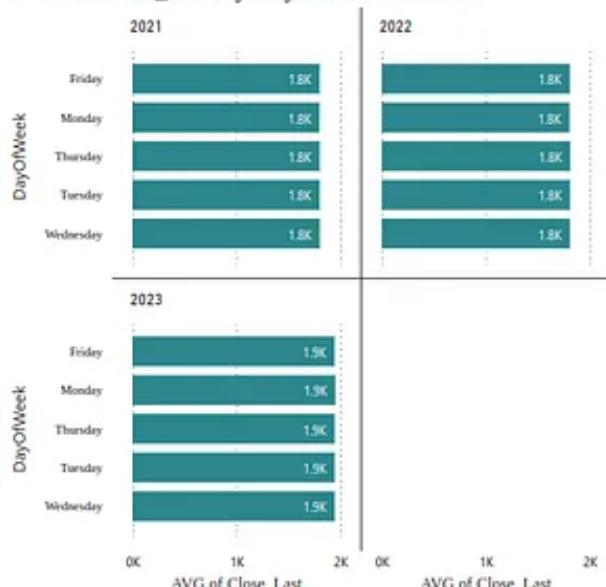


At 762,672.40, Wednesday had the highest Sum of Open and was 8.57% higher than Monday, which had the lowest Sum of Open at 702,456.40.

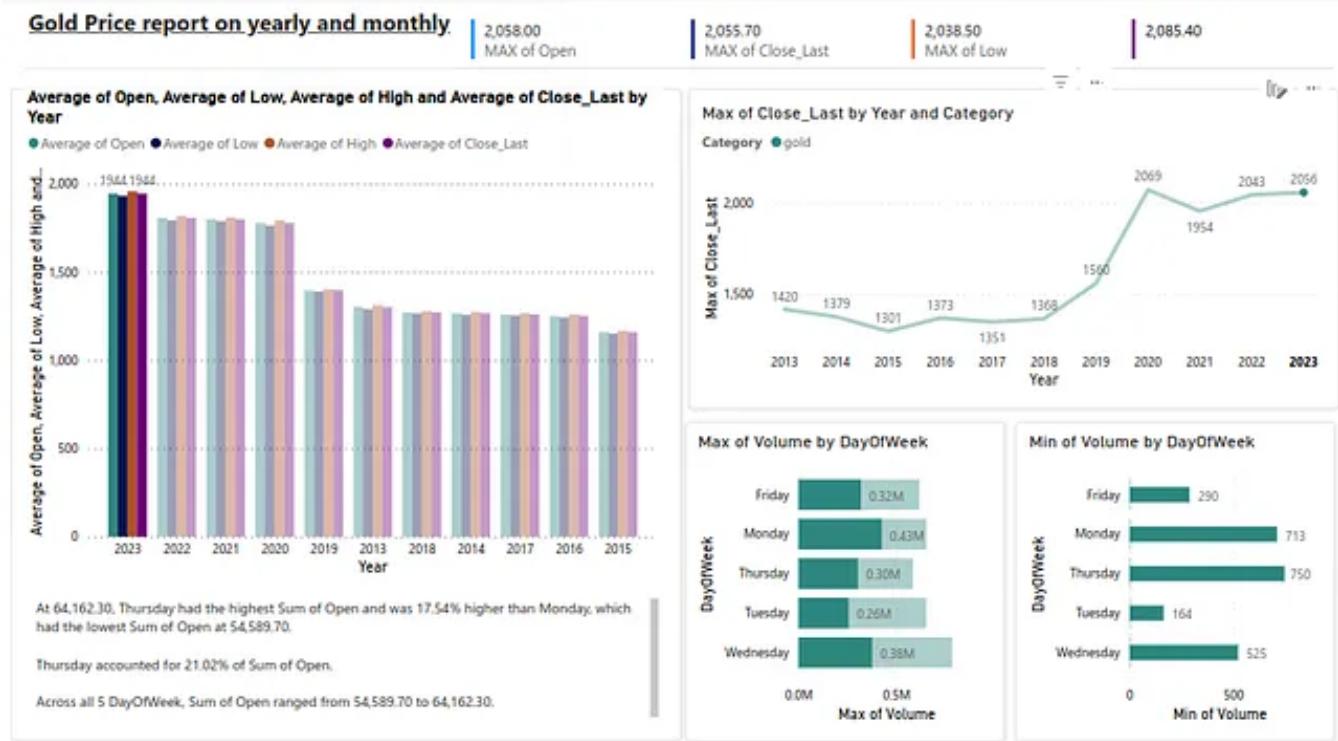
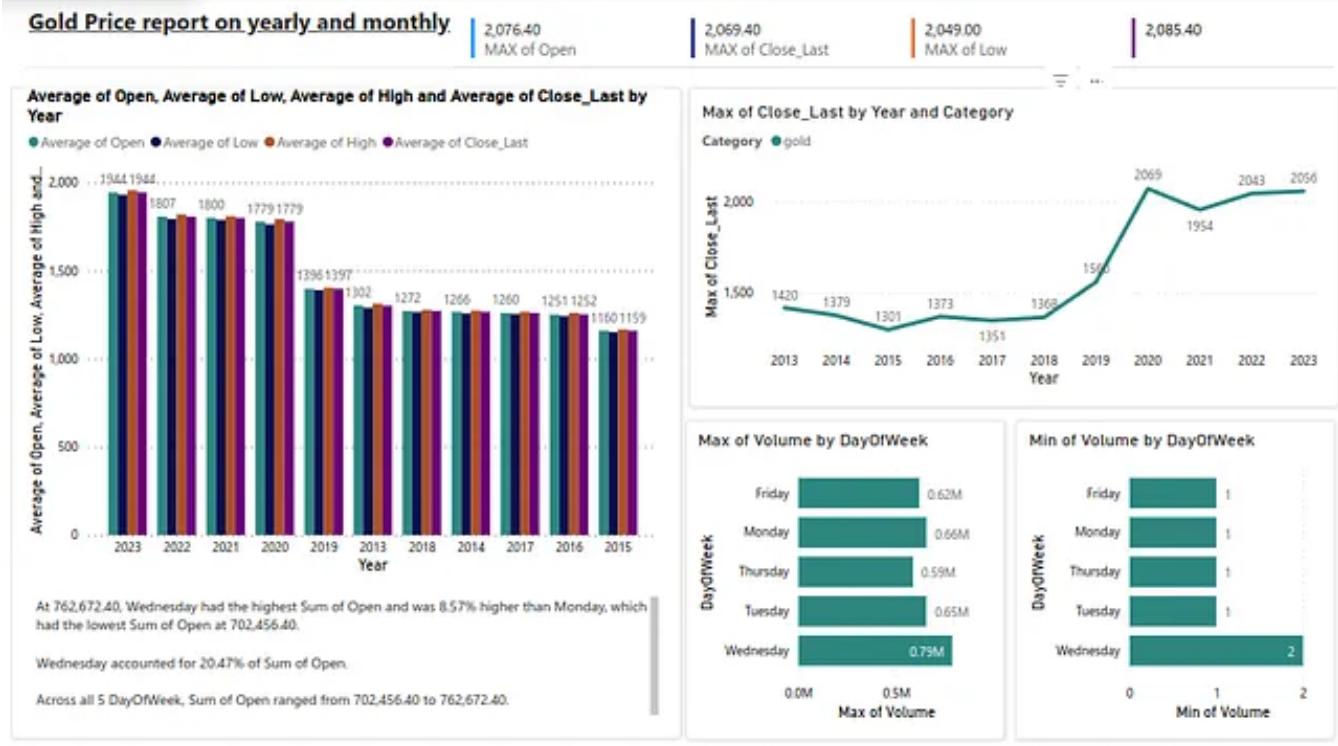
Wednesday accounted for 20.47% of Sum of Open.

Across all 5 DayOfWeek, Sum of Open ranged from 702,456.40 to 762,672.40.

### AVG of Close\_Last by DayOfWeek and Year



- Now on the second page of the report, we are visualizing the average of close\_last\_price, average of open\_price, average of high, and average of low of gold by year using a bar graph.
- Maximum of close\_last price by year
- And maximum and minimum volume of gold by weekdays.
- Also, we have small boxes representing the maximum low, high, open, and close gold prices.
- The fun thing in Power Bi is when we click on weekday from the column it selects that weekday and provides us insights into that day for the all visuals provided.



**Conclusion:** Analyzing gold price data with Azure services is a systematic process that involves data ingestion, preparation, warehousing, analysis, and visualization. Each step plays a vital role in unlocking valuable insights from the data. Azure Data Factory, Data Lake Gen 2, Synapse Analytics, and visualization tools work together seamlessly to enable data-driven decision-making in the dynamic realm of gold prices. By following these steps, organizations can stay competitive and make informed investment decisions based on data-backed insights.

Data Engineer

Data Analysis

Azure

Stock Market

Synapse

S

Edit profile

## Written by Syed Muhammad Raqim Ali Shah

0 Followers

More from Syed Muhammad Raqim Ali Shah



 Syed Muhammad Raqim Ali Shah

### Worth of My Time

People think “Time is Money”, But what we think is “Money is Time”. As we do work we earn money in exchange for our time. The worth of our...

5 min read · Aug 16