

Project Report DAMG 7275
Twitter: Summing up the Pulse of the Internet

Team 13:

Niraj Sai Prasad - NUID 001006514
Sindhu Swaroop - NUID 001006558
Vatsal Doshi - NUID 002776613
Avani Kala - NUID 002772623

Project Proposal

Topic: Twitter: Summing up the Pulse of the Internet

Data Model: Document (NoSQL)

Target Platform: Azure SQL Multi-Model

Objective/Scope:

- Scrap real-time tweets, Twitter user details, URLs and images off of Twitter
- Compare statistics across tweets and find the most trending type of tweets for a particular period of time
- Find the top 10 users who have the most tweets and look for any correlation between the number of tweets and their followers and following
- Perform location-based analysis on trending hashtags across regions (group by location)
- Perform analysis on the top 10 most followed Twitter users to understand what gender, age group and nationalities they most appeal to

Visualizations Tool: Tableau/ Power BI

Implementation

1) A brief description of the implementation process

The aim of our project is to use ETL tools to *extract* data from twitter, *transform* the data and *load* it onto our Azure database.

1. First, we wrote a Python script that includes API calls to Twitter to extract tweet data and some tweet details like twitter user, location, hashtags, tweet text and follower count. The script also performs some transformation on the data like data cleaning.
2. Next we created our SQL server, source blob, destination database, and batch pools. For the pools, we used a standard A2 windows VM instance with two nodes and two CPUs.
3. We then incorporated the python script in a batch job (“*runpythonscript*”) which is essentially the start of our data pipeline. This batch job runs the python script, extracts data from Twitter into data frames, puts the data into CSV files which are then automatically uploaded into Azure blobs (our data source).
4. Upon successful loading of data into the source, the second step in our pipeline is multiple jobs to copy the data from source (blob storage) to destination (database) using column mapping in Azure.
5. For the document data model, we linked a Cosmos DB resource instance to the data factory as a destination for the hashtag data.
6. For the graph data model, we created Nodes and Edges files to be pushed to our SQL destination database.

- As a third step in our pipeline, we also added a job to delete all CSV files from the blob once the data is copied into the destination.

We then plan to perform analysis and visualizations on this data using PowerBI.

2) Screenshots which capture key steps in implementing the database and other architecture components

Step 1: We created our own SQL server with the name `sqlserversindhu.database.windows.net`

The screenshot shows the Microsoft Azure portal's 'SQL servers' blade. A search bar at the top contains 'Subscription equals Azure for Students'. Below it, a table lists one record: 'sqlserversindhu' (Status: Available, Resource group: rgsindhu, Location: East US, Subscription: Azure for Students).

Server Properties:

The screenshot shows the 'sqlserversindhu' SQL server properties page. Under the 'Overview' section, it shows the server is available and part of the 'rgsindhu' resource group. It also lists the server admin (CloudSAcb6f9031), networking settings, Active Directory admin (saiprasad.n@northeastern.edu), and server name (sqlserversindhu.database.windows.net). The 'JSON View' button is visible in the top right.

Step 2: We created the source azure storage blob container as `tweetblob` in the SQL server.

The screenshot shows the 'Containers' blade for the 'storageaccountsindhu' storage account. The 'tweetblob' container is selected. It displays blob details like 'edge_df.csv', 'hashtagsdata.json', 'nodes.csv', etc., along with their modified times, access tiers, and lease states. The table has columns: Name, Modified, Access tier, Archive status, Blob type, Size, Lease state, and three-dot ellipsis.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
edge_df.csv	4/6/2023, 7:43:44 PM	Hot (inferred)		Block blob	669 B	Available
hashtagsdata.json	4/6/2023, 7:43:43 PM	Hot (inferred)		Block blob	9.25 kB	Available
nodes.csv	4/6/2023, 7:43:44 PM	Hot (inferred)		Block blob	7.06 kB	Available
referenced_tweet_table.csv	4/6/2023, 7:43:41 PM	Hot (inferred)		Block blob	3.46 kB	Available
tweet_url.csv	4/6/2023, 7:43:38 PM	Hot (inferred)		Block blob	7.63 kB	Available
twitter_user.csv	4/6/2023, 7:43:38 PM	Hot (inferred)		Block blob	20.8 kB	Available
twitterheader.csv	4/6/2023, 7:43:38 PM	Hot (inferred)		Block blob	20.31 kB	Available

Step 3: We created the destination azure SQL database as *db_destination_sindhu* in the SQL server.

The screenshot shows the Microsoft Azure portal interface for a SQL database named "db_destination_sindhu". The top navigation bar includes "Microsoft Azure", a search bar, and user information. Below the header, the database name is displayed with a "SQL database" icon. A breadcrumb trail shows "Home > db_destination_sindhu (sqlserversindhu/db_destination_sindhu)". The main content area has a "Search" bar and several actions: "Copy", "Restore", "Export", "Set server firewall", "Delete", "Connect with...", and "Feedback". A "JSON View" link is also present. On the left, a sidebar lists "Overview", "Activity log", "Tags", "Diagnose and solve problems", "Getting started", "Query editor (preview)", and "Settings". The "Overview" section displays resource group ("rgsindhu"), status ("Online"), location ("East US"), subscription ("Azure for Students"), and other details like "Server name" (sqlserversindhu.database.windows.net), "Elastic pool" (No elastic pool), and "Pricing tier" (General Purpose: Gen5, 2 vCores). It also shows the "Earliest restore point" (2023-04-05 16:47 UTC) and a "Tags (edit)" button.

Step 4: Next we created a Data Factory to build a pipeline to move the data from source to destination using ETL operations. We set the source of the data factory as *tweetblob* and destination as *db_destination_sindhu* and *sindhucosmosdb*.

The screenshot shows the Microsoft Azure portal interface for a Data Factory named "sindhudamg7275". The top navigation bar includes "Microsoft Azure | sindhudamg7275", a search bar, and user information. The main content area features a "Data factory" title and a 3D diagram illustrating data flow between various cloud services. Below the diagram, four buttons are shown: "Ingest" (Copy data at scale once or on a schedule.), "Orchestrate" (Code-free data pipelines.), "Transform data" (Transform your data using data flows.), and "Configure SSIS" (Manage & run your SSIS packages in the cloud.). A "Recent resources" section lists two items: "CopyPipeline_tweettest" (Pipeline, Last opened by you Yesterday at 9:09 PM) and "SourceDataset_mr" (Dataset, Last opened by you Yesterday at 9:09 PM).

Step 5: We then created a batch account and a batch job *runpythonbatch* to run the python script that extracts twitter data via API.

The screenshot shows the Microsoft Azure portal interface for "Batch accounts". The top navigation bar includes "Microsoft Azure", a search bar, and user information. The main content area shows a table of "Batch accounts" with one record: "runpythonbatch". The table columns are "Name", "Status", "Resource group", "Location", and "Subscription". The "runpythonbatch" row has "Online" status, "rgsindhu" resource group, "East US" location, and "Azure for Students" subscription. The table includes sorting and filtering options at the bottom.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (saiprasad.n@northeastern... NORTHEASTERN UNIVERSITY). Below the navigation bar, the URL is Home > Batch accounts > runpythonbatch.

The main content area displays the 'Batch accounts' blade for the 'runpythonbatch' account. On the left is a navigation menu with options like '+ Create', 'Manage view', 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Quick start', 'Properties', and 'Contact'. The 'Overview' section is selected.

runpythonbatch Batch account

Overview

Essentials

Setting	Value
Resource group (move)	rgsindhu
Status	Online
Location	East US
Subscription (move)	Azure for Students
Subscription ID	0d9d74f6-3b28-4341-a642-0a1d9fa08cc5
Tags (edit)	...
Account endpoint	runpythonbatch.eastus.batch.azure.com
Node management endpoint	e026dbc2-dc25-493f-9dec-a58336cab306.eastus.service.batch.azure.com
Identity type	None
Public network access	All networks
Pool allocation mode	Batch service

JSON View

Step 6: We configured a pool *runpythonpool* with two nodes and two CPUs each of which are small virtual machines that execute the jobs depending on which machine has the bandwidth.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (saiprasad.n@northeastern... NORTHEASTERN UNIVERSITY). Below the navigation bar, the URL is Home > Batch accounts > runpythonbatch > Pools.

The main content area displays the 'Pools' blade for the 'runpythonbatch' account. On the left is a navigation menu with options like '+ Create', 'Manage view', 'Features', 'Applications', 'Pools', 'Jobs', 'Job schedules', and 'Certificates'. The 'Pools' option is selected.

runpythonbatch | Pools

Pool ID	Dedicated nodes	Spot/low-priority n...	Current vCPUs	VM size	Allocation state
runpythonpool	2	0	2	STANDARD_A1_V2	Steady
runpythonpoolscript	0	0	0	STANDARD_A1_V2	Steady

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (saiprasad.n@northeastern... NORTHEASTERN UNIVERSITY). Below the navigation bar, the URL is Home > Batch accounts > runpythonbatch > Pools > runpythonpool > Nodes.

The main content area displays the 'Nodes' blade for the 'runpythonpool' pool. On the left is a navigation menu with options like 'Overview', 'Activity log', 'General', 'Properties', 'Nodes', and 'Settings'. The 'Nodes' option is selected.

runpythonpool | Nodes

Name	State	Allocation time	...
tvm..._2b6b30c84d9b5dcf3df7c7a9585de1de196b314637b93c1d12a85...	Idle	Thursday, April 6, 2023 at 13:34:03	...
tvm..._98ad45ffb08acbfb21edda8f1c18ce7ed5b4404aa2934e6d253ceb...	Idle	Thursday, April 6, 2023 at 13:34:03	...

Step 7: We created a NoSQL destination database in Azure Cosmos DB as *sindhucosmosdb*.

The screenshot shows the Microsoft Azure portal search results for the query "sindhucosmosdb". The results list one record: "sindhucosmosdb" from the "Azure for Students" subscription, located in the West US region. The status is "Online".

The screenshot shows the Azure Cosmos DB account details for "sindhucosmosdb". The account is a Free Tier account with 1000 RU/s and 25 GB of storage. It is part of the "rgsindh" resource group and is located in the West US region. The URI is <https://sindhucosmosdb.documents.azure.com:443/>. The throughput is provisioned at 1000 RU/s.

The screenshot shows the Data Explorer page for the "sindhucosmosdb" database. The "DATA" section is selected, showing a single container named "twitter_hashtags". The note indicates that notebooks are currently not available.

Step 8: The services linked to the Data Factory are as shown below:

The screenshot shows the 'Linked services' section in the Microsoft Azure Data Factory interface. It lists four services with their types and related counts:

Name	Type	Related	Annotations
AzureBatch1	Azure Batch	1	
AzureBlobStorage	Azure Blob Storage	9	
AzureSqlDatabase1	Azure SQL Database	6	
CosmosDbNoSql1	Azure Cosmos DB for NoSQL	1	

- Azurebatch1 refers to the batch job *runpythonbatch*
- AzureBlobStorage refers to the source storage blob *tweetblob*
- AzureSqlDatabase1 refers to the destination database *db_destination_sindhu*
- CosmosdbNosql1 refers to the NoSQL database *sindhucosmosdb*

Step 9: After running the batch job in the data pipeline, the *tweetblob* looks like this:

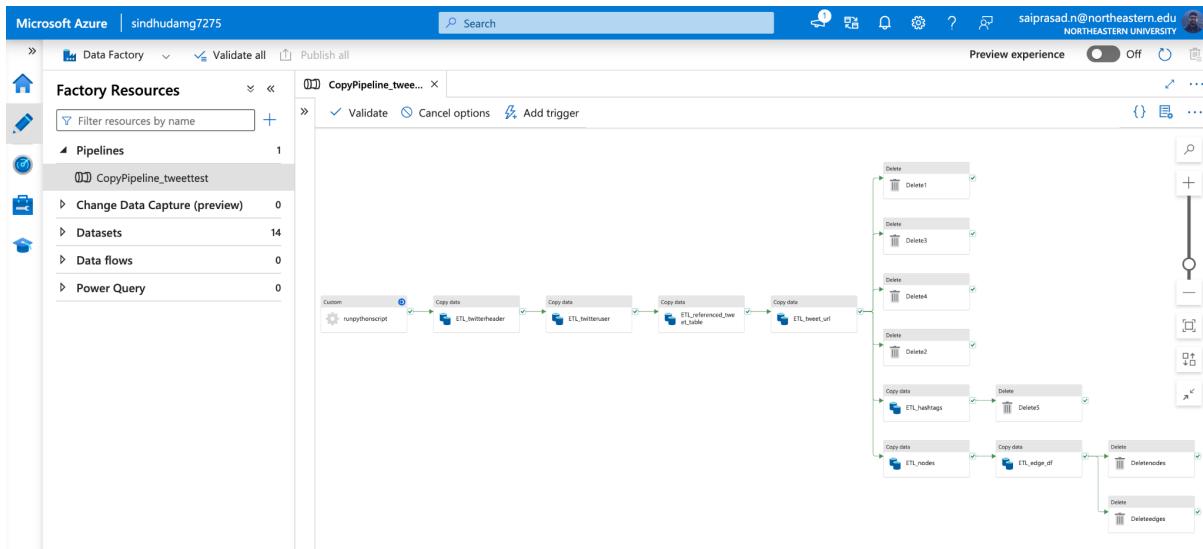
The screenshot shows the 'tweetblob' container in the Microsoft Azure Storage interface. It displays two blobs: 'logs' and 'test1.py'.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
logs					-	...
test1.py	4/6/2023, 2:35:31 PM	Hot (Inferred)		Block blob	7.99 KiB	Available

The screenshot shows the 'output' folder within the 'tweetblob' container in the Microsoft Azure Storage interface. It displays five blobs:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[...]					-	...
hashtagsdata.json	4/6/2023, 6:03:37 PM	Hot (Inferred)		Block blob	12.67 KiB	Available
referenced_tweet_table.csv	4/6/2023, 6:03:34 PM	Hot (Inferred)		Block blob	3.96 KiB	Available
tweet_url.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob	7.63 KiB	Available
twitter_user.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob	19.13 KiB	Available
twitterheader.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob	19.07 KiB	Available

Step 10: We created a data pipeline *CopyPipeline_tweetest* with 3 major steps to extract, copy data to destination and delete CSV files from source all in sequence.



Name	Type	Run start	Duration	Status	Integration runtime	Run ID
Deleteedges	Delete	2023-04-06T23:52:07.248723Z	00:00:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	d1d453d3-198c-47a6-9ea6-8cbdd822f
Deletenodes	Delete	2023-04-06T23:52:07.779979Z	00:00:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	d9562663-398f-434a-8c6c-141c3189
Delete5	Delete	2023-04-06T23:51:57.358548Z	00:00:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	c408b506-4f11-4ddd-8983-5b2bfb2ce
ETL_edge_df	Copy data	2023-04-06T23:51:54.499326Z	00:00:11	Succeeded	AutoResolveIntegrationRuntime (Eas)	4185b80d-ef5-4e74-b1b-23c57f04
Delete3	Delete	2023-04-06T23:51:41.6747614Z	00:00:08	Succeeded	AutoResolveIntegrationRuntime (Eas)	16f56df7-e84a-4735-a012-a570338c9
Delete1	Delete	2023-04-06T23:51:41.6747614Z	00:00:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	5f84759c-6358-4da3-96fd-5b12646a7
Delete4	Delete	2023-04-06T23:51:41.6747614Z	00:00:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	6e23058e-629-4454-911c-5d53580f6
ETL_hashtags	Copy data	2023-04-06T23:51:41.6747614Z	00:00:14	Succeeded	AutoResolveIntegrationRuntime (Eas)	57075ea-0bb5-437d-9f57-8b762a16
ETL_nodes	Copy data	2023-04-06T23:51:41.6591391Z	00:00:11	Succeeded	AutoResolveIntegrationRuntime (Eas)	73828006-4a2f-429b-be29-ad70a6088
Delete2	Delete	2023-04-06T23:51:41.6591391Z	00:00:14	Succeeded	AutoResolveIntegrationRuntime (Eas)	fe1a11e7-ba00-4eee-af2e-86e16abf
ETL_tweet_urrl	Copy data	2023-04-06T23:51:27.4388087Z	00:00:12	Succeeded	AutoResolveIntegrationRuntime (Eas)	103eaad-69f1-432b-bc5c-8329770d
ETL_referenced_tweet_table	Copy data	2023-04-06T23:50:23.2884395Z	00:01:03	Succeeded	AutoResolveIntegrationRuntime (Eas)	da9b7664-3e34-45cb-b99e-db7794d1
ETL_twitterson	Copy data	2023-04-06T23:49:28.5020233Z	00:00:54	Succeeded	AutoResolveIntegrationRuntime (Eas)	0d09a5d6-af5-4eb6-9e1d-923071a-
ETL_twitterheader	Copy data	2023-04-06T23:48:48.2893549Z	00:00:39	Succeeded	AutoResolveIntegrationRuntime (Eas)	5c707acc-f93c-d43e-af1-23cd68422c
runpythonscript	Custom	2023-04-06T23:48:12.784339Z	00:00:35	Succeeded	AutoResolveIntegrationRuntime (Eas)	5bd87875-e620-4b72-af6d1-b3b0e14f

Step 11: We performed destination data validation to ensure data flow was smooth and we can now perform analysis.

RDBMS and Graph Tables:

The screenshot shows a database interface with three tables:

- Messages** (Top Table):

	tweet_id	twitter_handle	parent_tweet_id	tweet	tweet_time	location
1	1644124149864972289	226994736	None	RT @UtdDistrict: Out of #...	2023-04-06 23:45:18+00:00	London, UK
2	1644124143787319300	1582299311920058374	None	RT @fortun3: This Liverp...	2023-04-06 23:45:16+00:00	London, England
- Twitter_Users** (Middle Table):

	twitter_handle	user_name	profile_img_url	description	follower_count	following_count	joined_on
1	944228945515433990	TEE. ™	http://pbs.twimg.com/prof...	H A T E I S H E A V Y L E...	701.0	1976.0	2017-12-22
2	1467119041567461377	Gunnertalk	http://pbs.twimg.com/prof...	Football is life, life is...	22.0	52.0	2021-12-04
- Edge_DF** (Bottom Table):

	_from	_to	type
1	nodes/FAYouthCup	nodes/Accra, Ghana	used_in...
2	nodes/FAYouthCup	nodes/Accra, Ghana	used_in...

Document Data on Cosmos:

The screenshot shows the Azure Cosmos DB Data Explorer interface for the 'sindhucosmosdb' account. The 'DATA' section is selected, and the 'twitter_hashtags' collection is displayed. The 'Items' table shows the following data:

id	/id
4d3f88f9...	4d3f88f9...
039a305...	039a305...
a302c6f6...	a302c6f6...
ae1c8e...	ae1c8e...
1a36f51f...	1a36f51f...
dd91e3bf...	dd91e3bf...

The right pane displays the JSON documents:

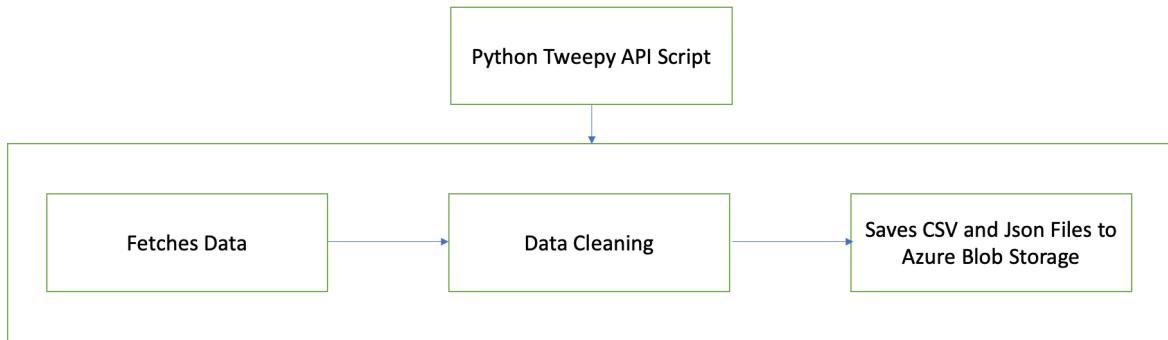
```

1  [
2    {
3      "username": "John kennedy 🇮🇳Hasaka's 🔥Finest🔥",
4      "tweet_location": "Uganda",
5      "tweet_text": "RT @adamkeys_: Throwback to this outrage"
6    },
7    {
8      "username": "Caroline 🌸",
9      "tweet_location": "Yorkshire, UK",
10     "tweet_text": "RT @Calderdale: We anticipate that roads"
11    },
12    {
13      "username": "HT",
14      "tweet_location": "Houston, TX",
15      "tweet_text": "@BeardedBeauner We're a QB needy team in"
16    },
17    {
18      "username": "Muyanja Ahmed",
19      "tweet_location": "Kampala, Uganda",
20      "tweet_text": "RT @now_arsenal: Granit Xhaka says Arsenal"
21    }
  ]

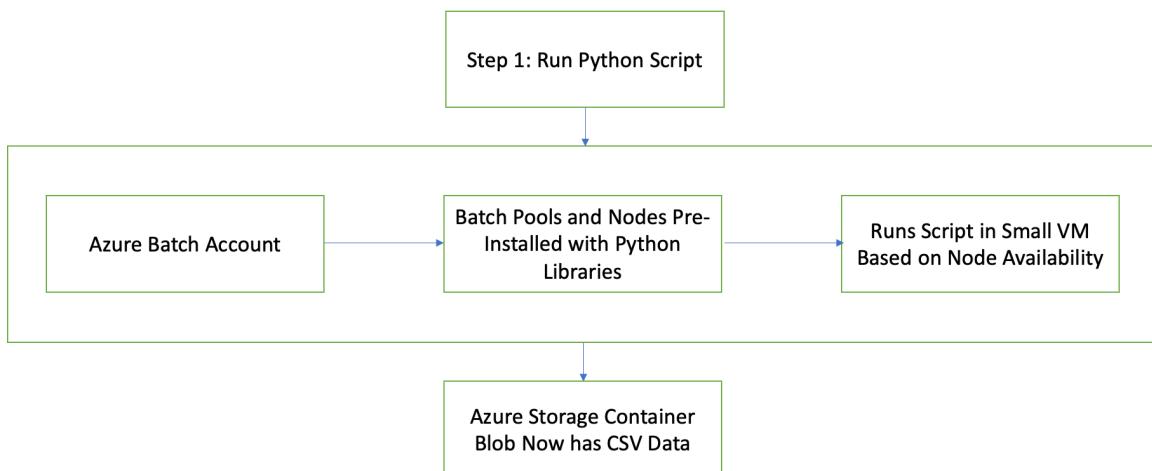
```

Architecture in Detail

Data Source



ADF

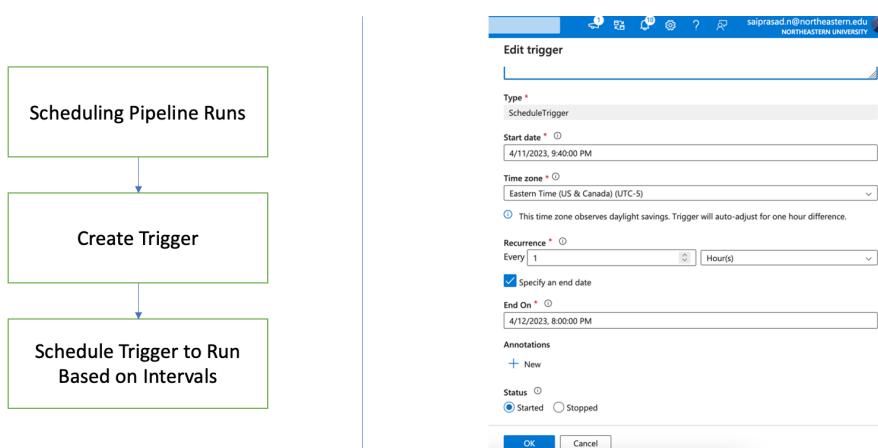


CSV and Json files in Blob Storage

Azure Storage Container - tweetblob / output						
Actions		Name	Modified	Access tier	Archive status	Blob type
<input type="checkbox"/>		[..]				---
<input type="checkbox"/>		hashtagsdata.json	4/6/2023, 6:03:37 PM	Hot (Inferred)		Block blob 12.67 KiB Available ---
<input type="checkbox"/>		referenced_tweet_table.csv	4/6/2023, 6:03:34 PM	Hot (Inferred)		Block blob 3.96 KiB Available ---
<input type="checkbox"/>		tweet_url.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob 7.63 KiB Available ---
<input type="checkbox"/>		twitter_user.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob 19.13 KiB Available ---
<input type="checkbox"/>		twitterheader.csv	4/6/2023, 6:03:31 PM	Hot (Inferred)		Block blob 19.07 KiB Available ---



Trigger



Data Destination

SQL Database:

- Twitter Header
- Twitter User
- Twitter URL
- Reference Tweet Table
- Hashtag Nodes
- Hashtag Edges

```

4 select top 2 * from twitterheaders;
5 select top 2 * from twitter_users;
6 select top 2 * from edge_df;

Results Messages
| tweet_id | twitter_handle | parent_tweet_id | tweet | tweet_time | location |
| 1644124149864972289 | 226994736 | None | RT @tdDistrict: Out of #... | 2023-04-06 23:45:18+00:00 | London, UK |
| 1644124143787319300 | 1582299311920058374 | None | RT @fortune3: This Liverp... | 2023-04-06 23:45:16+00:00 | London, England |

| twitter_handle | user_name | profile_img_url | description | follower_count | following_count | joined_on |
| 9442289465515433990 | TEE. | http://pbs.twimg.com/prof... | H A T E I S H E A V Y L E... | 701.0 | 1976.0 | 2017-12-22 |
| 1467119841567461377 | Gunnertalk | http://pbs.twimg.com/prof... | Football is life, life is... | 22.0 | 52.0 | 2021-12-04 |

| _from | _to | type |
| nodes/FAYouthCup | nodes/Accra, Ghana | used_inv... |
| nodes/FAYouthCup | nodes/Accra, Ghana | used_inv... |


```

Cosmos Database:

- Hashtags Data Json Items

```

SELECT * FROM c
id /id
1 4d3f8bf9...
2 039a305...
3 a302c6f6...
4 aec1c8e...
5 1a36f51f...
6 dd91e3bf...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...

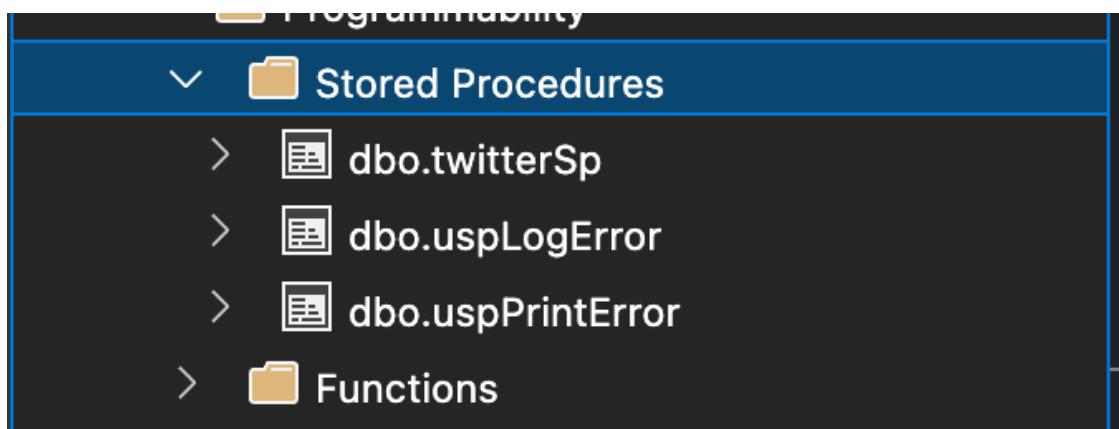

"afc": [
  {
    "username": "John kennedy ➤Masaka's ⚪Finest⚡️",
    "tweet_location": "Uganda",
    "tweet_text": "RT @adamkeys_: Throwback to this outrage"
  },
  {
    "username": "Caroline 🇬🇧",
    "tweet_location": "Yorkshire, UK",
    "tweet_text": "RT @Calderdale: We anticipate that roads"
  },
  {
    "username": "HT",
    "tweet_location": "Houston, TX",
    "tweet_text": "@BeardedBeauner We're a QB needy team in"
  },
  {
    "username": "Mayanja Ahmed",
    "tweet_location": "Kampala, Uganda",
    "tweet_text": "RT @www.arsenalT_Granit_Xhaka_says_Arsenal"
  }
]

```

Work with the Database

Functions Performed:

We created a stored procedure to archive data that is already in the database tables



Archive Tables

```
2  
3   select * from sys.tables where name like '%archive%';  
4
```

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date
1	edgeArchive	1458104235	NULL	1	0	U	USER_TABLE	2023-04-11 19:
2	nodeArchive	1474104292	NULL	1	0	U	USER_TABLE	2023-04-11 19:
3	referencedTweetArchive	1426104121	NULL	1	0	U	USER_TABLE	2023-04-11 19:
4	tweetUrlArchive	1394104007	NULL	1	0	U	USER_TABLE	2023-04-11 19:
5	twitterHeaderArchive	1378103950	NULL	1	0	U	USER_TABLE	2023-04-11 18:
6	twitterUserArchive	1410104064	NULL	1	0	U	USER_TABLE	2023-04-11 19:

Count of Data in Archive Tables

```
49  
50   select count(tweet_id) as cnt_records from tweetUrlArchive;
```

Results Messages

	cnt_records
1	276

The first step of the pipeline run ensures that the stored procedure above runs:

This screenshot shows the Microsoft Azure Data Factory pipeline configuration interface. The pipeline is named 'CopyPipeline_tweetdata'. The 'Sink' tab is selected. Under 'Sink dataset', 'DestinationDataset_twitterheader' is chosen. The 'Write behavior' is set to 'Insert'. The 'Bulk insert table lock' option is set to 'No'. The 'Table option' is set to 'Auto create table'. In the 'Pre-copy script' section, the script 'exec twitterSp;' is entered. The 'Write batch timeout' is set to 'e.g. 00:30:00'. The 'Write batch size' is set to a large value. The top navigation bar includes 'Validate all', 'Publish all', and a notification for one update.

We scheduled the ADF pipeline to run every hour so that we can run visualization tools over this.

This screenshot shows the Microsoft Azure Data Factory trigger configuration interface. The 'Triggers' section lists a single trigger named 'trigger_tweet_data' which is scheduled to run. The 'Edit trigger' dialog is open, showing the following details:

- Type:** ScheduleTrigger
- Start date:** 4/11/2023, 9:40:00 PM
- Time zone:** Eastern Time (US & Canada) (UTC-5)
- Recurrence:** Every 1 Hour(s)
- Status:** Started

The left sidebar shows other configuration options like General, Factory settings, and Source control.

Pipeline Runs

Microsoft Azure | sindhudamg7275

Trigger runs

All Schedule Tumbling window Storage events Custom events Refresh Edit columns

Local time : Last 24 hours Trigger name : trigger_tweet_data Status : All Runs : Latest runs Export to CSV

Trigger name ↑	Trigger type	Trigger time ↓	Status ↑	Pipelines	Run	Message	Properties	Run ↓
trigger_tweet_data	Schedule trigger	4/11/2023, 6:05:00 P	Succeeded	1	Original			0858
trigger_tweet_data	Schedule trigger	4/11/2023, 5:05:00 P	Succeeded	1	Original			0858

We also clean up the Azure blob storage after each run so that it does not clutter the container.

CopyPipeline_tweet... X

Activities Validate Debug Trigger (1) { }

Search activities

Move & transform

Synapse

Azure Data Explorer

Azure Function

Batch Service

Databricks

Data Lake Analytics

General

HDInsight

Iteration & conditionals

Machine Learning

Power Query

Delete Delete1

Delete Delete3

General Source Logging settings User properties

Dataset * Open New Preview data Learn

File path type File path in dataset Wildcard file path Prefix List of files

Start time (UTC)

End time (UTC)

Filter by last modified

Recursively

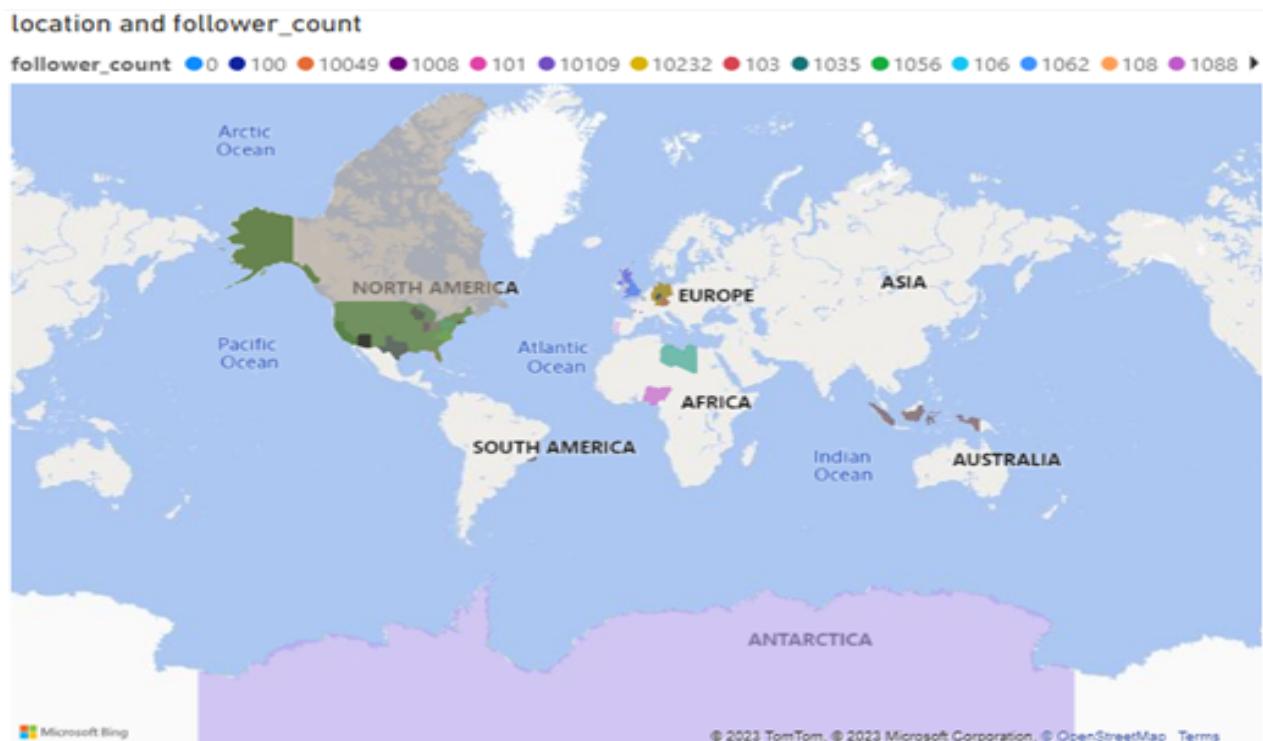
Max concurrent connections

Visualizations

1. Location-based analysis

We have performed a location-based analysis using Power BI, and have added the location to the map visual, and the number of users as the legend. It provides a visual representation of where your users are located and how many users are in each location.

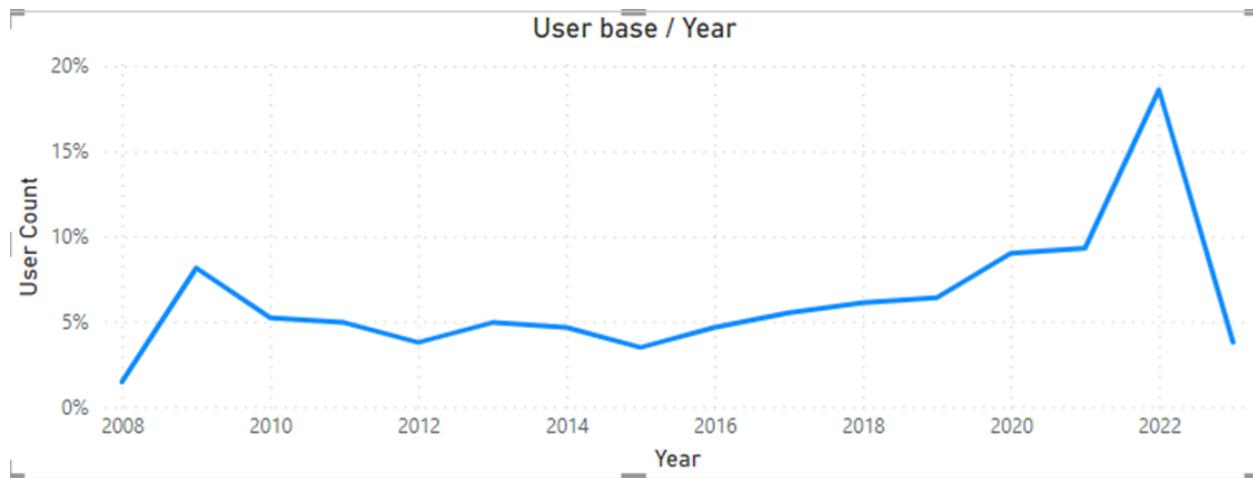
The outcome we got from this analysis is a better understanding of where users are located and how many users are in each location. We are also able to identify patterns or trends in user behavior across different locations, which can help inform marketing and sales strategies. However, the depth of insights we can gain will depend on the complexity of the analysis and the additional layers of data we incorporate into the visualizations.



2. User base on Twitter for each year

We have performed a time-series analysis on the user base of Twitter using Power BI. We have plotted the test date on the x-axis and the percentage growth of users on the y-axis. This type of analysis is useful for understanding how the user base of Twitter has grown or changed over time.

The outcome we can expect from this analysis is a better understanding of the trends and patterns in Twitter's user growth. By analyzing the percentage growth of users over time, we can identify periods of rapid growth or decline and the factors that contributed to these changes. This information can be used to inform business decisions, such as marketing strategies, product development, or investment opportunities. This type of analysis can be a valuable tool for understanding the trajectory of a company or product over time, and making data-driven decisions based on that insight.

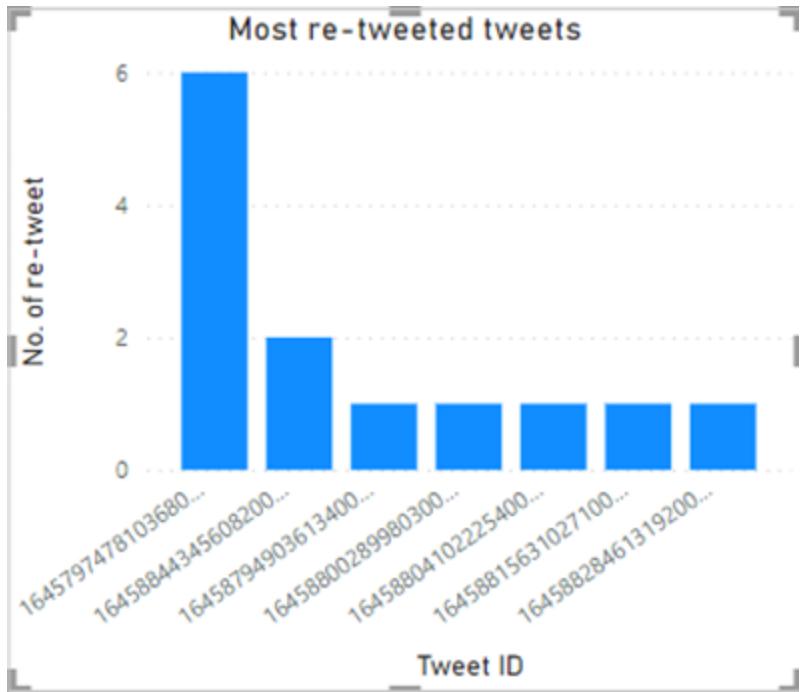


3. Analyze the number of retweets

We have performed an analysis of the most retweeted tweets using Power BI. We have used the parent tweet ID as the x-axis and the count of tweet IDs as the y-axis. This type of analysis can be useful for identifying the most popular or viral tweets on a specific topic or theme.

The outcome you can expect from this analysis is a better understanding of the tweets that have generated the most engagement and reach on Twitter. By analyzing the parent tweet ID and the count of tweet IDs, we can identify the tweets that have been retweeted the most and understand the content or messaging that resonated with users.

This information can be used to inform social media marketing strategies, as it provides insight into the types of content that are most likely to generate engagement and reach on Twitter. Additionally, by visualizing the data in Power BI, we can create informative and visually appealing presentations to share with stakeholders.



4. Graph data model for most popular hashtag

We have performed an analysis on the correlation matrix for the most popular hashtag using Power BI. We have used the network navigator visual and have added "_to" as the Source Node and "_from" as the Target Node. This type of analysis can be useful for identifying patterns and relationships between the most popular hashtag in a region.

The outcome you can expect from this analysis is a better understanding of the network of relationships between the most popular hashtag on the platform. By visualizing the correlation matrix in Power BI, we can identify which hashtags are most closely connected to each other, which users are most influential.

This information can be used to inform social media marketing strategies, as it provides insight into the influencers and key players in the platform's ecosystem. By gaining insight into the hashtags that have the most impact on the platform, we can refine our marketing strategies and better engage with our target audience.

Hashtags used at Location



5. Top 10 users with highest following

We have analyzed the top 10 most followed users on Twitter using Power BI. The outcome we can expect from this analysis is a better understanding of the distribution of followers among the top 10 most followed users on Twitter. By visualizing the data in a Treemap, we can easily identify which users have the most followers and compare their popularity to other users.

This information can be used to inform social media marketing strategies, as it provides insight into the most popular users on the platform and the types of content that are resonating with Twitter users. By gaining insight into which users have the most followers, companies can refine their social media marketing strategies and better engage with the targeted audience.

