

**Team Members:**

- 1. Niraj Sai Prasad - NUID 001006514**
- 2. Sindhu Swaroop - NUID 001006558**
- 3. Vatsal Doshi - NUID 002776613**
- 4. Avani Kala - NUID 002772623**

**Project Report for Topic:**

*Twitter: Summing up the Pulse of the Internet*

## P1: 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
- Perform sentiment analysis on tweets to find the general pulse of Twitterati
- 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

## P3: 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.
7. 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*

Name	Status	Resource group	Location	Subscription
sqlserversindhu	Available	rgsindhu	East US	Azure for Students

## Server Properties:

**sqlserversindhu** SQL server

**Overview**

**Essentials**

- Resource group ([move](#)) **rgsindhu**
- Status **Available**
- Location **East US**
- Subscription ([move](#)) **Azure for Students**
- Tags ([edit](#)) [Click here to add tags](#)

**Notifications (1)** **Features (6)**

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

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.

**db\_destination\_sindhu (sqlserversindhu/db\_destination\_sindhu)** SQL database

**Overview**

**Essentials**

Resource group ( <a href="#">move</a> )	<b>rgsindhu</b>	Server name	: sqlserversindhu.database.windows.net
Status	: Online	Elastic pool	: No elastic pool
Location	: East US	Connection strings	: Show database connection strings
Subscription ( <a href="#">move</a> )	: Azure for Students	Pricing tier	: General Purpose: Gen5, 2 vCores
Subscription ID	: 0d9d74f6-3b28-4341-a642-0a1d9fa08cc5	Earliest restore point	: 2023-04-05 16:47 UTC

**Tags** ([edit](#)) : Click here to add tags

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 Data Factory service interface. At the top, there's a navigation bar with 'Microsoft Azure' and the account name 'sindhudamg7275'. Below the navigation bar is a search bar and several icons for account management. The main area is titled 'Data factory' and has the name 'sindhudamg7275' displayed prominently. A large graphic illustrates the Data Factory process flow. Below the graphic, there are four main sections: '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). Underneath these sections, there's a section titled 'Recent resources' which lists two items: 'CopyPipeline\_tweettest' (Pipeline) and 'SourceDataset\_mnr' (Dataset), both last opened 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 first screenshot shows the 'Batch accounts' list in the Microsoft Azure portal. It displays a single record: 'runpythonbatch'. The second screenshot shows the detailed view of the 'runpythonbatch' account, including its 'Overview' tab and the 'Essentials' section. The 'Essentials' section provides details such as Resource group (move), Status (Online), Location (East US), Subscription (move), and Tags (edit).

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.

Microsoft Azure Search resources, services, and docs (G+)

Home > Batch accounts > runpythonbatch

### Batch accounts

Northeastern University (northeastern.onmicrosoft.com)

+ Create Manage view ...

Filter for any field... Name ↑

**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

Microsoft Azure Search resources, services, and docs (G+)

Home > Batch accounts > runpythonbatch | Pools > runpythonpool

### runpythonpool | Nodes

Pool

Overview Activity log General Properties Nodes Settings

Search State == all Add filter

Search for nodes by state Pagination effort limit Actual: 1

Name	State	Allocation time
tvm..._2b6b30c84d9b5dcf3df7c7a9585de1de196b314637b93c1d12a85...	Idle	Thursday, April 6, 2023 at 13:34:03
tvm..._98ad45ffb0acbfb21edda8ff1c18ce7ed5b4404aa2934e6d253ceb...	Idle	Thursday, April 6, 2023 at 13:34:03

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

Microsoft Azure Search resources, services, and docs (G+)

Home >

### Azure Cosmos DB

Northeastern University

+ Create Restore Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals Azure for Students Type equals all Resource group equals all Location equals all Add filter

Showing 1 to 1 of 1 records.

Name	Status	Subscription	Write region	Read Region
sindhucosmosdb	Online	Azure for Students	West US	West US

**sindhucosmosdb** Azure Cosmos DB account

**Essentials**

- Status: Online
- Resource group: (move) rgsindhu
- Subscription: (move) Azure for Students
- Subscription ID: 0d9d74f6-3b28-4341-a642-0a1d9fa08cc5
- Total throughput limit: 1000 RU/s
- Read Locations: West US
- Write Locations: West US
- URI: https://sindhucosmosdb.documents.azure.com:443/
- Free Tier Discount: Opted In
- Capacity mode: Provisioned throughput

**Containers**

ID	Database	Throughput (RU/s)
hashtags	twitter_hashtags	1000 (Max Throughput) (Shared)

**sindhucosmosdb | Data Explorer** Azure Cosmos DB account

**NOSQL API**

- DATA**
- twitter\_hashtags
- NOTEBOOKS**

Notebooks is currently not available. We are working on it.

**Welcome to Azure Cosmos DB**

Globally distributed, multi-model database service for any scale

Step 7: The services linked to the Data Factory are as shown below:

**Linked services**

Linked service defines the connection information to a data store or compute. [Learn more](#)

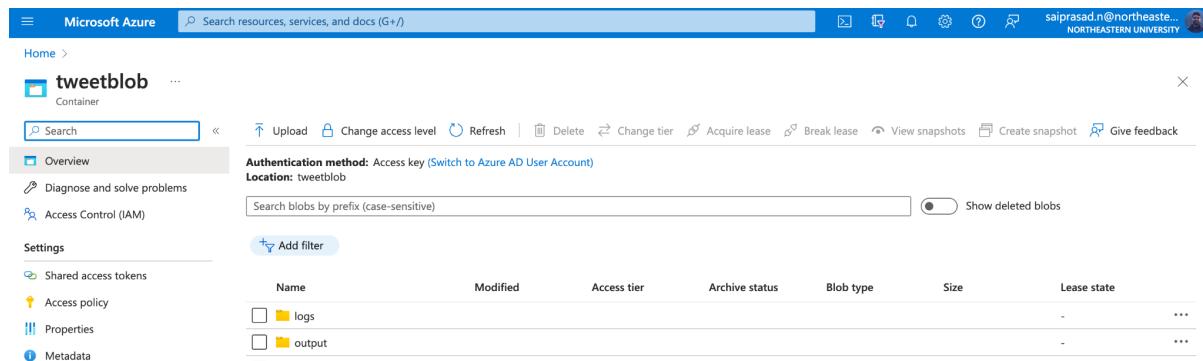
**New**

**Showing 1 - 4 of 4 items**

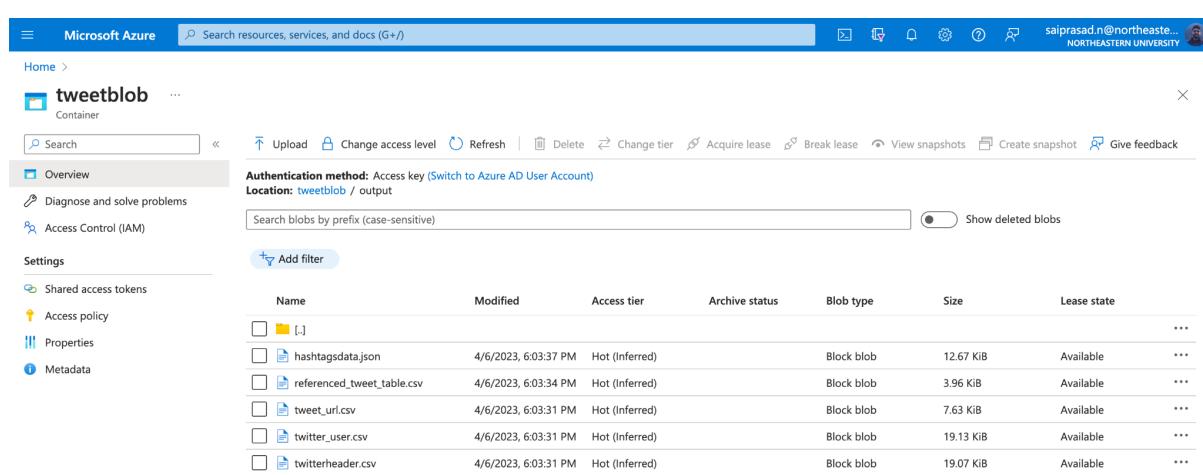
Name	Type	Related	Annotations
AzureBatch1	Azure Batch	1	
AzureBlobStorage	Azure Blob Storage	9	
AzureSqlDatabase1	Azure SQL Database	6	
CosmosDbNoSql1	Azure Cosmos DB for NoSQL	1	

- 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 8: After running the batch job in the data pipeline, the *tweetblob* looks like this:



The screenshot shows the Microsoft Azure Storage Blob Container Overview page for a container named "tweetblob". The left sidebar includes links for Home, Overview, Diagnose and solve problems, Access Control (IAM), Settings, Shared access tokens, Access policy, Properties, and Metadata. The main content area displays a table of blobs. The table has columns for Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. The blobs listed are "logs" (Modified 4/6/2023, 2:35:31 PM, Hot (Inferred), Block blob, 7.99 KiB, Available), "output" (Modified 4/6/2023, 2:35:31 PM, Hot (Inferred), Block blob, 7.99 KiB, Available), and "test1.py" (Modified 4/6/2023, 2:35:31 PM, Hot (Inferred), Block blob, 7.99 KiB, Available). A search bar at the top allows filtering by prefix.

The screenshot shows the Microsoft Azure Storage Blob Container Overview page for the same "tweetblob" container. The left sidebar is identical. The main content area displays a table of blobs. The blobs listed are "logs" (Modified 4/6/2023, 6:03:37 PM, Hot (Inferred), Block blob, 12.67 KiB, Available), "output" (Modified 4/6/2023, 6:03:34 PM, Hot (Inferred), Block blob, 3.96 KiB, Available), "tweet\_url.csv" (Modified 4/6/2023, 6:03:31 PM, Hot (Inferred), Block blob, 7.63 KiB, Available), "twitter\_user.csv" (Modified 4/6/2023, 6:03:31 PM, Hot (Inferred), Block blob, 19.13 KiB, Available), and "twitterheader.csv" (Modified 4/6/2023, 6:03:31 PM, Hot (Inferred), Block blob, 19.07 KiB, Available). A search bar at the top allows filtering by prefix.

Step 9: We created a data pipeline *CopyPipeline\_tweettest* with 3 major steps to extract, copy data to destination and delete CSV files from source all in sequence.

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar lists 'Factory Resources' including 'Pipelines' (1), 'Change Data Capture (preview)' (0), 'Datasets' (14), 'Data flows' (0), and 'Power Query' (0). The main area displays the 'CopyPipeline\_tweetest' pipeline. The pipeline consists of several stages: a 'Custom' activity followed by four 'Copy data' activities and three 'Delete' activities. The 'Copy data' activities are connected sequentially, with their outputs serving as inputs for the subsequent 'Delete' activities. The pipeline is currently in 'Validate' mode.

The screenshot shows the Microsoft Azure Data Factory interface. The top navigation bar includes 'Microsoft Azure', 'Data Factory', 'sindhudamg7275', a search bar, and account information for 'saiprasad.n@northeastern.edu' from 'NORTHEASTERN UNIVERSITY'. A 'Preview experience' toggle is set to 'Off'. The main area displays a pipeline run ID: 'e29b9cad-d7ed-4f19-899d-b34da96e0685'. The pipeline run details are shown in a table with columns: Name, Type, Run start, Duration, Status, Integration runtime, and Run ID. The table lists various steps: Deletededges (Delete), Deletednodes (Delete), Deletes (Delete), ETL\_edge\_df (Copy data), Delete3 (Delete), Delete1 (Delete), Delete4 (Delete), ETL\_hashtags (Copy data), ETL\_nodes (Copy data), Delete2 (Delete), ETL\_tweet\_url (Copy data), ETL\_referenced\_tweet\_table (Copy data), ETL\_twittereruse (Copy data), and runpythonscript (Custom). All steps show a status of 'Succeeded' and an integration runtime of 'AutoResolveIntegrationRuntime (Eas)'. The run ID column contains unique identifiers for each step.

Name	Type	Run start	Duration	Status	Integration runtime	Run ID
Deletededges	Delete	2023-04-06T23:52:07.2487236Z	00:00:03	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	d4d453d3-19bc-47a6-9ea6-8cb2d221
Deletednodes	Delete	2023-04-06T23:52:07.2799779Z	00:00:03	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	c46d2663-398f-434a-8656-c141c3389
Deletes	Delete	2023-04-06T23:51:57.3585458Z	00:00:03	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	4180b5b6-f411-4ddc-98b3-562b2fce
ETL_edge_df	Copy data	2023-04-06T23:51:54.4693264Z	00:00:11	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	4185bb0d-e4f7-4e7a-b1-23757f04
Delete3	Delete	2023-04-06T23:51:41.6747614Z	00:00:08	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	165fd57d-ef4a-4735-a012-a57033c9
Delete1	Delete	2023-04-06T23:51:41.6747614Z	00:00:08	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	584759c-6358-4da3-9ef0-5b126e4a7
Delete4	Delete	2023-04-06T23:51:41.6747614Z	00:00:03	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	d623058e-62a9-4454-931c-5d53580f
ETL_hashtags	Copy data	2023-04-06T23:51:41.6747614Z	00:00:14	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	5707c5ea-b055-437d-9575-8b762a19
ETL_nodes	Copy data	2023-04-06T23:51:41.6591391Z	00:00:11	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	7382b006-442f-4729-be29-ad70a088
Delete2	Delete	2023-04-06T23:51:41.6591391Z	00:00:14	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	f1e111ef-b400-4eeb-acbc-86e16ab0fc
ETL_tweet_url	Copy data	2023-04-06T23:51:27.4388087Z	00:00:12	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	103eaadd-d491-432b-bc3c-832977d0
ETL_referenced_tweet_table	Copy data	2023-04-06T23:50:28.848495Z	00:01:03	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	da9b7664-3e34-452b-9bfe-d7794d
ETL_twittereruse	Copy data	2023-04-06T23:49:25.5020332Z	00:00:54	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	0d9a5d6-4cf5-4b1d-9e1d-9263071a
ETL_twitterheader	Copy data	2023-04-06T23:48:48.3893549Z	00:00:39	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	5c707cc-f93c-4d3e-af1-23cd684a22
runpythonscript	Custom	2023-04-06T23:48:12.7843392Z	00:00:35	<span>✓ Succeeded</span>	AutoResolveIntegrationRuntime (Eas)	5bd87875-e620-4b72-42bf-ab3b6e0141

Step 10: We performed destination data validation to ensure data flow was smooth and we can now perform analysis.

## RDBMS and Graph Tables:

4	select top 2 * from twitterheader;																								
5	select top 2 * from twitter_user;																								
6	select top 2 * from edge_df;																								
<b>Results</b> Messages																									
<table border="1"> <thead> <tr> <th></th> <th>tweet_id</th> <th>twitter_handle</th> <th>parent_tweet_id</th> <th>tweet</th> <th>tweet_time</th> <th>location</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1644124149864972289</td> <td>226994736</td> <td>None</td> <td>RT @UtdDistrict: Out of #...</td> <td>2023-04-06 23:45:18+00:00</td> <td>London, UK</td> </tr> <tr> <td>2</td> <td>1644124143787319300</td> <td>1582299311920058374</td> <td>None</td> <td>RT @fortun3: This Liverp...</td> <td>2023-04-06 23:45:16+00:00</td> <td>London, England</td> </tr> </tbody> </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			
	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																			
<table border="1"> <thead> <tr> <th></th> <th>twitter_handle</th> <th>user_name</th> <th>profile_img_url</th> <th>description</th> <th>follower_count</th> <th>following_count</th> <th>joined_on</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>944228945515433990</td> <td>TFE.I™</td> <td>http://pbs.twimg.com/prof...</td> <td>H A T E I S H E A V Y L E...</td> <td>701.0</td> <td>1976.0</td> <td>2017-12-22</td> </tr> <tr> <td>2</td> <td>1467119041567461377</td> <td>Gunnertalk</td> <td>http://pbs.twimg.com/prof...</td> <td>Football is life, life is...</td> <td>22.0</td> <td>52.0</td> <td>2021-12-04</td> </tr> </tbody> </table>			twitter_handle	user_name	profile_img_url	description	follower_count	following_count	joined_on	1	944228945515433990	TFE.I™	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
	twitter_handle	user_name	profile_img_url	description	follower_count	following_count	joined_on																		
1	944228945515433990	TFE.I™	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																		
<table border="1"> <thead> <tr> <th></th> <th>_from</th> <th>_to</th> <th>type</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>nodes/FAYouthCup</td> <td>nodes/Accra, Ghana</td> <td>used_inrw</td> </tr> <tr> <td>2</td> <td>nodes/FAYouthCup</td> <td>nodes/Accra, Ghana</td> <td>used_inrw</td> </tr> </tbody> </table>			_from	_to	type	1	nodes/FAYouthCup	nodes/Accra, Ghana	used_inrw	2	nodes/FAYouthCup	nodes/Accra, Ghana	used_inrw												
	_from	_to	type																						
1	nodes/FAYouthCup	nodes/Accra, Ghana	used_inrw																						
2	nodes/FAYouthCup	nodes/Accra, Ghana	used_inrw																						

## Document Data on Cosmos:

The screenshot shows the Microsoft Azure Data Explorer interface for the 'sindhucosmosdb' database. The left sidebar includes links for Tags, Diagnose and solve problems, Cost Management, Quick start, Notifications, Data Explorer (which is selected), Settings, Features, Replicate data globally, Default consistency, Backup & Restore, Networking, CORS, Dedicated Gateway, and Keys. The main area has tabs for NOSQL API, DATA, and NOTEBOOKS. The DATA tab is active, showing a table structure for 'twitter\_hashtags'. The table has columns: id, /id, Scale, hashtags, Items, Settings, Stored Procedures, User Defined Functions, and Triggers. Below the table, a query editor window displays the following JSON document:

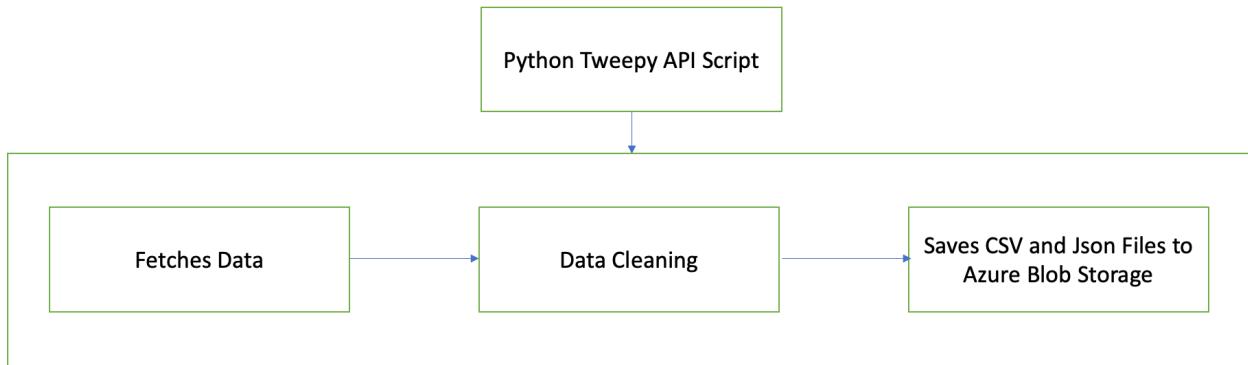
```

1 "afc": [
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 }

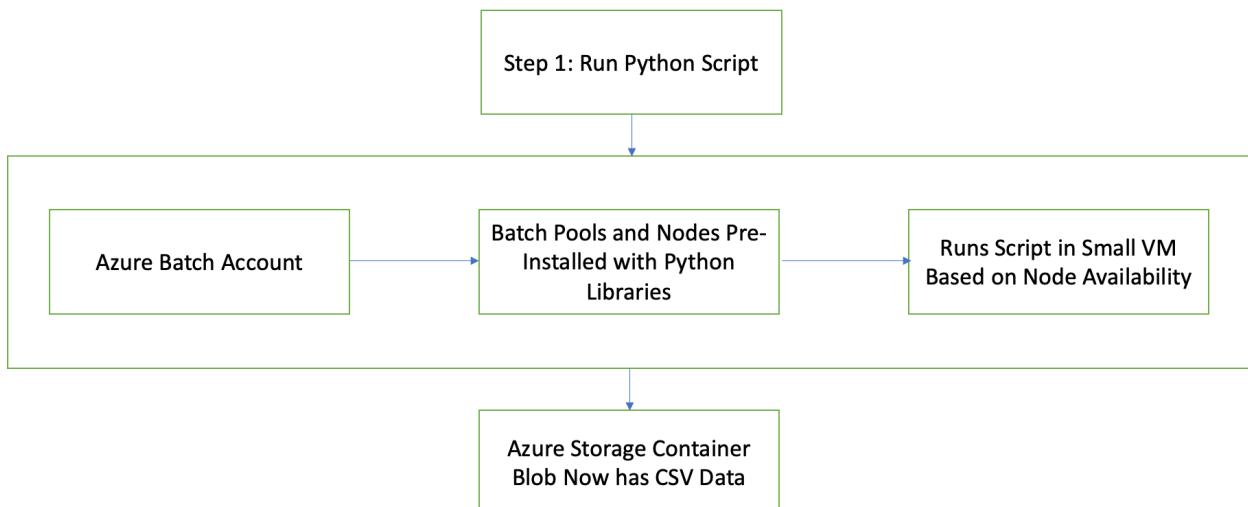
```

## Architecture in Detail:

### Data Source -



### ADF -



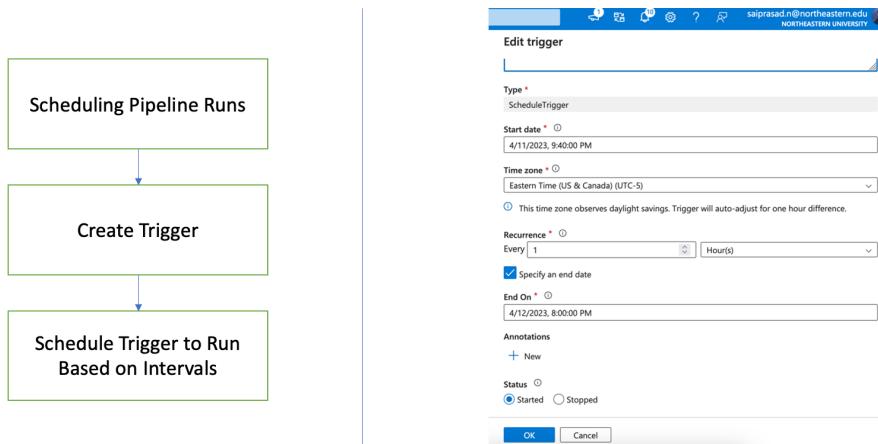
### CSV and Json files in Blob Storage:

A screenshot of the Azure Storage Container blob list interface. The top navigation bar includes options like Upload, Change access level, Refresh, Delete, Change tier, Acquire lease, Break lease, View snapshots, Create snapshot, and Give feedback. Below the bar, it shows the Authentication method as Access key and the Location as tweetblob / output. A search bar at the bottom left allows searching by prefix (case-sensitive). The main area displays a table of blobs with columns: Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. The table lists several files: hashtagsdata.json, referenced\_tweet\_table.csv, tweet\_url.csv, twitter\_user.csv, and twitterheader.csv, all of which are Block blob type and Available in terms of lease state. The table also includes a header row and a footer row with three dots.

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



### Trigger:



## Data Destination -

### SQL Database:

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

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. I	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_inw...
nodes/FAYouthCup	nodes/Accra, Ghana	used_inv...

### Cosmos Database:

- Hashtags Data Json Items

SELECT \* FROM c

id	/id
4d3f8bf9...	4d3f8bf9...
039a305...	039a305...
a302c6f6...	a302c6f6...
ae1c8e...	ae1c8e...
1a36f51f...	1a36f51f...
dd91e3bf...	dd91e3bf...

hashtags - I...

```

1   "afc": [
2     {
3       "username": "John kennedy >>Masaka'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    {
14      "username": "HT",
15      "tweet_location": "Houston, TX",
16      "tweet_text": "@BeardedBeaumer We're a QB needy team in"
17      ...
18    },
19    {
20      "username": "Mayana's Ahmed",
21      "tweet_location": "Kampala, Uganda",
22      "tweet_text": "RT @www.arsenalT_ Granit Xhaka says Arsenal"
23    }
  ]

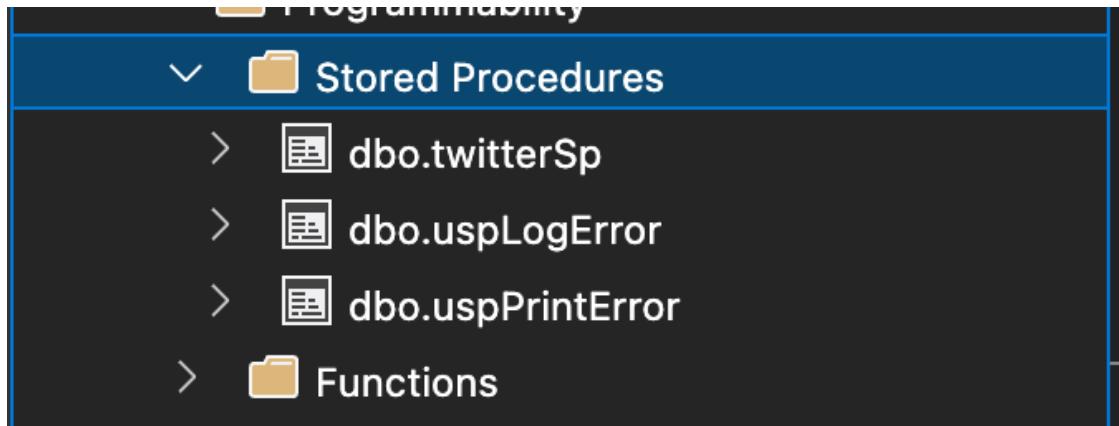
```

## P4: Work with the Database

### Functions Performed:

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

### Stored Procedure



### 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;
```

	cnt_records
1	276

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

The screenshot shows the Microsoft Azure Data Factory interface. A pipeline named 'CopyPipeline\_tweet...' is selected. The 'Sink' tab is active. Under 'Sink dataset', 'DestinationDataset\_twitterheader' is chosen. 'Write behavior' is set to 'Insert'. 'Bulk insert table lock' is set to 'No'. 'Table option' is set to 'Auto create table'. In the 'Pre-copy script' field, the following SQL command is entered:

```
exec twitterSp;
```

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

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar is expanded to show 'Triggers' under 'Author'. A trigger named 'trigger\_tweet\_data' is listed. On the right, the 'Edit trigger' dialog is open. The 'Type' is set to 'ScheduleTrigger'. The 'Start date' is '4/11/2023, 9:40:00 PM'. The 'Time zone' is 'Eastern Time (US & Canada) (UTC-5)'. The 'Recurrence' is set to 'Every 1 Hour(s)'. The 'Status' is 'Started'.

Pipeline Runs:

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a navigation menu with items like Dashboards, Runs, Pipeline runs, Trigger runs, Change Data Capture, Runtimes & sessions, Integration runtimes, Data flow debug, Notifications, and Alerts & metrics. The main content area is titled "Trigger runs" and displays a table of runs. The table has columns: Trigger name, Trigger type, Trigger time, Status, Pipelines, Run, Message, Properties, and Run ID. There are two entries:

Trigger name	Trigger type	Trigger time	Status	Pipelines	Run	Message	Properties	Run ID
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.

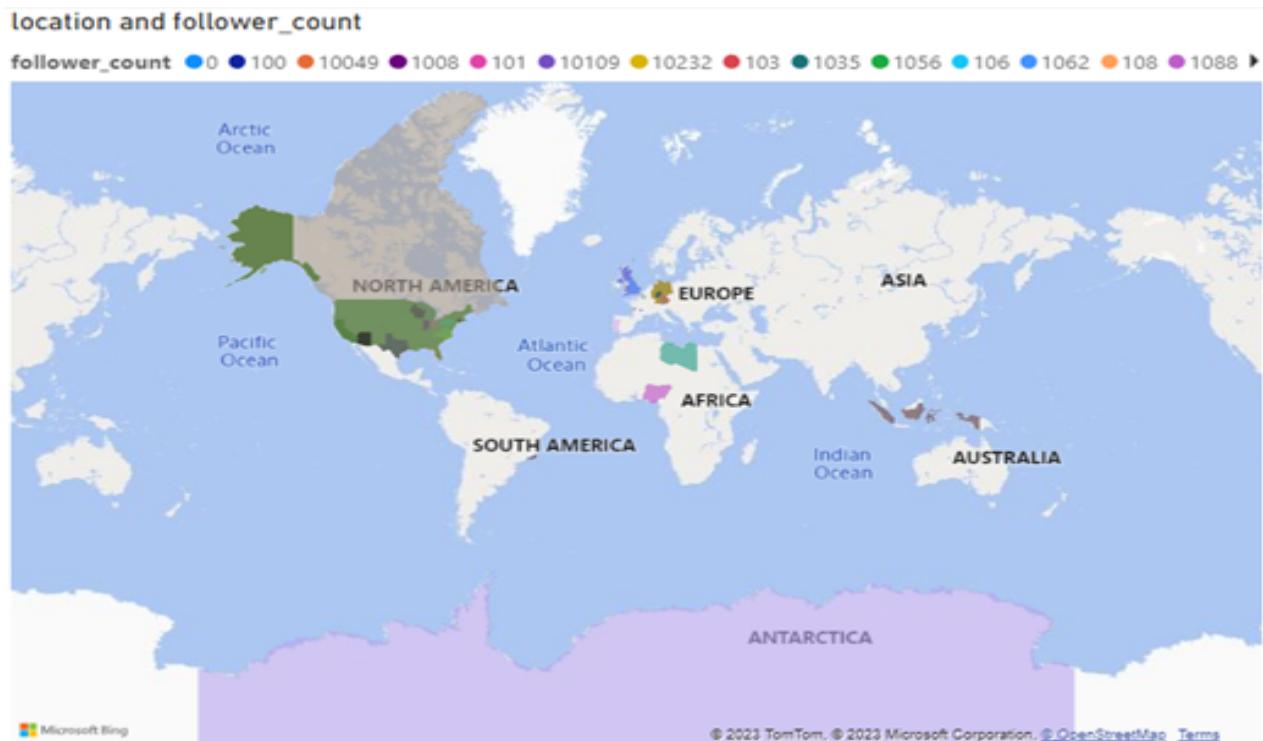
The screenshot shows the Azure Data Factory pipeline editor. On the left, there's a sidebar with a tree view of activities: Move & transform, Synapse, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, Machine Learning, and Power Query. The main area is titled "CopyPipeline\_tweet..." and shows a "Trigger (1)" section with a "Delete" activity named "Delete1". Below it is another "Delete" activity named "Delete3". The "Source" tab is selected in the configuration pane. It includes fields for Dataset (set to "SourceDataset\_twitterheader"), File path type (set to "File path in dataset"), Start time (UTC), End time (UTC), Filter by last modified, Recursively (checked), and Max concurrent connections.

P5:

## Visualizations

### 1. Location-based analysis

We have performed a location-based analysis on 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.



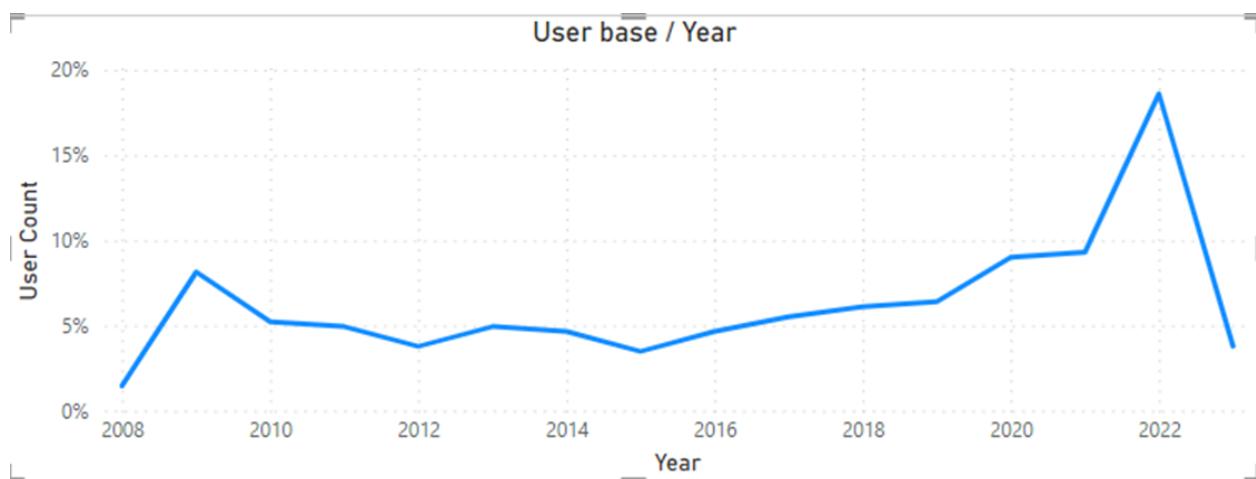
### 2. User base on Twitter for each year

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.

Furthermore, by visualizing the data in Power BI, we can create more compelling and informative presentations for stakeholders. 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.



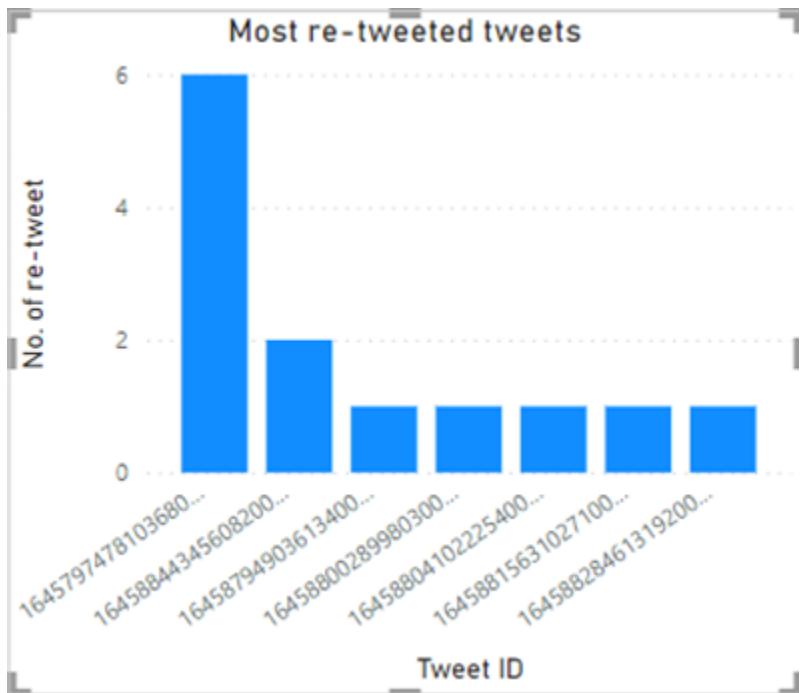
### 3. Analyze the number of retweets

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. Correlation matrix 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.

