

**PES UNIVERSITY**

100 feet Ring Road, BSK 3<sup>rd</sup> Stage  
Bengaluru 560085



Department of Computer Science and Engineering  
B. Tech. CSE - 6<sup>th</sup> Semester  
Jan – May 2022

UE19CS344  
DATABASE TECHNOLOGIES (DBT)  
**Project Report**

# Twitter Streaming Analysis

PES1UG19CS222: K K Tarun Kumar  
PES1UG19CS581: Vivek Ramesh  
PES1UG19CS143: Dhruv Menon

# Table of Contents

- 1.Introduction
- 2.Installation of Software
- 3.Input Data
  - a. Source
  - b.Description
- 4.Streaming Mode Experiment
  - a. Description
  - b.Windows
  - c. Results
- 5.Batch Mode Experiment
  - a. Description
  - b.Windows
  - c. Results
- 6.Comparison of Streaming & Batch mode
  - a. Results and Discussion
- 7.Conclusion
- 8.References

## Introduction:

This Database Technologies project deals with handling a stream of tweets, finding out the most popular hashtags in a particular batch/window and displaying them. The data is further also stored in a database for persistent storage, should there be any future requirements for the data.

The data is streamed through Kafka, both batch and stream processing are performed and the outputs are delivered. The data is then pulled by Spark (Structured Streaming is used) from Kafka and here the structured data is then finally stored in a MySQL database. This gives us an insight into how a pipeline can be created to pass data through for processing purposes.

**GITHUB LINK:** <https://github.com/tarunkumarrrr/UE19CS344---DBT---Twitter-Streaming-Analysis.git>

## Installation of Software:

- Spark v3.1.2
- Kafka v3.1.0
- Confluent for Kafka

## External python libraries required:

- confluent-kafka
- pandas
- numpy
- kafka-python
- pyspark
- mysql-connector

## Input Data:

- a) Source: The input dataset is called Sentiment140 which can be found here: [Sentiment140 dataset with 1.6 million tweets | Kaggle](#).
- b) Description: This dataset contains 1.6 million tweets. The schema is of the form:  
(Sentiment, Tweet).

# **Streaming Mode experiment:**

## **Description**

We conducted this experiment using Kafka. The input data is streamed into a topic using a kafka producer, which streams the data from a .csv file on the local machine, into the topic. We then set up a kafka consumer, which we used to subscribe to that particular topic. This data streams into this consumer, where a tumbling window calculates which hashtag is trending at that point. We performed this experiment using the following softwares:

- 1) kafka module (kafka.admin to publish)
- 2) confluent\_kafka module (confluent\_kafka.Consumer to subscribe)

## **Windows**

We have used a window size of 15 minutes. Therefore, every 15 minutes, a tumbling window operation takes place on the data so we can get trending hashtags.

## **Results**

We get the top trending hashtags every 15 minutes in case the tweets streamed during that time had hashtags, else we get an empty list for that window.

## **Batch Mode experiment:**

### **Description**

We conducted this experiment using Kafka. The input data is streamed into a topic using a kafka producer, which streams the data from a .csv file on the local machine, into the topic. We then set up a kafka consumer, which we used to subscribe to that particular topic. This data from the topic gets sent in batches to the consumer. Then, trending hashtags are then extracted from that data and displayed.

### **Batch Size**

We have implemented this experiment in such a way, that all the data is steamed in batches every 15 minutes. Therefore, the batch size depends on how much data has been sent to the topic in the past 15 minutes.

### **Results**

The top trending hashtags are received for each batch of tweets that is sent for processing at 15-minute intervals.

# Comparison of Streaming & Batch mode

## Batch mode:

```
peslug19cs222@tarun-kumar-k-k-cs222:~/Desktop/PE5/DBT/DBT_Project/For Batch and Stream$ python3 consumer_in_batch.py
['#diving', '#fb', '#datacenter', '#FF', '#beertweetup', '#wedding', '#noundlessunday', '#carpool', '#hoppusday', '#spymaster', '#Followfriday', '#hhhs', '#tcot', '#Coronation_Tweet']
['#communityone', '#haveyouever', '#iranelection']
['#cycling', '#biking', '#musicnonday', '#fb', '#jrfriday', '#6', '#LETS', '#Ireneber', '#15', '#golakers', '#Business', '#Marketing', '#squarespace', '#codaww', '#np1', '#NP2', '#xbox', '#WTF', '#Need']
```

(Output after 15 minutes)

## Stream mode:

```
peslug19cs222@tarun-kumar-k-k-cs222:~/Desktop/PE5/DBT/DBT_Project/For Batch and Stream$ python3 consumer.py
K311651723396.617[FAIL][rdkafka#consumer-1] [thrd:GroupCoordinator]: GroupCoordinator: tarun-kumar-k-k-cs222:9092: Failed to connect to broker at [tarun-kumar-k-k-cs222]:9092: Invalid argument (after 91
in state CONNECT)
K311651723401.708[FAIL][rdkafka#consumer-1] [thrd:tarun-kumar-k-k-cs222:9092/0]: tarun-kumar-k-k-cs222:9092/0: Failed to connect to broker at [tarun-kumar-k-k-cs222]:9092: Invalid argument (after 79ns i
state CONNECT)
K311651723402.805[FAIL][rdkafka#consumer-1] [thrd:tarun-kumar-k-k-cs222:9092/0]: tarun-kumar-k-k-cs222:9092/0: Failed to connect to broker at [tarun-kumar-k-k-cs222]:9092: Invalid argument (after 95ns i
state CONNECT, 1 identical error(s) suppressed)
b'4 Sooo readyyy for the summerr babyyyy!!!!!!!!!! '
[]
b'0 I had the awesonest dream ever!!!! Except co-starring the wrong person... '
[]
b'4 @mattgoyer Ahh I see. Thanks anyways! '
b'0 at work and ready to go home to my comfy cozy bed. maybe the boss man will let me leave....i feel miserable. '
b'0 @ArielRebel bonne nuit d'une autre quebecoise sheesh, my french stinks after being in the states so long '
b'4 resting my sore muscles and babying the first sunburn of the year. llife is good. '
b'0 Why is my baby growing up? Just tell me MHHH!!?
```

(a similar result for streaming mode after 15 minutes)

## Conclusion

We learnt about the following technologies: Kafka, Spark and MySQL and how they can be used in a pipeline to process data and perform further analysis with the data received. In our case we used tweets for our analysis but this can be extended to other domains where the data can be published to a Kafka topic and that topic can then be subscribed to by other technologies to receive whatever data is needed for processing/analysis.

