Techniques Used

Bloom Filter:

A bloom filter is a probabilistic data structure which is used to check whether an element is present in a set of words or not. It gives a definite answer if the element is not present. If the element is present it confirms it with a high probability. We create a bloom filter for the set of stop words and punctuations, which is used to check if an element is a stop word very efficiently.

Count Min Sketch:

Count Min Sketch is a probabilistic data structure that returns the frequency of an element in a stream of data. An element passes through multiple hash functions and the values generated are used to update the sketch data structure by incrementing the values in specific places. We use this to store the frequency of all the words in the twitter stream.

Heavy Hitters:

Heavy hitters are the most frequently occurring elements in a stream of data. We utilize the count min sketch to calculate the Top K heavy hitters more efficiently. We maintain a dictionary with K elements in it along with their frequencies. As soon as a new element comes, we get its frequency from the count min sketch and compare it with the minimum frequency element in the heavy hitters data structure. If the frequency of the current word is greater than the minimum frequency, we delete the minimum frequency element and add this element as a heavy hitter.

NLTK Tokenize:

We used the nltk library to tokenize the stream of tweets from twitter, to convert it into a stream of tokens after applying flatMap() operation. This library takes care of splitting the sentences into words properly, keeping in mind the various punctuations.

Confluent:

API used for connecting Twitter to Kafka and then Kafka to Elastic Search. It spawns all the required services like Zookeeper, Kafka e.t.c. We use config files in JSON format to source data from twitter using the TwitterSourceConnector and dumping into a Kafka topic.
We also use config files in JSON format to sink the data from the Kafka topic to the elastic search server by using the ElasticsearchSinkConnector.

Approach

1. Make Twitter developer accounts to receive access tokens and keys for app.
2. Configure confluent to spawn all the required services (Zookeeper, kafka e.t.c)
3. Configure twitter source config file to get data form twitter and dump in Kafka topic.
4. Configure elastic search sink config file to transfer data from Kafka to elasticsearch.
5. Now data will be flowing from Twitter to Kafka to Elastic Search.
6. We split the tasks in 2 phases now:

   Client to support various type of queries on elastic search

   Made a user interface for performing various type of queries on the streaming

twitter data which is now stored on the elastic search

Term Query, Match Query, Prefix Query and Fuzzy Query

Stream data from Kakfa to calculate the heavy hitters periodically

We used Apache Spark to generate streams of tweets from Kafka and then

a. **NLTK** to tokenize the stream of tweets into words
b. **Bloom Filters** to quickly filter the stop words in the stream
c. We updated the **count min sketch** for each word in the stream
d. Calculated the Top K **Heavy hitters** using the sketch from the stream.
e. Printed the Heavy hitters periodically

Results Summary

Search Client on Elastic Search

```
asaxena6@vm17-56:~$ python3 ElasticSearch.py
Enter the integer for type of query you want
 0. Match_All Query
 1. Term Query
 2. Match Query
 3. Prefix Query
 4. Fuzzy Query
2
Enter input for matching query
tennis
{'_shards': {'failed': 0, 'skipped': 0, 'successful': 5, 'total': 5},
 'hits': {'hits': [{'_id': '1121307071482347521',
                    '_index': 'twitterdatajson',
                    '_score': 0.9107011,
                    '_source': {'Text': "#Tennis: Andy Murray 'optimistic' he "
                                        'can play this year, says mother Judy '
                                        'https://t.co/quwgSS23CX'},
                    '_type': 'type.name=tweet'},
                   {'_id': '1121306475748478976',
                    '_index': 'twitterdatajson',
                    '_score': 0.81017023,
                    '_source': {'Text': 'Check out what I just added to my '
                                        'closet on Poshmark: 10ctw Diamond '
                                        'Tennis Bracelet. '
                                        'https://t.co/U1UAfkB6wg via '
                                        '@poshmarkapp #shopmycloset'},
```

Heavy Hitters Results

```
andy 3
final 3
-----------
tennis 22
https 19
rt 16
hunter 14
กจ็ดดด 7
uniqlo 7
x 7
โบรูโตะ 7
นารูโตะ 7
academia 7
```