```
from pyspark.sql import SparkSession
from pyspark.sql.functions import explode, split, col, lower, regexp_extract, udf
from pyspark.sql.types import StructType, StructField, StringType
import pandas as pd

spark = SparkSession.builder.appName("ColabSocialMediaStreaming").getOrCreate()
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell


```
from pyspark.sql import SparkSession
from pyspark.sql.functions import explode, split, col, lower, regexp_extract, udf
from pyspark.sql.types import StructType, StructField, StringType
import pandas as pd

spark = SparkSession.builder.appName("ColabSocialMediaStreaming").getOrCreate()
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell


```
# Simulated streaming tweets
data = [
    {"id":"1","text":"I love the new features! #awesome #spark","created_at":"2025-11-16T20:00:01"},
    {"id":"2","text":"This is the worst update ever. #fail #bug","created_at":"2025-11-16T20:00:05"},
    {"id":"3","text":"Had a great day working with data. #datascience #spark","created_at":"2025-11-16T20:00:08"},
    {"id":"4","text":"Why is this so slow? #performance #bug","created_at":"2025-11-16T20:00:12"},
    {"id":"5","text":"Fantastic improvements! Love it. #awesome #update","created_at":"2025-11-16T20:00:20"},
    {"id":"6","text":"Not impressed, found a problem. #bug #complaint","created_at":"2025-11-16T20:00:23"},
    {"id":"7","text":"Nice work from the team. #congrats #spark","created_at":"2025-11-16T20:00:30"},
    {"id":"8","text":"Neutral comment with no emotions","created_at":"2025-11-16T20:00:35"},
    {"id":"9","text":"Best release this year! #release #awesome","created_at":"2025-11-16T20:00:40"},
    {"id":"10","text":"Awful crashes still exist. #bug #fail","created_at":"2025-11-16T20:00:45"}
]

schema = StructType([
    StructField("id", StringType(), True),
    StructField("text", StringType(), True),
    StructField("created_at", StringType(), True)
])

df_full = spark.createDataFrame(data, schema)
df_full.show(truncate=False)
```

```
+---+-----------------------------------------------------+-------------------+
|id |text                                                 |created_at         |
+---+-----------------------------------------------------+-------------------+
```

```
|1  |I love the new features! #awesome #spark              |2025-11-16T20:00:01|
|2  |This is the worst update ever. #fail #bug             |2025-11-16T20:00:05|
|3  |Had a great day working with data. #datascience #spark|2025-11-16T20:00:08|
|4  |Why is this so slow? #performance #bug                |2025-11-16T20:00:12|
|5  |Fantastic improvements! Love it. #awesome #update     |2025-11-16T20:00:20|
|6  |Not impressed, found a problem. #bug #complaint       |2025-11-16T20:00:23|
|7  |Nice work from the team. #congrats #spark             |2025-11-16T20:00:30|
|8  |Neutral comment with no emotions                      |2025-11-16T20:00:35|
|9  |Best release this year! #release #awesome             |2025-11-16T20:00:40|
|10 |Awful crashes still exist. #bug #fail                 |2025-11-16T20:00:45|
+---+------------------------------------------------------+-------------------+
```

```
positive_words = ['good','great','happy','love','awesome','best','nice','fantastic','congrats']
negative_words = ['bad','sad','hate','terrible','worst','awful','angry','problem','fail','complaint']

def lex_sentiment(text):
    t = text.lower() if text else ""
    pos = sum(1 for w in positive_words if w in t)
    neg = sum(1 for w in negative_words if w in t)
    if pos > neg: return "positive"
    if neg > pos: return "negative"
    return "neutral"

sentiment_udf = udf(lex_sentiment, StringType())
```

```
batch_size = 3
batches = [data[i:i+batch_size] for i in range(0, len(data), batch_size)]

global_sentiment = {}
global_hashtags = {}

import re
def extract_hash(text):
    return re.findall(r'#\w+', text.lower())

for i, batch in enumerate(batches):
    print(f"\n=== Micro Batch {i+1} ===")
    df = spark.createDataFrame(batch, schema)

    df = df.withColumn("sentiment", sentiment_udf(col("text")))

    df_hash = df.withColumn("hashtag", explode(split(regexp_extract(lower(col("text")), r"(#\\w+)", 0),
    df_hash = df_hash.filter(col("hashtag") != "")

    pdf = df.toPandas()
    hdf = df_hash.toPandas()

    display(pdf)
    display(hdf)

    for row in pdf.itertuples():
        global_sentiment[row.sentiment] = global_sentiment.get(row.sentiment,0)+1

    for row in hdf.itertuples():
        global_hashtags[row.hashtag] = global_hashtags.get(row.hashtag,0)+1

print("\n\n=== FINAL AGGREGATED RESULTS ===")
print("Sentiment:", global_sentiment)
print("Hashtags:", global_hashtags)
```

```
=== Micro Batch 1 ===
```

	id	text	created_at	sentiment
0	1	I love the new features! #awesome #spark	2025-11-16T20:00:01	positive
1	2	This is the worst update ever. #fail #bug	2025-11-16T20:00:05	negative
2	3	Had a great day working with data. #datascienc...	2025-11-16T20:00:08	positive

id	text	created_at	sentiment	hashtag

```
=== Micro Batch 2 ===
```

	id	text	created_at	sentiment
0	4	Why is this so slow? #performance #bug	2025-11-16T20:00:12	neutral
1	5	Fantastic improvements! Love it. #awesome #update	2025-11-16T20:00:20	positive
2	6	Not impressed, found a problem. #bug #complaint	2025-11-16T20:00:23	negative

id	text	created_at	sentiment	hashtag

```
=== Micro Batch 3 ===
```

	id	text	created_at	sentiment
0	7	Nice work from the team. #congrats #spark	2025-11-16T20:00:30	positive
1	8	Neutral comment with no emotions	2025-11-16T20:00:35	neutral
2	9	Best release this year! #release #awesome	2025-11-16T20:00:40	positive

id	text	created_at	sentiment	hashtag

```
=== Micro Batch 4 ===
```

	id	text	created_at	sentiment
0	10	Awful crashes still exist. #bug #fail	2025-11-16T20:00:45	negative

id	text	created_at	sentiment	hashtag

```
=== FINAL AGGREGATED RESULTS ===
Sentiment: {'positive': 5, 'negative': 3, 'neutral': 2}
Hashtags: {}
```

Next steps: Generate code with pdf | New interactive sheet | Generate code with pdf | New interactive sheet | Generate code with pdf