3.

https://www.reddit.com/prefs/apps

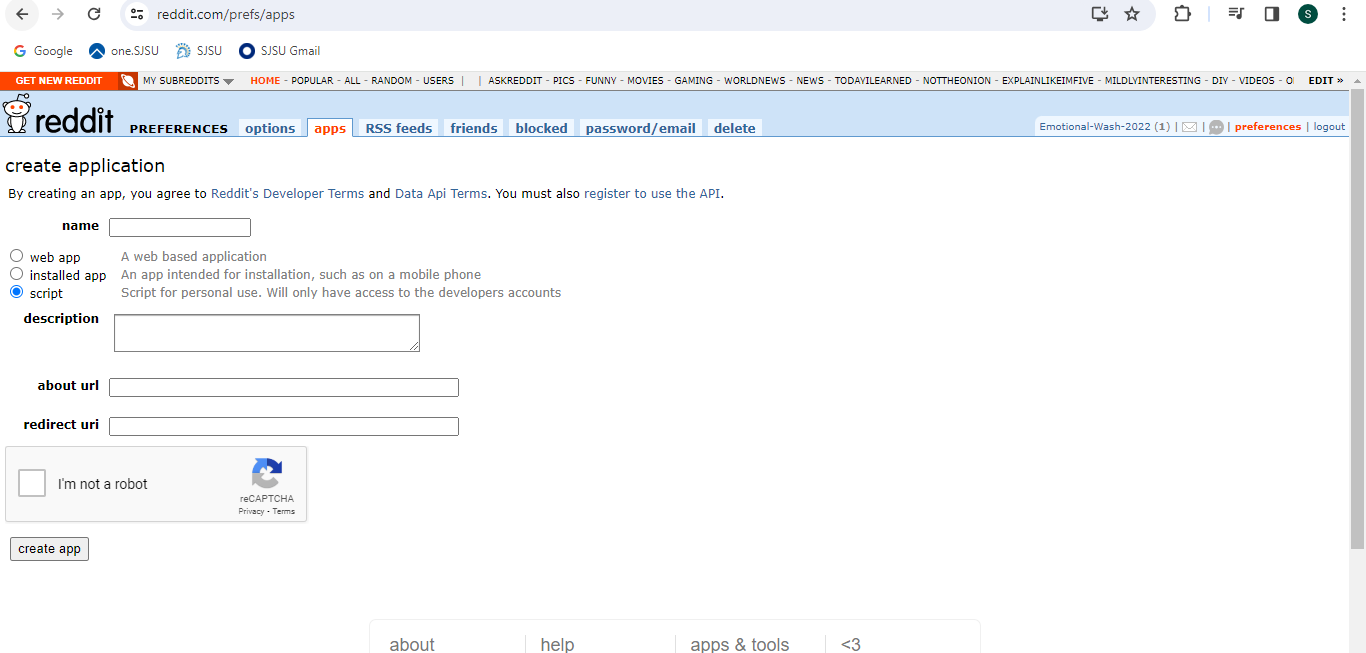
**Reddit: GrAfDlIemzbMRak3Rnodfeop6hIgpg**

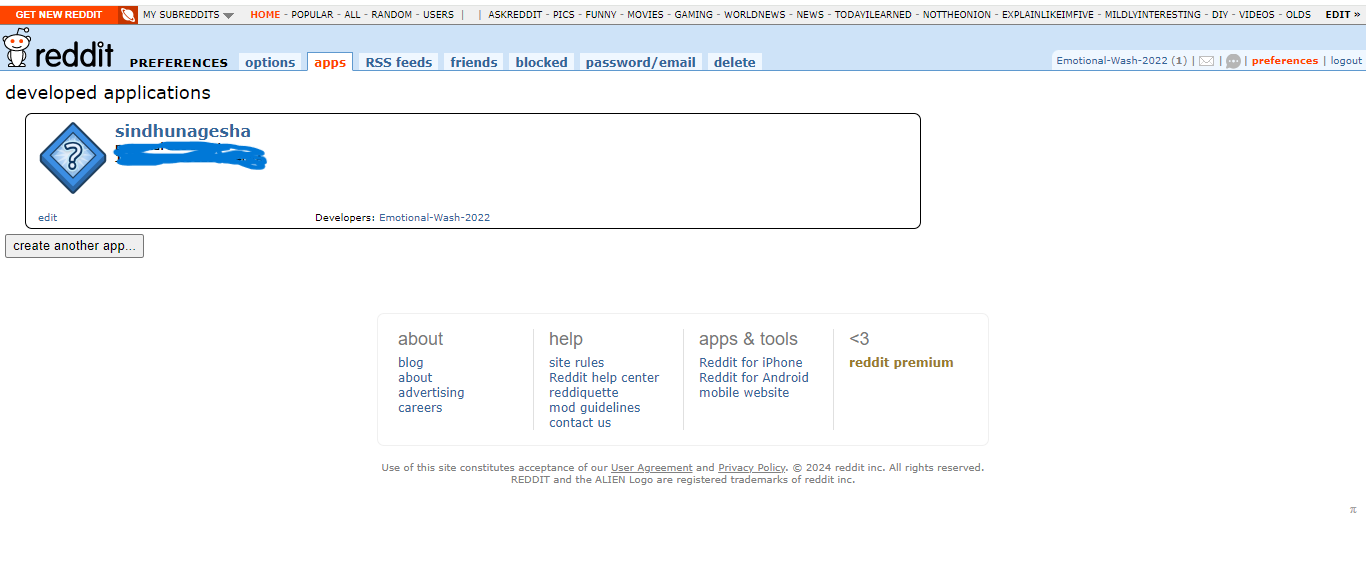
| **developers** | **: EdgeEnvironmental419** |
| --- | --- |

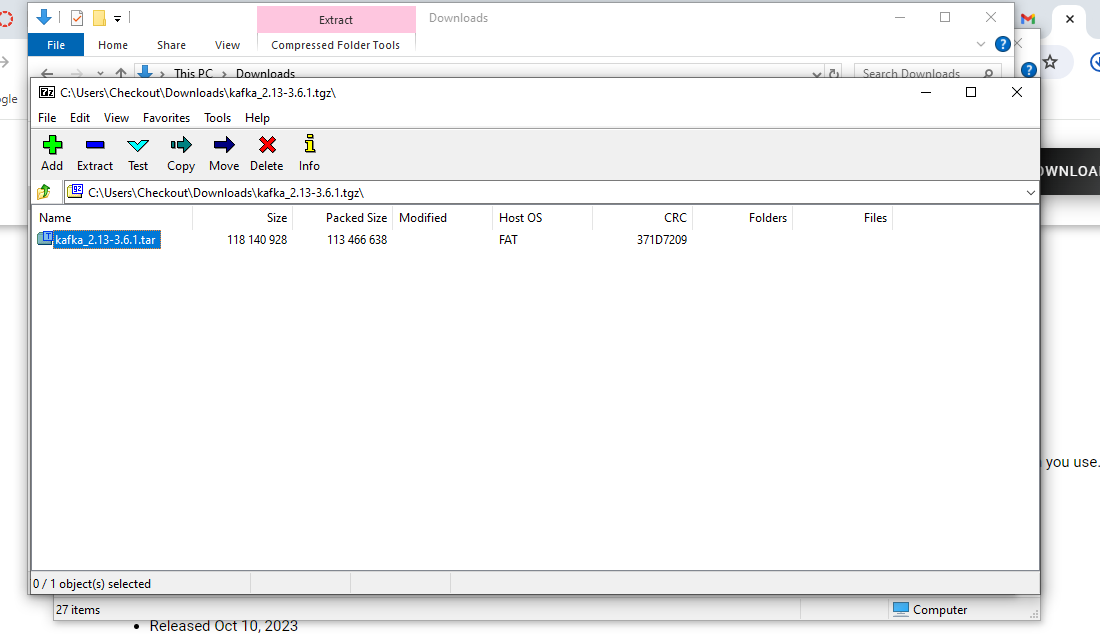
| **secret:** |  |
| --- | --- |

## **SINDHUNAGESHA**

### **personal use script: g5kxhCV2kwve1hoxLl4Njw**

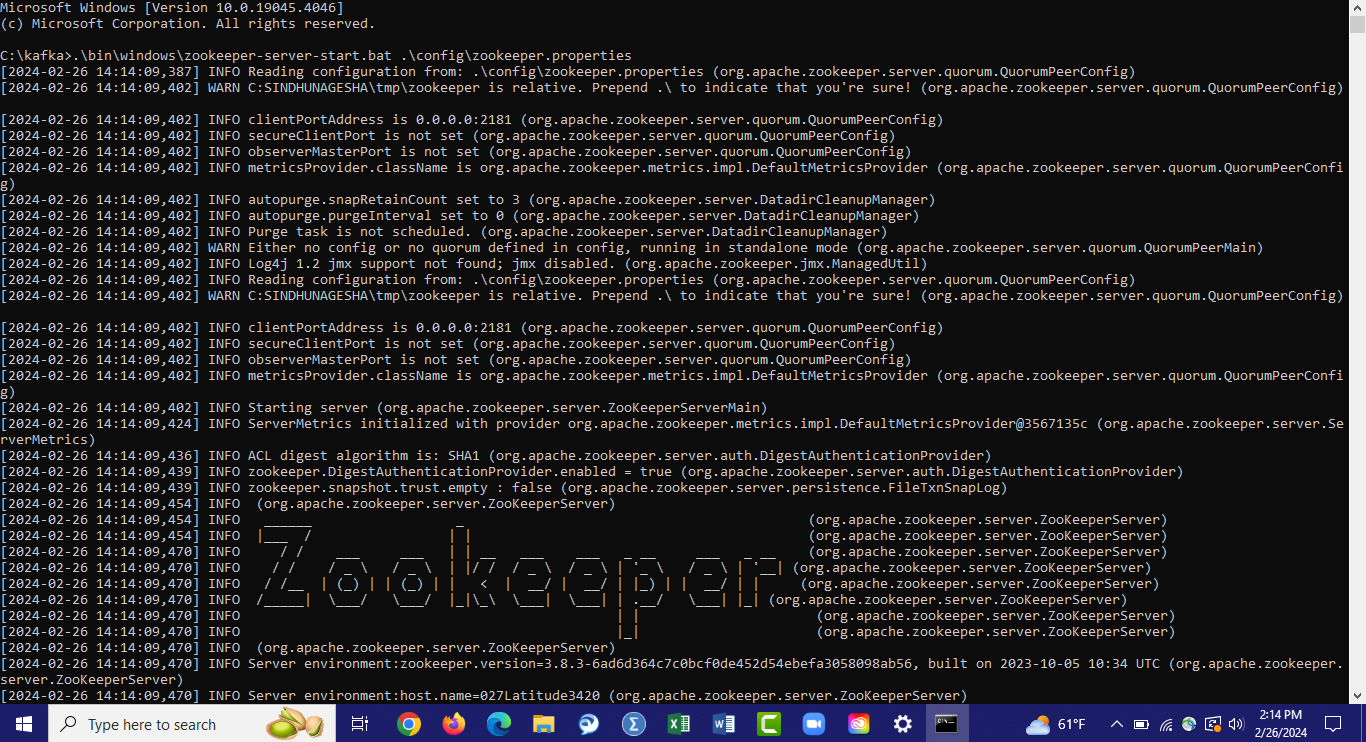
****

****

****

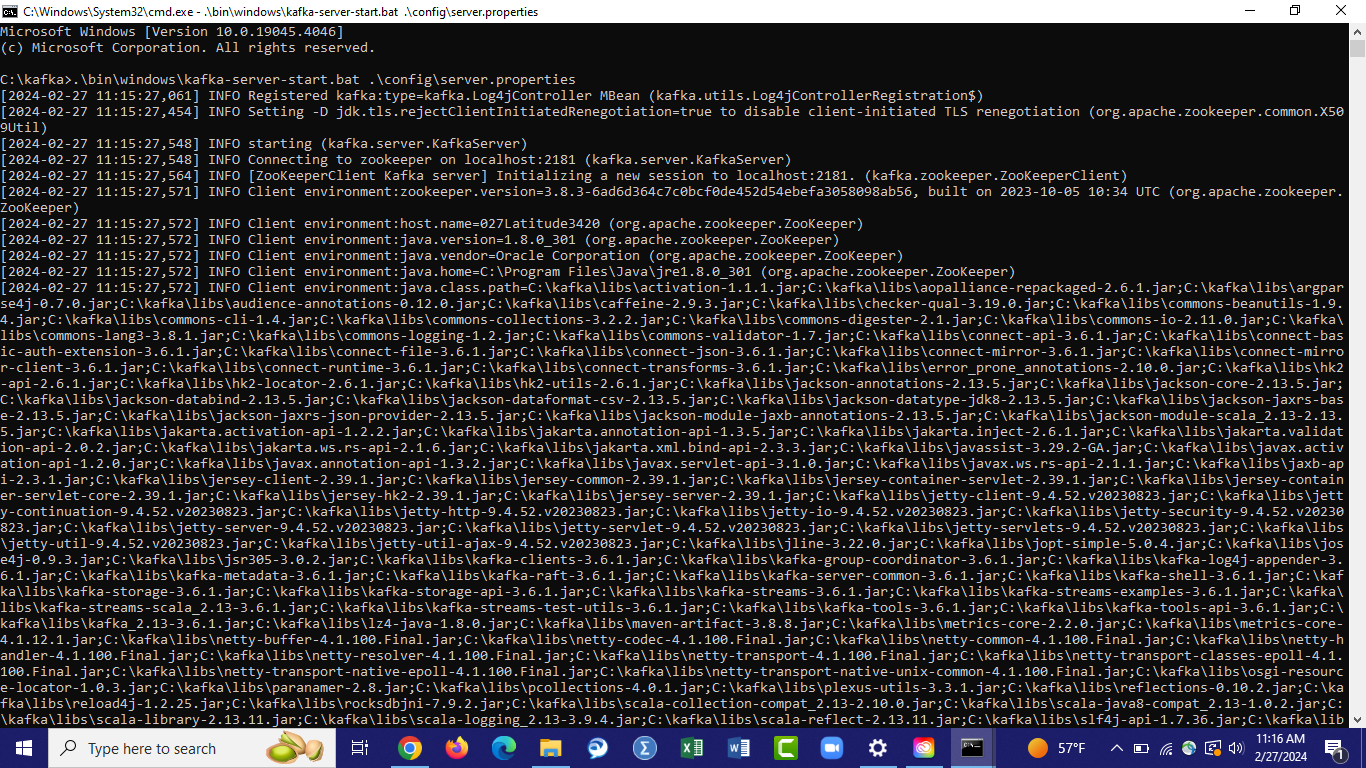
**.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties**

**To start zookeeper.**

****

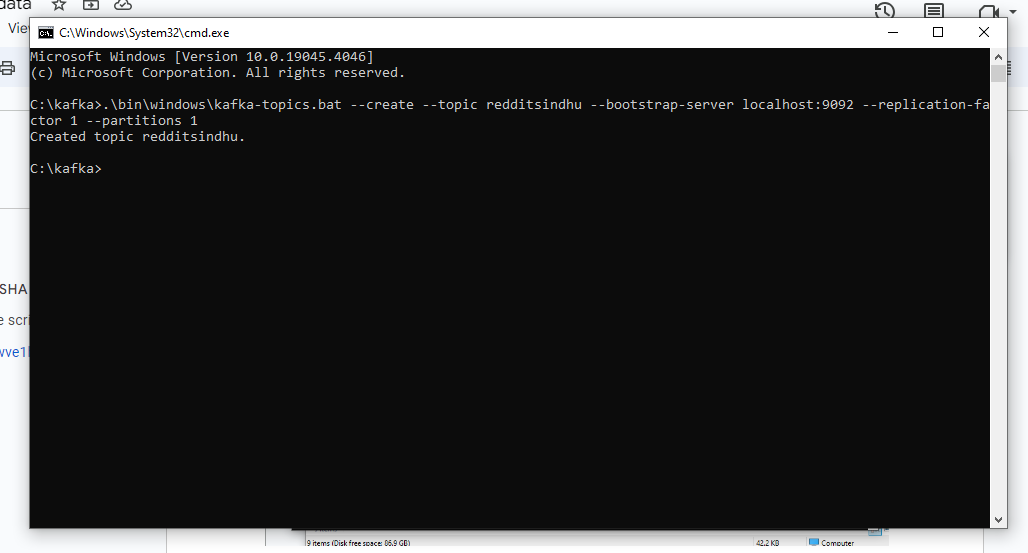
**To start kafka: .\bin\windows\kafka-server-start.bat .\config\server.properties**

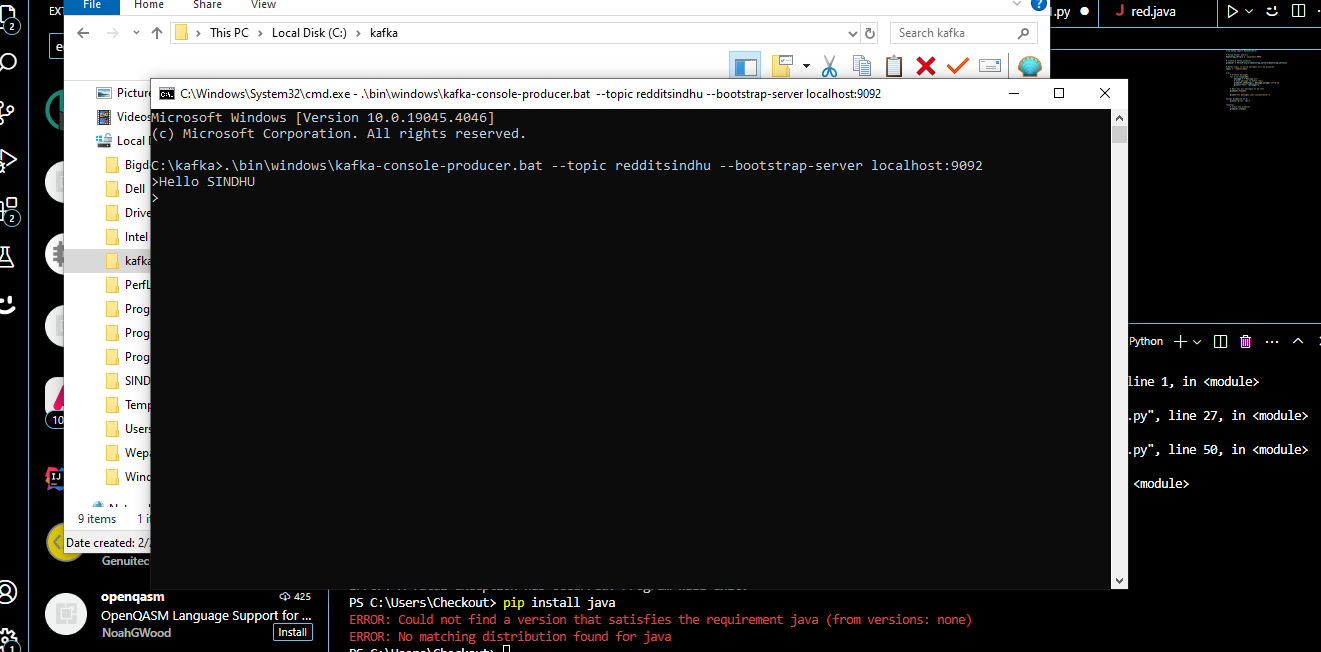
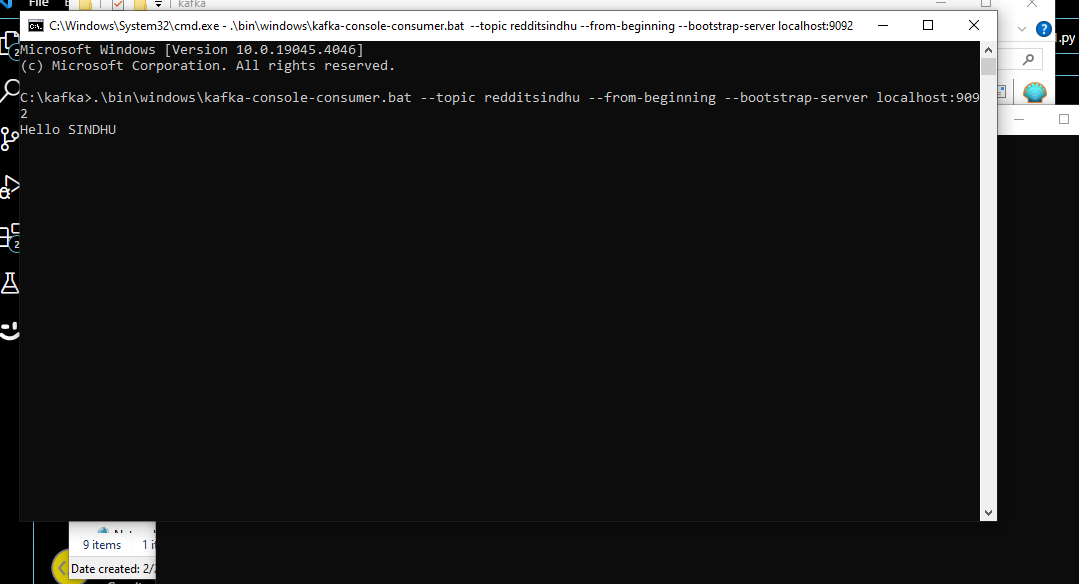
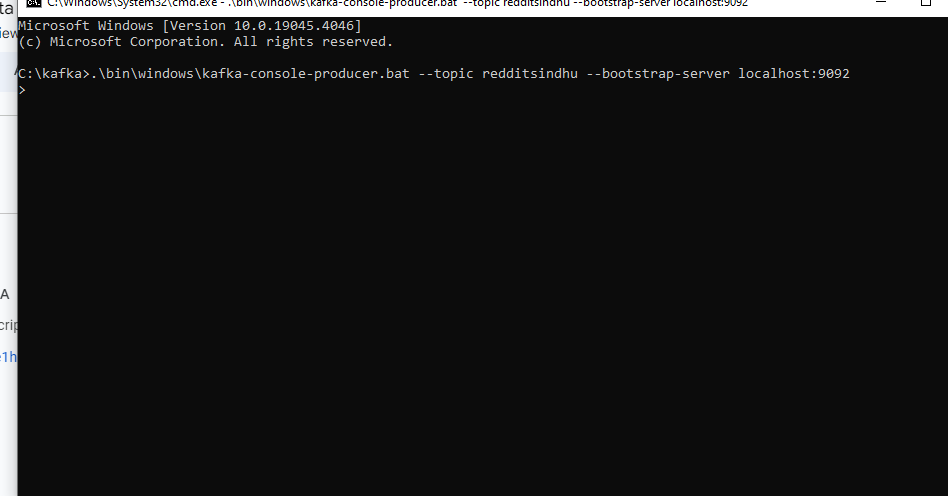
**Capture 17**

****

**Topic creation :**

**.\bin\windows\kafka-topics.bat --create --topic redditsindhu --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1**

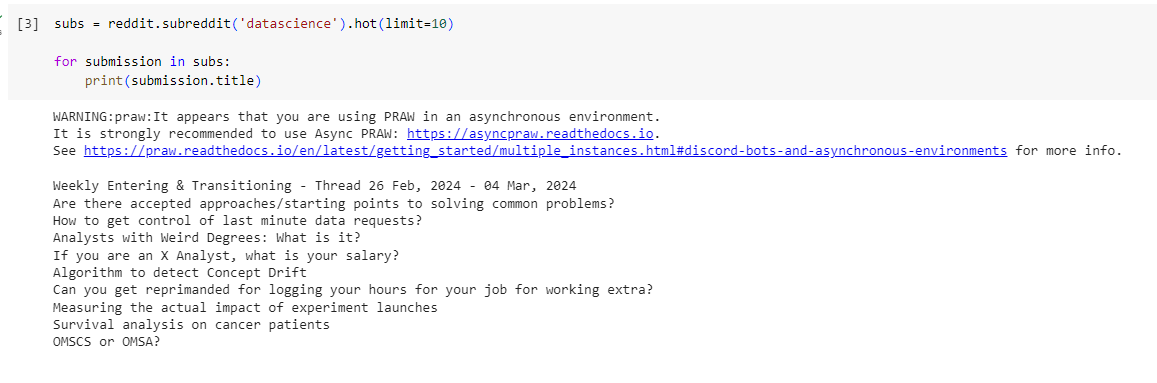
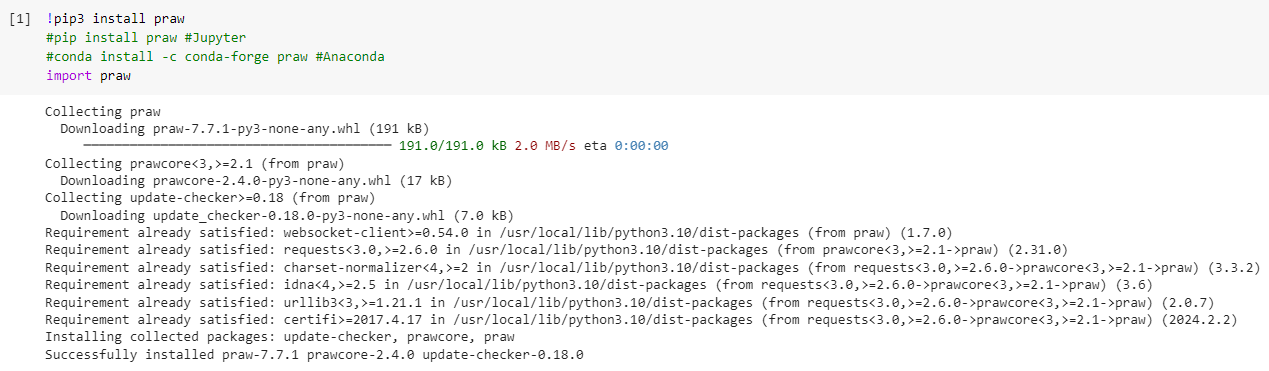
****

****

**2.a) Real-time Stream Consumer:**

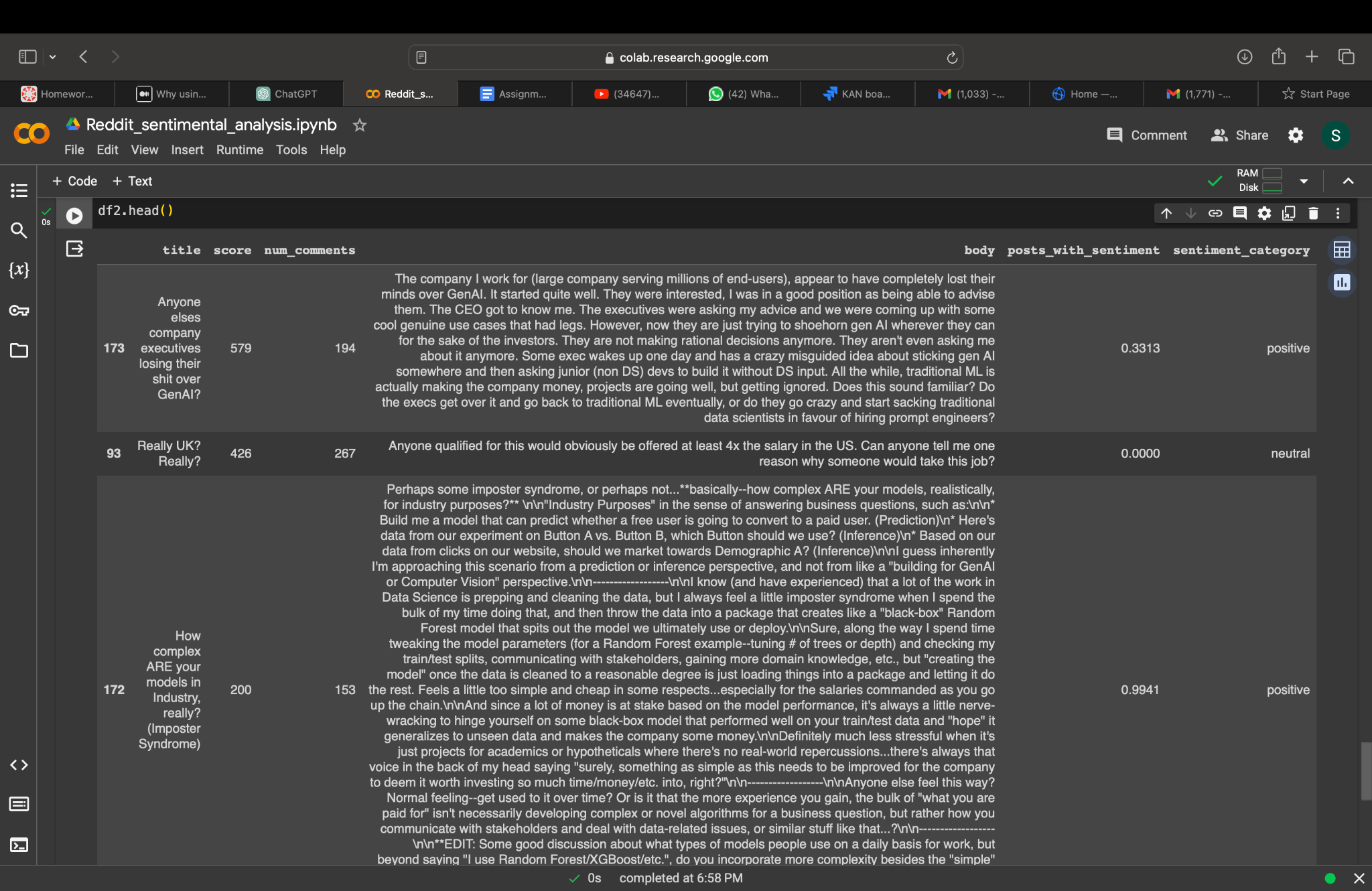
**Write a Python script using APIs to consume posts (data) from a specific keyword in real-time. Filter and pre-process the data (remove URLs, etc.)**

**FOr subreddit “datascience”**

****

**2.b) Real-time Streaming Data Analysis with Spark:**

**Sentiment Analysis -** Leverage a pre-trained sentiment analysis model (e.g., VADER) within Spark. Use PySpark to analyze the sentiment (positive, negative, neutral) of each post in the stream.

****

**2.c)Store and Access Data:**

**import csv**

**from confluent\_kafka import Producer**

**# Configuration**

**conf = {'bootstrap.servers': "localhost:9092"}**

**# Create Producer instance**

**producer = Producer(conf)**

**# Topic to produce to**

**topic = "your\_topic\_name"**

**# Path to your CSV file**

**csv\_file = "path/to/your/file.csv"**

**# Function to read CSV file and produce data to Kafka**

**def produce\_to\_kafka(file\_path, kafka\_producer, kafka\_topic):**

**with open(file\_path, 'r') as file:**

**csv\_reader = csv.reader(file)**

**headers = next(csv\_reader) # Skip headers if present**

**for row in csv\_reader:**

**# Assuming each row of CSV contains a single data point to be sent to Kafka**

**data = ','.join(row) # Adjust this according to your CSV structure**

**kafka\_producer.produce(kafka\_topic, value=data)**

**kafka\_producer.flush()**

**# Call the function to produce data to Kafka**

**produce\_to\_kafka(csv\_file, producer, topic)**

**import csv**

**from confluent\_kafka import Producer**

**def produce\_to\_kafka(csv\_file, bootstrap\_servers, topic):**

**producer = Producer({'bootstrap.servers': bootstrap\_servers})**

**with open(csv\_file, 'r') as file:**

**csv\_reader = csv.reader(file)**

**for row in csv\_reader:**

**key = None # Assuming no key**

**value = ','.join(row) # Assuming each row is a CSV record**

**producer.produce(topic, key=key, value=value)**

**producer.flush()**

**# Example usage:**

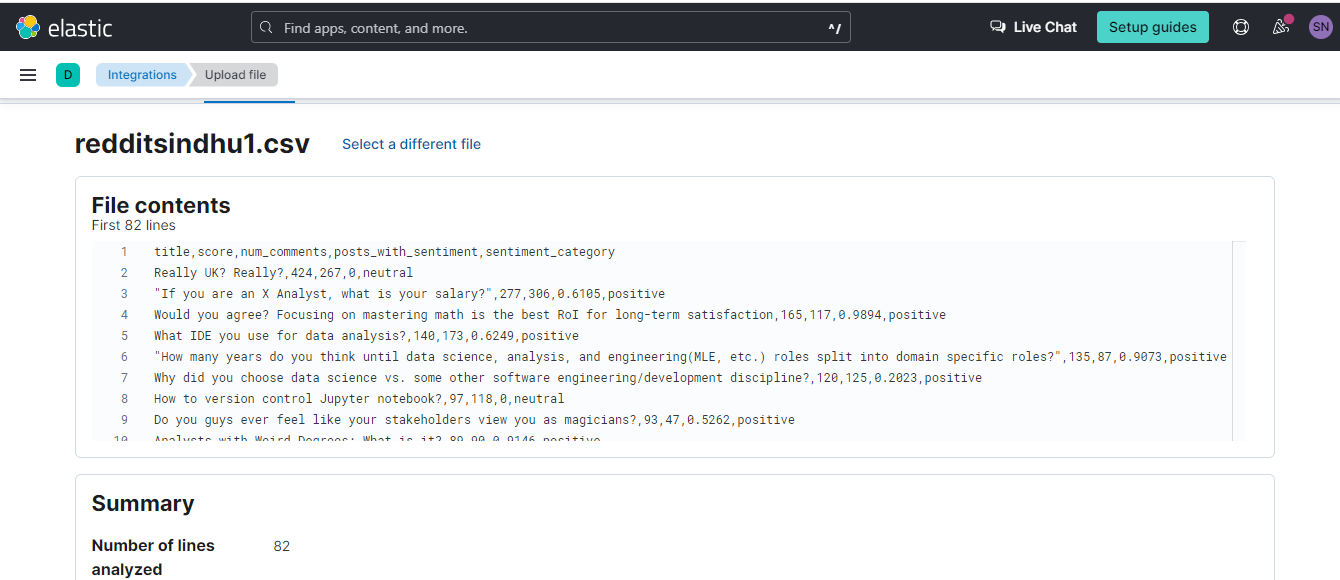
**csv\_file = 'your\_csv\_file.csv'**

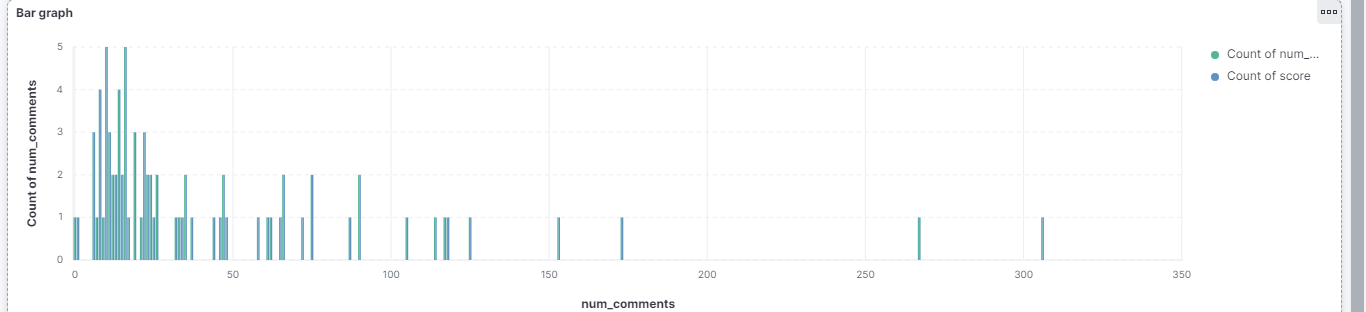
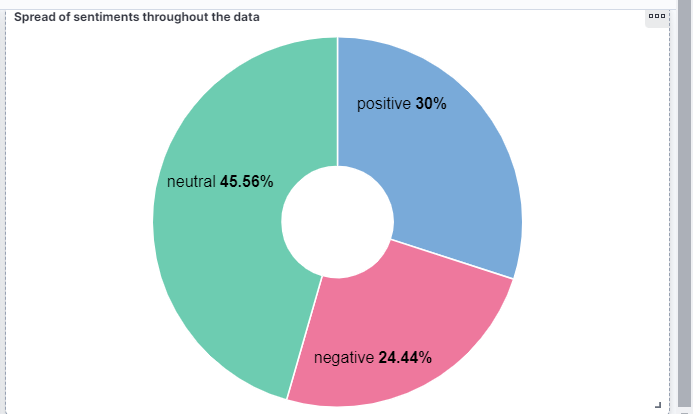
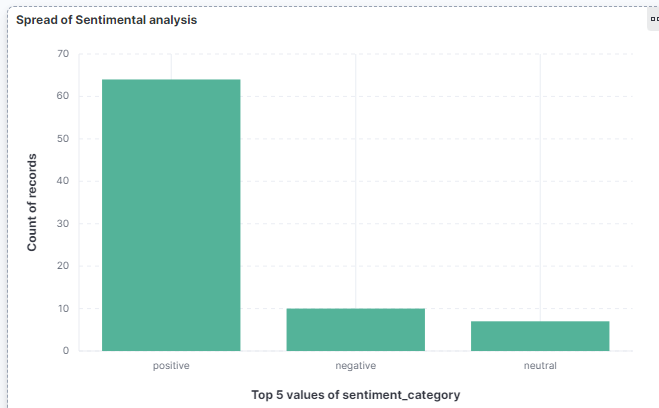
**bootstrap\_servers = 'localhost:9092'**

**topic = 'your\_topic\_name'**

**produce\_to\_kafka(csv\_file, bootstrap\_servers, topic)**

**2.d) Visualization:Create a simple dashboard using tools like Kibana to visualize the real-time data analysis.**

****

****

**2.e) Summarize your learning experience, challenges faced, and insights gained.**

1. **[5 points] Implement any one of the following:**

* **Analyze variations over time and identify trends.**
* **Incorporate geographical information and analyze sentiment by location.**
* **Use more advanced NLP techniques for improved sentiment accuracy.**
* **A comparative market trend analyzer with historic and real-time data using Alpaca API**
* **Write a blog post, create a video presentation, and share your work on GitHub to build your brand and engage with others.**

**import java.util.Properties**

**import org.apache.kafka.clients.producer.{KafkaProducer, ProducerRecord}**

**import scala.io.Source**

**object KafkaProducerExample extends App {**

**// Kafka broker properties**

**val props = new Properties()**

**props.put("bootstrap.servers", "<kafka\_broker\_address>")**

**props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer")**

**props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer")**

**// Create Kafka producer**

**val producer = new KafkaProducer[String, String](props)**

**// Read data from CSV and produce to Kafka**

**val csvFile = "your\_data.csv"**

**val topic = "<topic\_name>"**

**val bufferedSource = Source.fromFile(csvFile)**

**for (line <- bufferedSource.getLines) {**

**val record = new ProducerRecord[String, String](topic, line)**

**producer.send(record)**

**}**

**bufferedSource.close()**

**// Close producer**

**producer.close()**

**}**