1. Start Zookeeper in a terminal

zookeeper-server-start /opt/homebrew/etc/kafka/zookeeper.properties

1. Start Kafka in another terminal

kafka-server-start /opt/homebrew/etc/kafka/server.properties

If you find kafka shutting down problem run the following commands.  
lsof -i :9092

You will see the PID  
kill -9 <PID> #replace with that PID

1. Create a input csv file (streaming\_data.csv)

feature1,feature2,label

1.5,3.0,0

2.1,1.8,1

3.3,4.2,0

4.5,2.3,1

5.1,3.4,0

6.7,2.5,1

7.8,4.1,0

2.2,3.3,1

3.8,1.9,0

4.9,2.7,1

1. Create topic

bin/kafka-topics.sh --create --topic test-stream --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

1. Now create kafka streaming script file (stream-csv-to-kafka.sh)

# Adjust the file path to your CSV file

while IFS= read -r line; do

echo "$line" | kafka-console-producer --topic test-stream --bootstrap-server localhost:9092

done < "/Users/sivaprasanth/Documents/Big Data/ex10/streaming\_data.csv"

1. sudo ./stream-csv-to-kafka.sh
2. Create the python file (sparkstrem.py)

from pyspark.sql import SparkSession

from pyspark.ml.classification import LogisticRegression

from pyspark.ml.clustering import KMeans

from pyspark.ml.feature import VectorAssembler

from pyspark.sql.functions import col, split

# Initialize Spark session

spark = SparkSession.builder.appName("KafkaSparkCSVStreaming").getOrCreate()

# Read streaming data from Kafka

df = spark.readStream.format("kafka") \

.option("kafka.bootstrap.servers", "localhost:9092") \

.option("subscribe", "test-stream") \

.load()

# Kafka data comes as key-value pairs, we extract the value (CSV string)

df = df.selectExpr("CAST(value AS STRING)")

# Split the CSV values into columns (assuming CSV structure: feature1, feature2, label)

df = df.withColumn("value", split(col("value"), ","))

df = df.select(

col("value").getItem(0).cast("double").alias("feature1"),

col("value").getItem(1).cast("double").alias("feature2"),

col("value").getItem(2).cast("double").alias("label")

)

# VectorAssembler for features

assembler = VectorAssembler(inputCols=["feature1", "feature2"], outputCol="features")

df = assembler.transform(df)

# Memory sink to accumulate the data

query = df.writeStream.format("memory").queryName("streaming\_data").outputMode("append").start()

# Let the stream run for a while to collect data

import time

time.sleep(20) # Adjust time as needed to accumulate enough data

# Now, fetch the accumulated data from memory

static\_df = spark.sql("SELECT \* FROM streaming\_data")

# Logistic Regression for Classification

lr = LogisticRegression(featuresCol="features", labelCol="label")

lr\_model = lr.fit(static\_df)

# Perform classification prediction

classification\_predictions = lr\_model.transform(static\_df)

# KMeans Clustering

kmeans = KMeans(k=3, featuresCol="features")

kmeans\_model = kmeans.fit(static\_df)

# Perform clustering prediction

clustering\_predictions = kmeans\_model.transform(static\_df)

# Show the results

classification\_predictions.show()

clustering\_predictions.show()

# Stop the query once done

query.stop()

1. Now run the following command  
   sudo spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.0.1 sparkstream.py
2. After showing output make sure to stop zookeeper and kafka server  
   zookeeper-server-stop  
   kafa-server-stop