Spark Structured Streaming with Kafka

January 10, 2023

1 consumer.py

[ ]:
```
import json
import numpy as np
import joblib
import warnings
warnings.filterwarnings('ignore')

rob_scaler = joblib.load('../../models/rob_scaler.joblib')
rf_clf = joblib.load('../../models/rf_clf.joblib')

def consumer_task(transaction):
    # transaction = json.loads(message)
    transaction = json.loads(transaction)
    transaction["scaled_amount"]=rob_scaler.transform(np.
 ↪reshape(transaction["Amount"],(-1,1))).reshape(-1)[0]
    del transaction['Amount']
    pred = rf_clf.predict(np.array(list(transaction.values())).reshape(1,-1))

    if pred[0] == 0:
        prediction = "Valid Transaction"
    else:
        prediction = "Fraud Transaction**********"

    return prediction
```

2 spark_engine.py

[ ]:
```
from os import getcwd
from pyspark.sql import SparkSession
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType
from consumer import consumer_task # user-defined module
import warnings
warnings.filterwarnings('ignore')
```

```
# Create Spark Session
spark = SparkSession \
  .builder \
  .appName("CCFD_StrcturedStreaming") \
  .config("spark.serializer", "org.apache.spark.serializer.JavaSerializer") \
  .config("spark.streaming.receiver.writeAheadLog.enable", "true") \
  .getOrCreate()


spark.sparkContext.setLogLevel("ERROR")

# Read Kafka Streams

# Multiple Kafka Servers
# BOOTSTRAP_SERVERS = 'localhost:9092,localhost:9093,localhost:9094'

# Single Kafka Server
# BOOTSTRAP_SERVERS = 'localhost:9092'

# Specific TopicPartitions to consume.
# Format: json string '{"topicA":[0,1],"topicB":[2,4]}'

df = spark \
.readStream \
.format("kafka") \
.option("kafka.bootstrap.servers", 'localhost:9092') \
.option("assign", '{"topictest":[0,1,2]}') \
.option("startingOffsets", 'earliest') \
.option("failOnDataLoss","true") \
.load()


# Define UDF for prediction
prediction_udf = udf(consumer_task, returnType= StringType())

df2 = df.selectExpr("CAST(value AS STRING)")

# Appy UDF on received data
df3 = df2.withColumn("value2", prediction_udf("value"))

df4 = df3.selectExpr("CAST(value2 AS STRING)")

df5 = df4.writeStream \
  .outputMode("update") \
  .option("checkpointLocation", getcwd()+"/checkpoint_dir") \
  .format("console") \
  .trigger(processingTime= "1 seconds") \
```

```
  .queryName("CCFD_1producer") \
  .start()

df5.awaitTermination()
```

3 Submit Spark Job using spark-submit

```
$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.3.1
spark_engine.py
```

4 Documentation

Structured Streaming + Kafka Integration Guide