batch-processing

December 23, 2020

[1]:

```
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
import pandas as pd
import json
import time
import yaml
```

[2]:

```
ss = SparkSession.Builder() \
     .appName("SparkBatchStreamingKafka") \
     .master("spark://pre-batch-processing-spark-master:7077") \
     .config("spark.jars", "./jars/spark-streaming-kafka-0-8-assembly_2.11-2.4.
 ↪1.jar") \
     .config("spark.sql.warehouse.dir", "hdfs://namenode:9000/") \
     .getOrCreate()
```

[3]:

```
sc = ss.sparkContext
ssc = StreamingContext(sc, 5)
ss.sparkContext.setLogLevel('WARN')
```

[4]:

```
def handle_rdd1(rdd):
    if not rdd.isEmpty():
        global ss
        print(f"Recieved {len(rdd.collect())} records - transform 1")
        df = ss.createDataFrame(
            rdd,
            schema=[
                'ID',
                'ArrivalTime',
                'BusinessLeisure',
                'CabinCategory',
                'CreationDate',
                'CurrencyCode',
                'DepartureTime',
                'Destination',
                'OfficeIdCountry',
```

```
                'Origin',
                'TotalAmount',
                'nPAX',
                'Record'
            ])
        df.write.parquet(path='hdfs://namenode:9000/trips/trips.parquet',␣
 ↪mode='append')
def handle_rdd2(rdd):
    if not rdd.isEmpty():
        global ss
        print(f"Recieved {len(rdd.collect())} records - transform 2")
        df = ss.createDataFrame(
            rdd,
            schema=[
                'ID',
                'ArrivalTime',
                'BusinessLeisure',
                'CabinCategory',
                'CreationDate',
                'CurrencyCode',
                'DepartureTime',
                'Destination',
                'OfficeIdCountry',
                'Origin',
                'TotalAmount',
                'nPAX',
                'Record'
            ])
        df.write.parquet(path='hdfs://namenode:9000/trips/processed_trips.
 ↪parquet', mode='append')
```

[5]:

```
def read_yaml(filename: str):
    with open(filename, 'r') as stream:
        try:
            return yaml.safe_load(stream)
        except yaml.YAMLError as exc:
            print(exc)
```

[6]:

```
mapping_and_statistic = read_yaml('mapping_and_statistic.yml')
```

[7]:

```
def json_to_list(s):
    t = json.loads(s)
    results = []
    for k, v in t.items():
        results.append(v)
    results.append(s)
    return results
```

```
def get_continous(x, m):
    if str(x) == 'nan':
        return 0.0
    else:
        x = float(x)
        return (x - m['statistic']['mean']) / m['statistic']['std']

def get_categorical(x, m):
    if str(x) == 'nan':
        return 0.0
    else:
        v = m['mapping'][str(x)]
        return (v - m['statistic']['mean']) / m['statistic']['std']

def json_to_processed_data(s):
    t = json.loads(s)
    return [
        t['ID'],
        get_continous(t['ArrivalTime'], mapping_and_statistic['ArrivalTime']),
        get_categorical(t['BusinessLeisure'],␣
 ↪mapping_and_statistic['BusinessLeisure']),
        get_categorical(t['CabinCategory'],␣
 ↪mapping_and_statistic['CabinCategory']),
        get_continous(t['CreationDate'], mapping_and_statistic['CreationDate']),
        get_categorical(t['CurrencyCode'],␣
 ↪mapping_and_statistic['CurrencyCode']),
        get_continous(t['DepartureTime'],␣
 ↪mapping_and_statistic['DepartureTime']),
        get_categorical(t['Destination'], mapping_and_statistic['Destination']),
        get_categorical(t['OfficeIdCountry'],␣
 ↪mapping_and_statistic['OfficeIdCountry']),
        get_categorical(t['Origin'], mapping_and_statistic['Origin']),
        get_continous(t['TotalAmount'], mapping_and_statistic['TotalAmount']),
        get_continous(t['nPAX'], mapping_and_statistic['nPAX']),
        s
    ]
```

[ ]:

```
ks = KafkaUtils.createDirectStream(
    ssc, ['trips'], {'metadata.broker.list': 'kafka-broker-1:
 ↪9093,kafka-broker-2:9093'})
lines = ks.map(lambda x: x[1])

transform1 = lines.map(lambda tripInfo: json_to_list(tripInfo))
transform1.foreachRDD(handle_rdd1)
```

```
transform2 = lines.map(lambda tripInfo: json_to_processed_data(tripInfo))
transform2.foreachRDD(handle_rdd2)

ssc.start()
ssc.awaitTermination()
```

```
Recieved 41 records - transform 1
Recieved 41 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
```

```
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 50 records - transform 1
Recieved 50 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
Recieved 49 records - transform 1
Recieved 49 records - transform 2
```

[ ]:

```
```