Assignment 9.1

February 28, 2022

0.1 Assignment 9.1

```python
[1]: import os
     import shutil
     import json
     from pathlib import Path

     import pandas as pd

     from kafka import KafkaProducer, KafkaAdminClient
     from kafka.admin.new_topic import NewTopic
     from kafka.errors import TopicAlreadyExistsError

     from pyspark.sql import SparkSession
     from pyspark.streaming import StreamingContext
     from pyspark import SparkConf
     from pyspark.sql.functions import window, from_json, col
     from pyspark.sql.types import StringType, TimestampType, DoubleType,␣
      ↪StructField, StructType
     from pyspark.sql.functions import udf

     current_dir = Path(os.getcwd()).absolute()
     checkpoint_dir = current_dir.joinpath('checkpoints')
     locations_checkpoint_dir = checkpoint_dir.joinpath('locations')
     accelerations_checkpoint_dir = checkpoint_dir.joinpath('accelerations')

     if locations_checkpoint_dir.exists():
         shutil.rmtree(locations_checkpoint_dir)

     if accelerations_checkpoint_dir.exists():
         shutil.rmtree(accelerations_checkpoint_dir)

     locations_checkpoint_dir.mkdir(parents=True, exist_ok=True)
     accelerations_checkpoint_dir.mkdir(parents=True, exist_ok=True)
```

0.1.1 Configuration Parameters

TODO: Change the configuration prameters to the appropriate values for your setup.

```
[2]: config = dict(
         bootstrap_servers=['kafka.kafka.svc.cluster.local:9092'],
         first_name='Bilal',
         last_name='Kudaimi'
     )

     config['client_id'] = '{}{}'.format(
         config['last_name'],
         config['first_name']
     )
     config['topic_prefix'] = '{}{}'.format(
         config['last_name'],
         config['first_name']
     )

     config['locations_topic'] = '{}-locations'.format(config['topic_prefix'])
     config['accelerations_topic'] = '{}-accelerations'.
      →format(config['topic_prefix'])
     config['simple_topic'] = '{}-simple'.format(config['topic_prefix'])

     config
```

```
[2]: {'bootstrap_servers': ['kafka.kafka.svc.cluster.local:9092'],
      'first_name': 'Bilal',
      'last_name': 'Kudaimi',
      'client_id': 'KudaimiBilal',
      'topic_prefix': 'KudaimiBilal',
      'locations_topic': 'KudaimiBilal-locations',
      'accelerations_topic': 'KudaimiBilal-accelerations',
      'simple_topic': 'KudaimiBilal-simple'}
```

0.1.2 Create Topic Utility Function

The create_kafka_topic helps create a Kafka topic based on your configuration settings. For instance, if your first name is *John* and your last name is *Doe*, create_kafka_topic('locations') will create a topic with the name DoeJohn-locations. The function will not create the topic if it already exists.

```
[3]: def create_kafka_topic(topic_name, config=config, num_partitions=1,␣
      →replication_factor=1):
         bootstrap_servers = config['bootstrap_servers']
         client_id = config['client_id']
         topic_prefix = config['topic_prefix']
         name = '{}-{}'.format(topic_prefix, topic_name)

         admin_client = KafkaAdminClient(
             bootstrap_servers=bootstrap_servers,
```

```
            client_id=client_id
        )

        topic = NewTopic(
            name=name,
            num_partitions=num_partitions,
            replication_factor=replication_factor
        )

        topic_list = [topic]
        try:
            admin_client.create_topics(new_topics=topic_list)
            print('Created topic "{}"'.format(name))
        except TopicAlreadyExistsError as e:
            print('Topic "{}" already exists'.format(name))

create_kafka_topic('simple')
```

```
Created topic "KudaimiBilal-simple"
```

[4]:
```
spark = SparkSession\
    .builder\
    .appName("Assignment09")\
    .getOrCreate()

df_locations = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "kafka.kafka.svc.cluster.local:9092") \
    .option("subscribe", config['locations_topic']) \
    .load()
```

TODO: Create a data frame called `df_accelerations` that reads from the accelerations topic you published to in assignment 8. In order to read data from this topic, make sure that you are running the notebook you created in assignment 8 that publishes acceleration and location data to the `LastnameFirstname-simple` topic.

[5]:
```
df_accelerations = ''
```

TODO: Create two streaming queries, `ds_locations` and `ds_accelerations` that publish to the `LastnameFirstname-simple` topic. See http://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#starting-streaming-queries and http://spark.apache.org/docs/latest/structured-streaming-kafka-integration.html for more information.

[6]:
```
ds_locations = ''

ds_accelerations = ''
```

3

```
try:
    ds_locations.awaitTermination()
    ds_accelerations.awaitTermination()
except KeyboardInterrupt:
    print("STOPPING STREAMING DATA")
```

                                    ␣
↪--

        AttributeError   Traceback (most recent call␣
↪last)

        <ipython-input-6-e547d4e2a8a7> in <module>
          4
          5 try:
    ----> 6   ds_locations.awaitTermination()
          7   ds_accelerations.awaitTermination()
          8 except KeyboardInterrupt:

        AttributeError: 'str' object has no attribute 'awaitTermination'