Machine Learning on streaming data using Kafka

```
!pip install tensorflow-io==0.17.0
!pip install tensorflow==2.4.0
!pip install kafka-python
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publ
Collecting tensorflow-io==0.17.0
  Downloading tensorflow_io-0.17.0-cp37-cp37m-manylinux2010_x86_64.whl (25.3 MB)
     |████████████████████████████████| 25.3 MB 1.5 MB/s
Collecting tensorflow<2.5.0,>=2.4.0
  Downloading tensorflow-2.4.4-cp37-cp37m-manylinux2010_x86_64.whl (394.5 MB)
     |████████████████████████████████| 394.5 MB 40 kB/s
Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist-packages (fr
Collecting wrapt~=1.12.1
  Downloading wrapt-1.12.1.tar.gz (27 kB)
Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7/dist-packag
Collecting absl-py~=0.10
  Downloading absl_py-0.15.0-py3-none-any.whl (132 kB)
     |████████████████████████████████| 132 kB 61.7 MB/s
Collecting h5py~=2.10.0
  Downloading h5py-2.10.0-cp37-cp37m-manylinux1_x86_64.whl (2.9 MB)
     |████████████████████████████████| 2.9 MB 52.0 MB/s
Collecting typing-extensions~=3.7.4
  Downloading typing_extensions-3.7.4.3-py3-none-any.whl (22 kB)
Collecting termcolor~=1.1.0
  Downloading termcolor-1.1.0.tar.gz (3.9 kB)
Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist-packages (fr
Collecting numpy~=1.19.2
  Downloading numpy-1.19.5-cp37-cp37m-manylinux2010_x86_64.whl (14.8 MB)
     |████████████████████████████████| 14.8 MB 53.5 MB/s
Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: tensorboard~=2.4 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python3.7/di
Collecting tensorflow-estimator<2.5.0,>=2.4.0
  Downloading tensorflow_estimator-2.4.0-py2.py3-none-any.whl (462 kB)
     |████████████████████████████████| 462 kB 54.3 MB/s
Collecting gast==0.3.3
  Downloading gast-0.3.3-py2.py3-none-any.whl (9.7 kB)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages
Collecting grpcio~=1.32.0
  Downloading grpcio-1.32.0-cp37-cp37m-manylinux2014_x86_64.whl (3.8 MB)
     |████████████████████████████████| 3.8 MB 54.7 MB/s
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/p
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (fron
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lil
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages
Building wheels for collected packages: termcolor, wrapt
  Building wheel for termcolor (setup.py) ... done
  Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4848 sha25
  Stored in directory: /root/.cache/pip/wheels/3f/e3/ec/8a8336ff196023622fbcb36de0c5a5c
  Building wheel for wrapt (setup.py) ... done
  Created wheel for wrapt: filename=wrapt-1.12.1-cp37-cp37m-linux_x86_64.whl size=68715
  Stored in directory: /root/.cache/pip/wheels/62/76/4c/aa25851149f3f6d9785f6c869387ad8
Successfully built termcolor wrapt
Installing collected packages: typing-extensions, numpy, grpcio, absl-py, wrapt, termcol
  Attempting uninstall: typing-extensions
    Found existing installation: typing-extensions 4.1.1
    Uninstalling typing-extensions-4.1.1:
      Successfully uninstalled typing-extensions-4.1.1
  Attempting uninstall: numpy
    Found existing installation: numpy 1.21.6
    Uninstalling numpy-1.21.6:
      Successfully uninstalled numpy-1.21.6
  Attempting uninstall: grpcio
    Found existing installation: grpcio 1.49.1
    Uninstalling grpcio-1.49.1:
      Successfully uninstalled grpcio-1.49.1
  Attempting uninstall: absl-py
    Found existing installation: absl-py 1.3.0
    Uninstalling absl-py-1.3.0:
      Successfully uninstalled absl-py-1.3.0
  Attempting uninstall: wrapt
    Found existing installation: wrapt 1.14.1
    Uninstalling wrapt-1.14.1:
      Successfully uninstalled wrapt-1.14.1
  Attempting uninstall: termcolor
    Found existing installation: termcolor 2.0.1
    Uninstalling termcolor-2.0.1:
      Successfully uninstalled termcolor-2.0.1
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.9.0
    Uninstalling tensorflow-estimator-2.9.0:
      Successfully uninstalled tensorflow-estimator-2.9.0
  Attempting uninstall: h5py
    Found existing installation: h5py 3.1.0
    Uninstalling h5py-3.1.0:
      Successfully uninstalled h5py-3.1.0
  Attempting uninstall: gast
    Found existing installation: gast 0.4.0
    Uninstalling gast-0.4.0:
      Successfully uninstalled gast-0.4.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.9.2
    Uninstalling tensorflow-2.9.2:
      Successfully uninstalled tensorflow-2.9.2
ERROR: pip's dependency resolver does not currently take into account all the packages 
xarray-einstats 0.2.2 requires numpy>=1.21, but you have numpy 1.19.5 which is incompat
jaxlib 0.3.22+cuda11.cudnn805 requires numpy>=1.20, but you have numpy 1.19.5 which is 
jax 0.3.23 requires numpy>=1.20, but you have numpy 1.19.5 which is incompatible.
cmdstanpy 1.0.7 requires numpy>=1.21, but you have numpy 1.19.5 which is incompatible
```

```
    cmdstanpy 1.0.7 requires numpy>=1.21, but you have numpy 1.19.5 which is incompatible.
    Successfully installed absl-py-0.15.0 gast-0.3.3 grpcio-1.32.0 h5py-2.10.0 numpy-1.19.5
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publ
    Collecting tensorflow==2.4.0
      Downloading tensorflow-2.4.0-cp37-cp37m-manylinux2010_x86_64.whl (394.7 MB)
         |████████████████████████████████| 394.7 MB 18 kB/s
    Requirement already satisfied: tensorboard~=2.4 in /usr/local/lib/python3.7/dist-package
    Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist-packages (fr
    Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: h5py~=2.10.0 in /usr/local/lib/python3.7/dist-packages (f
    Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: gast==0.3.3 in /usr/local/lib/python3.7/dist-packages (fr
```

```
!tar -xzf kafka_2.13-2.7.2.tgz
```

```
    Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7/dist-packag
```

```
!./kafka_2.13-2.7.2/bin/zookeeper-server-start.sh -daemon ./kafka_2.13-2.7.2/config/zookeeper
!./kafka_2.13-2.7.2/bin/kafka-server-start.sh -daemon ./kafka_2.13-2.7.2/config/server.proper
!echo "Waiting for 10 secs until kafka and zookeeper services are up and running"
!sleep 10
```

```
    Waiting for 10 secs until kafka and zookeeper services are up and running
    Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7/dist-packa
```

```
!./kafka_2.13-2.7.2/bin/kafka-topics.sh --create --bootstrap-server 127.0.0.1:9092 --replicat
!./kafka_2.13-2.7.2/bin/kafka-topics.sh --create --bootstrap-server 127.0.0.1:9092 --replicat
```

```
    Error while executing topic command : Topic 'newdata-train' already exists.
    [2022-10-24 09:55:40,673] ERROR org.apache.kafka.common.errors.TopicExistsException: Top
     (kafka.admin.TopicCommand$)
    Error while executing topic command : Topic 'newdata-test' already exists.
    [2022-10-24 09:55:43,795] ERROR org.apache.kafka.common.errors.TopicExistsException: Top
     (kafka.admin.TopicCommand$)
```

```
    Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-
```

```
!./kafka_2.13-2.7.2/bin/kafka-topics.sh --describe --bootstrap-server 127.0.0.1:9092 --topic
!./kafka_2.13-2.7.2/bin/kafka-topics.sh --describe --bootstrap-server 127.0.0.1:9092 --topic
```

```
    Topic: newdata-train    PartitionCount: 1       ReplicationFactor: 1    Configs: segment
            Topic: newdata-train    Partition: 0    Leader: 0       Replicas: 0     Isr: 0
    Topic: newdata-test     PartitionCount: 2       ReplicationFactor: 1    Configs: segment
            Topic: newdata-test     Partition: 0    Leader: 0       Replicas: 0     Isr: 0
            Topic: newdata-test     Partition: 1    Leader: 0       Replicas: 0     Isr: 0
```

```
          Successfully uninstalled tensorflow-2.4.4
```

```
import os
from datetime import datetime
import time
import threading
import json
from kafka import KafkaProducer
from kafka.errors import KafkaError
from sklearn.model_selection import train_test_split
import pandas as pd
import tensorflow as tf
import tensorflow_io as tfio


print("tensorflow-io version: {}".format(tfio.__version__))
print("tensorflow version: {}".format(tf.__version__))
```

```
     tensorflow-io version: 0.17.0
     tensorflow version: 2.4.0
```

```
COLUMNS = [
           'sex',
           'age',

           'type',

           'induration_diameter',
           'treatment'

           ]
```

```
newdata_iterator = pd.read_csv('ml.csv', header=None, names=COLUMNS, chunksize=100000)
newdata_df = next(newdata_iterator)
newdata_df.head()
```

	sex	age	type	induration_diameter	treatment
0	1	34	34	34	1
1	1	32	4	32	1
2	1	12	2	12	1
3	2	11	66	11	0
4	2	12	3	12	0

```
# Number of datapoints and columns
len(newdata_df), len(newdata_df.columns)
```

```
     (96, 5)
```

```
# Number of datapoints belonging to each class (0: background noise, 1: signal)
len(newdata_df[newdata_df["treatment"]==0]), len(newdata_df[newdata_df["treatment"]==1])
```

```
     (48, 48)
```

```
# Split the dataset

train_df, test_df = train_test_split(newdata_df, test_size=0.4, shuffle=True)
print("Number of training samples: ",len(train_df))
print("Number of testing sample: ",len(test_df))

x_train_df = train_df.drop(["treatment"], axis=1)
y_train_df = train_df["treatment"]

x_test_df = test_df.drop(["treatment"], axis=1)
y_test_df = test_df["treatment"]

# The labels are set as the kafka message keys so as to store data
# in multiple-partitions. Thus, enabling efficient data retrieval
# using the consumer groups.
x_train = list(filter(None, x_train_df.to_csv(index=False).split("\n")[1:]))
print(x_train)
y_train = list(filter(None, y_train_df.to_csv(index=False).split("\n")[1:]))

x_test = list(filter(None, x_test_df.to_csv(index=False).split("\n")[1:]))
y_test = list(filter(None, y_test_df.to_csv(index=False).split("\n")[1:]))
```

```
Number of training samples:  57
Number of testing sample:  39
['1,34,34,34', '1,12,2,12', '2,13,2,13', '2,11,66,11', '1,32,4,32', '2,12,3,12', '2,12,3
```

```
NUM_COLUMNS = len(x_train_df.columns)
len(x_train), len(y_train), len(x_test), len(y_test)
```

```
(57, 57, 39, 39)
```

```
# Store the train and test data in kafka

def error_callback(exc):
    raise Exception('Error while sendig data to kafka: {0}'.format(str(exc)))

def write_to_kafka(topic_name, items):
  count=0
  producer = KafkaProducer(bootstrap_servers=['127.0.0.1:9092'])
  for message, key in items:
    producer.send(topic_name, key=key.encode('utf-8'), value=message.encode('utf-8')).add_err
    count+=1
  producer.flush()
  print("Wrote {0} messages into topic: {1}".format(count, topic_name))

write_to_kafka("newdata-train", zip(x_train, y_train))
write_to_kafka("newdata-test", zip(x_test, y_test))
```

```
Wrote 57 messages into topic: newdata-train
Wrote 39 messages into topic: newdata-test
```

```
def decode_kafka_item(item):
  message = tf.io.decode_csv(item.message, [[0.0] for i in range(NUM_COLUMNS)])
  key = tf.strings.to_number(item.key)
  return (message, key)

BATCH_SIZE=64
SHUFFLE_BUFFER_SIZE=64
train_ds = tfio.IODataset.from_kafka('newdata-train', partition=0, offset=0)
train_ds = train_ds.shuffle(buffer_size=SHUFFLE_BUFFER_SIZE)
train_ds = train_ds.map(decode_kafka_item)
train_ds = train_ds.batch(BATCH_SIZE)
```

```
# Set the parameters

OPTIMIZER="adam"
LOSS=tf.keras.losses.BinaryCrossentropy(from_logits=True)
METRICS=['accuracy']
EPOCHS=10
```

```
# design/build the model
print(NUM_COLUMNS)
model = tf.keras.Sequential([
  tf.keras.layers.Input(shape=(NUM_COLUMNS,)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(256, activation='relu'),
  tf.keras.layers.Dropout(0.4),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.4),
  tf.keras.layers.Dense(1, activation='sigmoid')
])

print(model.summary())
```

```
4
Model: "sequential_8"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
dense_32 (Dense)            (None, 128)               640
_________________________________________________________________
dropout_24 (Dropout)        (None, 128)               0
_________________________________________________________________
dense_33 (Dense)            (None, 256)               33024
_________________________________________________________________
dropout_25 (Dropout)        (None, 256)               0
_________________________________________________________________
dense_34 (Dense)            (None, 128)               32896
```

```
_________________________________________________________________
dropout_26 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_35 (Dense)             (None, 1)                 129
=================================================================
Total params: 66,689
Trainable params: 66,689
Non-trainable params: 0
_________________________________________________________________
None
```

```
# compile the model
model.compile(optimizer=OPTIMIZER, loss=LOSS, metrics=METRICS)
```

```
print(train_ds)

# fit the model
model.fit(train_ds, epochs=EPOCHS)
```

```
<BatchDataset shapes: ((None, 4), (None,)), types: (tf.float32, tf.float32)>
Epoch 1/10
1/1 [==============================] - 1s 1s/step - loss: 0.4359 - accuracy: 0.6667
Epoch 2/10
1/1 [==============================] - 0s 475ms/step - loss: 0.8530 - accuracy: 0.6667
Epoch 3/10
1/1 [==============================] - 0s 473ms/step - loss: 5.0097 - accuracy: 0.6667
Epoch 4/10
1/1 [==============================] - 0s 482ms/step - loss: 1.6433 - accuracy: 0.6667
Epoch 5/10
1/1 [==============================] - 0s 483ms/step - loss: 1.8732 - accuracy: 0.3333
Epoch 6/10
1/1 [==============================] - 0s 481ms/step - loss: 0.6113 - accuracy: 0.6667
Epoch 7/10
1/1 [==============================] - 0s 477ms/step - loss: 1.2339 - accuracy: 0.3333
Epoch 8/10
1/1 [==============================] - 0s 482ms/step - loss: 0.4002 - accuracy: 0.6667
Epoch 9/10
1/1 [==============================] - 0s 478ms/step - loss: 0.3881 - accuracy: 0.6667
Epoch 10/10
1/1 [==============================] - 0s 483ms/step - loss: 0.0636 - accuracy: 1.0000
<tensorflow.python.keras.callbacks.History at 0x7fa4af755690>
```

```
test_ds = tfio.experimental.streaming.KafkaGroupIODataset(
    topics=["newdata-test"],
    group_id="testcg",
    servers="127.0.0.1:9092",
    stream_timeout=10000,
    configuration=[
        "session.timeout.ms=7000",
        "max.poll.interval.ms=8000",
        "auto.offset.reset=earliest"
    ],
)

def decode_kafka_test_item(raw_message, raw_key):
  message = tf.io.decode_csv(raw_message, [[0.0] for i in range(NUM_COLUMNS)])
  key = tf.strings.to_number(raw_key)
  return (message, key)

test_ds = test_ds.map(decode_kafka_test_item)
test_ds = test_ds.batch(BATCH_SIZE)


res = model.evaluate(test_ds)
print("test loss, test acc:", res)
```

```
1/1 [==============================] - 11s 11s/step - loss: 1.6361 - accuracy: 0.8056
test loss, test acc: [1.6361463069915771, 0.8055555820465088]
```

```
!./kafka_2.13-2.7.2/bin/kafka-consumer-groups.sh --bootstrap-server 127.0.0.1:9092 --describe
```

```
GROUP           TOPIC           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG
testcg          newdata-test    0          8               8               0
testcg          newdata-test    1          31              31              0
```

```
online_train_ds = tfio.experimental.streaming.KafkaBatchIODataset(
    topics=["newdata-train"],
    group_id="cgonline",
    servers="127.0.0.1:9092",
    stream_timeout=10000, # in milliseconds, to block indefinitely, set it to -1.
    configuration=[
        "session.timeout.ms=7000",
        "max.poll.interval.ms=8000",
        "auto.offset.reset=earliest"
    ],
)


def decode_kafka_online_item(raw_message, raw_key):
  message = tf.io.decode_csv(raw_message, [[0.0] for i in range(NUM_COLUMNS)])
  key = tf.strings.to_number(raw_key)
  return (message, key)

for mini_ds in online_train_ds:
  mini_ds = mini_ds.shuffle(buffer_size=32)
  mini_ds = mini_ds.map(decode_kafka_online_item)
  mini_ds = mini_ds.batch(32)
  if len(mini_ds) > 0:
    model.fit(mini_ds, epochs=3)
```

```
Epoch 1/3
2/2 [==============================] - 0s 8ms/step - loss: 1.8841 - accuracy: 0.7593
Epoch 2/3
2/2 [==============================] - 0s 8ms/step - loss: 3.4117 - accuracy: 0.6111
Epoch 3/3
2/2 [==============================] - 0s 8ms/step - loss: 1.4926 - accuracy: 0.6481
```

Task 1: Execute the above code properly with the given dataset.

Task 2: Make a report about,

-> detailed analysis of the code

-> How did you execute the task using Kafka, and why is Kafka important in this machine learning model?

Task 3: Feed a new dataset into Kafka. Utilizing the dataset, train and test your choice of machine learning model and solve any issues that may arise in the code