Homework 5 (Streaming)

The aim of this homework is to get familiar with **Spark Structured Streaming** and obtain knowledge of processing real time data

The first part is to prepare streaming data for processing based on **Homework 4**.

Additionally to the existing data from Homework 4 create an additional Kafka topic with new events user events **buys**. The file **yoochoose-buys.dat** (CSV file with coma delimiter) comprising the clicks of the users over the items. Each record/line in the file has the following fields/format

* + Session ID - the id of the session. In one session there are one or many buying events.
  + Timestamp - the time when the buy occurred.
  + Item ID – the unique identifier of item.
  + Price – the price of the item.
  + Quantity – how many of this item were bought.

The Session ID in buys will always exist in the clicks data – the records with the same Session ID together form the sequence of click events of a certain user during the session. The session could be short (few minutes) or very long (few hours), it could have one click or hundreds of clicks. All depends on the activity of the user.

Please download sample data from here: <https://s3-eu-west-1.amazonaws.com/yc-rdata/yoochoose-data.7z>

Read data from **yoochoose-clicks.dat** and **yoochoose-buys.dat** files and store it to two different Kafka topics **clickstream** and **buystream** accordingly, using Session ID as the key and all other fields as value. Reorder the data using timestamp field so clicks and buys are added to the topics in chronological order.

The seconds part is to create a Spark Streaming application to consume and process the given data:

1. Read data from **clickstream** and **buystream** topics
2. Calculate average conversion (buys/clicks – per session) for each valid category within the next sliding intervals 1 day, 1 hour and for the whole data history at the moment of aggregation. Use [**Spark Structued Streaming API**](https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html)for data aggregation. Use **windowing**, **watermarking**, **joining** operations.
3. Pick top N categories with the highest conversion for all three intervals.
4. Report these values to a downstream service (Simulate REST API interaction or store value to a RDBMS instance, write your custom sink). Values have to be reported every minute.