How to Create Discretized Streams (Dstreams)

What are some sources of DStreams?

- *Basic sources*: Sources directly available in the StreamingContext API. Examples: file systems, and socket connections.
- *Advanced sources*: Sources like Kafka, Flume, Kinesis, etc. are available through extra utility classes. These require linking against extra dependencies as discussed in the linking section.

Example: Twitter as a DStream Source

- In our first example, we set up a twitter stream, and created the Spark socket stream with the following lines

```
sc = SparkContext()
ssc = StreamingContext(sc, 10)
socket_stream = ssc.socketTextStream("127.0.0.1", 5555)
```

Example: text files as a Stream

`textFileStream(`*`dataDirectory`*`)`

Creates an input stream from new text files that enter a specific directory.

```
def simple_text_to_stream(ssc):
        ssc.textFileStream('/data').pprint()
```

Parameters

- dataDirectory – filepath for a folder with new files being added after the start of the stream

Example: Queue of RDDs as a Stream

queueStream(*rdds, oneAtATime=True, default=None*)

Creates an input stream from an queue of RDDs or list. In each batch, it will process either one or all of the RDDs returned by the queue.

```
def simple_queue_one_at_a_time(ssc):
        ssc.queueStream([range(5), ['a', 'b'], ['c’]], oneAtATime=True).pprint()
```

Parameters

- rdds – Queue of RDDs
- oneAtATime – pick one rdd each time or pick all of them once.
- default – The default rdd if no more in rdds

To the Code