package com.twitter.timelines.data\_processing.ml\_util.aggregation\_framework.job

import com.twitter.ml.api.DataRecord

import com.twitter.summingbird.Counter

/\*\*

\* A summingbird Counter which is associated with a predicate which operates on

\* [[com.twitter.ml.api.DataRecord]] instances.

\*

\* For example, for a data record which represents a Tweet, one could define a predicate

\* which checks whether the Tweet contains a binary feature representing the presence of

\* an image. The counter can then be used to represent the the count of Tweets with

\* images processed.

\*

\* @param predicate a predicate which gates the counter

\* @param counter a summingbird Counter instance

\*/

case class DataRecordFeatureCounter(predicate: DataRecord => Boolean, counter: Counter)

object DataRecordFeatureCounter {

/\*\*

\* Increments the counter if the record satisfies the predicate

\*

\* @param recordCounter a data record counter

\* @param record a data record

\*/

def apply(recordCounter: DataRecordFeatureCounter, record: DataRecord): Unit =

if (recordCounter.predicate(record)) recordCounter.counter.incr()

/\*\*

\* Defines a feature counter with a predicate that is always true

\*

\* @param counter a summingbird Counter instance

\* @return a data record counter

\*/

def any(counter: Counter): DataRecordFeatureCounter =

DataRecordFeatureCounter({ \_: DataRecord => true }, counter)

}