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

import com.twitter.ml.api.\_

import com.twitter.ml.api.FeatureContext

import scala.collection.JavaConverters.\_

/\*

\* A really bad default merge policy that picks all the aggregate

\* features corresponding to the first sparse key value in the list.

\* Does not rename any of the aggregate features for simplicity.

\* Avoid using this merge policy if at all possible.

\*/

object PickFirstRecordPolicy extends SparseBinaryMergePolicy {

val dataRecordMerger: DataRecordMerger = new DataRecordMerger

override def mergeRecord(

mutableInputRecord: DataRecord,

aggregateRecords: List[DataRecord],

aggregateContext: FeatureContext

): Unit =

aggregateRecords.headOption

.foreach(aggregateRecord => dataRecordMerger.merge(mutableInputRecord, aggregateRecord))

override def aggregateFeaturesPostMerge(aggregateContext: FeatureContext): Set[Feature[\_]] =

aggregateContext.getAllFeatures.asScala.toSet

}