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

import com.twitter.timelines.data\_processing.ml\_util.aggregation\_framework.TypedAggregateGroup

import com.twitter.ml.api.Feature

trait OnlineAggregationConfigTrait {

def ProdAggregates: Set[TypedAggregateGroup[\_]]

def StagingAggregates: Set[TypedAggregateGroup[\_]]

def ProdCommonAggregates: Set[TypedAggregateGroup[\_]]

/\*\*

\* AggregateToCompute: This defines the complete set of aggregates to be

\* computed by the aggregation job and to be stored in memcache.

\*/

def AggregatesToCompute: Set[TypedAggregateGroup[\_]]

/\*\*

\* ProdFeatures: This defines the subset of aggregates to be extracted

\* and hydrated (or adapted) by callers to the aggregates features cache.

\* This should only contain production aggregates and aggregates on

\* product specific engagements.

\* ProdCommonFeatures: Similar to ProdFeatures but containing user-level

\* aggregate features. This is provided to PredictionService just

\* once per user.

\*/

lazy val ProdFeatures: Set[Feature[\_]] = ProdAggregates.flatMap(\_.allOutputFeatures)

lazy val ProdCommonFeatures: Set[Feature[\_]] = ProdCommonAggregates.flatMap(\_.allOutputFeatures)

}