package com.twitter.timelines.prediction.features.p\_home\_latest

import com.twitter.ml.api.Feature.{Continuous, Discrete}

import com.twitter.dal.personal\_data.thriftjava.PersonalDataType.\_

import scala.collection.JavaConverters.\_

object HomeLatestUserFeatures {

val LAST\_LOGIN\_TIMESTAMP\_MS =

new Discrete("home\_latest.user\_feature.last\_login\_timestamp\_ms", Set(PrivateTimestamp).asJava)

}

object HomeLatestUserAggregatesFeatures {

/\*\*

\* Used as `timestampFeature` in `OfflineAggregateSource` required by feature aggregations, set to

\* the `dateRange` end timestamp by default

\*/

val AGGREGATE\_TIMESTAMP\_MS =

new Discrete("home\_latest.user\_feature.aggregate\_timestamp\_ms", Set(PrivateTimestamp).asJava)

val HOME\_TOP\_IMPRESSIONS =

new Continuous("home\_latest.user\_feature.home\_top\_impressions", Set(CountOfImpression).asJava)

val HOME\_LATEST\_IMPRESSIONS =

new Continuous(

"home\_latest.user\_feature.home\_latest\_impressions",

Set(CountOfImpression).asJava)

val HOME\_TOP\_LAST\_LOGIN\_TIMESTAMP\_MS =

new Discrete(

"home\_latest.user\_feature.home\_top\_last\_login\_timestamp\_ms",

Set(PrivateTimestamp).asJava)

val HOME\_LATEST\_LAST\_LOGIN\_TIMESTAMP\_MS =

new Discrete(

"home\_latest.user\_feature.home\_latest\_last\_login\_timestamp\_ms",

Set(PrivateTimestamp).asJava)

val HOME\_LATEST\_MOST\_RECENT\_CLICK\_TIMESTAMP\_MS =

new Discrete(

"home\_latest.user\_feature.home\_latest\_most\_recent\_click\_timestamp\_ms",

Set(PrivateTimestamp).asJava)

}

case class HomeLatestUserFeatures(userId: Long, lastLoginTimestampMs: Long)

case class HomeLatestUserAggregatesFeatures(

userId: Long,

aggregateTimestampMs: Long,

homeTopImpressions: Option[Double],

homeLatestImpressions: Option[Double],

homeTopLastLoginTimestampMs: Option[Long],

homeLatestLastLoginTimestampMs: Option[Long],

homeLatestMostRecentClickTimestampMs: Option[Long])