package com.twitter.timelines.prediction.common.aggregates

import com.twitter.ml.api.constant.SharedFeatures.TIMESTAMP

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

import com.twitter.timelines.prediction.features.p\_home\_latest.HomeLatestUserAggregatesFeatures

import timelines.data\_processing.ad\_hoc.recap.data\_record\_preparation.RecapDataRecordsAggMinimalJavaDataset

/\*\*

\* Any update here should be in sync with [[TimelinesFeatureGroups]] and [[AggMinimalDataRecordGeneratorJob]].

\*/

object TimelinesAggregationSources {

/\*\*

\* This is the recap data records after post-processing in [[GenerateRecapAggMinimalDataRecordsJob]]

\*/

val timelinesDailyRecapMinimalSource = OfflineAggregateSource(

name = "timelines\_daily\_recap",

timestampFeature = TIMESTAMP,

dalDataSet = Some(RecapDataRecordsAggMinimalJavaDataset),

scaldingSuffixType = Some("dal"),

withValidation = true

)

val timelinesDailyTwitterWideSource = OfflineAggregateSource(

name = "timelines\_daily\_twitter\_wide",

timestampFeature = TIMESTAMP,

scaldingHdfsPath = Some("/user/timelines/processed/suggests/recap/twitter\_wide\_data\_records"),

scaldingSuffixType = Some("daily"),

withValidation = true

)

val timelinesDailyListTimelineSource = OfflineAggregateSource(

name = "timelines\_daily\_list\_timeline",

timestampFeature = TIMESTAMP,

scaldingHdfsPath = Some("/user/timelines/processed/suggests/recap/all\_features/list"),

scaldingSuffixType = Some("hourly"),

withValidation = true

)

val timelinesDailyHomeLatestSource = OfflineAggregateSource(

name = "timelines\_daily\_home\_latest",

timestampFeature = HomeLatestUserAggregatesFeatures.AGGREGATE\_TIMESTAMP\_MS,

scaldingHdfsPath = Some("/user/timelines/processed/p\_home\_latest/user\_aggregates"),

scaldingSuffixType = Some("daily")

)

}