package com.twitter.timelines.prediction.features.request\_context

import com.twitter.ml.api.FeatureContext

import com.twitter.ml.api.Feature.\_

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

import scala.collection.JavaConverters.\_

object RequestContextFeatures {

val COUNTRY\_CODE =

new Text("request\_context.country\_code", Set(PrivateCountryOrRegion, InferredCountry).asJava)

val LANGUAGE\_CODE = new Text(

"request\_context.language\_code",

Set(GeneralSettings, ProvidedLanguage, InferredLanguage).asJava)

val REQUEST\_PROVENANCE = new Text("request\_context.request\_provenance", Set(AppUsage).asJava)

val DISPLAY\_WIDTH = new Continuous("request\_context.display\_width", Set(OtherDeviceInfo).asJava)

val DISPLAY\_HEIGHT = new Continuous("request\_context.display\_height", Set(OtherDeviceInfo).asJava)

val DISPLAY\_DPI = new Continuous("request\_context.display\_dpi", Set(OtherDeviceInfo).asJava)

// the following features are not Continuous Features because for e.g. continuity between

// 23 and 0 hours cannot be handled that way. instead, we will treat each slice of hours/days

// independently, like a set of sparse binary features.

val TIMESTAMP\_GMT\_HOUR =

new Discrete("request\_context.timestamp\_gmt\_hour", Set(PrivateTimestamp).asJava)

val TIMESTAMP\_GMT\_DOW =

new Discrete("request\_context.timestamp\_gmt\_dow", Set(PrivateTimestamp).asJava)

val IS\_GET\_INITIAL = new Binary("request\_context.is\_get\_initial")

val IS\_GET\_MIDDLE = new Binary("request\_context.is\_get\_middle")

val IS\_GET\_NEWER = new Binary("request\_context.is\_get\_newer")

val IS\_GET\_OLDER = new Binary("request\_context.is\_get\_older")

// the following features are not Binary Features because the source field is Option[Boolean],

// and we want to distinguish Some(false) from None. None will be converted to -1.

val IS\_POLLING = new Discrete("request\_context.is\_polling")

val IS\_SESSION\_START = new Discrete("request\_context.is\_session\_start")

// Helps distinguish requests from "home" vs "home\_latest" (reverse chron home view).

val TIMELINE\_KIND = new Text("request\_context.timeline\_kind")

val featureContext = new FeatureContext(

COUNTRY\_CODE,

LANGUAGE\_CODE,

REQUEST\_PROVENANCE,

DISPLAY\_WIDTH,

DISPLAY\_HEIGHT,

DISPLAY\_DPI,

TIMESTAMP\_GMT\_HOUR,

TIMESTAMP\_GMT\_DOW,

IS\_GET\_INITIAL,

IS\_GET\_MIDDLE,

IS\_GET\_NEWER,

IS\_GET\_OLDER,

IS\_POLLING,

IS\_SESSION\_START,

TIMELINE\_KIND

)

}