package com.twitter.product\_mixer.core.functional\_component.scorer

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.BaseBulkCandidateFeatureHydrator

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

import com.twitter.product\_mixer.core.model.common.SupportsConditionally

import com.twitter.product\_mixer.core.model.common.UniversalNoun

import com.twitter.product\_mixer.core.model.common.identifier.ScorerIdentifier

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.stitch.Stitch

/\*\* Scores the provided `candidates` \*/

trait Scorer[-Query <: PipelineQuery, -Candidate <: UniversalNoun[Any]]

extends BaseBulkCandidateFeatureHydrator[Query, Candidate, Feature[\_, \_]]

with SupportsConditionally[Query] {

/\*\* @see [[ScorerIdentifier]] \*/

override val identifier: ScorerIdentifier

/\*\*

\* Features returned by the Scorer

\*/

def features: Set[Feature[\_, \_]]

/\*\*

\* Scores the provided `candidates`

\*

\* @note the returned Seq of [[FeatureMap]] must contain all the input 'candidates'

\* and be in the same order as the input 'candidates'

\*\*/

def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[Candidate]]

): Stitch[Seq[FeatureMap]]

}