package com.twitter.search.common.util.ml.prediction\_engine;

import com.google.common.base.Preconditions;

import com.twitter.ml.api.Feature;

/\*\*

\* Score accumulator for legacy (non-schema-based) features. It provides methods to add features

\* using Feature objects.

\*

\* @deprecated This class is retired and we suggest to switch to schema-based features.

\*/

@Deprecated

public abstract class BaseLegacyScoreAccumulator<D> extends BaseScoreAccumulator<D> {

public BaseLegacyScoreAccumulator(LightweightLinearModel model) {

super(model);

Preconditions.checkState(!model.isSchemaBased(),

"Cannot create LegacyScoreAccumulator with a schema-based model: %s", model.getName());

}

/\*\*

\* Add to the score the weight of a binary feature (if it's present).

\*

\* @deprecated This function is retired and we suggest to switch to addSchemaBooleanFeatures in

\* SchemaBasedScoreAccumulator.

\*/

@Deprecated

protected BaseLegacyScoreAccumulator addBinaryFeature(Feature<Boolean> feature,

boolean value) {

if (value) {

Double weight = model.binaryFeatures.get(feature);

if (weight != null) {

score += weight;

}

}

return this;

}

/\*\*

\* Add to the score the weight of a continuous feature.

\* <p>

\* If the model uses real valued features, it multiplies its weight by the provided value.

\* Otherwise, it tries to find the discretized feature and adds its weight to the score.

\*

\* @deprecated This function is retired and we suggest to switch to addSchemaContinuousFeatures in

\* SchemaBasedScoreAccumulator.

\*/

@Deprecated

protected BaseLegacyScoreAccumulator addContinuousFeature(Feature<Double> feature,

double value) {

Double weightFromContinuous = model.continuousFeatures.get(feature);

if (weightFromContinuous != null) {

score += weightFromContinuous \* value;

} else {

DiscretizedFeature discretizedFeature = model.discretizedFeatures.get(feature);

if (discretizedFeature != null) {

// Use only the weight of the discretized feature (there's no need to multiply it)

score += discretizedFeature.getWeight(value);

}

}

return this;

}

}