package com.twitter.search.common.encoding.features;

import com.google.common.base.Preconditions;

/\*\*

\* A byte normalizer that restricts the values to the given range before normalizing them.

\*/

public class ClampByteNormalizer extends ByteNormalizer {

private final int minUnnormalizedValue;

private final int maxUnnormalizedValue;

/\*\*

\* Creates a new ClampByteNormalizer instance.

\*

\* @param minValue The smallest allowed unnormalized value.

\* @param maxValue The largest allowed unnormalized value.

\*/

public ClampByteNormalizer(int minUnnormalizedValue, int maxUnnormalizedValue) {

Preconditions.checkState(minUnnormalizedValue <= maxUnnormalizedValue);

Preconditions.checkState(minUnnormalizedValue >= 0);

Preconditions.checkState(maxUnnormalizedValue <= 255);

this.minUnnormalizedValue = minUnnormalizedValue;

this.maxUnnormalizedValue = maxUnnormalizedValue;

}

@Override

public byte normalize(double val) {

int adjustedValue = (int) val;

if (adjustedValue < minUnnormalizedValue) {

adjustedValue = minUnnormalizedValue;

}

if (adjustedValue > maxUnnormalizedValue) {

adjustedValue = maxUnnormalizedValue;

}

return ByteNormalizer.intToUnsignedByte(adjustedValue);

}

@Override

public double unnormLowerBound(byte norm) {

return ByteNormalizer.unsignedByteToInt(norm);

}

@Override

public double unnormUpperBound(byte norm) {

return ByteNormalizer.unsignedByteToInt(norm) + 1;

}

}