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

import com.google.common.base.Preconditions;

/\*\*

\* Normalizes values as follows:

\* Positive numbers normalize to (1 + round(log\_baseN(value))).

\* Negative numbers throw.

\* 0 will normalize to 0.

\* The log base is 2 by default.

\*/

public class LogByteNormalizer extends ByteNormalizer {

private static final double DEFAULT\_BASE = 2;

private final double base;

private final double logBase;

public LogByteNormalizer(double base) {

Preconditions.checkArgument(base > 0);

this.base = base;

logBase = Math.log(base);

}

public LogByteNormalizer() {

this(DEFAULT\_BASE);

}

@Override

public byte normalize(double val) {

if (val < 0) {

throw new IllegalArgumentException("Can't log-normalize negative value " + val);

} else if (val == 0) {

return 0;

} else {

long logVal = 1 + (long) Math.floor(Math.log(val) / logBase);

return logVal > Byte.MAX\_VALUE ? Byte.MAX\_VALUE : (byte) logVal;

}

}

@Override

public double unnormLowerBound(byte norm) {

return norm < 0

? Double.NEGATIVE\_INFINITY

: Math.floor(Math.pow(base, norm - 1));

}

@Override

public double unnormUpperBound(byte norm) {

return norm == Byte.MAX\_VALUE

? Double.POSITIVE\_INFINITY

: Math.floor(Math.pow(base, norm));

}

}