#pragma once

#ifdef \_\_cplusplus

#include <twml/DataRecord.h>

#include <twml/HashedDataRecord.h>

#include <twml/Tensor.h>

namespace twml {

template<class RecordType>

class GenericBatchPredictionRequest {

static\_assert(std::is\_same<RecordType, HashedDataRecord>::value ||

std::is\_same<RecordType, DataRecord>::value,

"RecordType has to be HashedDatarecord or DataRecord");

public:

typedef typename RecordType::Reader Reader;

GenericBatchPredictionRequest(int numOfLabels=0, int numOfWeights=0):

m\_common\_features(), m\_requests(),

num\_labels(numOfLabels), num\_weights(numOfWeights)

{}

void decode(Reader &reader);

std::vector<RecordType>& requests() {

return m\_requests;

}

RecordType& common() {

return m\_common\_features;

}

private:

RecordType m\_common\_features;

std::vector<RecordType> m\_requests;

int num\_labels;

int num\_weights;

};

using HashedBatchPredictionRequest = GenericBatchPredictionRequest<HashedDataRecord>;

using BatchPredictionRequest = GenericBatchPredictionRequest<DataRecord>;

}

#endif