Diabetes Prediction Software - Test Cases with Results

PASS CASES

Test Name: Record::getFeatures - normal Test Description: Test getFeatures() on Record with features [120.0, 30.0, 45.0].
Pass / Fail: Pass
Comments: Returned correct features list.

Test Name: Record::getFeatures - empty Test Description: Test getFeatures() on Record with empty features list.
Pass / Fail: Pass
Comments: Returned empty list as expected.

Test Name: Record::getLabel - true label Test Description: Test getLabel() on Record with label true.
Pass / Fail: Pass
Comments: Returned true as expected.

Test Name: Record::getLabel - false label Test Description: Test getLabel() on Record with label false.
Pass / Fail: Pass
Comments: Returned false as expected.

Test Name: RecordCollector::loadFromCSV - valid file Test Description: Load two records from a valid CSV and check retrieval.
Pass / Fail: Pass
Comments: Records loaded correctly with normalized features.

Test Name: RecordCollector::loadFromCSV - non-existent file Test Description: Attempt to load from a non-existent CSV file.
Pass / Fail: Pass
Comments: Correctly returned false and kept records empty.

Test Name: RecordCollector::handleMissingData - partial missing Test Description: Fill missing feature using average from another record.
Pass / Fail: Pass
Comments: Missing data filled with mean value (normalized to 0.0).

Test Name: RecordCollector::handleMissingData - all missing Test Description: Handle case where all features are missing in a record.
Pass / Fail: Pass
Comments: All missing features set to 0.0 and normalized correctly.

Test Name: RecordCollector::normalizeInput - normal Test Description: Normalize an input after loading a dataset.
Pass / Fail: Pass
Comments: Input normalized within expected z-score range.

Test Name: Node::split - normal Test Description: Split records at a node based on a threshold.
Pass / Fail: Pass
Comments: Correctly split records into left and right children.

Test Name: Node::split - all one side Test Description: Split records where all go to one side of the threshold.
Pass / Fail: Pass
Comments: Left child null, right child contains all records.

Test Name: Node::split - empty records Test Description: Split an empty list of records at a node.
Pass / Fail: Pass
Comments: Node became a leaf with default label false.

Test Name: Tree::build - normal Test Description: Build a tree with multiple records and verify splits.
Pass / Fail: Pass
Comments: Root node created with appropriate splits.

Test Name: Tree::build - empty records Test Description: Build a tree with an empty record list.
Pass / Fail: Pass
Comments: Root node is a leaf with default label false.

Test Name: RandomForest::train - normal Test Description: Train a RandomForest with 5 trees on a dataset.
Pass / Fail: Pass
Comments: 5 trees created, each with a root node.

Test Name: RandomForest::train - empty records Test Description: Train a RandomForest with 5 trees on an empty dataset.
Pass / Fail: Pass
Comments: 5 trees created, each with a leaf root node (label false).

Test Name: RandomForest::predict - tie Test Description: Predict with 2 trees, one true, one false (tie).
Pass / Fail: Pass
Comments: Returned true as default tiebreaker.

FAIL CASES

Test Name: RecordCollector::normalizeInput - untrained Test Description: Attempt to normalize input before loading a dataset.

Pass / Fail: Fail
Comments: Expected IllegalStateException, but no exception was thrown.

Test Name: Tree::predict - normal Test Description: Test Tree prediction on a trained tree with diabetic-like features.
Pass / Fail: Fail
Comments: Expected prediction to be true, but returned false.

Test Name: RandomForest::predict - normal Test Description: Test RandomForest prediction on diabetic-like features.
Pass / Fail: Fail
Comments: Expected prediction to be true, but returned false.

Test Name: RandomForest::analyzePrediction - normal Test Description: Analyze prediction to include result breakdown.
Pass / Fail: Fail
Comments: Expected analysis to include "Prediction: Diabetic", but prediction was false.