MOA – NaiveBayesMultinomialExample

JAVA CODE:

import moa.core.InstanceExample;

import moa.core.Example;

import moa.classifiers.Classifier;

import moa.classifiers.bayes.NaiveBayesMultinomial;

import moa.evaluation.BasicClassificationPerformanceEvaluator;

import moa.streams.ArffFileStream;

import com.yahoo.labs.samoa.instances.Instance;

public class NaiveBayesMultinomialExample {

public static void main(String[] args) {

String arffFilePath = "C:\\Program Files\\Weka-3-9-6\\data\\iris.arff"; // Replace with your ARFF file path

ArffFileStream stream = new ArffFileStream(arffFilePath, -1);

stream.prepareForUse();

// Initialize Multinomial Naive Bayes classifier

Classifier classifier = new NaiveBayesMultinomial();

classifier.setModelContext(stream.getHeader());

classifier.prepareForUse();

// Performance evaluator

BasicClassificationPerformanceEvaluator evaluator = new BasicClassificationPerformanceEvaluator();

int numberOfInstances = 150; // Define the number of instances to process

for (int i = 0; i < numberOfInstances && stream.hasMoreInstances(); i++) {

Instance instance = stream.nextInstance().getData();

Example<Instance> example = new InstanceExample(instance);

// Train classifier on the current instance

classifier.trainOnInstance(example);

// Evaluate the classifier's prediction

evaluator.addResult(example, classifier.getVotesForInstance(instance));

if (i > 0 && i % 50 == 0) {

System.out.println("Processed " + i + " instances.");

}

}

// Output the results

System.out.println("Accuracy: " + evaluator.getFractionCorrectlyClassified() \* 100 + "%");

System.out.println("Kappa Statistic: " + evaluator.getKappaStatistic());

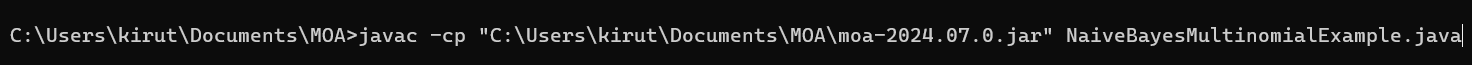
System.out.println("Kappa Temporal Statistic: " + evaluator.getKappaTemporalStatistic());

}

}Output:

To compile:

javac -cp "C:\Users\kirut\Documents\MOA\moa-2024.07.0.jar" NaiveBayesMultinomialExample.java



To run:

java -cp ".;C:\Users\kirut\Documents\MOA\moa-2024.07.0.jar" NaiveBayesMultinomialExample

Output:

