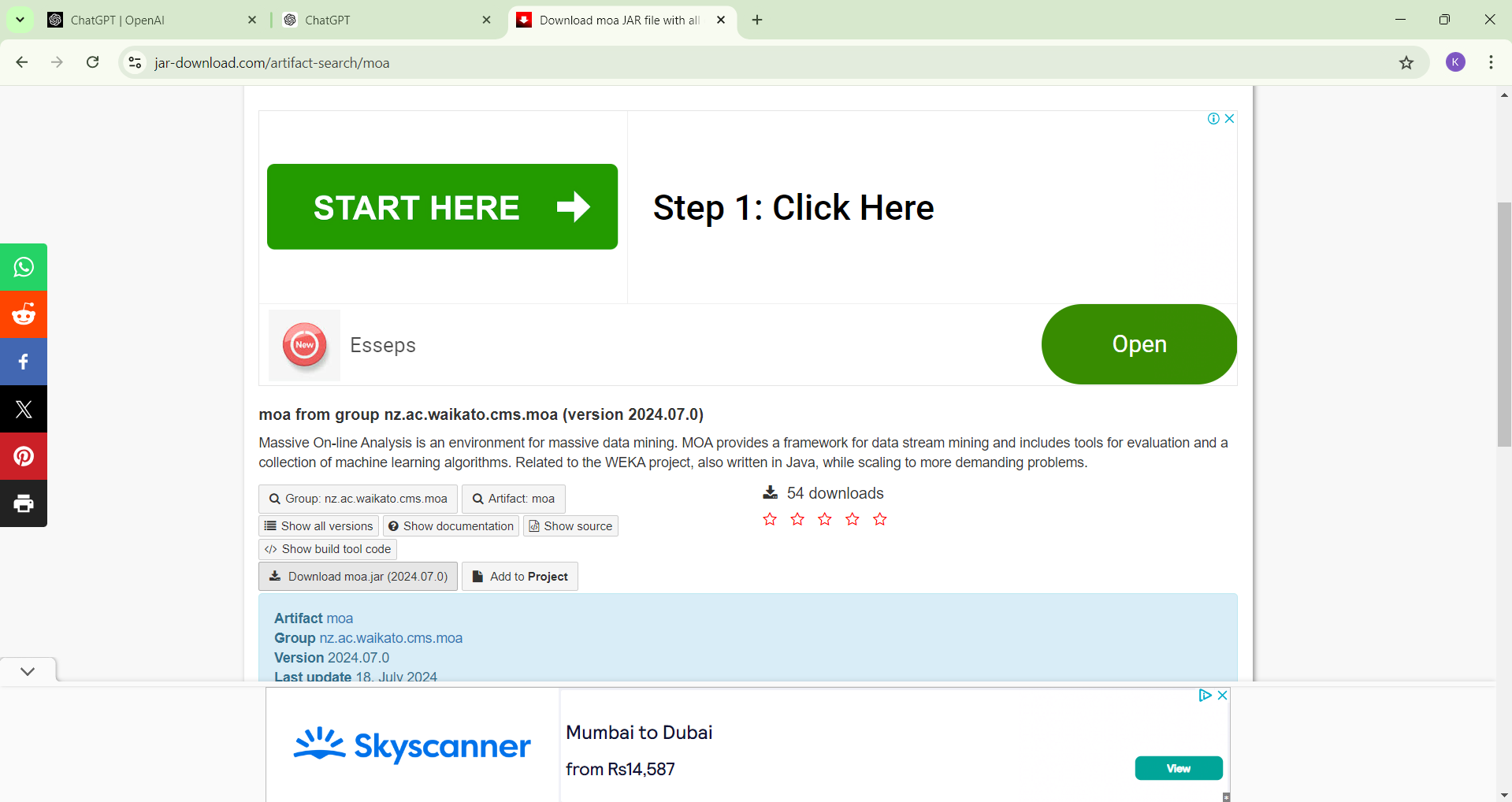
**MOA API implementation in java.**

**Stream Classifier – Naïve Bayes**

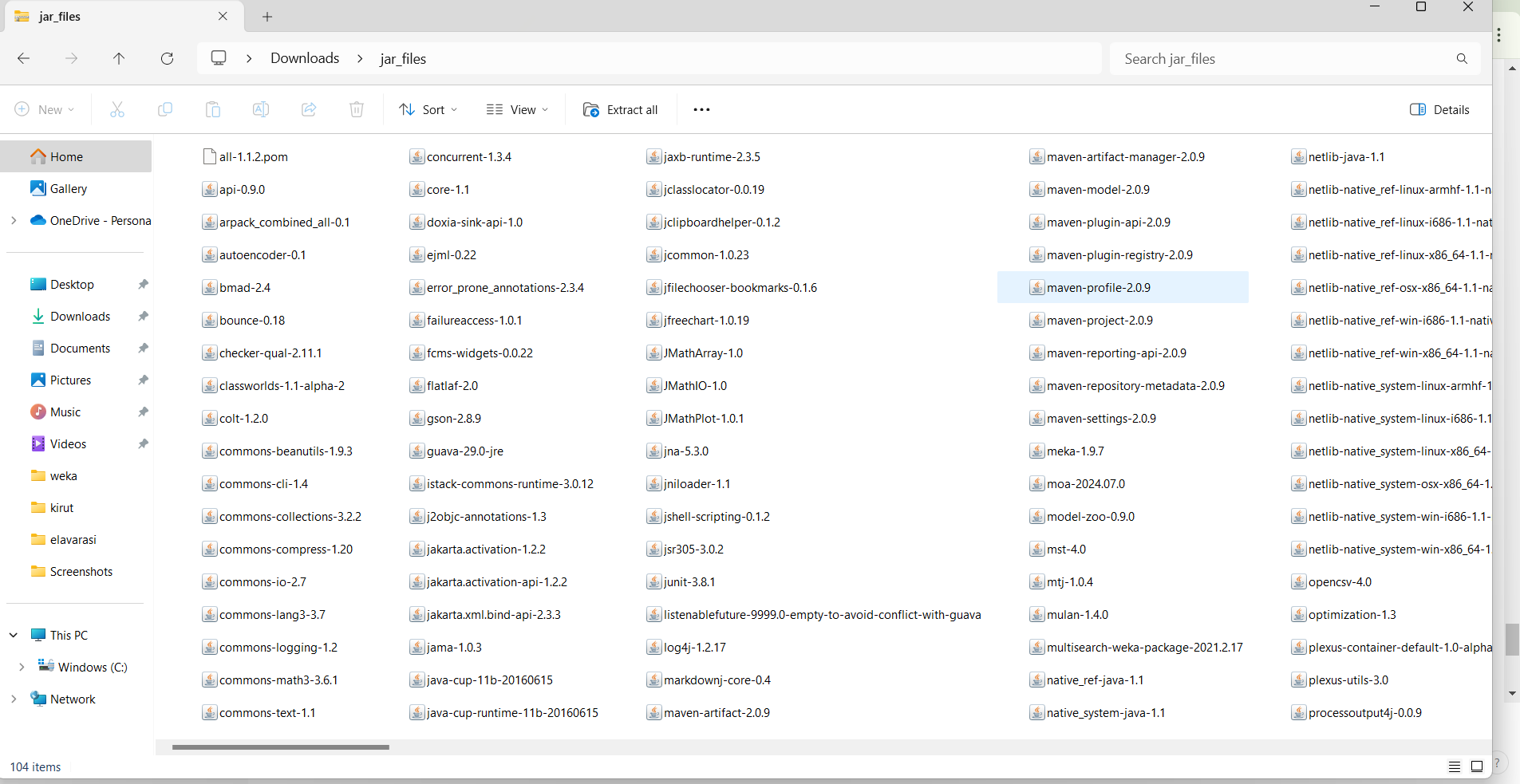
**In notepad:**

**Launch moa jar in your system**

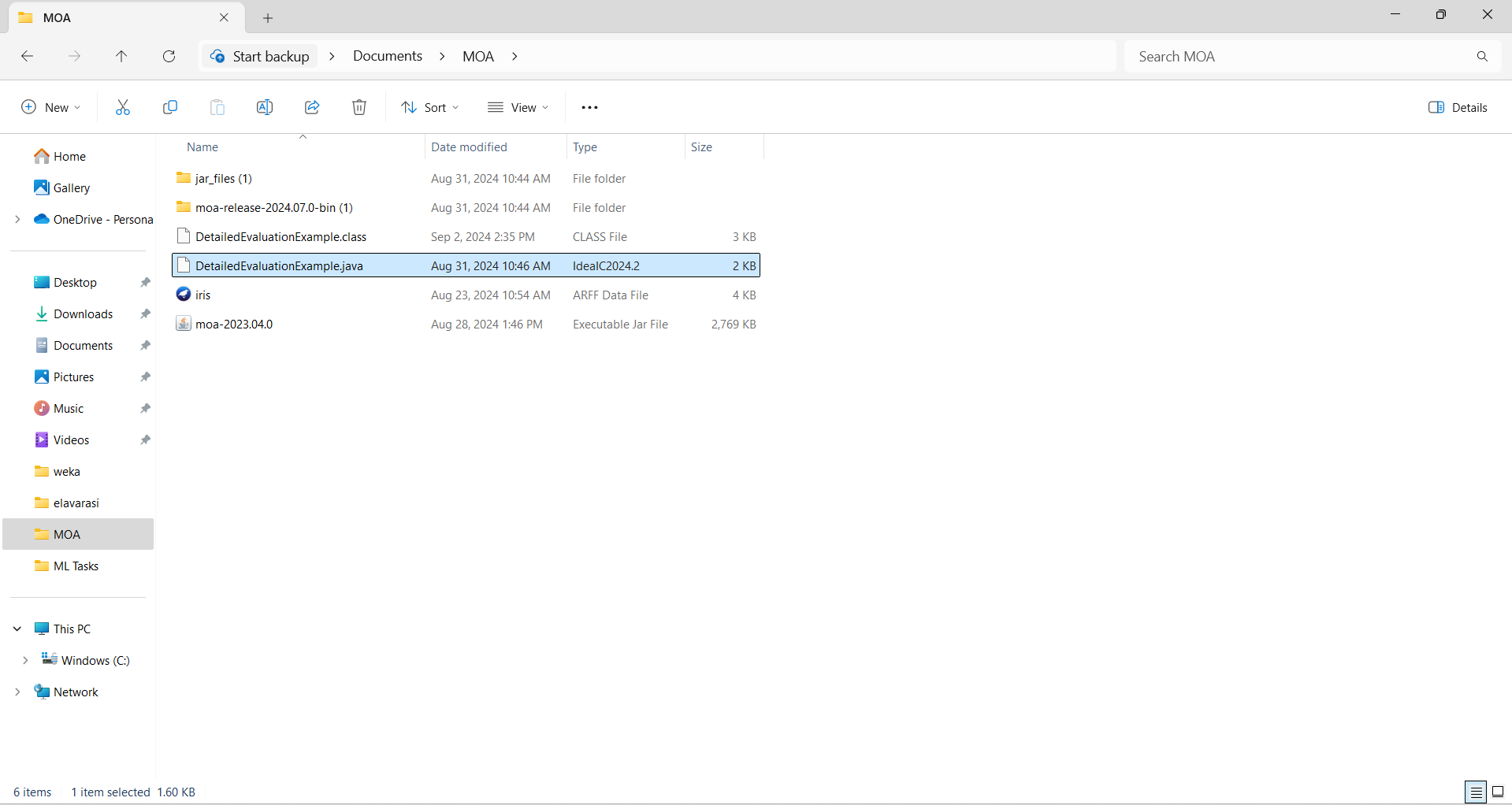
Download moa jar file from the moa official websites.



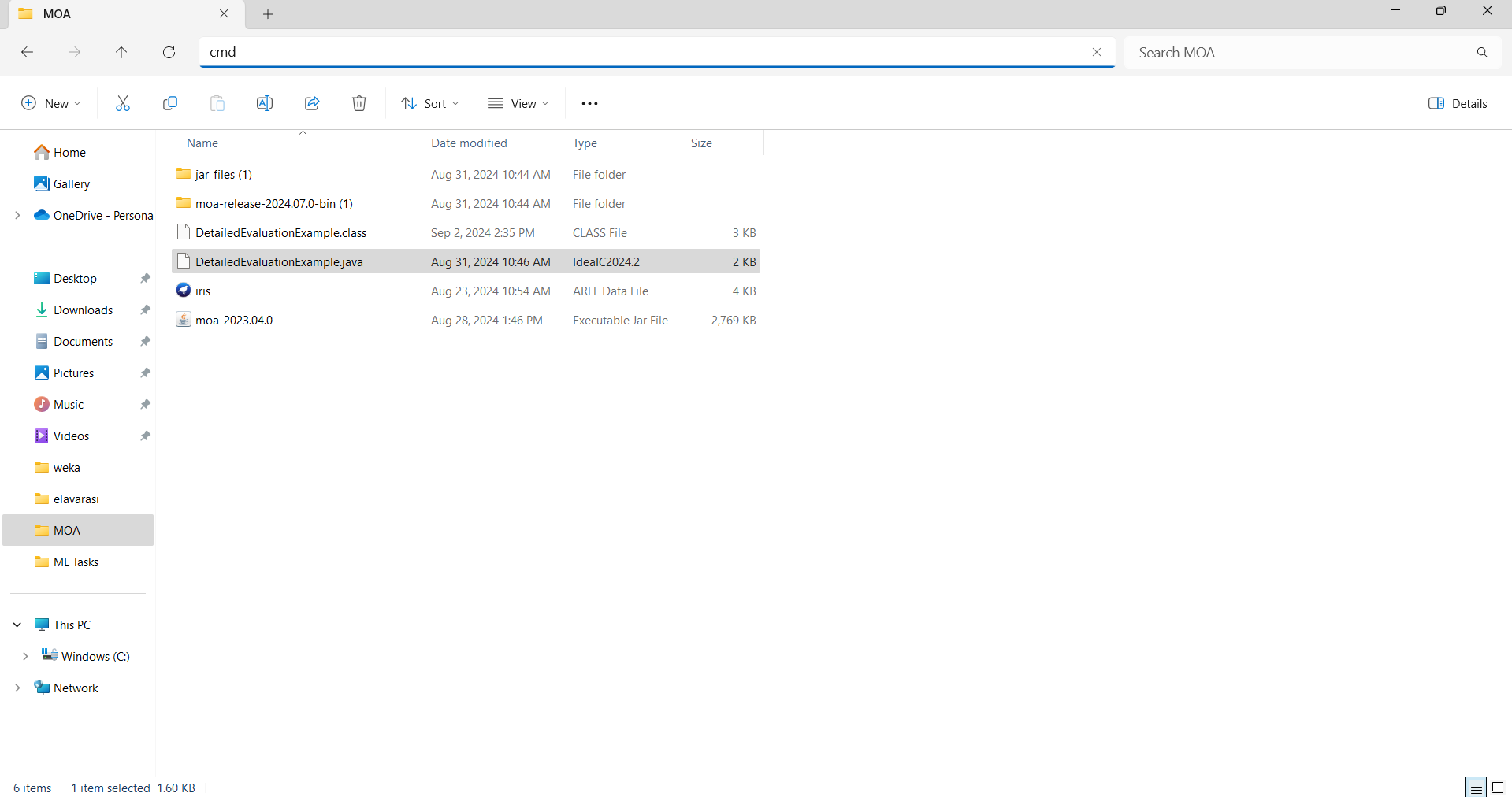
Extract all the jar file.



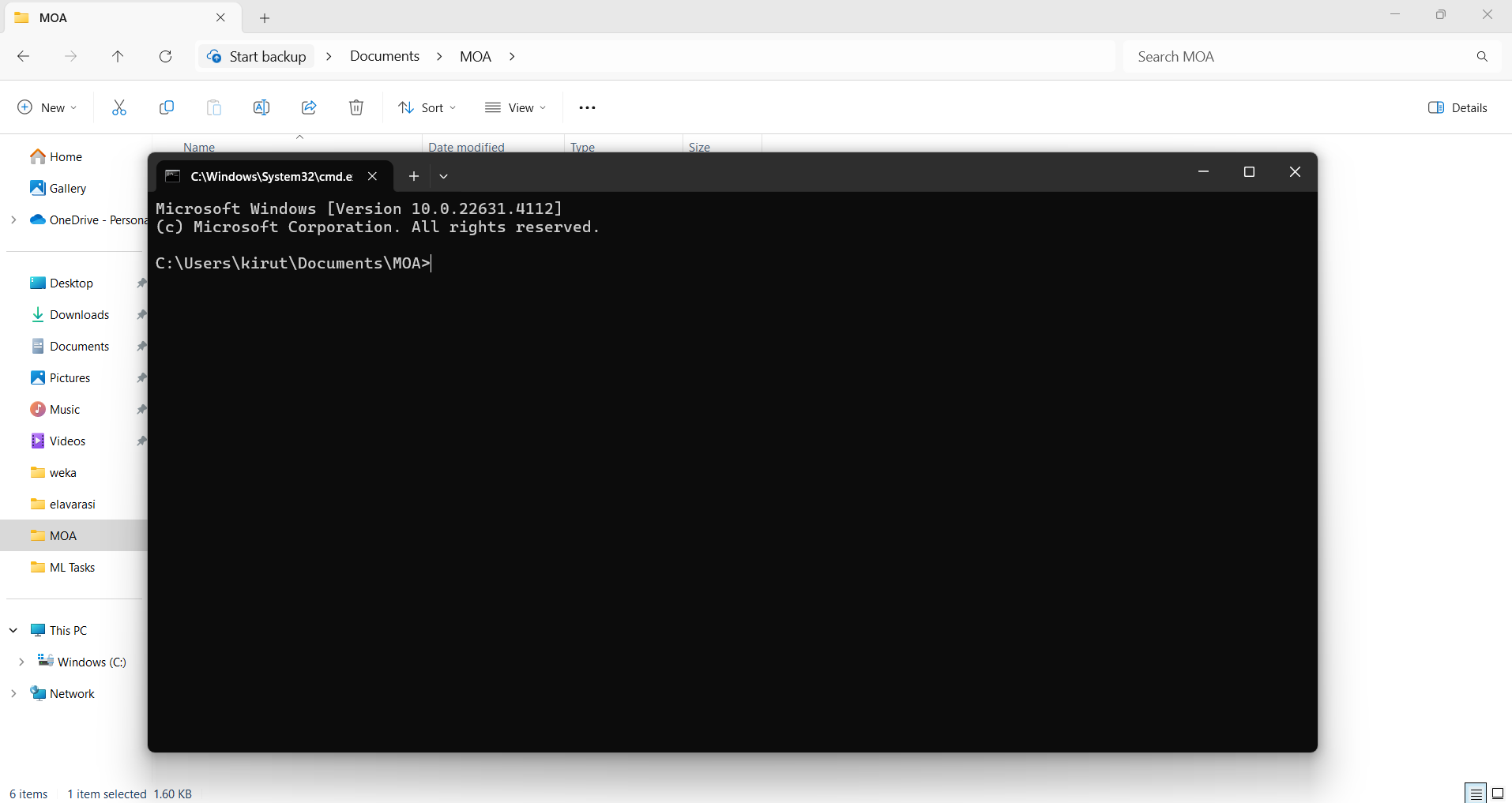
* Open notepad
* Write the program in the notepad
* Save the file where the datasets is stored and the moa is located.



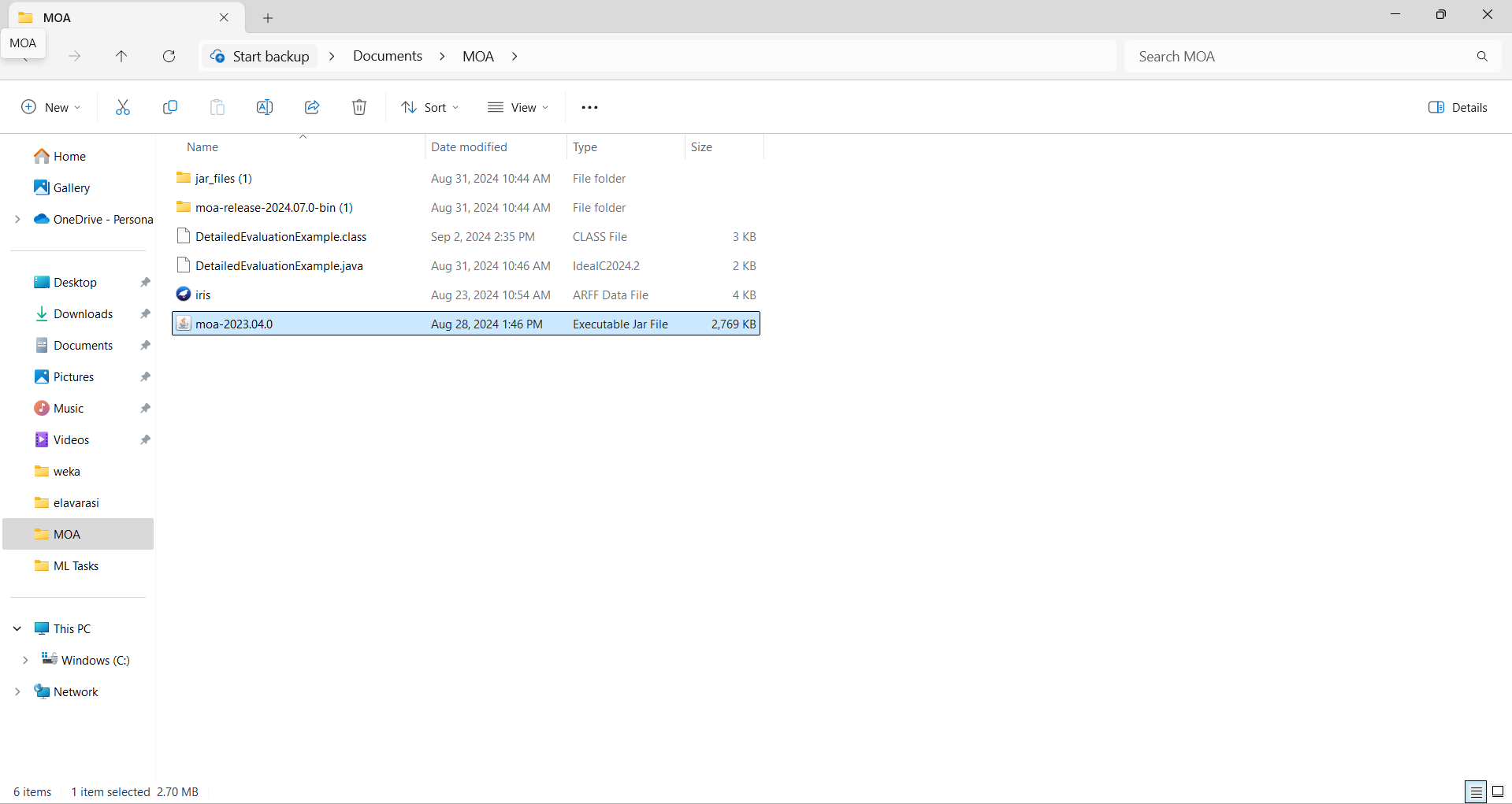
* Keep dataset , moa and the program in the same folder.
* Type command prompt in the location



* Open command prompt.



* Write command line to compile the program.
* Copy the path of moa jar file. Right click the moa jar file to copy the path.

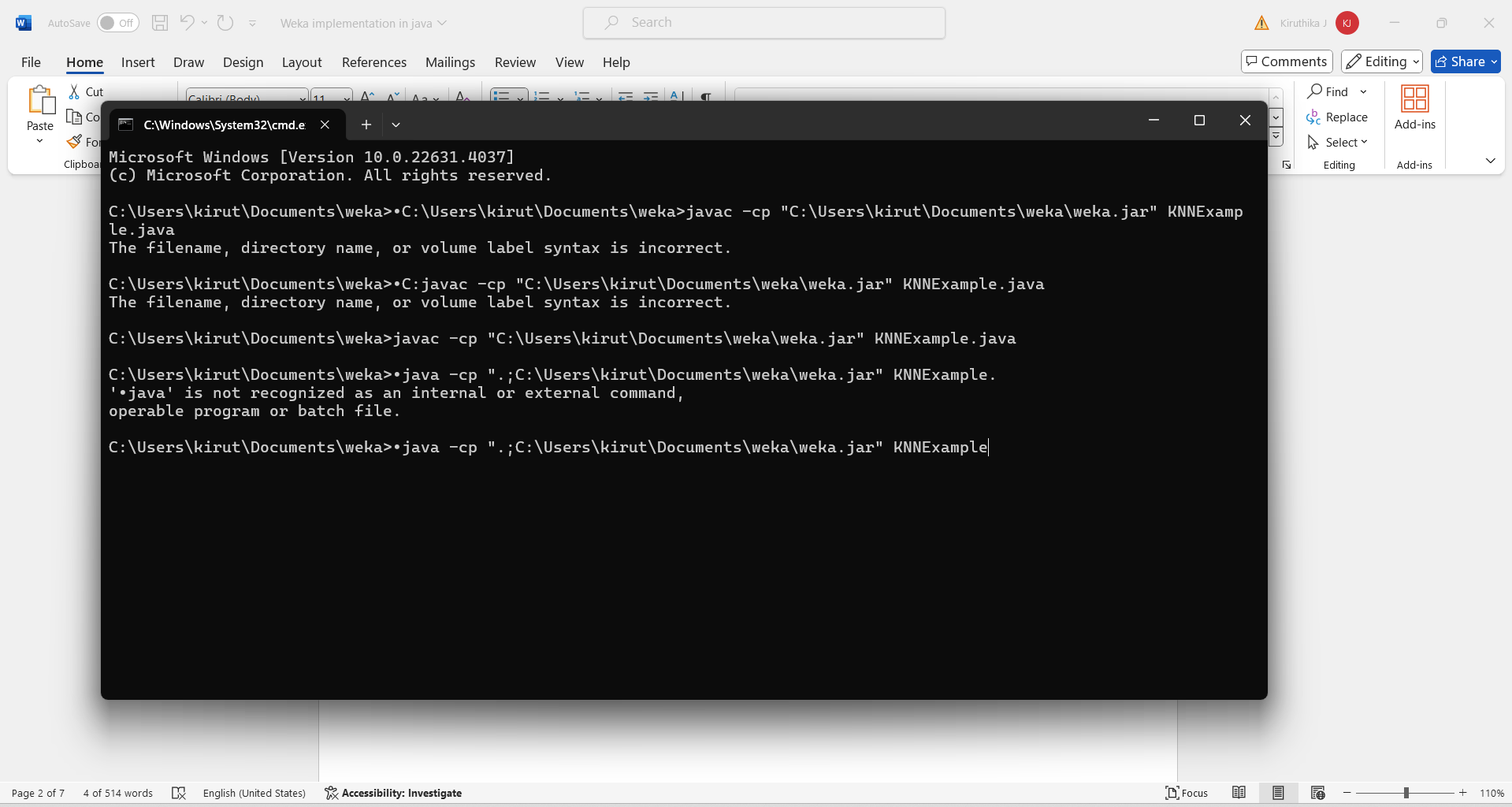


**Command for compile.**

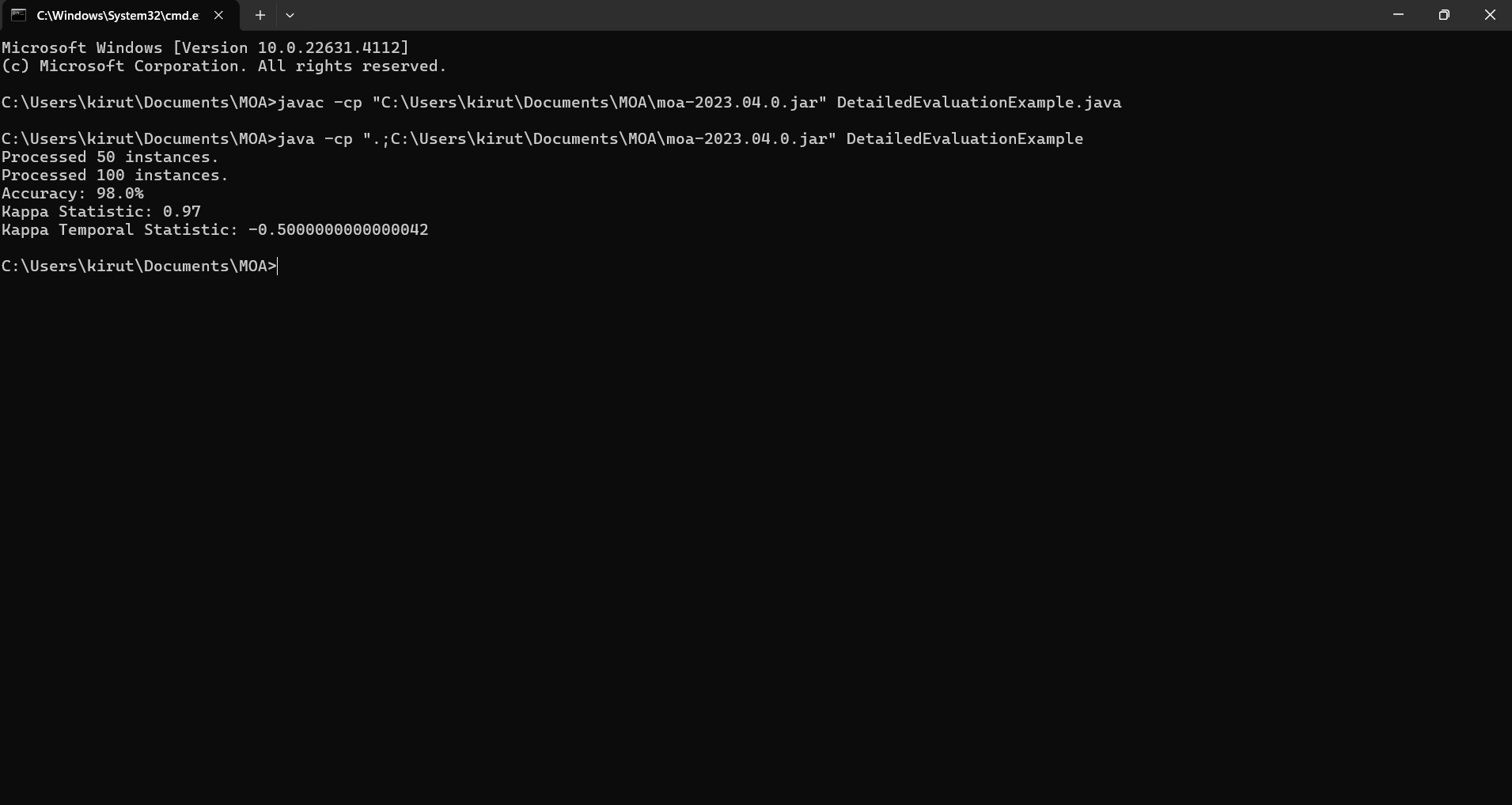
javac -cp "C:\Users\kirut\Documents\MOA\moa-2023.04.0.jar" DetailedEvaluationExample.java



* **Command for run.**
* java -cp ".;C:\Users\kirut\Documents\MOA\moa-2023.04.0.jar" DetailedEvaluationExample.



Click enter to run the program



Program:

import moa.core.InstanceExample;

import moa.core.Example;

import moa.classifiers.Classifier;

import moa.classifiers.bayes.NaiveBayes;

import moa.evaluation.BasicClassificationPerformanceEvaluator;

import moa.streams.ArffFileStream;

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

public class DetailedEvaluationExample {

public static void main(String[] args) {

String arffFilePath = "C:\\Program Files\\Weka-3-9-6\\data\\iris.arff";

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

stream.prepareForUse();

Classifier classifier = new NaiveBayes();

classifier.setModelContext(stream.getHeader());

classifier.prepareForUse();

BasicClassificationPerformanceEvaluator evaluator = new BasicClassificationPerformanceEvaluator();

int numberOfInstances = 150;

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

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

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

classifier.trainOnInstance(example);

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

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

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

}

}

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

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

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

}

}

NOTES:

To run the different algorithm , change the class in import statement and the arff file we want to use.