ANKARA YILDIRIM BEYAZIT UNIVERSITY

Faculty of Engineering and Natural Sciences
Department of Computer Engineering

SUPPORT VECTOR MACHINE

MEMBERS OF GROUP 13:

19050161004-Ali Can GÜNEŞ

19050111039-Dilek YALÇIN

19050161003-Furkan EKİZ

19050111013-Semra ATEŞ

19050111001-Taha KUZUDİŞLİ

SUPERVISOR:

Asst. Prof. Fadi YILMAZ

CENG - 303 Design and Analysis of Algorithms

January 18, 2023

Introduction

The aim of the project is to use only two properties, width and length, to classify apples as ripe or immature in the two-dimensional apple dataset. The SVM algorithm we created will use these two features to find the best hyperplane that separates ripe apples from unripe apples. The location of the hyperplane will be determined by the width and length of the apples, and apples with the width and length falling on one side of the hyperplane will be classified as ripe, while apples on the other side will be classified as unripe.

Project Development Process

Before and during the project, all processes from the planning of the project to the development and reporting of the project were gathered in common working areas as group members and carried out together by all group members. In addition, the GitHub link of the project was created by only one group member on behalf of all group members and shared in the report.

GitHub: https://github.com/cangnss/java-svm-project

What is SVM (Support Vector Machine)?

Support Vector Machine, is a machine learning algorithm used for categorizing data. The algorithm uses a boundary called a "hyperplane" to separate different groups of data. The goal is to find the hyperplane that best separates the data into different groups, while also maximizing the space between the closest data points of each group. If the SVM is two-dimensional, it means it only uses two characteristics, such as width and length, to classify the data. The SVM algorithm will use these characteristics to find the ideal hyperplane that separates the groups. The position of the hyperplane is determined by the values of the width and length of the data, and any new data that falls on one side of the hyperplane will be classified into one group, while those on the other side will be classified into another.

Detailed Function and Class Descriptions

createDataset method:

This method creates a dataset of 2000 random data points in the form of an ARFF file, which stands for Attribute-Relation File Format. It is a file format used for storing data in a format that is readable by the Weka machine learning software.

readDataset method:

This method reads an ARFF file named "2D_dataset.arff" and returns the dataset as an "Instances" object.

LinearSVM class:

This Java class is defined as a class called "LinearSVM" that trains a linear support vector machine (SVM) model using data provided in an Instances object and corresponding labels. Required Java classes such as "Arrays" from java.util package, Instance and Instances from weka.core package are imported. A method named "train()" is defined inside the "LinearSVM" class that takes two arguments.

train method:

This method initializes a variable "n" to the number of attributes in the data and creates a pair of array "weights" with the same number of elements and fills it with 0. It then sets two variables named "learningRate" to 0.1 and "maxIterations" to 100. These variables will be used in the training process. It then enters a for loop that iterates over the number of "maxIterations". Inside the outer loop, it enters another for loop that iterates over the number of instances in the data. In the inner loop, it takes the sample at index i from the data and calculates the estimate using the "predict()" method, passing the sample and the weight array. It then calculates the error between the sample's label and the prediction. It then updates the weight by adding the product of the learning rate, error, and attribute value for each attribute of the sample. Returns the weights after the outer loop is complete.

predict method:

The method first initializes a variable "prediction" to 0, which will later be used to store the dot product of the instance's attribute values and the weight array. Then, it performs a loop over all the attributes of the instance, and for each attribute, it multiplies the attribute's value with the corresponding weight, and adds this product to the "prediction" variable. The loop is followed by a simple conditional statement that compares the value of "prediction" with 0, if prediction is greater than 0, the method returns 1, otherwise it returns -1. The dot product of the attribute values and the weight array represents the value of the decision function of the SVM. The decision function assigns a score to each input example, and if the score is greater than 0, the example is classified as a positive example and if the score is less than 0, it is classified as a negative example.

plotData method:

This method is used to create a scatter plot of the given data points using the JFreeChart library. It takes a two-dimensional array of data points as input, where each row represents a point with x and y coordinates. The method loops through the array and adds each point to a series, then it creates a scatter plot of the series, creates a new frame, sets the scatter plot as the content of the frame, and makes it visible on the screen. In short, this method plots a scatter plot of given data points using JFreeChart library and shows it on the screen.

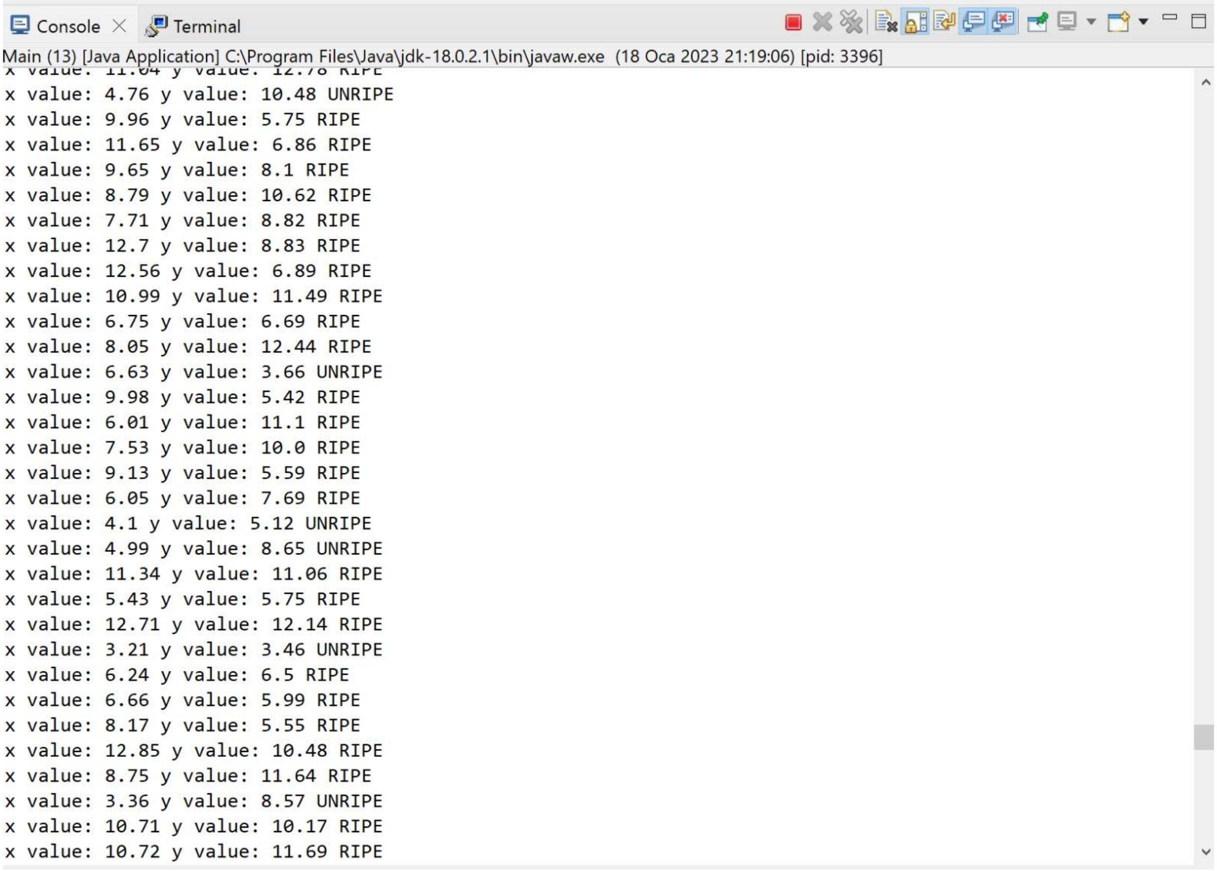
Main class:

The Main class uses other predefined classes and methods to create a dataset, read the dataset, train a linear SVM model, and make predictions on a test sample.

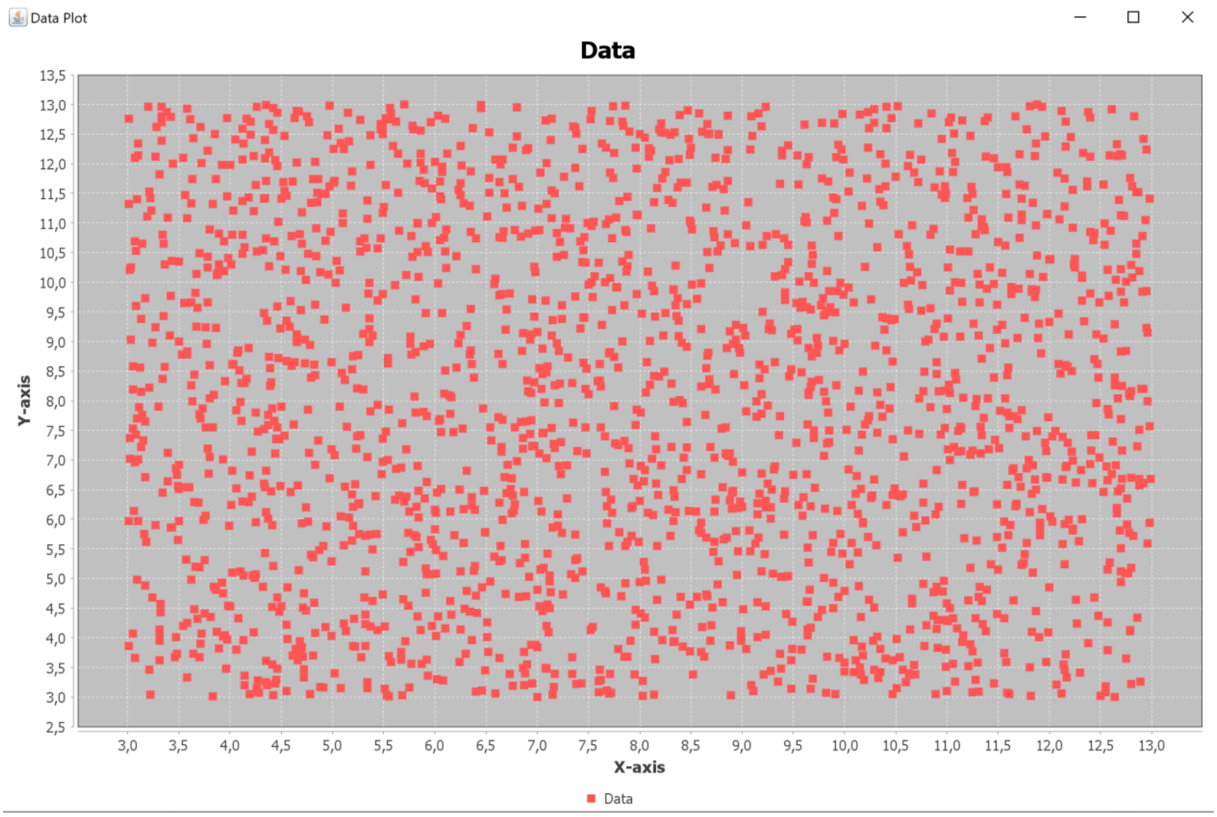
To explain what this method does simply;

It starts by reading the dataset from file and assigns it to an Instances object called "data". It creates a new object of the LinearSVM class called "ls". It creates an int array called "labels" with the same number of elements as the number of instances in the data, and assigns a label of 1 if the value of the x attribute is greater than or equal to 5 and value of y attribute is greater than or equal to 4, otherwise assigns a label of -1. It trains the linear SVM model by calling the train() method of the LinearSVM object, passing in the data and labels as arguments, and assigns the returned weights to a double array called "weights". It creates a double array called "dataArr" with the same number of elements as the number of instances in the data, and assigns the values of the x and y attributes to corresponding element. It creates a new instance from the data set and sets the value of x to 4 and y to 5. It makes a prediction on the test.

Sample Run



```
Main (13) [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe  (18 Oca 2023 21:19:06) [pid: 3396]
x value: 11.04 y value: 12.78 RIPE
x value: 4.76 y value: 10.48 UNRIPE
x value: 9.96 y value: 5.75 RIPE
x value: 11.65 y value: 6.86 RIPE
x value: 9.65 y value: 8.1 RIPE
x value: 8.79 y value: 10.62 RIPE
x value: 7.71 y value: 8.82 RIPE
x value: 12.7 y value: 8.83 RIPE
x value: 12.56 y value: 6.89 RIPE
x value: 10.99 y value: 11.49 RIPE
x value: 6.75 y value: 6.69 RIPE
x value: 8.05 y value: 12.44 RIPE
x value: 6.63 y value: 3.66 UNRIPE
x value: 9.98 y value: 5.42 RIPE
x value: 6.01 y value: 11.1 RIPE
x value: 7.53 y value: 10.0 RIPE
x value: 9.13 y value: 5.59 RIPE
x value: 6.05 y value: 7.69 RIPE
x value: 4.1 y value: 5.12 UNRIPE
x value: 4.99 y value: 8.65 UNRIPE
x value: 11.34 y value: 11.06 RIPE
x value: 5.43 y value: 5.75 RIPE
x value: 12.71 y value: 12.14 RIPE
x value: 3.21 y value: 3.46 UNRIPE
x value: 6.24 y value: 6.5 RIPE
x value: 6.66 y value: 5.99 RIPE
x value: 8.17 y value: 5.55 RIPE
x value: 12.85 y value: 10.48 RIPE
x value: 8.75 y value: 11.64 RIPE
x value: 3.36 y value: 8.57 UNRIPE
x value: 10.71 y value: 10.17 RIPE
x value: 10.72 y value: 11.69 RIPE
```



References

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

https://stackoverflow.com/questions/15625359/support-vector-machine-for-java

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

https://www.geeksforgeeks.org/support-vector-machine-algorithm/

https://youtu.be/SIU_GWaEils

https://youtu.be/_YPScrckx28

https://www.youtube.com/watch?v=1NxnPkZM9bc

https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e