

Ujian Akhir Semester

Machine Learning

Muhammad Zaid Abdillah - 2110191013

Data Pasien Penyakit Hipertensi

Usia	Berat Badan	Kelamin	Hipertensi
muda	overweight	pria	ya
muda	underweight	pria	tidak
muda	average	wanita	tidak
tua	average	pria	tidak
tua	overweight	pria	ya
muda	underweight	pria	tidak
tua	overweight	wanita	ya
tua	average	pria	tidak

Assignment

1. Lakukan Learning untuk mendapatkan Weights yang terbaik
2. Lakukan pengujian dengan data learning dan hitunglah error ratio-nya

Source Code

```
private void readFile() {  
    int label;  
    int[] f = new int[13];  
    String line;  
    try {  
        BufferedReader reader = new BufferedReader(new FileReader("../heart.csv"));  
        while ((line = reader.readLine()) != null) {  
            String[] barisData = line.split(regex: ",");  
            for (int i = 0; i < 13; i++) {  
                f[i] = (int) Double.parseDouble(barisData[i]);  
            }  
            label = Integer.parseInt(barisData[13]);  
            listDataSet.add(new DataSet(f, label));  
        }  
    } catch (FileNotFoundException ex) {  
        ex.printStackTrace();  
    } catch (IOException ex) {  
        ex.printStackTrace();  
    }  
}
```

Source Code

```
public void proses() {  
    int jumlahEpoch = 0;  
  
    System.out.println("Case Study Heart");  
    readFile();  
    randomW();  
  
    error = trainingPerceptron();  
    jumlahEpoch++;  
    while (jumlahEpoch < 2000) {  
        error = trainingPerceptron();  
        jumlahEpoch++;  
    }  
  
    System.out.println("W optimal : ");  
    for (int j = 0; j < 13; j++) {  
        System.out.println("w[" + j + "] = " + w[j]);  
    }  
    System.out.println("Jumlah epoch : " + jumlahEpoch);  
  
    System.out.println("\nRata rata error : " + testing());  
}
```

Source Code

```
public void randomW() {  
    Random random = new Random();  
  
    for(int j = 0; j < 3; j++){  
        w[j] = random.nextInt(1 + 1) - 1;  
        w[j] = w[j]/10;  
    }  
}
```

```
public int trainingPerceptron() {  
    int output = 0, classLabel, jumlah = 0;  
    int x[] = new int[13];  
    double summation = 0;  
  
    for (int i = 0; i < listDataSet.size(); i++) {  
        classLabel = listDataSet.get(i).getClassLabel();  
        x = listDataSet.get(i).getX();  
  
        summation += 1 * w[0];  
        for (int j = 0; j < 13; j++) {  
            summation += x[j] * w[j + 1];  
        }  
  
        if (summation < 0) {  
            output = 1;  
        } else if (summation >= 0) {  
            output = 2;  
        }  
  
        error = getError(classLabel, output);  
  
        if (error != 0) {  
            getWBaru(error, x);  
            return error;  
        }  
    }  
  
    return error;  
}
```

Source Code

```
private double testing(){
    int output = 0, classLabel, jumlah = 0;
    int x[] = new int[13];
    double summation = 0, presentaseError;
    double errorBaru[] = new double[300];

    for (int i = 0; i < listDataSet.size(); i++) {
        classLabel = listDataSet.get(i).getClassLabel();
        x = listDataSet.get(i).getX();

        summation += 1 * w[0];
        for (int j = 0; j < 13; j++) {
            summation += x[j] * w[j + 1];
        }

        if (summation < 0) {
            output = 1;
        } else if (summation >= 0) {
            output = 2;
        }

        errorBaru[i] = getError(classLabel, output);
    }

    return presentaseError(errorBaru);
}
```

Source Code

```
public int getError(int classLabel, int output) {  
    return Math.abs(classLabel - output);  
}  
  
public void getWBaru(int error, int x[]) {  
    w[0] = w[0] + (0.1 * 1) * error;  
    for (int i = 0; i < 13; i++) {  
        w[i + 1] = w[i + 1] + (0.1 * x[i] * error);  
    }  
}
```


Source Code

```
private double presentaseError(double[] errorBaru){  
    double presentaseError = 0;  
  
    for(int i = 0; i < errorBaru.length; i++){  
        presentaseError += errorBaru[i];  
    }  
  
    return presentaseError/errorBaru.length*100;  
}
```

Output Case Study Diabetes

Case Study Heart

W optimal :

w[0] = 199.899999999999293

w[1] = 13399.9000000000409

w[2] = 199.899999999999293

w[3] = 799.9999999999717

w[4] = 31999.9

w[5] = 57199.89999999807

w[6] = 0.0

w[7] = 399.99999999998585

w[8] = 21599.999999999196

w[9] = 199.99999999999292

w[10] = 199.99999999999292

w[11] = 399.99999999998585

w[12] = 599.90000000000003

Jumlah epoch : 2000

Presentase error : 50.0%

BUILD SUCCESSFUL (total time: 0 seconds)