

```

1 package classes;
2
3 import java.io.*;
4 import java.text.DecimalFormat;
5 import java.util.ArrayList;
6 import java.util.List;
7 import java.util.Random;
8
9 /**
10  * * created by lgcaobianco on 21/03/18 **
11  */
12 public class Perceptron {
13     List<double[]> matrizPontos = new ArrayList<double[]>(
14 );
15     List<double[]> conjuntoTeste = new ArrayList<double[]>(
16 );
17     private double[][] matrizPesosInicial = new double[4][
18 1];
19     private double[][] matrizPesosFinal = new double[4][1]
20 ;
21     private int contadorEpocas;
22
23     public void imprimirmatrizPesosInicial() {
24         for (int i = 0; i < this.matrizPesosInicial.length
25 ; i++) {
26             for (int j = 0; j < this.matrizPesosInicial[i]
27 .length; j++) {
28                 System.out.print(this.matrizPesosInicial[i
29 ][j] + " ");
30             }
31             System.out.println();
32         }
33     }
34
35     public void imprimirMatrizPontos() {
36         for (int i = 0; i < this.matrizPontos.size(); i++)
37         {
38             for (int j = 0; j < this.matrizPontos.get(i).
39 length; j++) {
40                 System.out.print(this.matrizPontos.get(i)[
41 j] + " ");
42             }
43             System.out.println();
44         }
45     }
46
47     public void construirMatrizPontos(String nomeArquivo,
48 String extensaoArquivo) {
49         LeInformacoes informacoes = new LeInformacoes(

```

```

39 nomeArquivo, extensaoArquivo);
40     this.matrizPontos = informacoes.extrairPontos();
41 }
42
43 public void construirConjuntoTeste(String nomeArquivo,
String extensaoArquivo) {
44     LeInformacoes informacoes = new LeInformacoes(
nomeArquivo, extensaoArquivo);
45     this.conjuntoTeste = informacoes.extrairPontos();
46 }
47
48 public void construirmatrizPesosInicial() {
49     matrizPesosInicial[0][0] = -1;
50     for (int i = 1; i < this.matrizPesosInicial.length
; i++) {
51         for (int j = 0; j < this.matrizPesosInicial[i]
.length; j++) {
52             Random r = new Random();
53             this.matrizPesosInicial[i][j] = r.
nextDouble();
54             this.matrizPesosFinal[i][j] = this.
matrizPesosInicial[i][j];
55         }
56     }
57
58 }
59
60 public int ativacao(double somatorio) {
61     int classificacao; //passar o somatório pelo g(u)
62     if (somatorio >= 0) {
63         classificacao = 1;
64     } else {
65         classificacao = -1;
66     }
67     return classificacao;
68 }
69
70
71 public void treinarPerceptron() {
72     double somatorio, taxaAprendizagem = 0.01,
classificacao;
73     int erro, iMaximo = 0;
74
75     percorreLinhasConjuntoTreinamento:
76     for (int i = 0; i < this.matrizPontos.size(); ) {
77         somatorio = 0;
78         for (int j = 0; j < this.matrizPontos.get(i).
length - 1; j++) {
79             somatorio += (this.matrizPontos.get(i)[j]
* this.matrizPesosFinal[j][0]);

```

```

80         }
81
82         classificacao = ativacao(somatorio);
83
84         if (classificacao == this.matrizPontos.get(i)
85 [3]) {
86             i++;
87         } else { //se classificacao nao coincide com
88 d_i, ajustar coeficientes
89             erro = (int) (this.matrizPontos.get(i)[3]
90 - classificacao);
91             for (int j = 0; j < this.
92 matrizPesosInicial.length; j++) {
93                 this.matrizPesosFinal[j][0] += (
94 taxaAprendizagem * erro * this.matrizPontos.get(i)[j]);
95             }
96             this.contadorEpocas++;
97             i = 0;
98         }
99     }
100 }
101
102     public void classificarVetores() {
103         double somatorio;
104
105         for (int i = 0; i < this.conjuntoTeste.size(); i
106 ++)) {
107             somatorio = 0;
108             for (int j = 0; j < this.conjuntoTeste.get(i)
109 .length; j++) {
110                 somatorio += this.conjuntoTeste.get(i)[j]
111 * this.matrizPesosFinal[j][0];
112             }
113             int classificacao = ativacao(somatorio);
114             System.out.println("A linha: " + (i + 1) + "
115 foi classificada como: " + classificacao);
116         }
117     }
118
119     public static void iniciarTabelaLatex() {
120         try (Writer writer = new BufferedWriter(new
121 OutputStreamWriter(
122             new FileOutputStream("matrizSaida.tex"),

```

```

119 "utf-8")) {
120     writer.write("\\documentclass{article}" +
System.getProperty("line.separator"));
121     writer.write("\\usepackage{multirow}" +
System.getProperty("line.separator"));
122     writer.write("\\usepackage[utf8]{inputenc}" +
System.getProperty("line.separator"));
123     writer.write("\\usepackage[bottom]{footmisc}"
+ System.getProperty("line.separator"));
124     writer.write("\\usepackage{landscape}" + System.
getProperty("line.separator"));
125     writer.write("\\renewcommand\\tablename{
Tabela}" + System.getProperty("line.separator"));
126     writer.write("\\begin{document}" + System.
getProperty("line.separator"));
127     writer.write("\\begin{landscape}" + System.
getProperty("line.separator"));
128     writer.write("\\begin{table}" + System.
getProperty("line.separator"));
129     writer.write("\\centering" + System.
getProperty("line.separator"));
130     writer.write("\\begin{tabular}{llllllllll}" +
System.getProperty("line.separator"));
131     writer.write("\\hline" + System.getProperty("
line.separator"));
132     writer.write("\\multicolumn{1}{|c|}{\\
multirow{2}{*}{Treinamento}} & \\multicolumn{4}{l|}{Vetor
de Pesos Inicial} " +
133         "& \\multicolumn{4}{l|}{Vetor de
Pesos Final} & \\multicolumn{1}{l|}{\\multirow{2}{*}{Núm
. de Épocas}} \\\\ " +
134         "\\cline{2-9}\\n \\multicolumn{1}{|c
|}{ } & \\multicolumn{1}{l|}{w_0} & \\multicolumn{1}{l|}{
w_1} & " +
135         "\\multicolumn{1}{l|}{w_2} & \\
multicolumn{1}{l|}{w_3} & \\multicolumn{1}{l|}{w_0} & " +
136         "\\multicolumn{1}{l|}{w_1} & \\
multicolumn{1}{l|}{w_2} & \\multicolumn{1}{l|}{w_3} & " +
137         "\\multicolumn{1}{l|}{ } \\\\ \\hline"
);
138
139     } catch (IOException e) {
140         e.printStackTrace();
141     }
142 }
143
144 public void preencherTabelaLatex(Perceptron p, int
contaQuantidadeTreinamentos) {
145     try (Writer writer = new FileWriter("matrizSaida.
tex", true)) {

```

```

146         writer.append("\\multicolumn{1}{|l|}" + (
            contaQuantidadeTreinamentos+1) + "} &");
147
148         for (int i = 0; i < 4; i++) {
149             writer.append("\\multicolumn{1}{|l|}" +
                new DecimalFormat("#.####").format(p.matrizPesosInicial[i]
                    ][0]) + "} & ");
150         }
151
152         for (int i = 0; i < 4; i++) {
153             writer.append("\\multicolumn{1}{|l|}" +
                new DecimalFormat("#.####").format(p.matrizPesosFinal[i][
                    0]) + "} & ");
154         }
155
156         writer.append("\\multicolumn{1}{l|} {" + p.
            contadorEpocas + "}");
157         writer.append("\\\\\\" + "\\hline" + System.
            getProperty("line.separator"));
158     } catch (IOException e) {
159         e.printStackTrace();
160     }
161 }
162
163 public static void finalizarTabelaLatex() {
164     try (Writer writer = new FileWriter("matrizSaida.
        tex", true)) {
165         writer.append("\\end{tabular}" + System.
            getProperty("line.separator"));
166         writer.append("\\caption{Tabela em \\LaTeX{}
            gerada automaticamente pelo software em Java}" + System.
            getProperty("line.separator"));
167         writer.append("\\end{table}" + System.
            getProperty("line.separator"));
168         writer.append("\\end{landscape}" + System.
            getProperty("line.separator"));
169         writer.append("\\end{document}" + System.
            getProperty("line.separator"));
170
171     } catch (IOException e) {
172         e.printStackTrace();
173     }
174 }
175
176 }
177
178

```

```
1 package classes;
2 /**
3  * * created by lgcaobianco on 21/03/18 **
4  */
5
6 import java.io.BufferedReader;
7 import java.io.FileReader;
8 import java.io.IOException;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class LeInformacoes {
13
14     private String nomeArquivo;
15     private String formato;
16     private String separadorValor;
17
18     private String getNomeArquivo() {
19         return nomeArquivo;
20     }
21
22     private String getFormato() {
23         return formato;
24     }
25
26     private String getSeparadorValor() {
27         return separadorValor;
28     }
29
30     public LeInformacoes(String nomeArquivo, String
31     formato) {
32         this.nomeArquivo = nomeArquivo;
33         this.formato = formato;
34         switch (formato) {
35             case ".csv":
36                 this.separadorValor = ",";
37                 break;
38             case ".txt":
39                 this.separadorValor = " ";
40                 break;
41             default:
42                 System.out.println("Formato ainda não
43                 suportado");
44                 System.exit(1);
45                 break;
46         }
47     }
48 }
```

```
49
50     public List<double[]> extrairPontos() {
51         List<double[]> matrizPontos = new ArrayList<double
52         []>();
53         String linhaLida = "";
54         BufferedReader stream = null;
55         try {
56             stream = new BufferedReader(new FileReader(
57             getNomeArquivo() + getFormato()));
58             while ((linhaLida = stream.readLine()) != null
59             ) {
60                 String[] temporario = linhaLida.split(
61                 getSeparadorValor());
62                 double[] numerosSeparados = new double[
63                 temporario.length];
64                 for (int i = 0; i < temporario.length; i++
65                 ) {
66                     numerosSeparados[i] = Double.
67                     parseDouble(temporario[i]);
68                 }
69                 matrizPontos.add(numerosSeparados);
70             }
71         } catch (IOException e) {
72             e.printStackTrace();
73             System.exit(1);
74         } finally {
75             if (stream != null) {
76                 try {
77                     stream.close();
78                 } catch (IOException e) {
79                     e.printStackTrace();
80                     System.exit(1);
81                 }
82             }
83         }
84     }
85     return matrizPontos;
86 }
```