Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Algoritmos para Bioinformática I - Primeiro Semestre 2020

Trabalhos Práticos 3 - Mineração de Dados utilizando álgebra linear

Aluno: Guilherme de Abreu Lima Buitrago Miranda - Matrícula: 2018054788

1. Introdução

Um dos principais fatores discutidos em consultas médicas de rotina é representado pelos chamados fatores de risco, ou seja, aspectos da saúde do paciente que aumentam a probabilidade de desenvolvimento de outra doença ou piora no quadro clínico já existente. Contudo, é pertinente questionar: como são encontrados os fatores de risco para determinada doença?

Além de observar características biológicas para definir tais correlações, profissionais da saúde podem fazer uso de ferramentas computacionais para encontrar ligações latentes entre certos quadros clínicos. Em particular, soluções recentes usando técnicas de mineração de dados e álgebra linear têm obtido bastante sucesso ao tentar encontrar tais relacionamentos, e serão objeto de estudo no presente trabalho.

Não obstante, é importante destacar que tais técnicas não ficam restritas apenas ao contexto biológico e, neste mesmo trabalho, serão usadas para dividir em grupos e classificar tipos de vinho, por exemplo. Assim, em primeiro lugar, explica-se as entradas esperadas e as saídas obtidas. Posteriormente, aborda-se o sentido de cada uma das informações obtidas, assim como os métodos utilizados pelo algoritmo e suas respectivas fundamentações teóricas. Por último, debate-se as conclusões obtidas a partir do trabalho e suas implicações no cotidiano.

2. Entradas e Saídas

Como entradas para o trabalho, tem-se os seguintes datasets, em formato *.data,* presentes no repositório de aprendizado de máquina da UCI: *Cancer Wisconsin (Prognostic)*, *Wine* e *Iris.* Assim, para integrá-los com a ferramenta

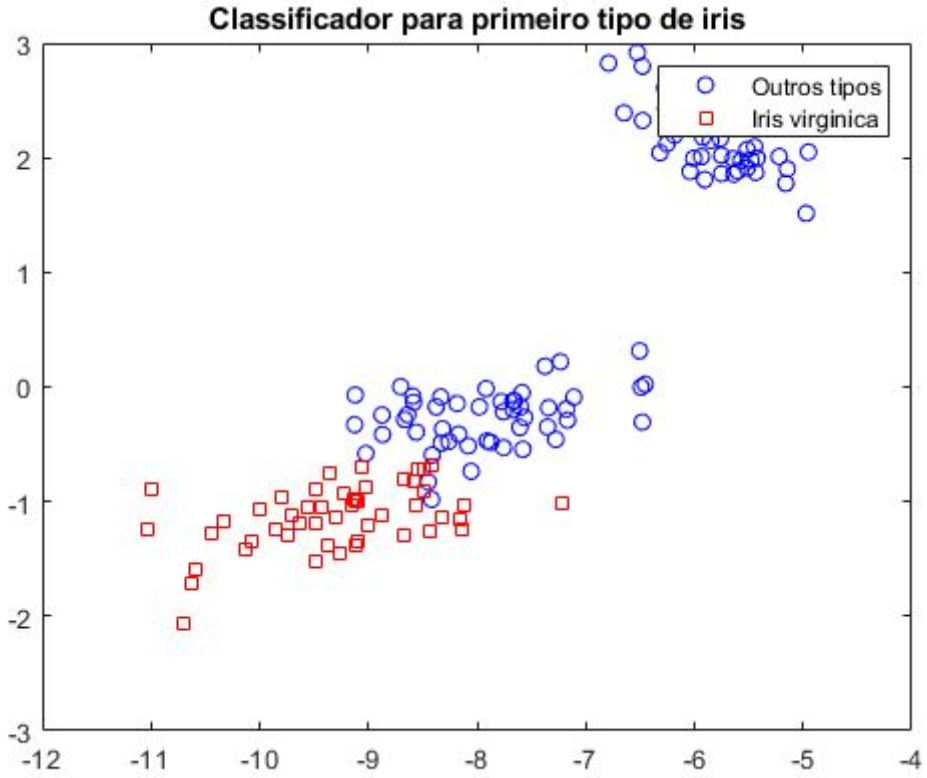
MATLAB, é necessário utilizar a função "*Import Data*" do programa, selecionando o arquivo desejado no diretório correspondente.

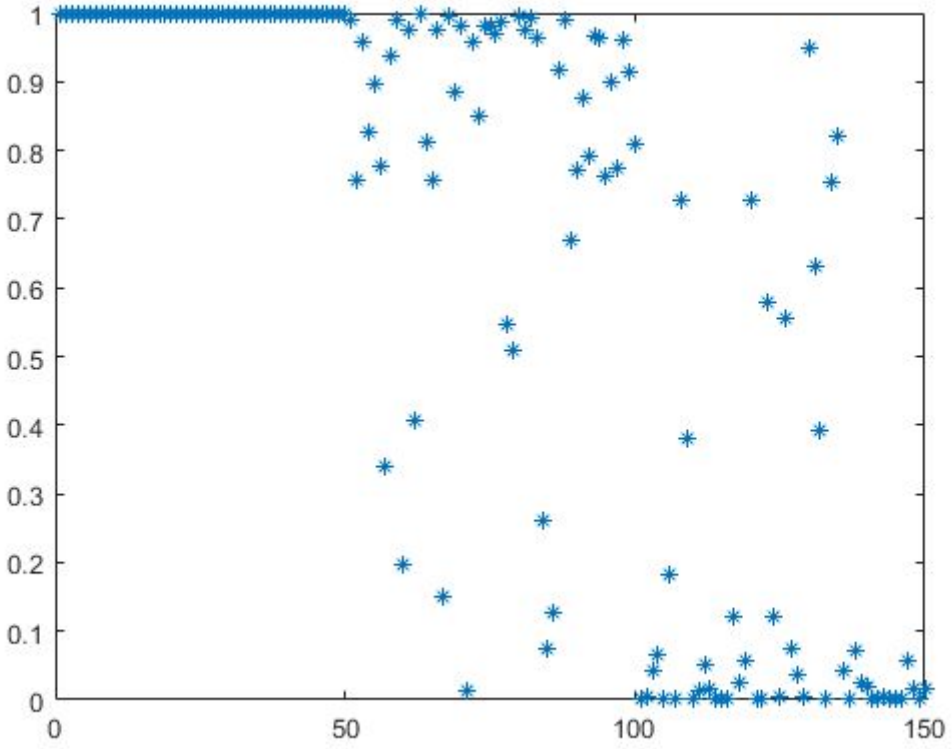
Em seguida, é preciso selecionar a parte desejada do arquivo, ou seja, exclui-se as colunas que fazem a identificação dos indivíduos (1ª e 2ª coluna do dataset *Cancer Wisconsin (Prognostic),* 1ª coluna do dataset *Wine* e 5ª coluna do dataset *Iris)*. Por fim, é necessário alterar o modo do *import* de *table* para *matrix* a fim de que, dessa forma, seja possível realizar operações de álgebra linear com os dados fornecidos.

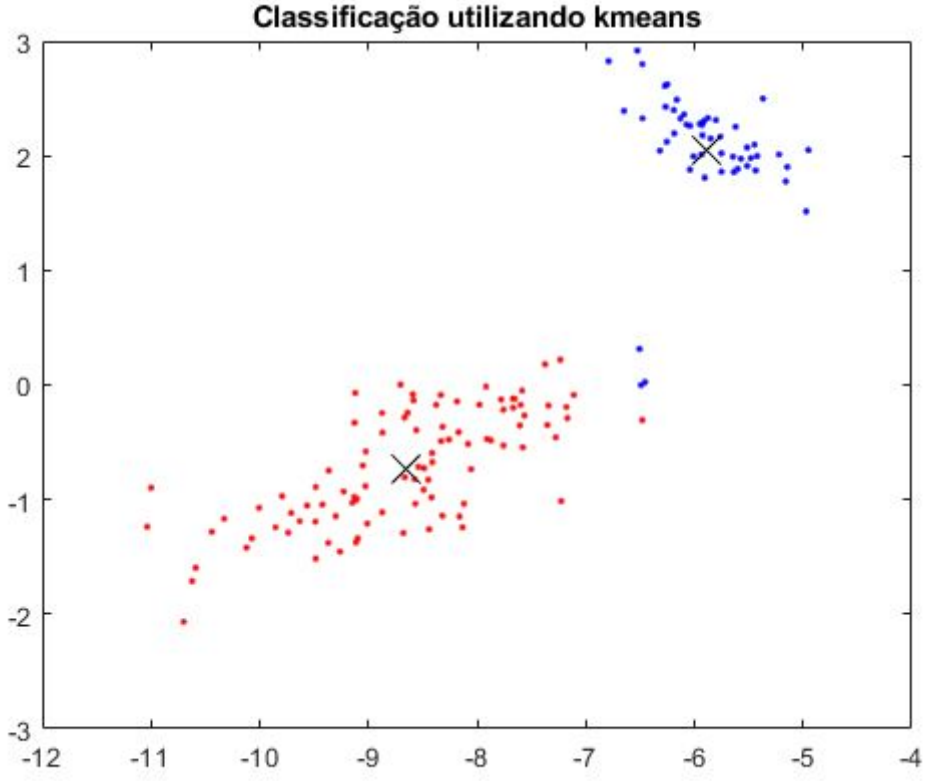
Para cada um dos *Datasets,* são esperadas três saídas: um gráfico com um classificador para os tipos (de íris, de vinho e de câncer), um gráfico de probabilidades de cada indivíduo estar em um grupo específico e um gráfico de classificação usando o algoritmo *kmeans.* Para o último, além dos indivíduos pintados com a respectiva cor de seu grupo, também está plotado a centróide obtida pela execução da solução.

Assim, obtém-se para:

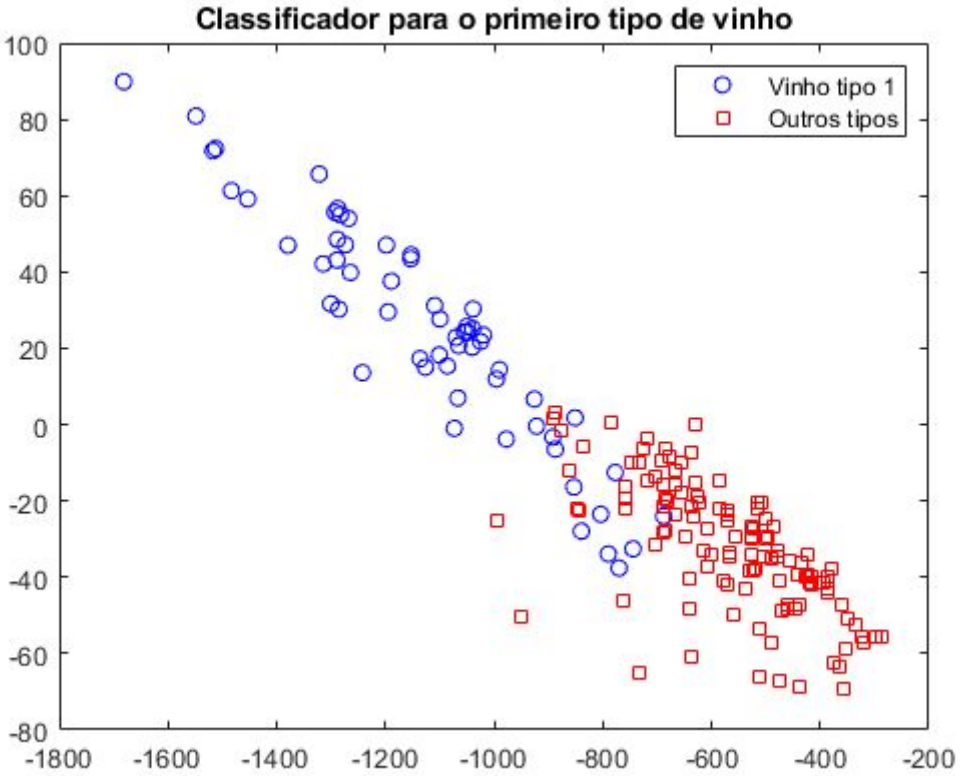
a) *Cancer Wisconsin (Prognostic)*

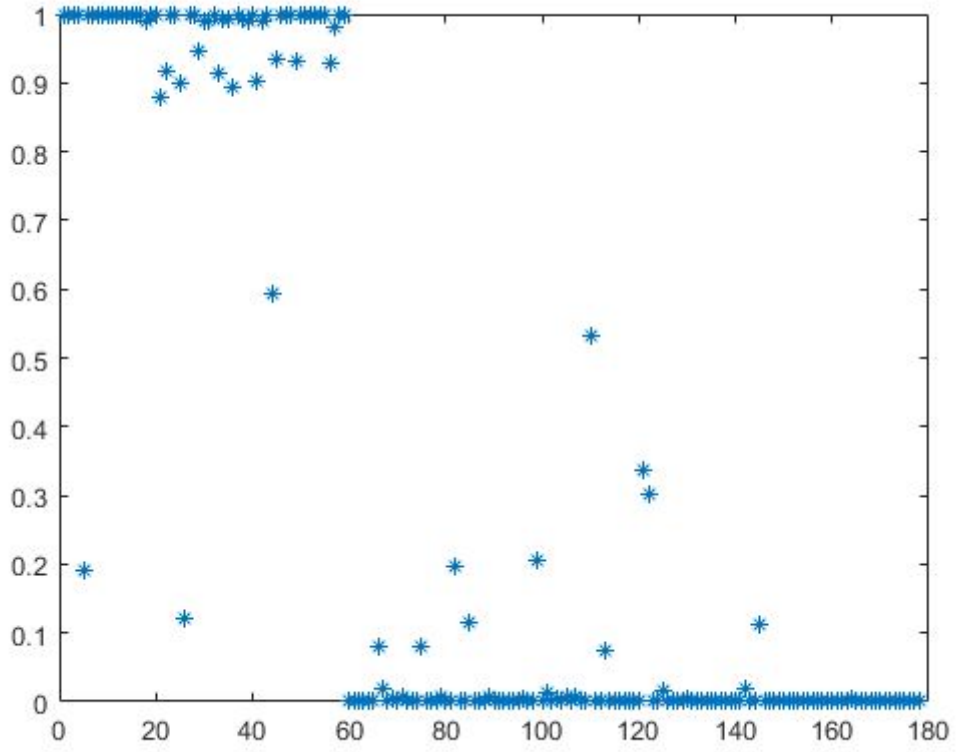


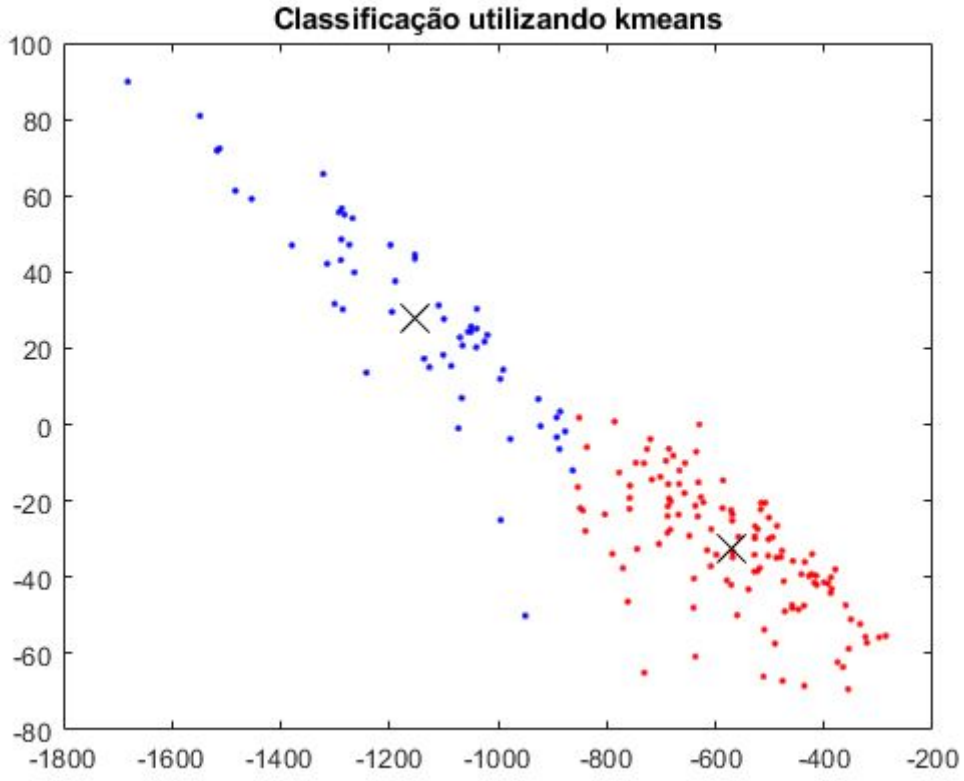

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150


Classificação utilizando kmeans
3
2
1
0
-1
-2
-3
-12
-11
-10
-9
-8
-7
-6
-5
-4

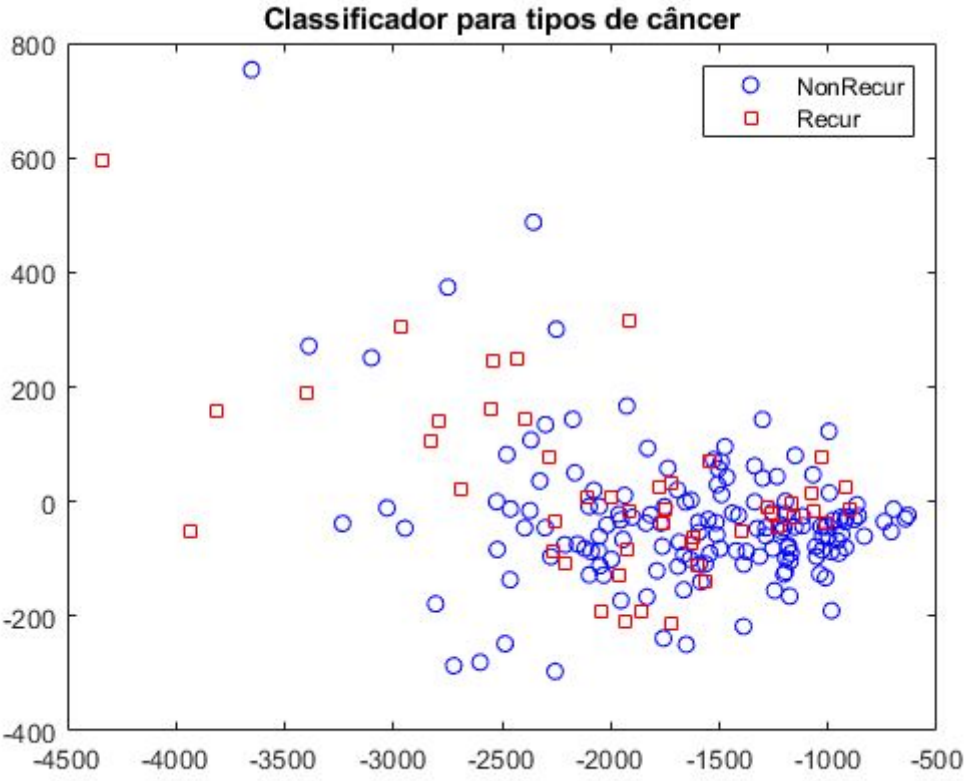
b) *Wine*

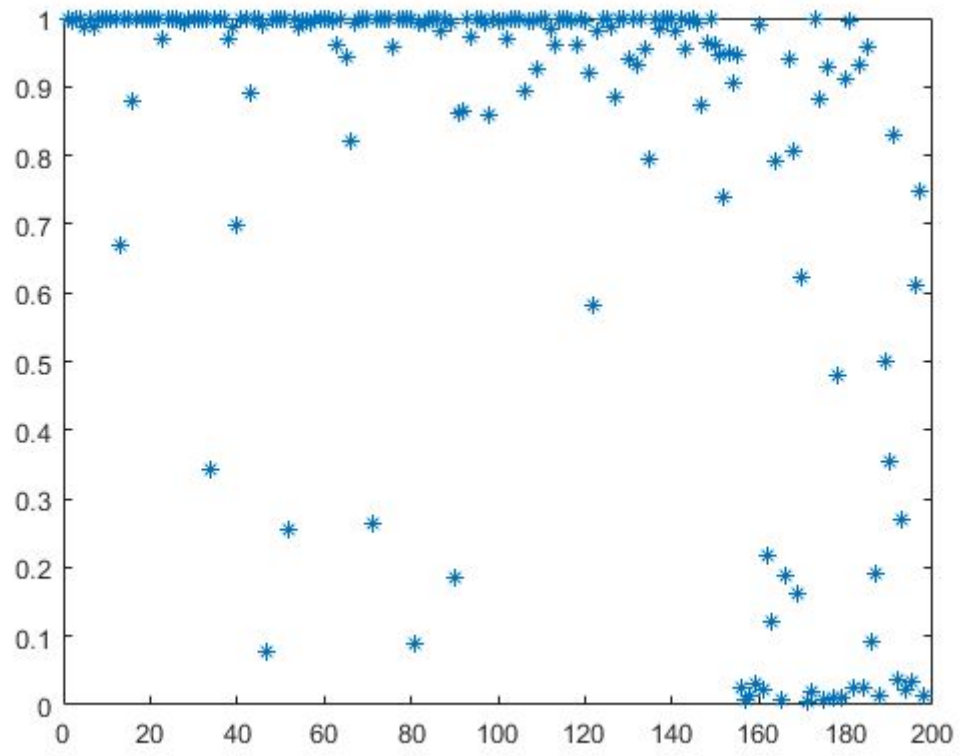

Classificador para o primeiro tipo de vinho
Vinho tipo 1
Outros tipos
100
80
60
40
20
0
-20
-40
-60
-80
-1800
-1600
-1400
-1200
-1000
-800
-600
-400
-200

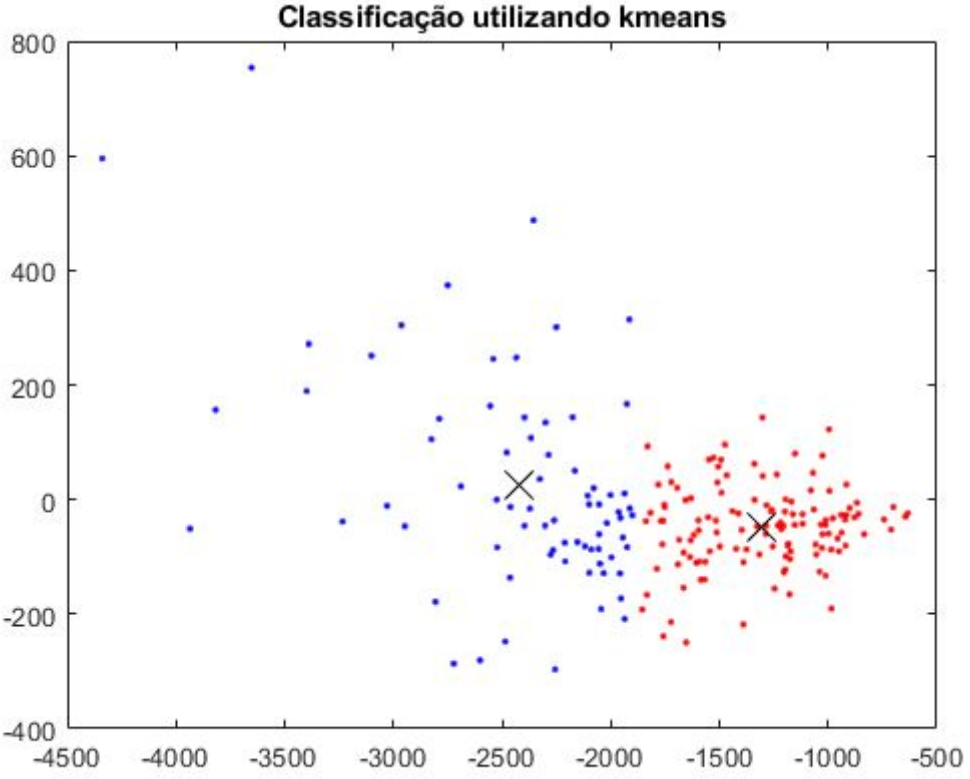

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
160
180


Classificação utilizando kmeans
100
80
60
40
20
0
-20
-40
-60
-80
-1800
-1600
-1400
-1200
-1000
-800
-600
-400
-200

c) *Iris*


Classificador para tipos de câncer
NonRecur
Recur
800
600
400
200
0
-200
-400
-4500
-4000
-3500
-3000
-2500
-2000
-1500
-1000
-500


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
160
180
200


Classificação utilizando kmeans
800
600
400
200
0
-200
-400
-4500
-4000
-3500
-3000
-2500
-2000
-1500
-1000
-500

3. Metodologia

Após obter a representação dos indivíduos conforme seu respectivo conjunto de *features* (características) como no *vector space model*, o primeiro passo da solução é realizar a decomposição por valores singulares (SVD). Posteriormente, para fins de consulta, plota-se os valores relativos de S, com o objetivo de perceber quais são os padrões mais importantes para o problema em questão.

Em sequência, observando os dados já obtidos, estima-se a quantidade de padrões que deseja-se utilizar, assim como a quantidade de grupos e, então, procede-se a separação dos mesmos utilizando o algoritmo *kmeans.* Utilizando os dados já obtidos pela decomposição SVD, o algoritmo em questão procede à separação observando as semelhanças e as diferenças entre as entidades. Quando mais parecidos são os vinhos, por exemplo, maior a probabilidade de ambos serem agrupados no mesmo conjunto. Em contrapartida, se dois pacientes possuírem fatores de risco muito díspares, a chance de serem agrupados juntos é pequena.

Assim, conforme observado nas imagens contidas na seção 2, as três separações obtidas a partir do algoritmo *kmeans* utilizando os *datasets* supracitados foram bastante satisfatórias. Portanto, tem-se um indicativo positivo para a possibilidade de se construir um algoritmo preditivo para tais grupos.

Portanto, por último, cria-se um modelo de regressão logística com o objetivo de classificar uma das categorias da respectiva base de dados trabalhada no momento. Assim, é possível afirmar, dadas determinadas *features* (características), se aquele indivíduo pertence ou não a um grupo A (por exemplo, ao grupo de recuperados do câncer).

4. Conclusões

O trabalho mostrou-se muito valioso para que se observasse como a técnica de decomposição SVD combinada com o algoritmo *kmeans* e o modelo de regressão logística se comporta quando aplicada em diferentes contextos, dentro e fora da área de bioinformática.

Além disso, é interessante observar, na prática, os poderes da regressão logística. Em particular, é curioso levar em consideração que soluções como essas são utilizadas por grandes sistemas, como as pontuações de crédito, a medição de taxas de sucesso de campanhas de *marketing* e a previsão de vendas de um

determinado produto. Assim, nada mais justo do que adaptações para utilizá-la na biologia.

5. Referências Bibliográficas

- Datasets *Cancer Wisconsin (Prognostic), Wine* e *Iris* - http://archive.ics.uci.edu/ml/datasets.php
- Kmeans plot centroid error - MATLAB Answers - https://se.mathworks.com/matlabcentral/answers/416463-kmeans-plot-centroid-error
- Kmeans - Help Center - https://www.mathworks.com/help/stats/kmeans.html

6. Anexos

Script utilizado para a análise do problema *Cancer Wisconsin (Prognostic)*:

```
wpbc = readtable('wpbc.data.csv');
wpbc = sortrows(wpbc, 2)
wpbc = wpbc{:,3:end}
wpbc(isnan(wpbc))=0;
A = wpbc';
[U, S, V] = svd(A);
s = diag(S);
s = s*(1/sum(s));
plot(s, '*');
Aux = S*V';
x = Aux(1, :);
y = Aux(2, :);
plot(x(1:151), y(1:151), 'ob')
hold on
plot(x(152:end), y(152:end), 'sr')
title ('Classificador para tipos de câncer')
legend('NonRecur', 'Recur')
hold off

lgCh1 = log(0.9999/(1-0.9999));
lgCh0 = log(0.0001/(1-0.001));
b = zeros(198, 1);
b(1:151) = lgCh1;
b(152:end) = lgCh0;
alpha = A'\b;
```

```
aux = A'*alpha;
num = exp(aux);
p = num./(1+num);
figure
plot(p, '*')

figure
Aux_kmeans = transpose(Aux(1:2, :));
[idx, C] = kmeans(Aux_kmeans, 2);
plot(Aux_kmeans(idx==1,1),Aux_kmeans(idx==1,2),'r.')
hold on
plot(Aux_kmeans(idx==2,1),Aux_kmeans(idx==2,2),'b.')
plot(C(:,1),C(:,2),'kx', 'MarkerSize',15);
title ('Classificação utilizando kmeans')
```

Script utilizado para a análise do problema *Wine*:

```
A = wine';
[U, S, V] = svd(A);
s = diag(S);
s = s*(1/sum(s));
plot(s, '*');
Aux = S*V';
x = Aux(1, :);
y = Aux(2, :);
plot(x(1:59), y(1:59), 'ob')
hold on
plot(x(60:end), y(60:end), 'sr')
title ('Classificador para o primeiro tipo de vinho')
legend('Vinho tipo 1', 'Outros tipos')
hold off

lgCh1 = log(0.9999/(1-0.9999));
lgCh0 = log(0.0001/(1-0.001));
b = zeros(178, 1);
b(1:59) = lgCh1;
b(60:end) = lgCh0;
alpha = A'\b;

aux = A'*alpha;
num = exp(aux);
```

```
p = num./(1+num);
figure
plot(p, '*')

figure
Aux_kmeans = transpose(Aux(1:2, :));
[idx, C] = kmeans(Aux_kmeans, 2);
plot(Aux_kmeans(idx==1,1),Aux_kmeans(idx==1,2),'r.')
hold on
plot(Aux_kmeans(idx==2,1),Aux_kmeans(idx==2,2),'b.')
plot(C(:,1),C(:,2),'kx', 'MarkerSize',15)
title ('Classificação utilizando kmeans')
```

Script utilizado para a análise do problema *Iris*:

```
A = iris';
[U, S, V] = svd(A);
s = diag(S);
s = s*(1/sum(s));
plot(s, '*');
Aux = S*V';
x = Aux(1, :);
y = Aux(2, :);
plot(x(1:100), y(1:100), 'ob')
hold on
plot(x(101:end), y(101:end), 'sr')
title ('Classificador para primeiro tipo de iris')
legend('Outros tipos', 'Iris virginica')
hold off

lgCh1 = log(0.9999/(1-0.9999));
lgCh0 = log(0.0001/(1-0.001));
b = zeros(150, 1);
b(1:100) = lgCh1;
b(101:end) = lgCh0;
alpha = A'\b;

aux = A'*alpha;
num = exp(aux);
p = num./(1+num);
figure
plot(p, '*')
```

```
figure
Aux_kmeans = transpose(Aux(1:2, :));
[idx, C] = kmeans(Aux_kmeans, 2);
plot(Aux_kmeans(idx==1,1),Aux_kmeans(idx==1,2),'r.')
hold on
plot(Aux_kmeans(idx==2,1),Aux_kmeans(idx==2,2),'b.')
plot(C(:,1),C(:,2),'kx', 'MarkerSize',15)
title ('Classificação utilizando kmeans')
```