Universidade Federal do Rio Grande do Sul
Instituto de Informática Aplicada
Biologia Computacional - Lista 8
Prof. Dr. Márcio Dorn

Nome: Vicente Merlo
Matrícula: 244950

Implementei o K-Means seguindo os passos do tutorial, apenas usando a biblioteca numpy para realizar a distância euclidiana. Criei a classe KMeans, que possui os métodos:
generate_first: para gerar a primeira etapa do algoritmo, com pontos aleatórios
generate: para realizar uma iteração dos clusters e predições, conforme os passos 2 e 3 do guia.
add_point: adiciona uma amostra ao KMeans, para ser usada no aprendizado.
E fornece os atributos **predictions** e **clusters**, que são atualizados em cada etapa de iteração e determinam a qual cluster uma amostra pertence, e quais os centróides dos clusters do algoritmo.
A classe é chamada com:

```
means = KMeans(2) # sendo K = 2
means.add_point([1,1,1])
means.generate_first()
means.generate()
print means.predictions
```

Usando o arquivo da etapa7, modificando para usar o meu KMeans ao invés do sklearn, ainda com a pandas, obtive os centróides, com **10 iterações** (ou seja, 10 chamadas de *means.generate*), quando **K=2**:

(centróides no arquivo centroides_k2.txt, muito grande para colocar aqui)

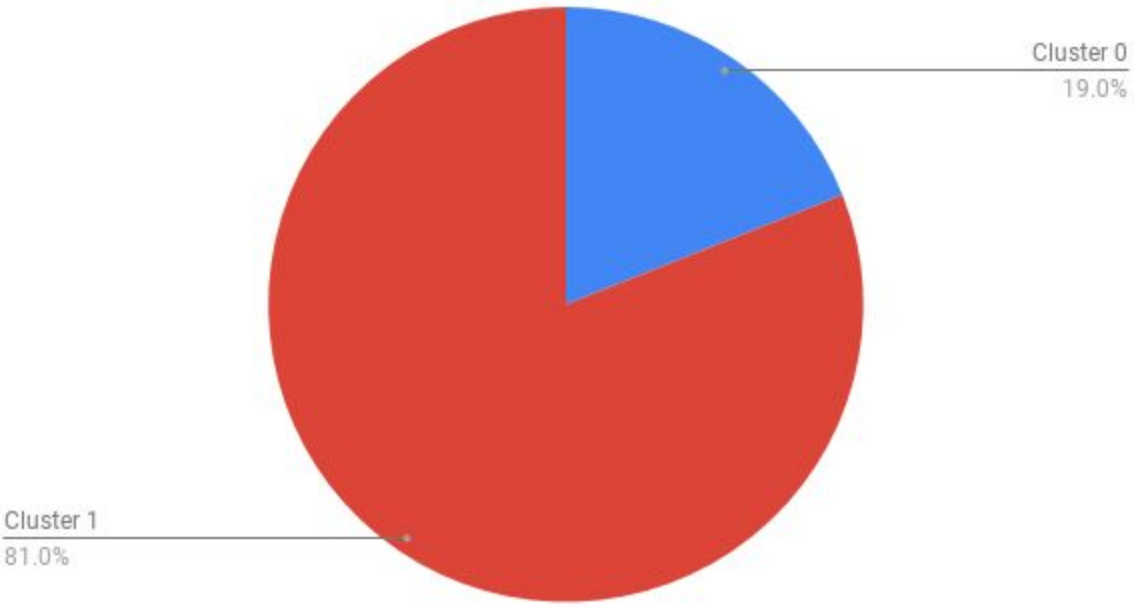
Nesse caso, a predição dividiu as amostras em 2 conjuntos:

cluster 0: 35
cluster1: 37

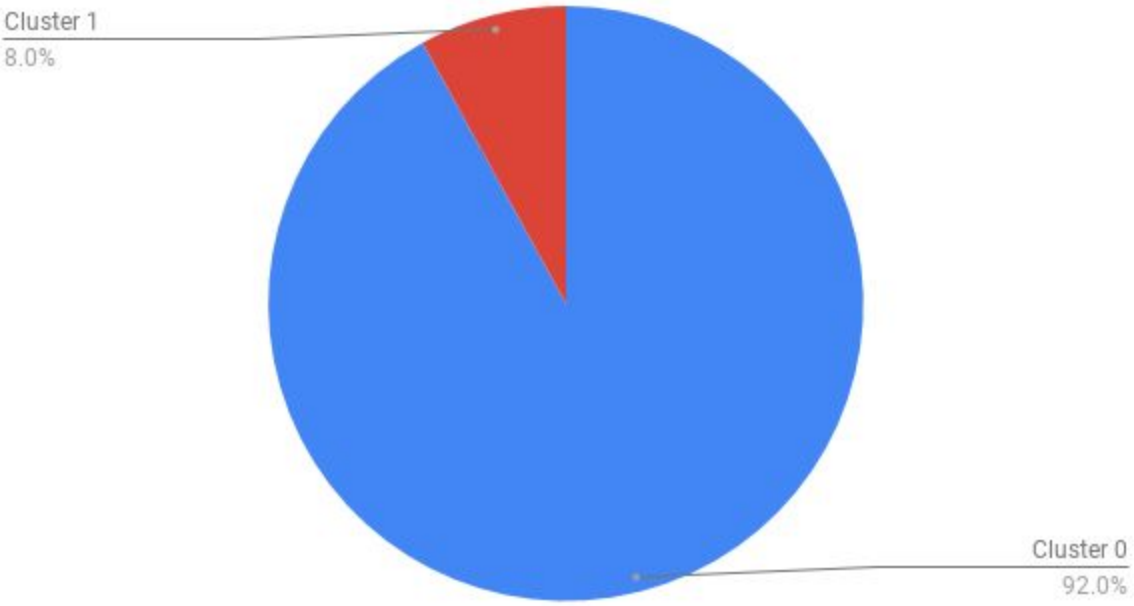
Sendo a distribuição das amostras:

Cluster 0:
ALL: 9.0 - 19.0% do total de ALL das amostras
AML: 23.0 - 92.0% do total de AML das amostras
Cluster 1:
ALL: 38.0 - 81.0% do total de ALL das amostras
AML: 2.0 - 8.0% do total de AML das amostras

ALL



AML



Na etapa passada, considerei “taxa de acerto” como o número de amostras corretamente classificadas pelo algoritmo. Porém, esse entendimento mudou: não existe uma taxa de acerto para os clusters, existem apenas amostras pertecentens ou não a esses clusters. Então, deixo apenas a distribuição.
Como o algoritmo é não-deterministico, com um início aleatório, acho coerente repetí-lo algumas vezes:

Sendo % no total das amostras da respectiva label.

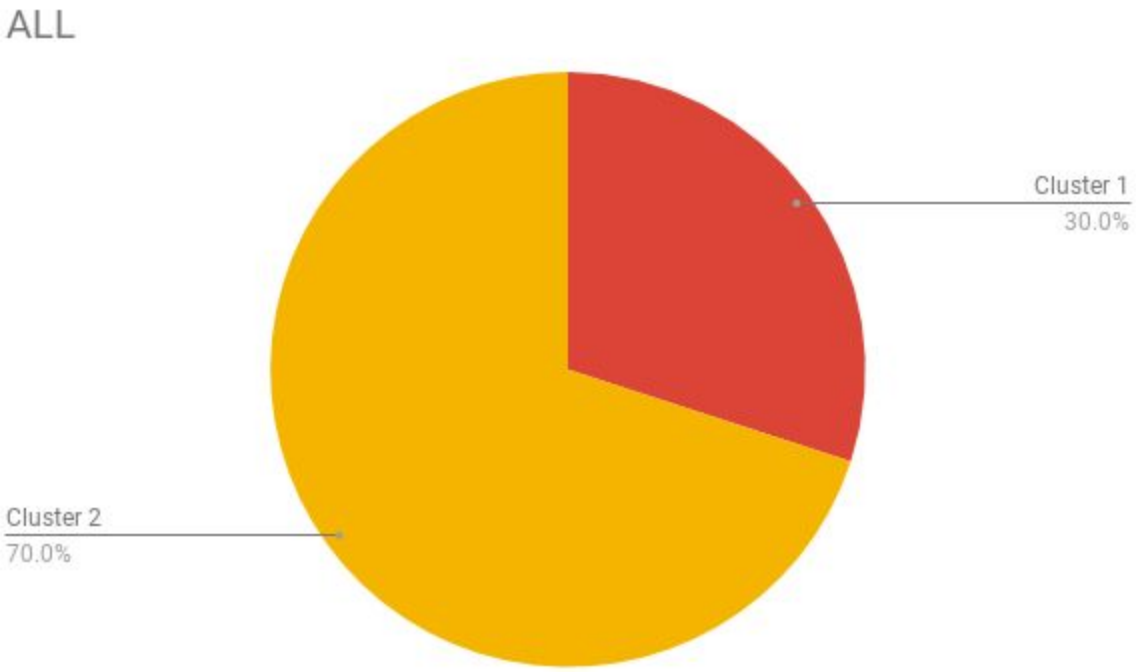
Round 1
Cluster 0:
ALL: 31.0 - 66.0%
AML: 1.0 - 4.0%
Cluster 1:
ALL: 16.0 - 34.0%
AML: 24.0 - 96.0%
Round 2
Cluster 0:
ALL: 47.0 - 100.0%
AML: 1.0 - 4.0%
Cluster 1:
ALL: 0.0 - 0.0%
AML: 24.0 - 96.0%
Round 3
Cluster 0:
ALL: 36.0 - 77.0%
AML: 0.0 - 0.0%
Cluster 1:
ALL: 11.0 - 23.0%
AML: 25.0 - 100.0%
Round 4
Cluster 0:
ALL: 1.0 - 2.0%
AML: 0.0 - 0.0%
Cluster 1:
ALL: 46.0 - 98.0%
AML: 25.0 - 100.0%

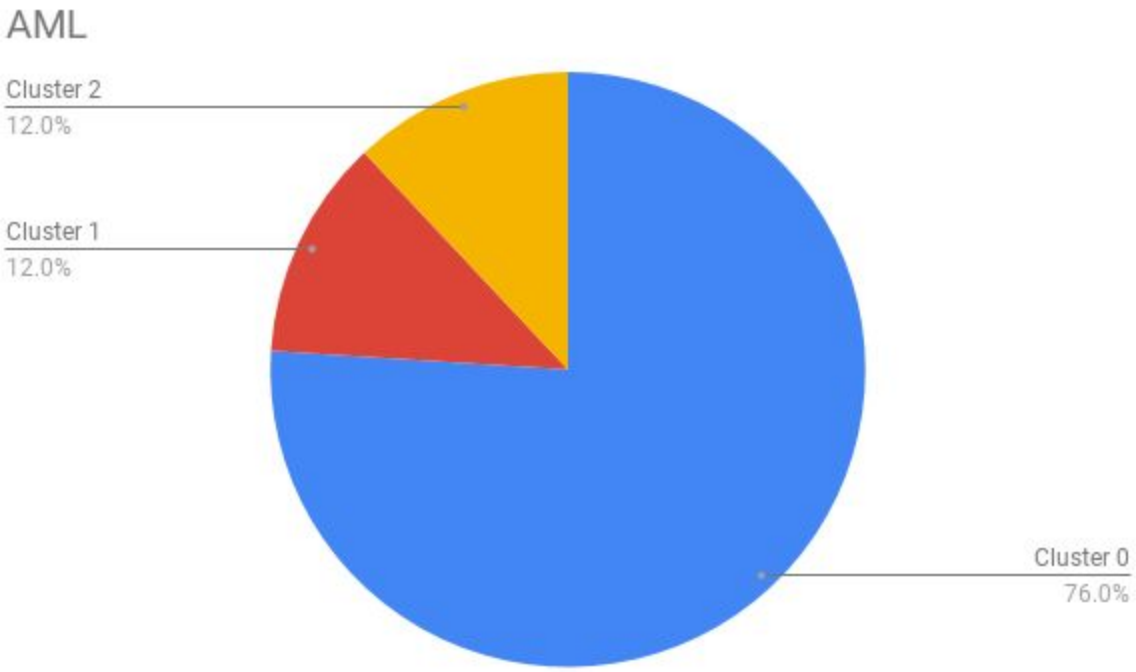
Agora, com **K=3**, mantendo **10 iterações**, temos os centróides:

(centróides no arquivo centroides_k3.txt, muito grande para colocar aqui)

E os resultados foram:

Cluster 0:
ALL: 0.0 - 0.0%
AML: 19.0 - 76.0%
Cluster 1:
ALL: 14.0 - 30.0%
AML: 3.0 - 12.0%
Cluster 2:
ALL: 33.0 - 70.0%
AML: 3.0 - 12.0%





Sendo % no total das amostras da respectiva label.
Também rodei mais 4x o algoritmo, já que ele é não-determinístico.

Round 1
Cluster 0:
  ALL: 24.0 - 51.0%
  AML: 22.0 - 88.0%
Cluster 1:
  ALL: 0.0 - 0.0%
  AML: 1.0 - 4.0%
Cluster 2:
  ALL: 23.0 - 49.0%
  AML: 2.0 - 8.0%
Round 2
Cluster 0:
  ALL: 3.0 - 6.0%
  AML: 6.0 - 24.0%
Cluster 1:
  ALL: 23.0 - 49.0%
  AML: 3.0 - 12.0%
Cluster 2:
  ALL: 21.0 - 45.0%
  AML: 16.0 - 64.0%

Round 3
Cluster 0:
  ALL: 21.0 - 45.0%
  AML: 2.0 - 8.0%
Cluster 1:
  ALL: 24.0 - 51.0%
  AML: 0.0 - 0.0%
Cluster 2:
  ALL: 2.0 - 4.0%
  AML: 23.0 - 92.0%
Round 4
Cluster 0:
  ALL: 22.0 - 47.0%
  AML: 2.0 - 8.0%
Cluster 1:
  ALL: 2.0 - 4.0%
  AML: 21.0 - 84.0%
Cluster 2:
  ALL: 23.0 - 49.0%
  AML: 2.0 - 8.0%

Então, a classificação quando K=2 deu origem a dois grupos bem distintos, chegando a classificar corretamente 100% dos AML. Com K=3, também há clusters que têm distinção grande, ocorrendo casos em que AML é classificado fora dele corretamente 100% dos casos (ou seja, 0% de ocorrências de AML no cluster), e 92% dentro de outro cluster do mesmo round.