

CURSO BÁSICO DE
INTELIGÊNCIA
ARTIFICIAL E
BATE-PAPO COM
CONVIDADOS
ESPECIAIS



INTELIGÊNCIA ARTIFICIAL PARA TODOS

DE 08/06 A 12/08



COM OS PROFESSORES DO
LABORATÓRIO ARIA/UFPB:
TELMO FILHO, THAÍS
GAUDENCIO E YURI
MALHEIROS

 Centro de Informática
UFPB

 Departamento de
ESTATÍSTICA

 artificial
intelligence
applications

- CURSO SEM PRÉ-REQUISITOS
- [HTTP://ARIA.CI.UFPB.BR/IAPARATODOS](http://aria.ci.ufpb.br/iaparatos)
- INSCRIÇÃO PARA CERTIFICADO - DE 01/06
A 07/06: [HTTP://BIT.LY/SIGEVENTOS](http://bit.ly/sigeventos)
- ENCONTROS: SEGUNDAS E QUARTAS
- HORÁRIO: 19:00 ÀS 20:00



[Início](#) [Sobre](#) [Projetos](#) [Membros](#) [Parceiros](#) [Publicações](#) [Contato](#)

LABORATÓRIO DE APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

As experiências definem a aprendizagem. Assim, o ARIA constrói experiências para máquinas e para pessoas, formando especialistas na área de inteligência artificial e ciência de dados, desenvolvendo aplicações e pesquisando seus métodos.

SAIBA MAIS

SE INSCREVE E JÁ APERTA NO SININHO, QUE VOCÊS PASSAM A RECEBER AS NOTIFICAÇÕES.

NOSSOS ENCONTROS DURARÃO 1 HORA E, ASSIM QUE POSSÍVEL, DEIXAREMOS OS VÍDEOS GRAVADOS NO CANAL.

NÃO PRECISA SE PREOCUPAR EM ESTAR LIGADO ÀS 19:00, MAS ESTANDO, ROLA TIRAR DÚVIDA E PARTICIPAR, O QUE JÁ DEIXA A AULA MAIS ANIMADA.

**SOBRE O MATERIAL DE ACOMPANHAMENTO: O ALUNO PRECISA SE LOGAR EM:
CLASSROOM.GOOGLE.COM**

DEPOIS, CLICAR EM PARTICIPAR DA TURMA (ÍCONE COM UM MAIS +)

POR FIM, ENTRAR COM O CÓDIGO DA TURMA: PXV3ANW

TAMBÉM EM: [HTTPS://ARIA.CI.UFPB.BR/IA-PARA-TODOS-MATERIAL/](https://aria.ci.ufpb.br/ia-para-todos-material/)

ESPERAMOS QUE VOCÊS REALMENTE CURTAM O CURSO E APROVEITEM-NO AO MÁXIMO. NÃO DEIXEM DE INTERAGIR CONOSCO, TAMBÉM, POR E-MAIL OU MENSAGEM NO NOSSO INSTAGRAM (@APRENDIZAGEMDEMAQUINA)

Aprendizagem não-supervisionada

O algoritmo de aprendizagem recebe um conjunto com exemplos cujos rótulos não são conhecidos

O objetivo é encontrar padrões previamente desconhecidos nos dados

Aprendizagem não-supervisionada

Atividades não-supervisionadas incluem:

Agrupamento

Estimação de densidades

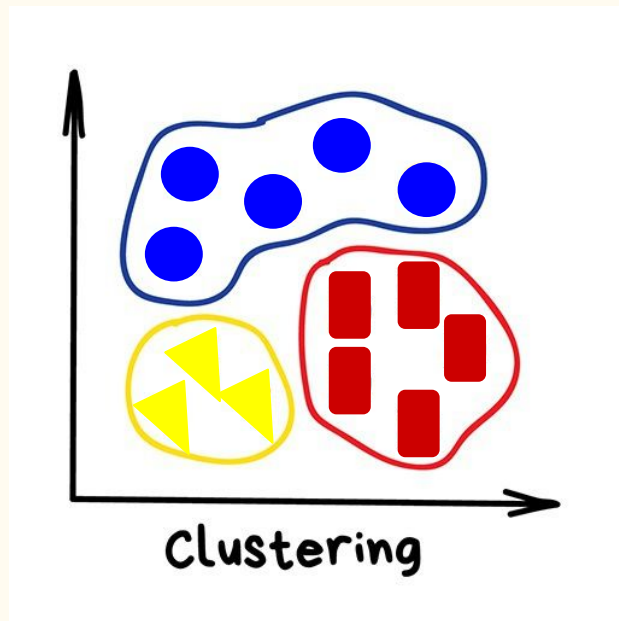
Descoberta de variáveis mais importantes

Agrupamento

Na tarefa de agrupamento, buscamos dividir um conjunto de dados em grupos cujos elementos são

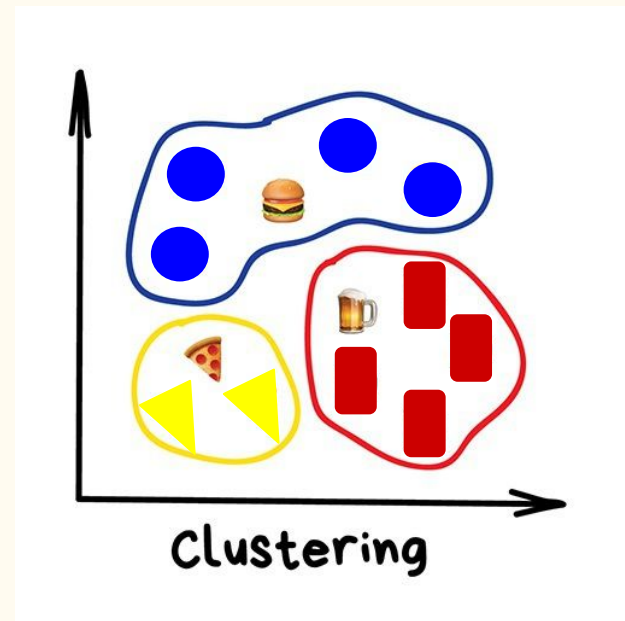
Parecidos entre si

Diferentes dos outros grupos



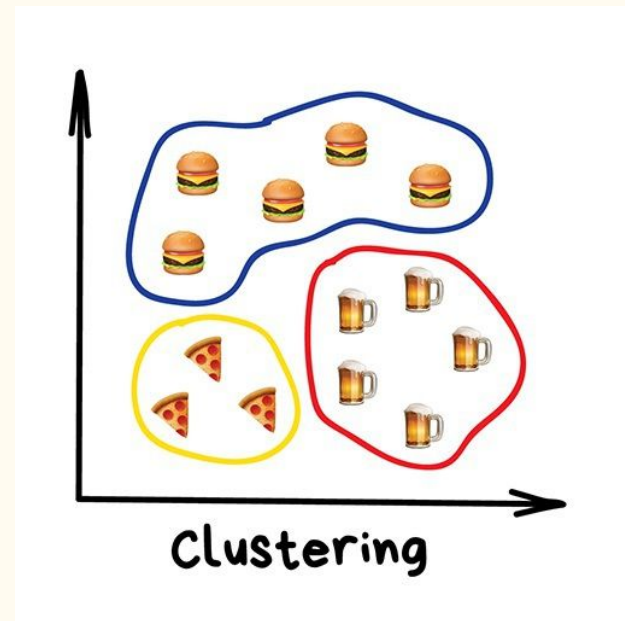
Agrupamento

Após a realização do agrupamento, os grupos encontrados podem ser **investigados** para que os padrões sejam descobertos de fato



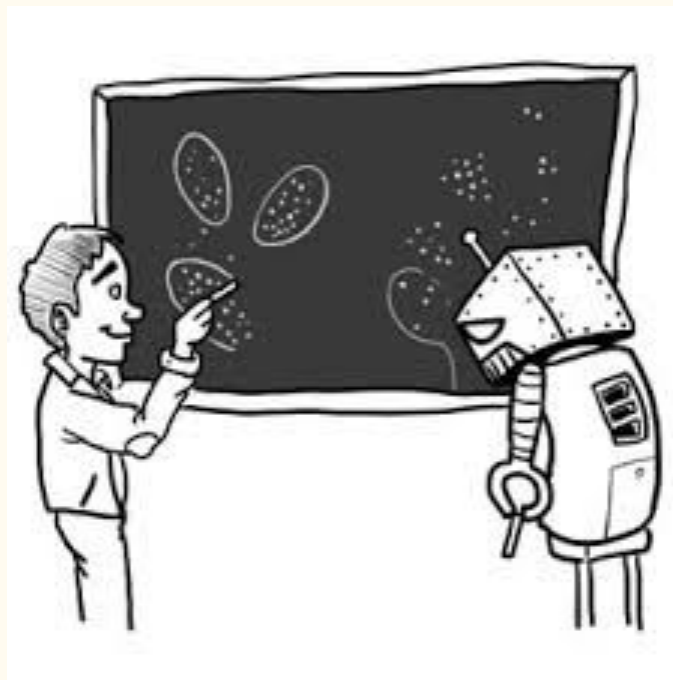
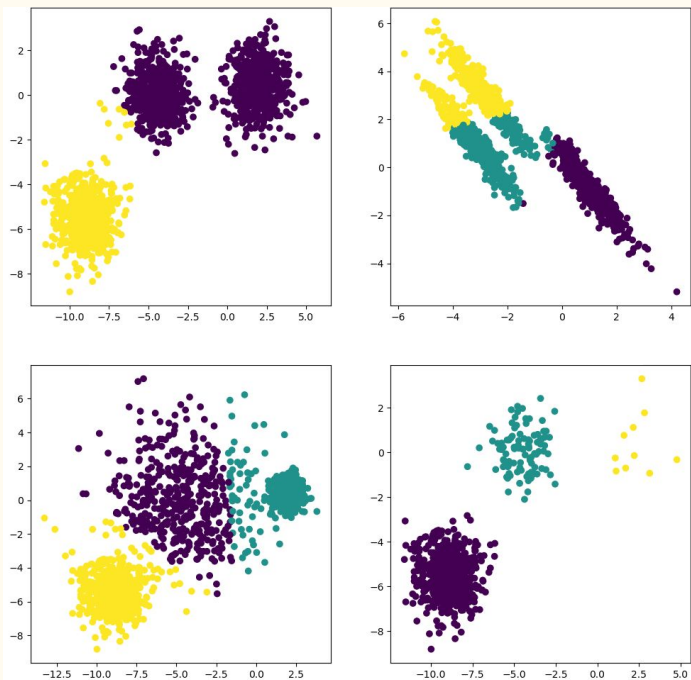
Agrupamento

Após a realização do agrupamento, os grupos encontrados podem ser **investigados** para que os padrões sejam descobertos de fato



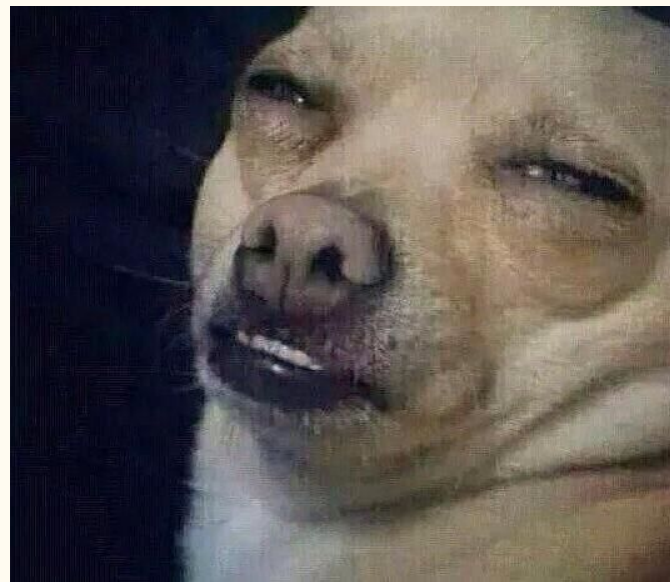
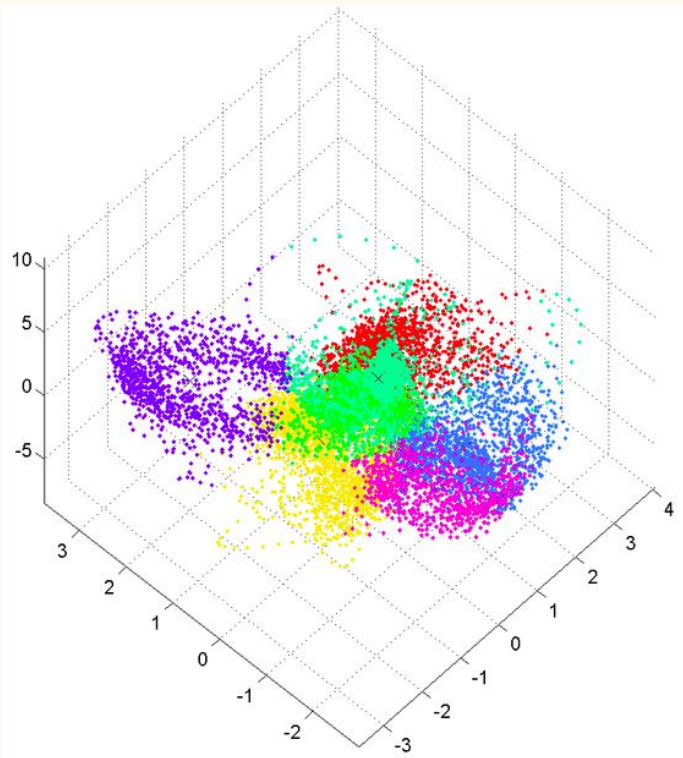
Agrupamento

Em conjuntos de dados 2D, essa tarefa pode ser visualmente resolvida por humanos até com mais sucesso do que a máquina:



Agrupamento

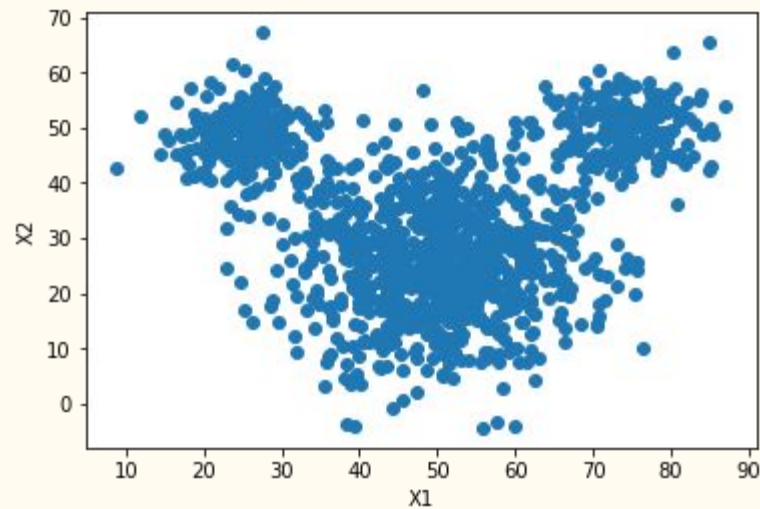
As coisas começam a complicar quando vamos para 3 dimensões ou mais:



Agrupamento

Vamos começar com um exemplo simples:

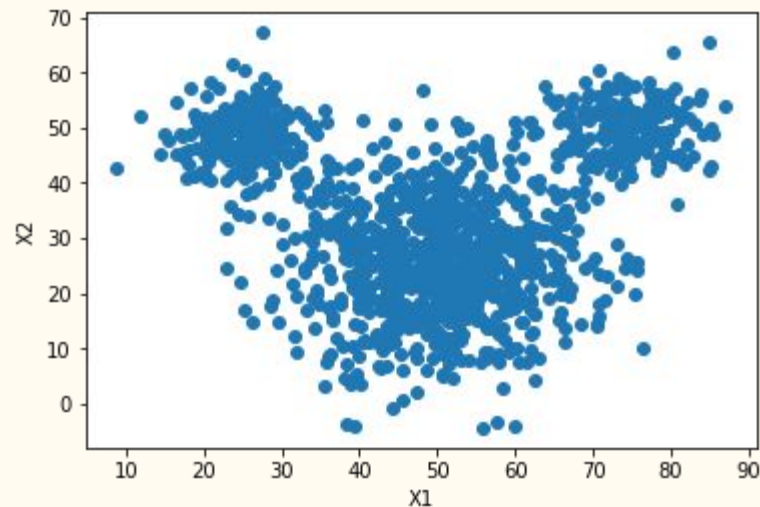
O conjunto ao lado foi gerado de forma simulada



Agrupamento

Vamos começar com um exemplo simples:

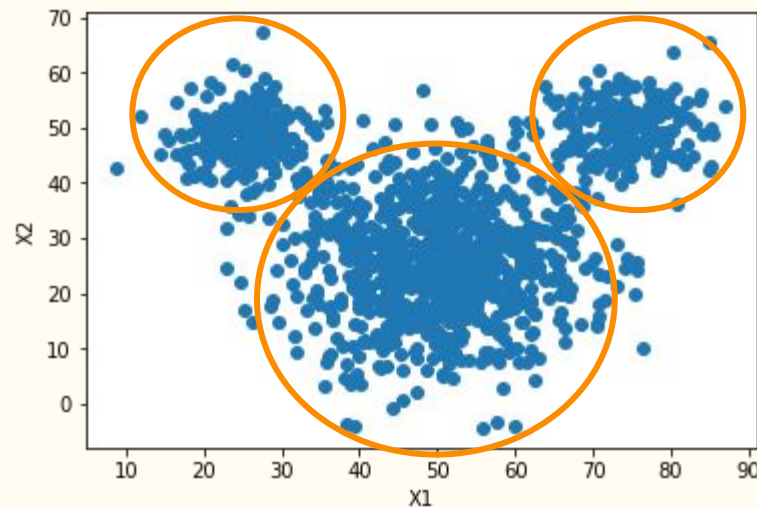
Podemos identificar 3 grupos ao observar os dados



Agrupamento

Vamos começar com um exemplo simples:

Podemos identificar 3 grupos ao observar os dados

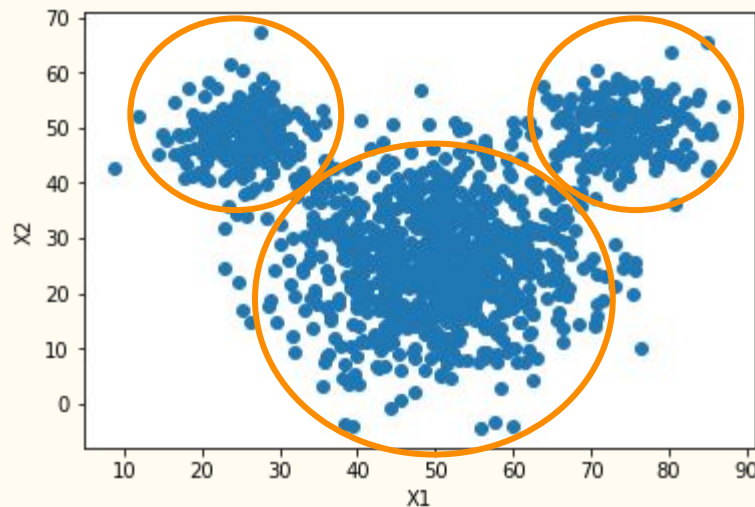


Agrupamento

Vamos começar com um exemplo simples:

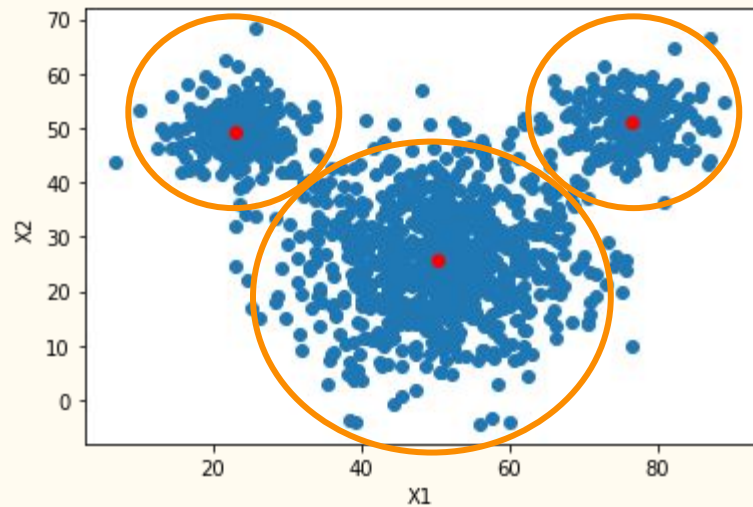
Podemos identificar 3 grupos ao observar os dados

Mas como **automatizar** esse processo?



Como automatizar esse processo?

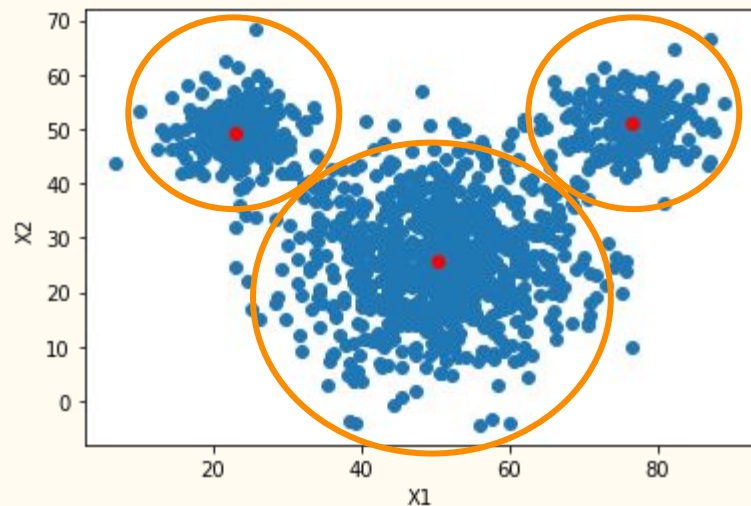
Podemos definir cada grupo como o conjunto dos objetos mais próximos a um elemento **central**:



Como automatizar esse processo?

Fazemos isso **facilmente**, mas o computador precisará **buscar** esses elementos

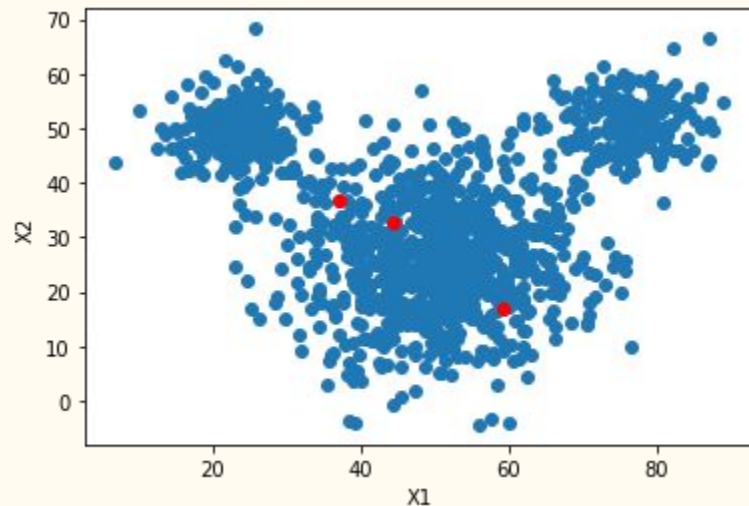
E toda busca precisa de um **ponto de partida**



Como automatizar esse processo?

Um bom ponto de partida é escolher **aleatoriamente** 3 elementos do nosso conjunto

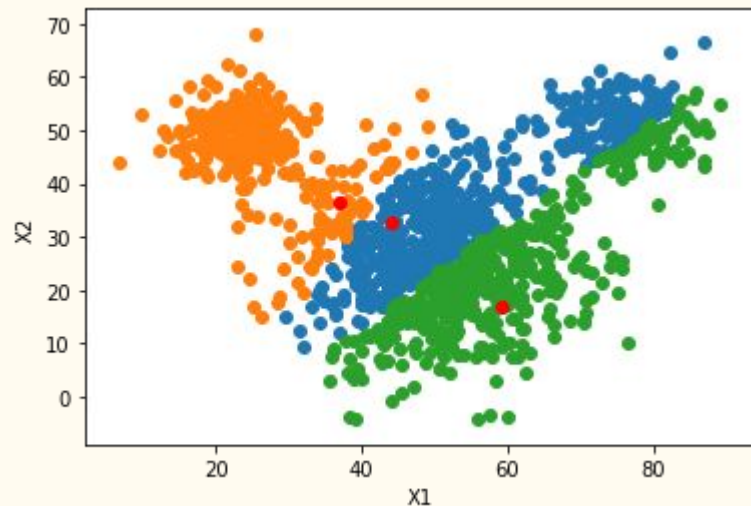
Vamos chamar esses elementos de **médias**



Como automatizar esse processo?

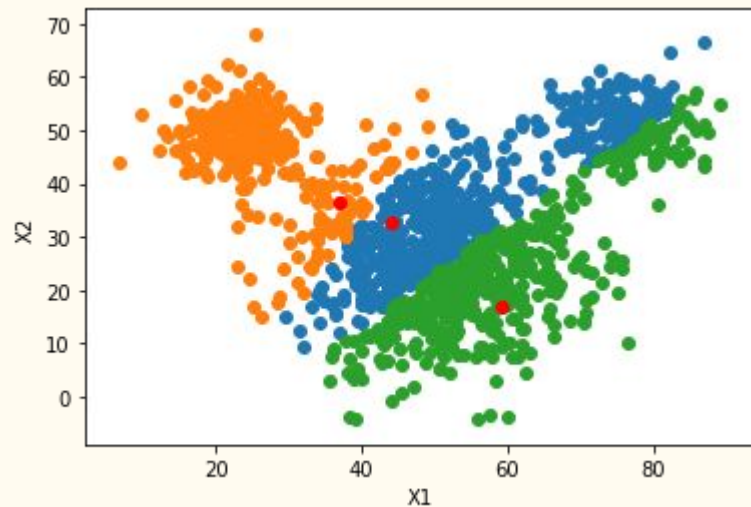
Um bom ponto de partida é escolher **aleatoriamente** 3 elementos do nosso conjunto

Temos como resultado o seguinte agrupamento inicial:



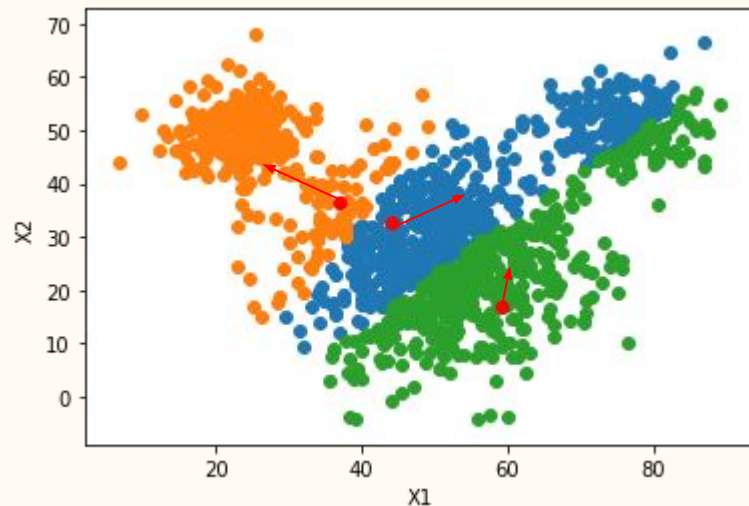
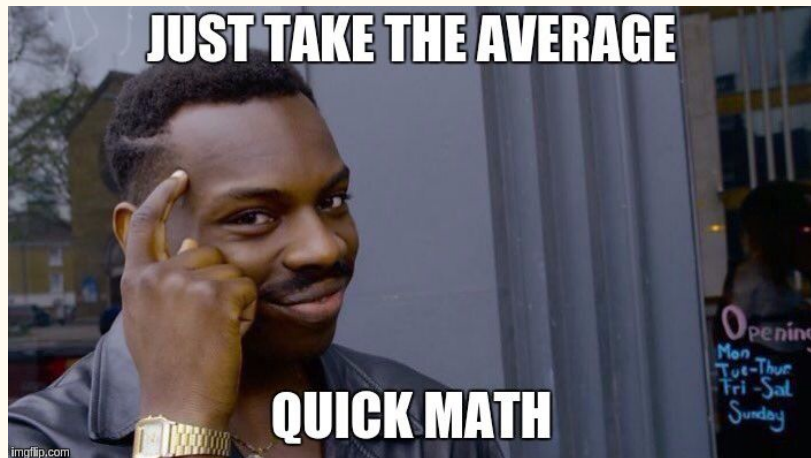
Como automatizar esse processo?

Esse resultado parece bem **ruim**. O que faremos para melhorá-lo?



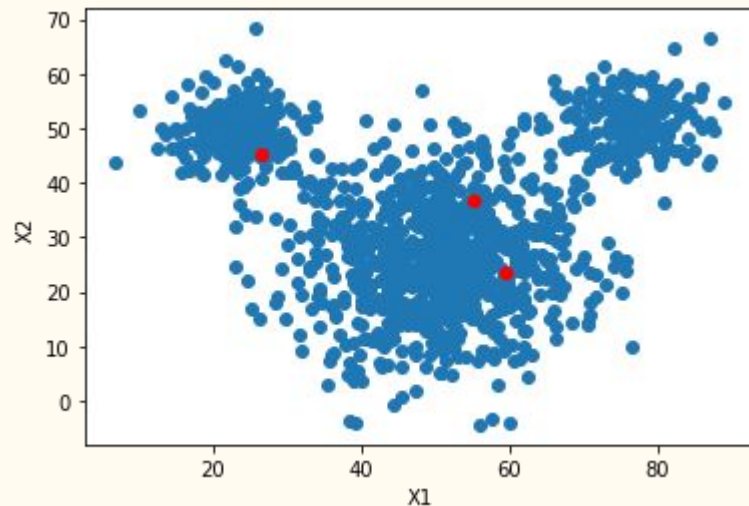
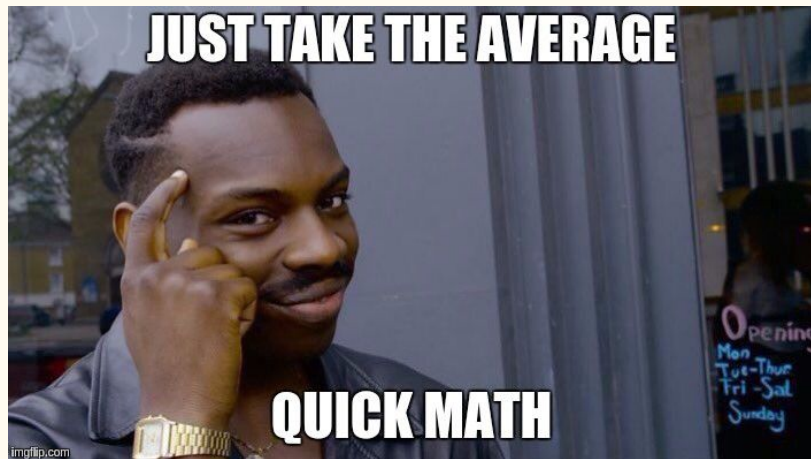
Como automatizar esse processo?

Parece que se substituirmos cada **média** pelo ponto médio de cada área colorida, nos aproximaremos dos grupos desejados



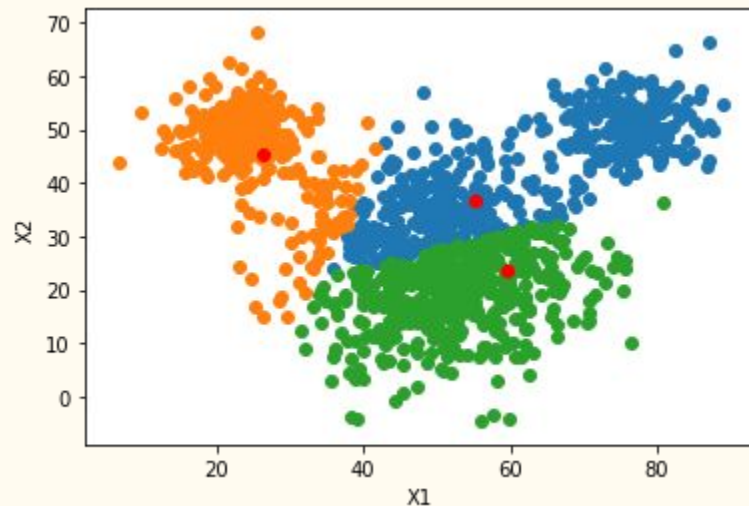
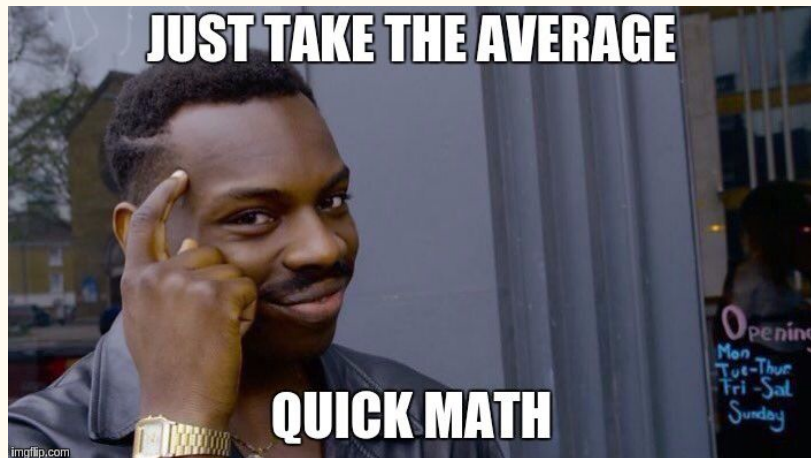
Como automatizar esse processo?

Parece que se substituirmos cada **média** pelo ponto médio de cada área colorida, nos aproximaremos dos grupos desejados



Como automatizar esse processo?

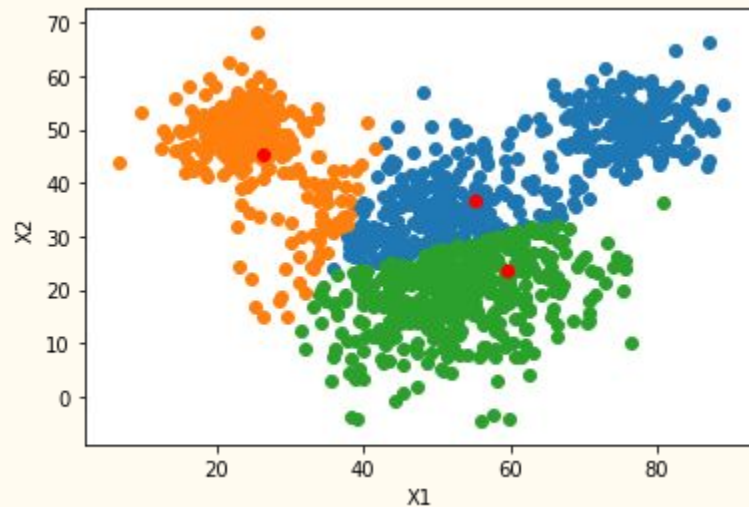
Parece que se substituirmos cada **média** pelo ponto médio de cada área colorida, nos aproximaremos dos grupos desejados



Como automatizar esse processo?

Já melhorou, então vamos seguir com esses passos

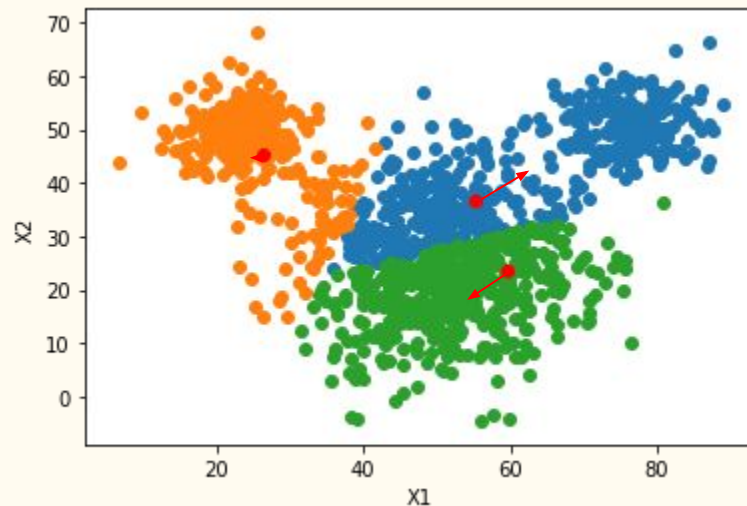
Novamente, vamos substituir nossas **médias** atuais pelos pontos médios das áreas coloridas correspondentes



Como automatizar esse processo?

Já melhorou, então vamos seguir com esses passos

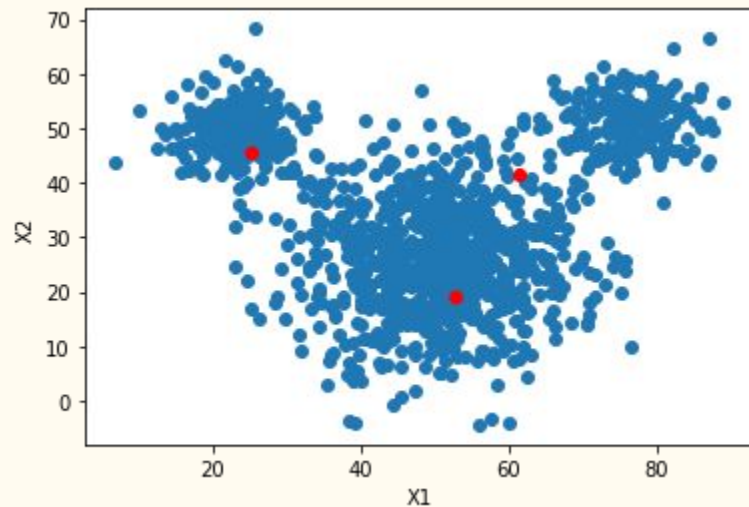
Novamente, vamos substituir nossas **médias** atuais pelos pontos médios das áreas coloridas correspondentes



Como automatizar esse processo?

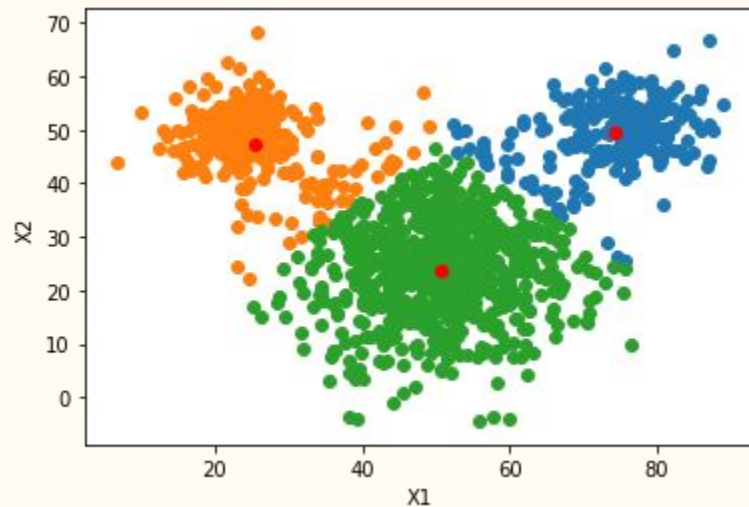
Já melhorou, então vamos seguir com esses passos

Novamente, vamos substituir nossas **médias** atuais pelos pontos médios das áreas coloridas correspondentes



Como automatizar esse processo?

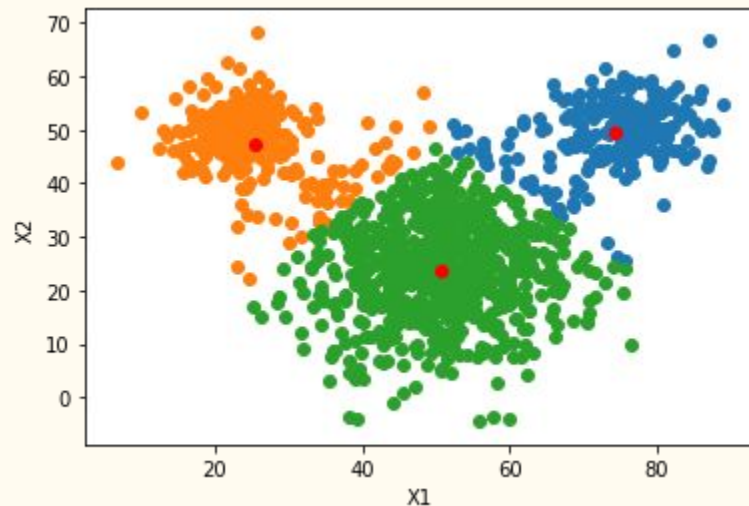
Seguindo esses passos por mais algumas repetições, chegamos ao seguinte resultado:



Como automatizar esse processo?

Note que os grupos não ficaram separados como esperávamos visualmente

Os elementos estão divididos pelas suas distâncias em relação às médias



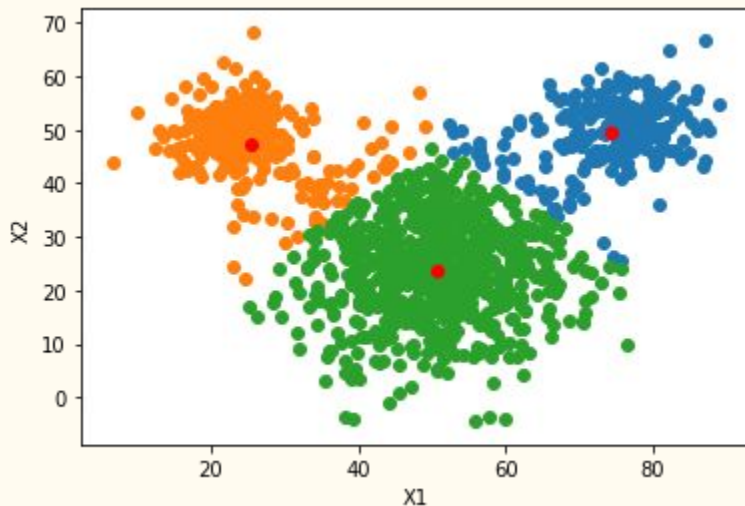
K-médias

O algoritmo K-médias

O algoritmo que acabamos de desenvolver é chamado de K-médias

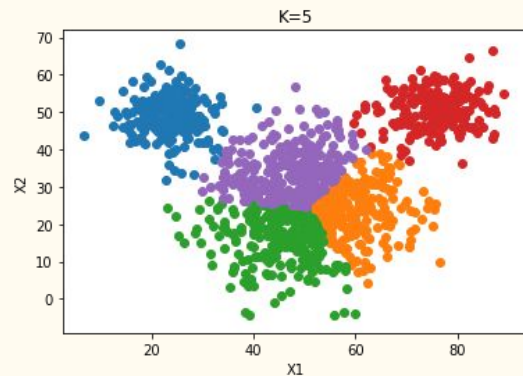
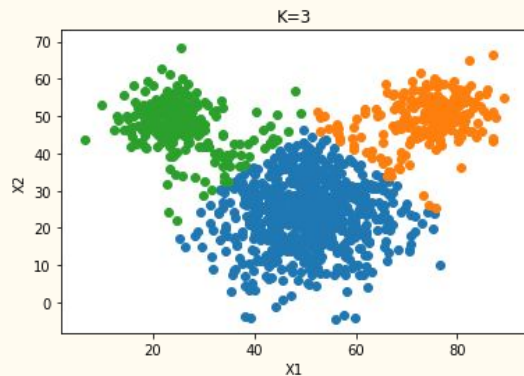
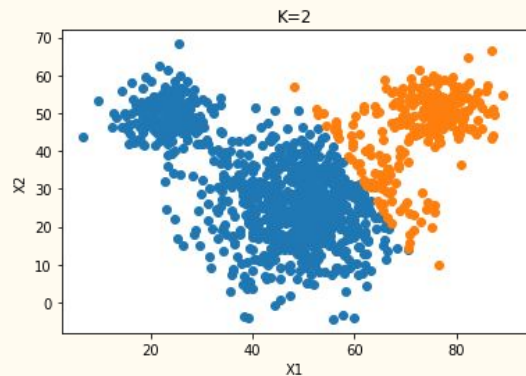
K é um parâmetro que indica o número de grupos que a máquina tentará encontrar nos dados

```
from sklearn.cluster import KMeans  
  
km = KMeans(n_clusters=3) # K=3  
km.fit(X)
```



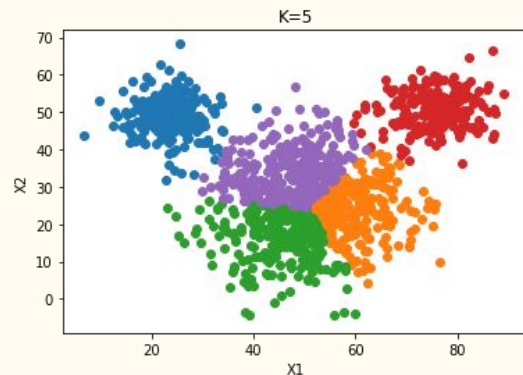
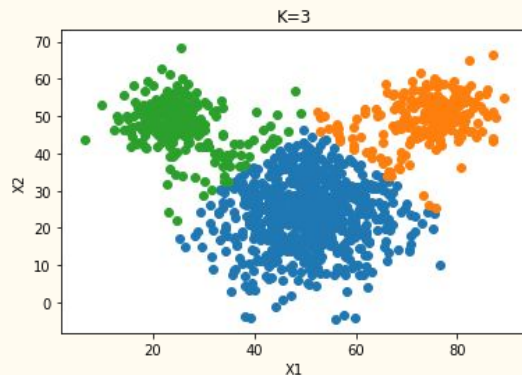
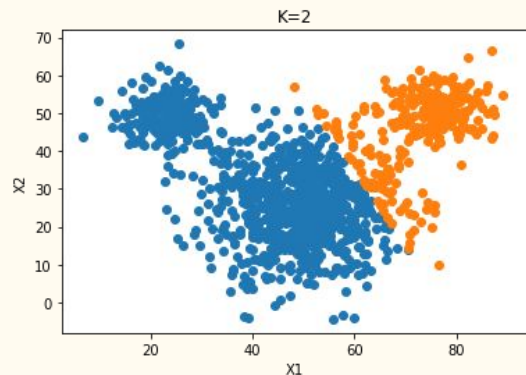
A importância do parâmetro K

O valor de K é extremamente importante e pode ser conhecido a priori por especialistas ou otimizado a partir dos dados



A importância do parâmetro K

Além disso, não importa o real número de grupos nos dados, o algoritmo vai tentar encontrar o número de grupos especificado por K



Prós e contras do K-médias

Prós:

Simplicidade

Velocidade

Contras:

Prefere encontrar grupos circulares e de mesmo tamanho

Dependência do valor de K



Agrupamento hierárquico

Agrupamento hierárquico

Busca construir uma hierarquia com os grupos

Dividido em dois tipos:

Aglomerativo: começa com cada elemento em um grupo separado, unindo-os gradativamente até que todos estejam juntos

Agrupamento hierárquico

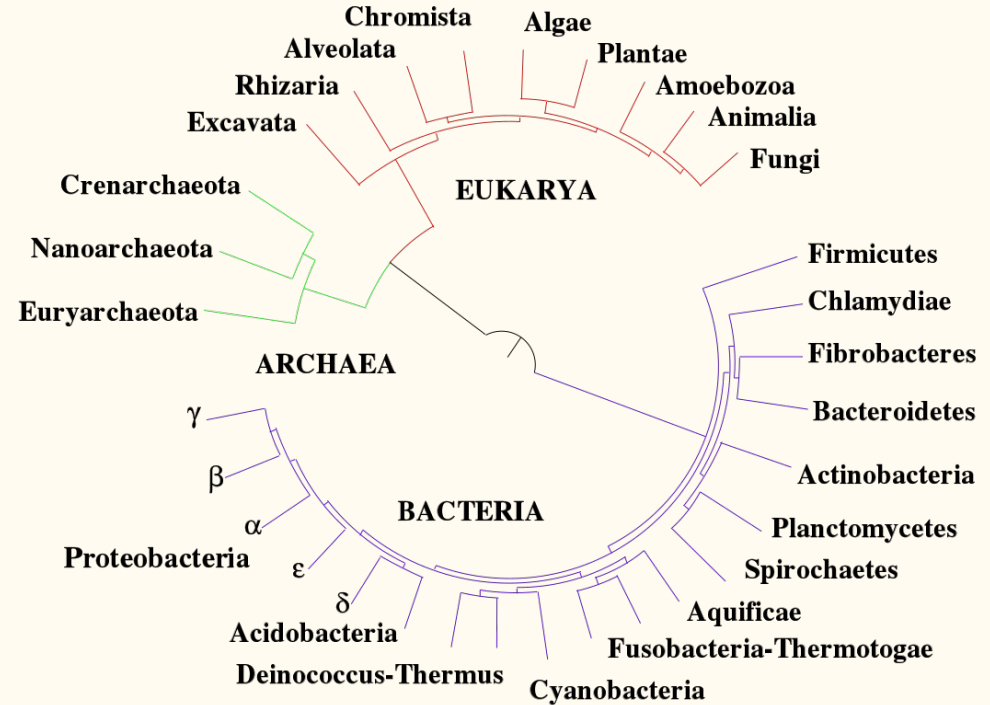
Busca construir uma hierarquia com os grupos

Dividido em dois tipos:

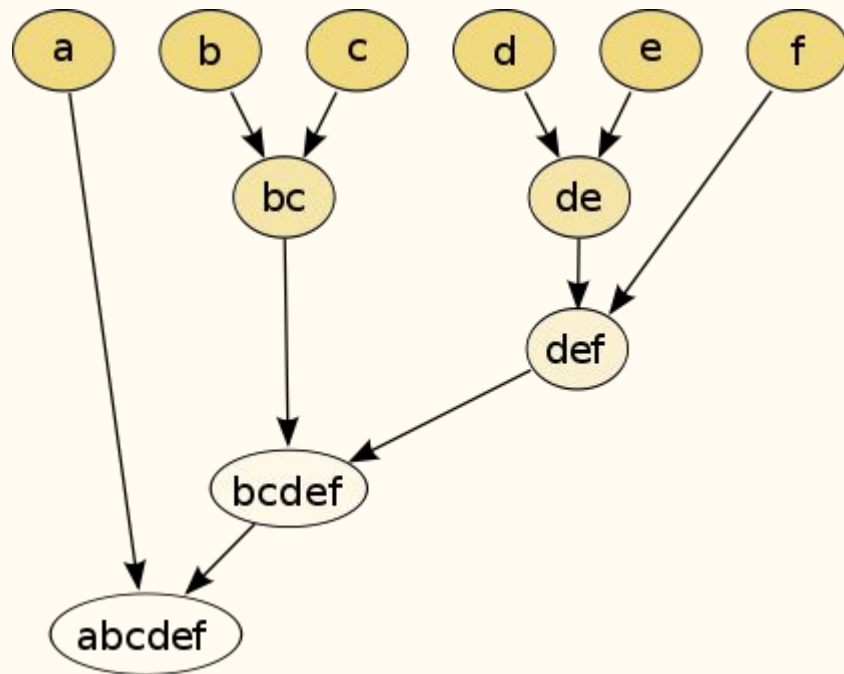
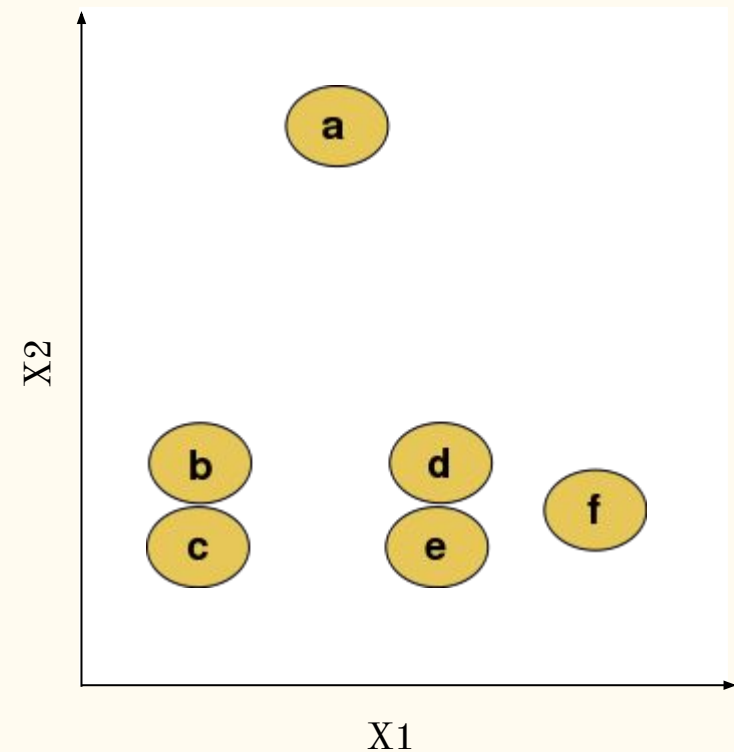
Divisivo: começa com todos os elementos em um único grupo e os divide gradativamente até que cada elemento esteja em um grupo separado

Agrupamento hierárquico

Os resultados do agrupamento hierárquico são geralmente apresentados na forma de um **dendrograma**



Como funciona o agrupamento hierárquico?



Prós e contras do agrupamento hierárquico

Prós:

Não é necessário informar o número de grupos a priori

Visualização dos resultados no dendrograma

Contras:

Treinamento muito custoso (lento)

Análise de componentes principais

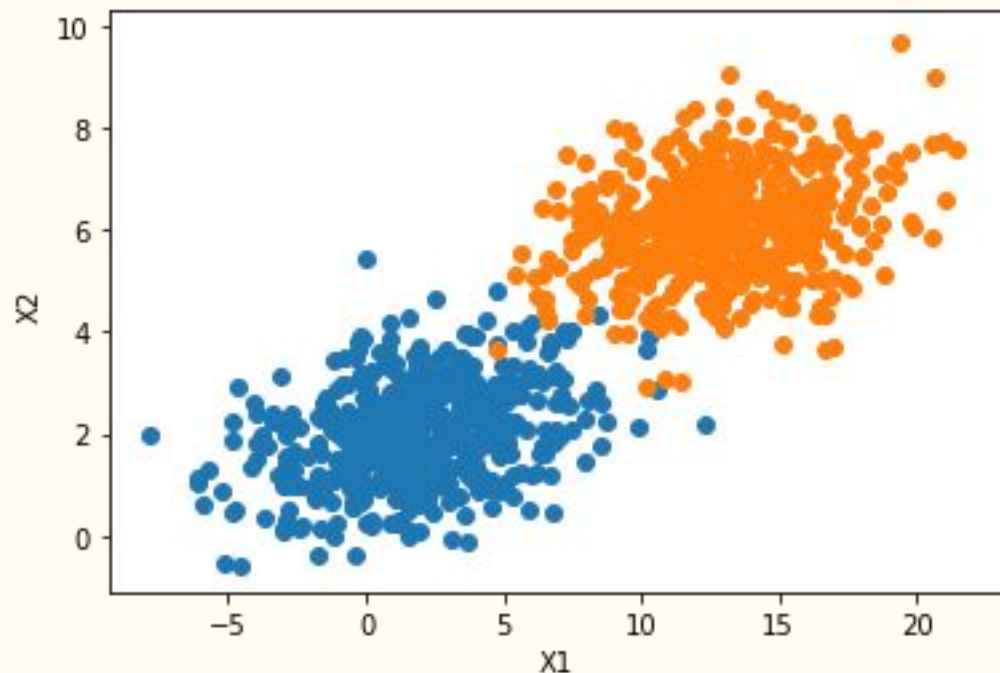
Análise de componentes principais

Dado um conjunto de dados, será que todas as variáveis são necessárias para tomar uma decisão/reconhecer padrões?

Será que podemos transformar os dados de forma que precisemos de menos variáveis para tomar nossas decisões?

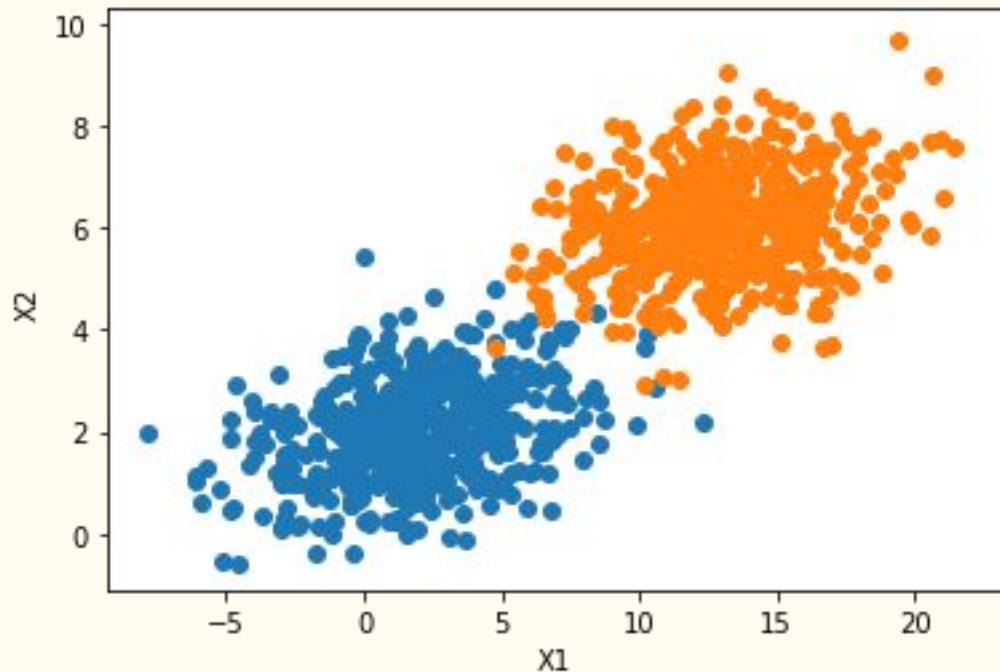
Análise de componentes principais

Neste gráfico, temos dois grupos já demarcados



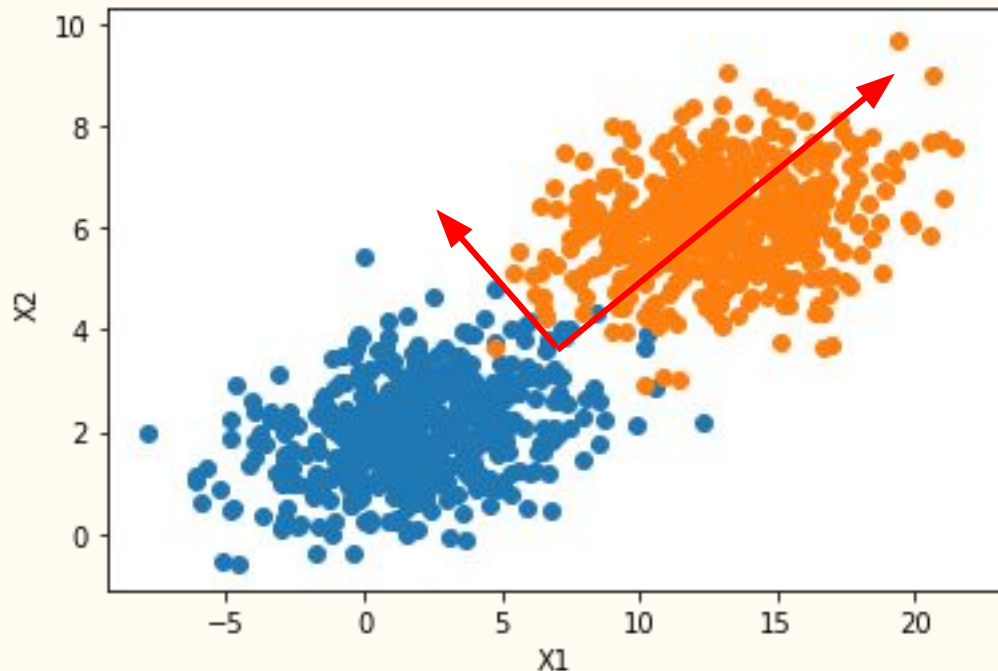
Análise de componentes principais

O grupo laranja se localiza mais à direita e mais acima em relação ao grupo azul



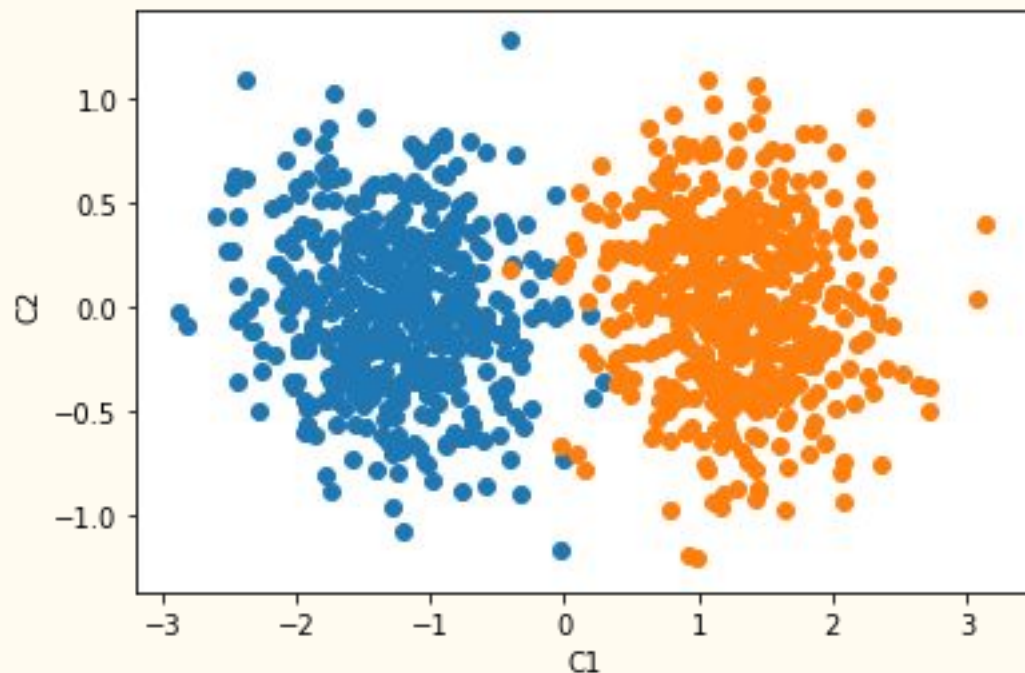
Análise de componentes principais

E se a gente pudesse mexer nos **eixos** do gráfico, de forma que o conjunto de dados não estivesse na diagonal?



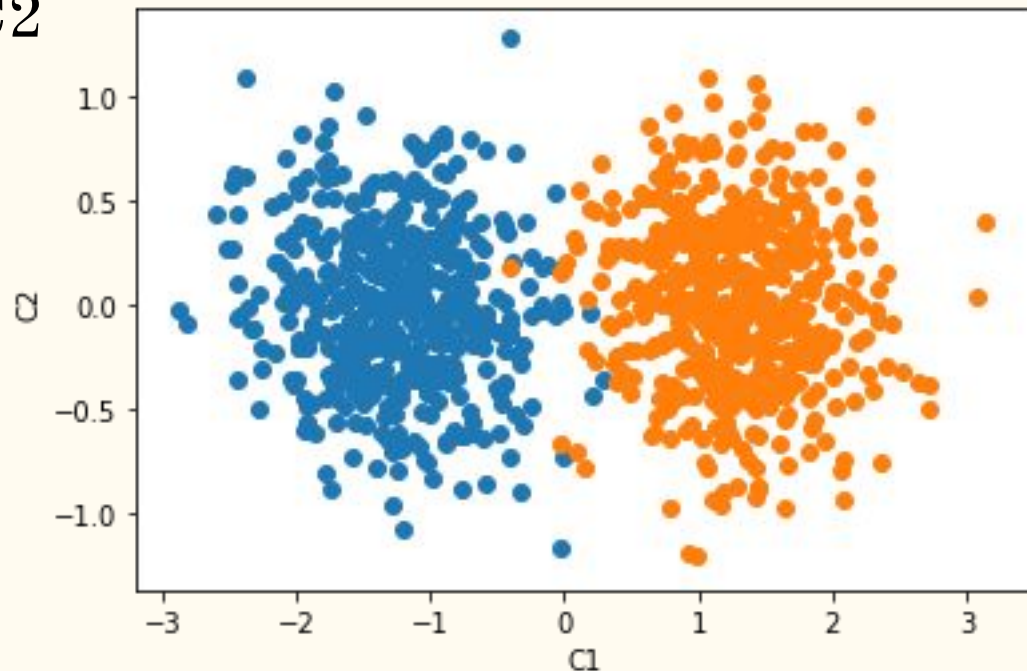
Análise de componentes principais

Os novos eixos são chamados de **componentes** (C1 e C2)



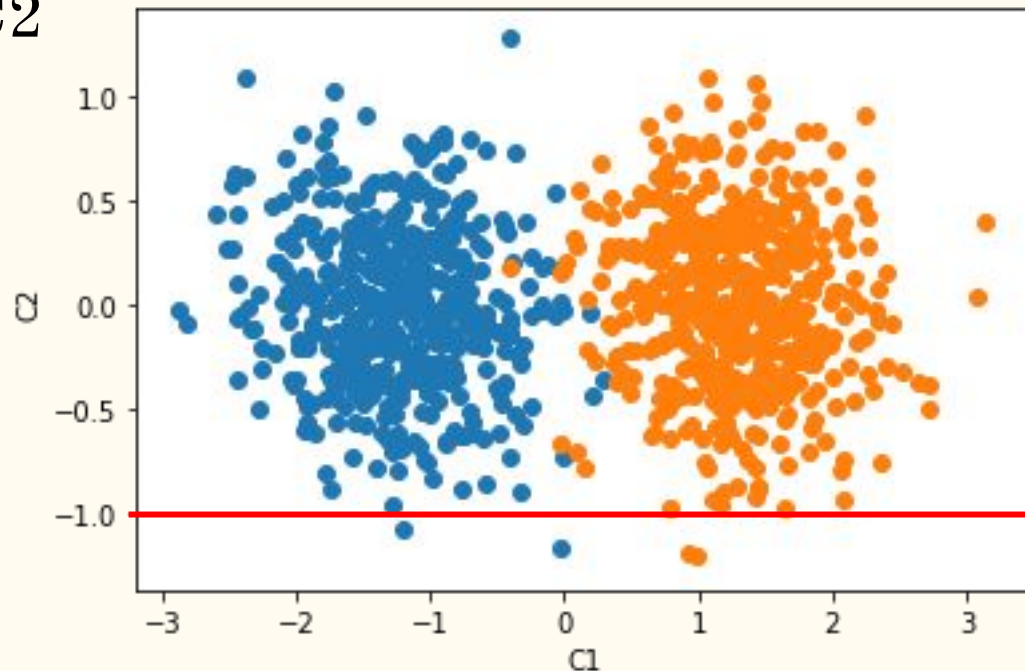
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



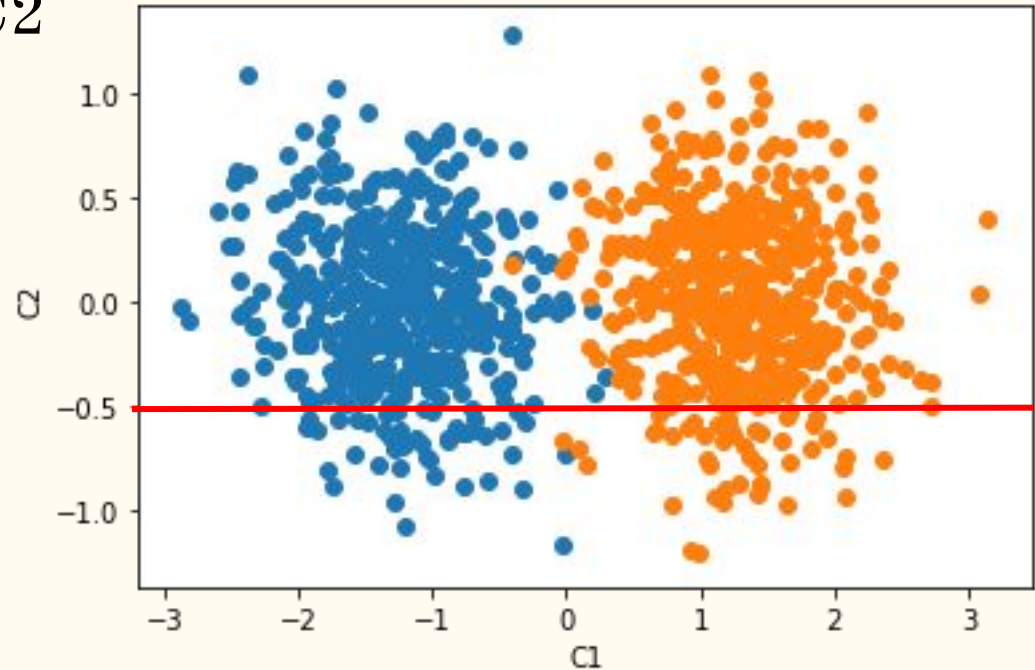
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



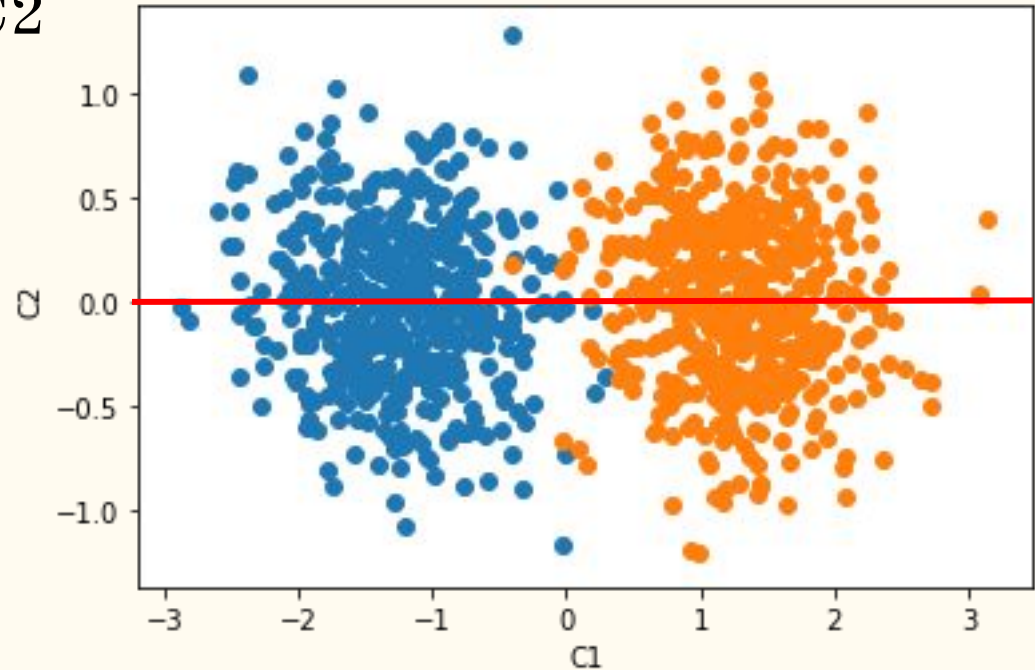
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



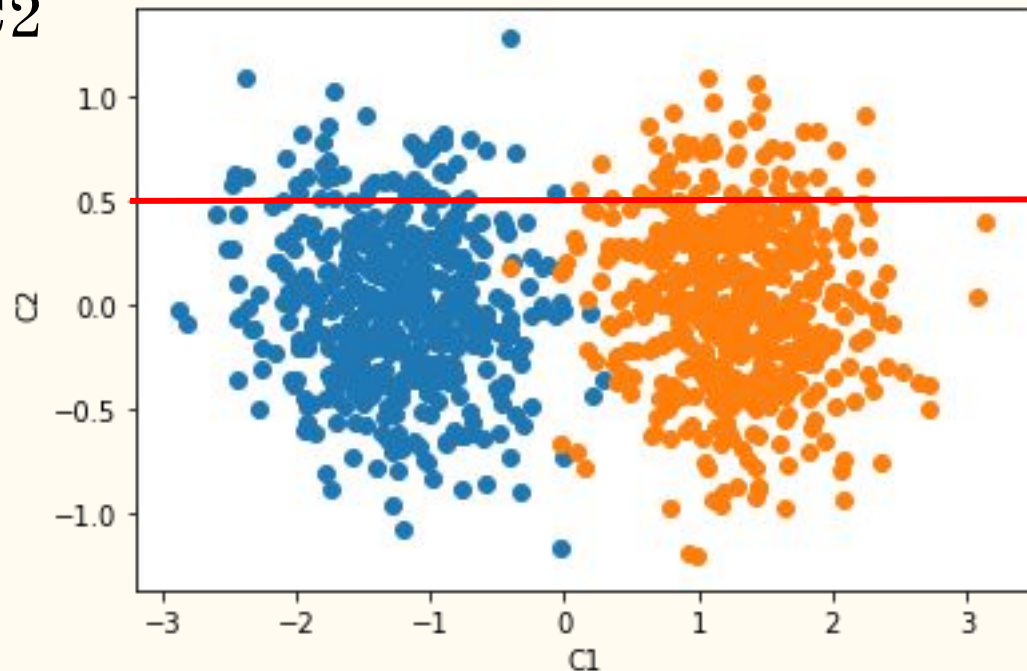
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



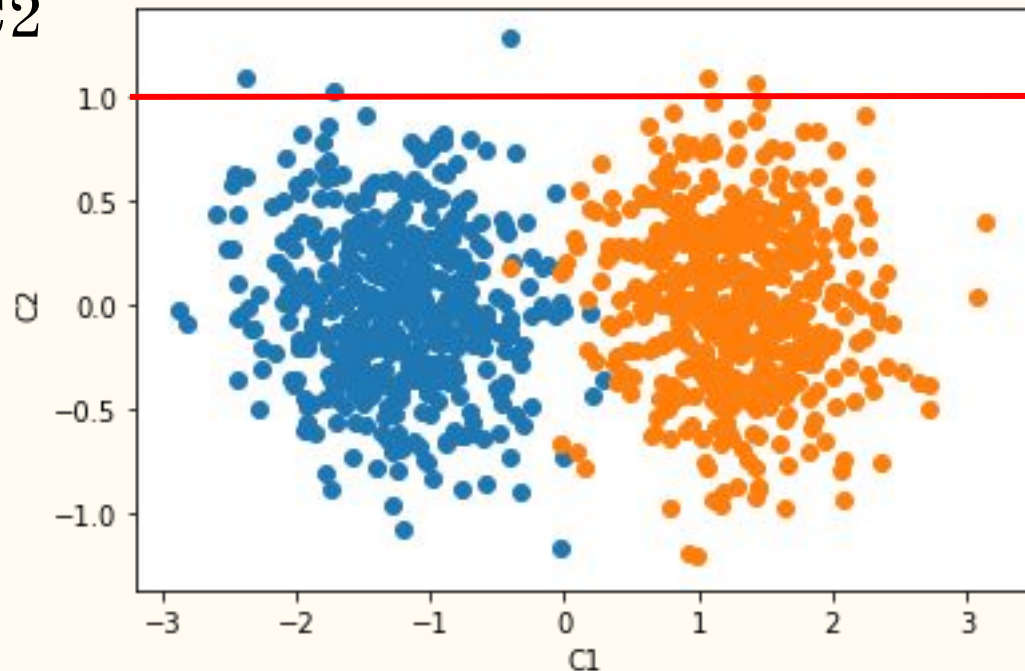
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



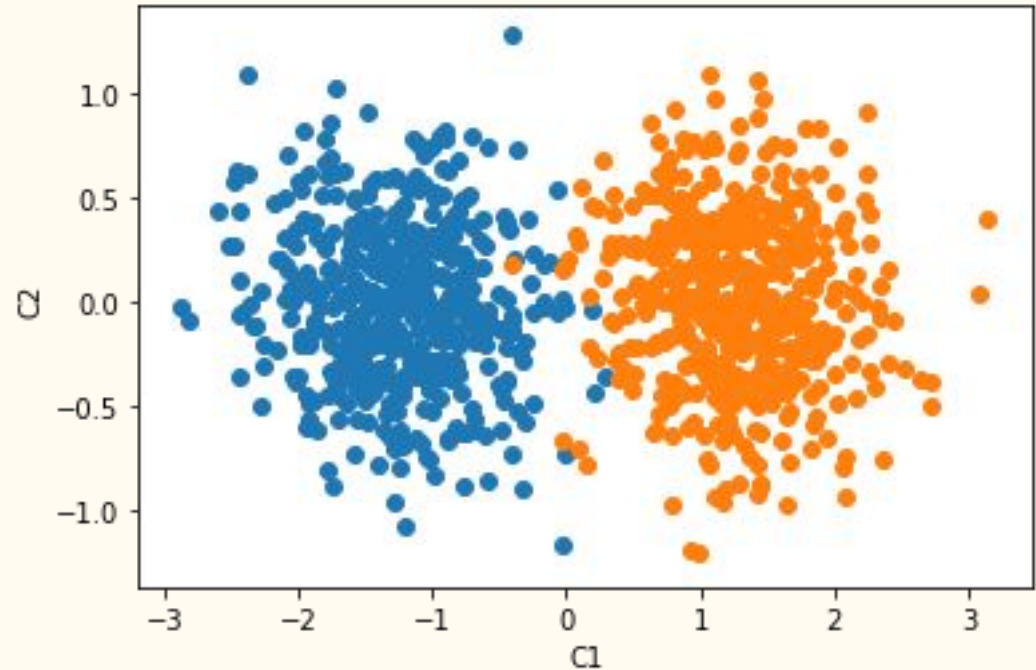
Análise de componentes principais

Olhando para os componentes, os grupos não parecem separáveis de acordo com C2



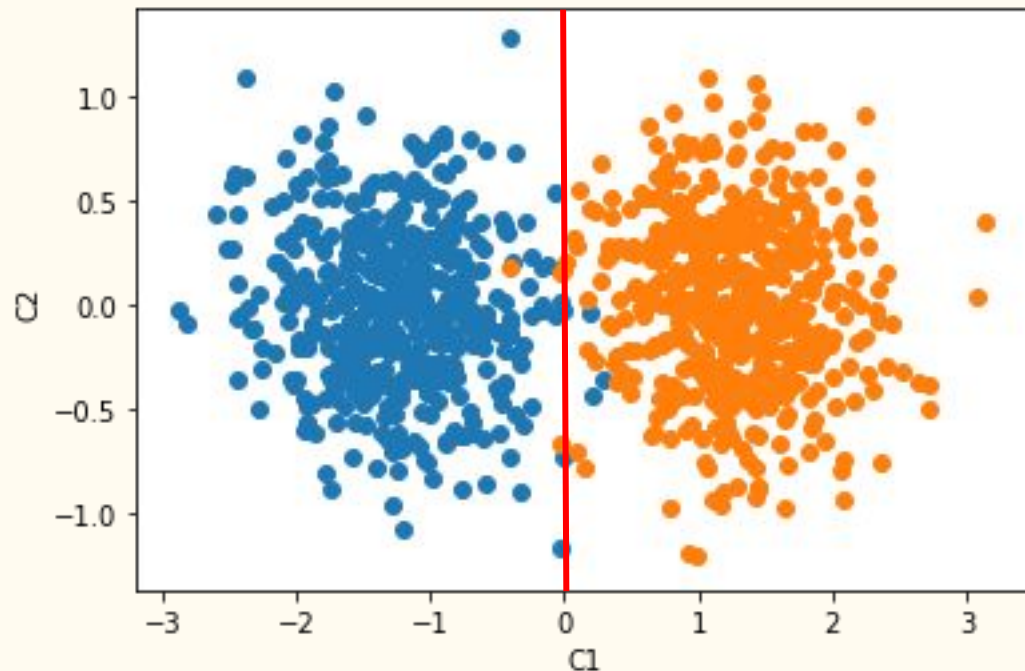
Análise de componentes principais

O componente C1 por outro lado, oferece uma boa separação para os grupos



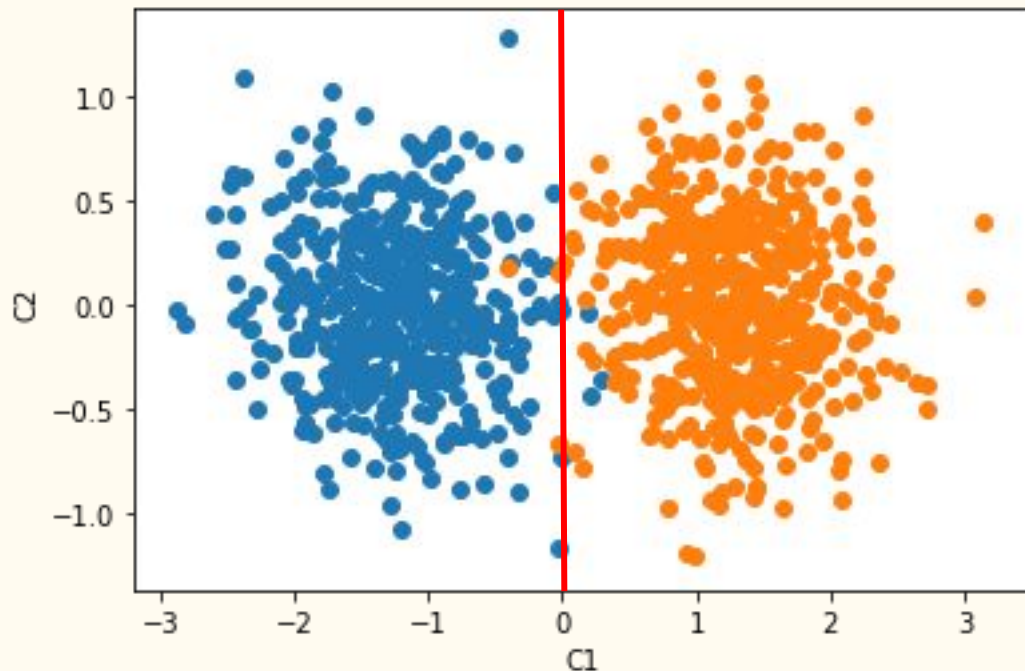
Análise de componentes principais

O componente C1 por outro lado, oferece uma boa separação para os grupos



Análise de componentes principais

C1 é chamado de primeiro componente principal e consegue explicar quase toda a **variância** (97%) do conjunto



Análise de componentes principais

A análise de componentes principais (**PCA** - Principal Component Analysis) consegue encontrar os novos eixos, chamados de componentes, automaticamente

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler # recomenda-se normalizar os dados

ss = StandardScaler()
X_ = ss.fit_transform(X)

pca = PCA(n_components=2)
X_transformed = pca.fit_transform(X_)
print(pca.explained_variance_ratio_)
```

Análise de componentes principais

A partir daí, podemos usar apenas os componentes que explicam uma fatia maior da variância do nosso conjunto nas nossas análises

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler # recomenda-se normalizar os dados

ss = StandardScaler()
X_ = ss.fit_transform(X)

pca = PCA(n_components=2)
X_transformed = pca.fit_transform(X_)
print(pca.explained_variance_ratio_)
```

FIM!





08/07

19:00

**YouTube
CANAL ARIA**

Bate-papo com Marianne Linhares

Da graduação para a DeepMind

Graduada em Ciência da Computação na UFCG, durante a graduação estagiou na Google (NY), Kunumi e DeepMind (Londres), além de laboratórios da UFCG. Hoje é Research Engineer na DeepMind (Londres).

@aprendizagemdemaquina