

Introdução ao Aprendizado de Máquina

Prof^a. Samira Santos da Silva

(UEMG)

21 de Outubro de 2020

Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

O que é Aprendizagem de Máquina?

Aprendizagem de Máquina ou Machine Learning (ML)

- ▶ Área da **Inteligência Artificial** onde desenvolve-se algoritmos para ensinar uma máquina a desempenhar determinadas tarefas.
- ▶ Um algoritmo de ML recebe um **conjunto de dados de entrada** e, baseado nos padrões encontrados, **gera saídas**.
- ▶ Cada entrada possui suas *features*.
 - ▶ Extrair as features dos dados é o **ponto inicial** para um algoritmo de ML.

História

- ▶ O termo “**machine learning**” foi criado por **Arthur Samuel**, engenheiro do MIT, em 1959.
- ▶ Ele descrevia este conceito como “um campo de estudo que dá aos computadores **a habilidade de aprender** sem terem sido programados para tal”.
- ▶ Samuel trabalhava em um projeto para criar uma **máquina autônoma** com estas características.

História



Figura: Arthur Samuel e um IBM 700 na década de 1950.

História

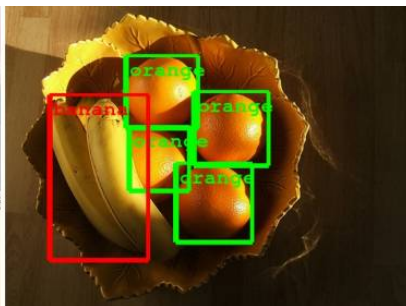
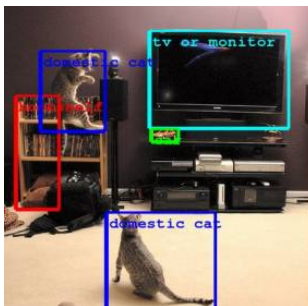
- ▶ Mas foi somente com o advento da **Internet** que ML começou a tomar forma.
- ▶ Com **tanta informação** coletada e armazenada na Web, foi necessário **meios de organizar** esse conteúdo gigantesco de forma **automatizada**.

- ▶ Existem **diversas aplicações** para Machine Learning.
- ▶ Alguns exemplos: Reconhecimento de Padrões, Recomendação, Mineração de Texto, Análise de Sentimentos, etc.

Reconhecimento de Padrões

- ▶ **Atribuir** uma **classe** (padrão) a um conjunto de dados **desconhecidos**.
- ▶ Exemplos:
 - ▶ Reconhecimento de padrões em imagens.
 - ▶ Reconhecimento de padrões em textos.
 - ▶ Reconhecimento de padrões em áudios.

Reconhecimento de Padrões



Recomendação

- ▶ Um **sistema de recomendação** combina técnicas computacionais para **selecionar itens personalizados** com base nos **interesses** dos usuários e conforme o **contexto** no qual estão inseridos.
- ▶ Ex. de itens: livros, filmes, notícias, música, vídeos, anúncios, links patrocinados, produtos de uma loja virtual, etc.
- ▶ Recomendações são exibidas de acordo com um “**score**” (nota).

Recomendação



Mineração de Textos

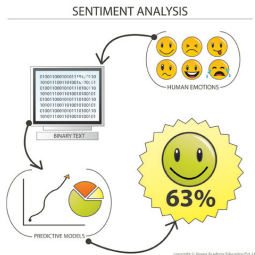
- ▶ Explorar **grandes quantidades de dados** à procura de **padrões consistentes** para detectar **relacionamentos sistemáticos entre variáveis** gerando informação.
- ▶ Ex: Supermercado pode minerar seus dados de compras e acabar percebendo que quem compra cerveja geralmente compra fraldas.
 - ▶ Então por que não colocar cervejas e fraldas próximos?

Mineração de Textos



Análise de Sentimentos

- ▶ Processo para conseguir **identificar sentimentos** embutidos em **textos** (geralmente de mídias sociais).
 - ▶ É positivo?
 - ▶ É negativo?



Exemplos Práticos de Aplicações

1 - Indicar a qualidade de um vinho



Exemplos Práticos de Aplicações

2 - Escolher o melhor filme para o seu final de semana



Exemplos Práticos de Aplicações

3 - Tirar motoristas de grandes congestionamentos



Exemplos Práticos de Aplicações

4 - Reconhecimento de placas de carro em imagens



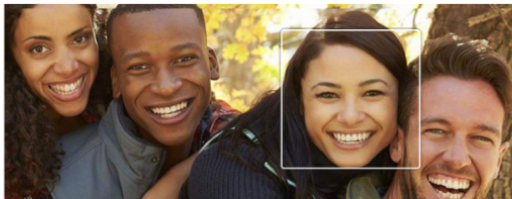
Exemplos Práticos de Aplicações

5 - Reconhecimento de Faces no Facebook



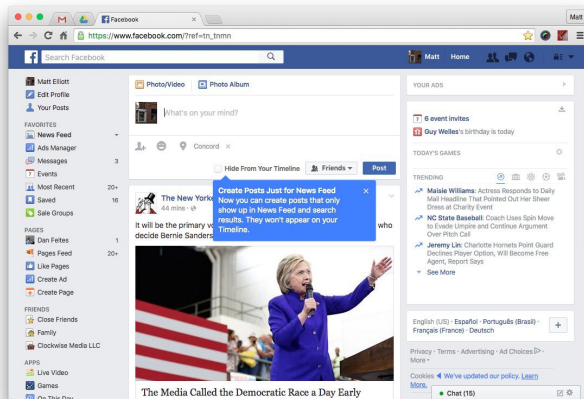
Gary Chavez added a photo you might ...
be in.

about a minute ago · 



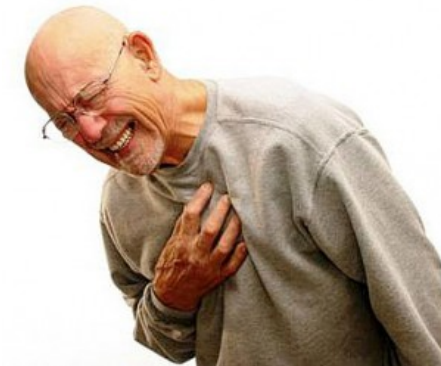
Exemplos Práticos de Aplicações

6 - Exibição de Postagens do seu Interesse



Exemplos Práticos de Aplicações

7 - Previsão de Derrame



Exemplos Práticos de Aplicações

8 - Carros Autônomos



Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

O que é uma Feature?

Feature

- ▶ Feature é uma **característica** que descreve um objeto.
- ▶ Qualquer atributo de um objeto pode ser tratado como feature: um número, um texto, uma data, um booleano etc.

O que é uma Feature?

Feature

- ▶ No objeto Pessoa, vemos vários atributos que o descreve.
- ▶ Esses atributos são suas features:



Pessoa 1	
1. Dia	7
2. Mês	1
3. Ano	1986
4. Peso	80
5. Altura	1,76
6. Nº irmãos	1
7. Nº irmãs	0
...	...
65. Sexo	M

O que é uma Feature?

Feature

- ▶ Na tabela abaixo, temos um conjunto maior de dados, onde:
 - ▶ Cada coluna é uma feature que descreve a linha;
 - ▶ Cada linha é uma entrada e tem seu conjunto de features.

Nome	Data de Nascimento			Peso	Altura	Sexo
	Dia	Mês	Ano			
Pessoa 1	05	02	1982	85	1,72	M
Pessoa 2	13	10	1990	60	1,50	F
Pessoa 3	05	01	1972	75	1,63	M

O que é uma Feature?

Feature

- ▶ As features são as **entradas** dos **algoritmos de ML**.
- ▶ Quanto **mais detalhes** o algoritmo tiver sobre uma entrada, mais **facilmente** achará **padrões** nos dados.
- ▶ Features **ruins** podem prejudicar o desempenho do algoritmo.
- ▶ Features **boas** são a chave para o sucesso de um algoritmo.

O que é uma Feature?

Feature

- ▶ Grande parte do trabalho em ML consiste em **trabalhar os dados e gerar boas features** em cima deles.
 - ▶ Engenharia de features ou feature engineering.
- ▶ Existem **diversas técnicas para gerar features**:
 - ▶ Conhecendo a natureza dos dados;
 - ▶ Aplicando matemática e estatística para criá-las em cima dos dados.

Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

Extração de Features

Extração de Features

- ▶ Tem o **objetivo** retirar de imagens, vídeos, textos ou áudios informações que descrevem esses tipos de mídia.
- ▶ O **resultado** da extração de características é um **vetor de atributos** descrevendo cada objeto.
- ▶ Existem **métodos adequados** para cada tipo de **mídia**.

Extração de Features em Imagens

Extração de Features Imagens

- ▶ Na extração de features em imagens, por exemplo, existem **descritores especializados** em representar:
 - ▶ Cores dos Objetos;
 - ▶ Formas dos Objetos;
 - ▶ Textura dos Objetos;
 - ▶ Relacionamentos entre Objetos;
 - ▶ etc.

Extração de Features em Imagens

Descrição de Cor - Exemplo: Histograma

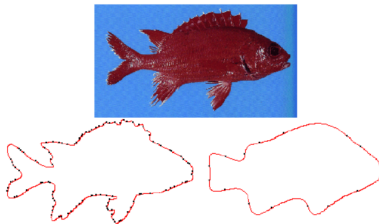
- ▶ Vetor $[h_1, \dots, h_n]$ onde cada h_j contém a quantidade de pixels de cor j na imagem.
- ▶ Para comparação de histogramas utiliza-se:

$$D(H, H') = \frac{\sum_i \min(h_i, h'_i)}{\sum_i h'_i}$$

Extração de Features em Imagens

Descrição de Forma - Exemplo: Algoritmo detector de bordas

- ▶ Curvaturas obtidas a partir do contorno, com redução de mudanças (ruídos) na curvatura.



Descrição de Textura - Exemplo: Haralick Features

- ▶ Haralick propõe o uso de 14 medidas estatísticas como features.

Haralick *et al.*^[19]

- Contrast
- Correlation
- Difference entropy
- Difference variance
- Energy
- Entropy
- Homogeneity
- Information measure of correlation 2
- Information measure of correlation 1
- Max correlation coefficient
- Sum average
- Sum entropy
- Sum variance
- Variance

Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

Aprendizagem Supervisionada x Não-supervisionada

- ▶ Tendo nossas **features em mãos** podemos aplicar **diversos algoritmos de aprendizado** nelas.
- ▶ Existem dois grandes grupos de algoritmos em ML: os de **aprendizagem supervisionada** e os de **aprendizagem não-supervisionada**.

Aprendizagem Supervisionada

- ▶ Tem-se um **conjunto de entradas A** que já **possuem as saídas**.
- ▶ Deseja-se **prever a saída** de um **conjunto de entrada B**.
- ▶ Se A é um **conjunto de tamanho considerável**, conhecer suas saídas permite com seja possível encontrar **padrões** que **relacionam entradas com saídas**.
- ▶ Assim, é possível **prever as saídas do conjunto B** com base nesses **padrões previamente encontrados**.

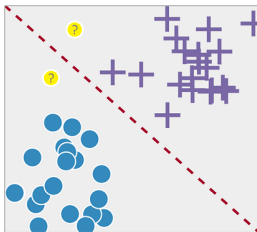
Aprendizagem Supervisionada

Aprendizagem Supervisionada

- ▶ Algoritmos que realizam este procedimento são denominados **supervisionados**.
- ▶ Algoritmos supervisionados são divididos em dois grupos: **classificação** e **regressão**.

Classificação

- ▶ É quando queremos prever uma **classificação**.
- ▶ As **classes** utilizadas no **aprendizado** devem ser as **mesmas** dos dados cujas classes são desconhecidas.



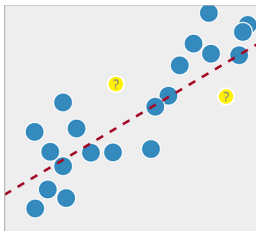
Classificação - Ex: classificar um e-mail como spam ou não spam.

- ▶ Coletou-se vários e-mails já **classificados** como **spam** ou **não spam**;
- ▶ **Treinou-se** um algoritmo que seria capaz de encontrar os **padrões** determinantes para **cada uma das classes**;
- ▶ Teria-se um **algoritmo** capaz de **ler um novo e-mail** e **classificar** como **spam** ou **não** baseado em suas características.

Aprendizagem Supervisionada

Regressão

- ▶ Quando queremos **prever um valor** e não uma classe.
- ▶ Os **dados** usados no **aprendizado** possuem também valores ao invés de classes.



Regressão - Ex: determinar o preço de uma casa

- ▶ Coletou-se as features de **várias casas** e os seus **preços** (bolinhas azuis);
- ▶ Treinou-se um **algoritmo** capaz de criar uma **relação** entre as **features da casa** e o seu **preço** (linha vermelha);
- ▶ Logo, esse algoritmo é capaz de determinar o **preço de uma nova casa** (bolinhas amarelas) baseado em suas **características**.

Regressão

- ▶ No treinamento supervisionado, sempre utilizase **dados com saídas semelhantes** as que desejamos encontrar.
 - ▶ No exemplo 1: vários e-mails rotulados como **spam** e **não spam** foram usados no treinamento.
 - ▶ No exemplo 2: várias casas com seus **preços** foram usadas nos treinamento.

Aprendizagem Não-supervisionada

Aprendizagem Não-supervisionada

- ▶ Tem-se um conjunto de **entradas B sem as saídas** que deseja-se.
- ▶ Com base nas características desses dados, é possível gerar
 - ▶ Um agrupamento;
 - ▶ Processá-los a fim de gerar novas formas de expressar essas características.
- ▶ Algoritmos não-supervisionados podem ser divididos em dois grupos: **redução de dimensionalidade** e **clusterização**.

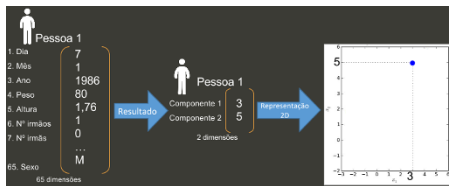
Redução de dimensionalidade

- ▶ Reduzir um conjunto de dados de **alta dimensão** para um **número menor de dimensões** de forma que **represente o máximo possível** os dados originais.
- ▶ A **dimensão** é a quantidade de features consideradas.
- ▶ Deve-se remover features que são **irrelevantes** ou **redundantes**.

Aprendizagem Não-supervisionada

Redução de dimensionalidade

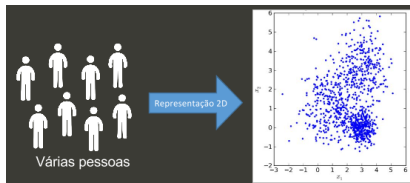
- ▶ O objeto Pessoa contém 65 features, ou seja, **65 dimensões**.
- ▶ Após passar essas features pelo algoritmo, **reduzimos a 2 dimensões**.
- ▶ As novas features 1 e 2 representam o máximo possível as 65 features originais.



Aprendizagem Não-supervisionada

Redução de dimensionalidade

- ▶ É possível fazer a **mesma redução** pra **várias pessoas**.
- ▶ Com **2 features** é possível **enxergar melhor a distribuição dos dados**.
- ▶ Pessoas com **mesmas características** podem ficar mais **próximas** nesse tipo de representação. Enquanto pessoas muito **diferentes** ficam mais **distantes**.



Redução de dimensionalidade

- ▶ Reduzimos para 2 dimensões para **melhor visualização**.
- ▶ Entretanto, algoritmos são capazes de lidar com **quantas dimensões forem necessárias**.
- ▶ Esse tipo de técnica pode ser usada para **facilitar a análise dos dados**, como no exemplo.

Redução de dimensionalidade

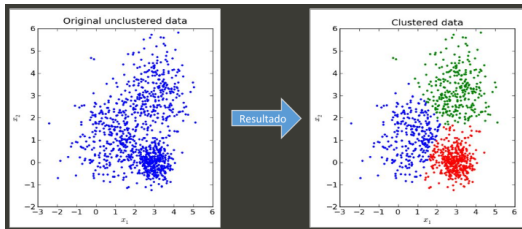
- ▶ Também utilizado quando features de alta dimensão impedem algoritmos de funcionarem corretamente.
- ▶ Isso ocorre, principalmente, se **número de features** > **número de entradas**.
 - ▶ Mal da dimensionalidade.
- ▶ Nesse caso, reduz-se features para um número mais aceitável e então faz-se o treinamento normalmente.

Clusterização

- ▶ **Agrupamento dos dados** baseado em suas características.
- ▶ Os grupos formados são chamados de **clusters**.
- ▶ Poderíamos querer agrupar dados em 3 grupos.
- ▶ Um algoritmo de **clusterização** seria capaz de **analisar os dados** e **identificar esses grupos** baseado nas **características** desses dados.

Clusterização

- ▶ É uma técnica muito **poderosa** e tem uma **aplicabilidade** muito alta.



Clusterização - Exemplos

- ▶ Identificação de **clientes similares** e com isso ser mais assertivo ao oferecer um novo produto;
- ▶ Agrupamento de **pacientes** com os **mesmos sintomas**;
- ▶ Classificação de documentos;
- ▶ **Qualquer agrupamento** de uma grande quantidade de dados baseado em suas **características**.

Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

Base de Dados: Iris Dataset

Iris Dataset

- ▶ Problema proposto: através das características fornecidas de uma flor, descobrir de qual dos 3 tipos de flores se trata.
- ▶ 150 entradas.
- ▶ 4 features.



Iris Setosa



Iris Virginica



Iris Versicolor

Base de Dados: Iris Dataset

Iris Dataset

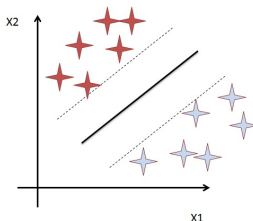
- Features: Comprimento da Sépala, Largura da Sépala, Comprimento da Pétala, Largura da Pétala.

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Algoritmo de Classificação: SVM

SVM

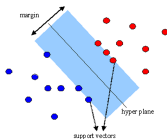
- ▶ Algoritmo de classificação **supervisionado** que tenta criar uma **linha** (ou fronteira) que melhor separa os dados.
- ▶ Tipicamente, chamamos esta “linha” de “**hiperplano**”.
- ▶ O **melhor hiperplano** é que resulta em **maior distância marginal**.



Algoritmo de Classificação: SVM

SVM

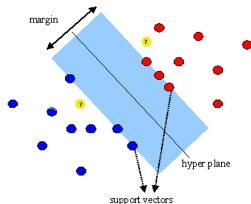
- ▶ **Vetores de suporte** são calculados para então escolher o melhor hiperplano.
- ▶ **Vetores de suporte** são **pontos** que estão **mais próximos** das linhas. A distância entre esses pontos é chamada de **margem**.
- ▶ O algoritmo escolhe um **hiperplano** que tem uma **margem maior**.



Algoritmo de Classificação: SVM

SVM

- ▶ Ao receber uma **amostra desconhecida**, verifica-se que **lado da margem** ela está e sua **distância**.
- ▶ Desta forma, é possível determinar **sua classe**.



Execução do Algoritmo SVM

```
from sklearn import datasets
```

Execução do Algoritmo SVM

```
from sklearn import datasets
```

```
iris= datasets.load_iris()
```

Execução do Algoritmo SVM

```
from sklearn import datasets
```

```
iris= datasets.load_iris()
```

```
print(iris.data)
```

Execução do Algoritmo SVM

```
print(iris.target)
```

Execução do Algoritmo SVM

```
print(iris.target)
```

```
print(iris.data.shape)
```


Execução do Algoritmo SVM

```
print(iris.target)
```

```
print(iris.data.shape)
```

```
print(iris.target.shape)
```

Execução do Algoritmo SVM

```
from sklearn.model_selection import train_test_split
```

Execução do Algoritmo SVM

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split (iris.data,  
iris.target, test_size=0.4, random_state=0)
```

Execução do Algoritmo SVM

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split (iris.data,  
iris.target, test_size=0.4, random_state=0)
```

```
from sklearn import svm
```

Execução do Algoritmo SVM

```
clf = svm.SVC(kernel= 'linear', C=1).fit(X_train, y_train)
```

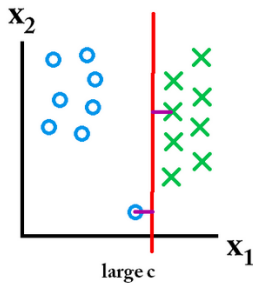
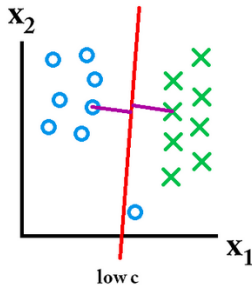
Execução do Algoritmo SVM

```
clf = svm.SVC(kernel= 'linear', C=1).fit(X_train, y_train)
```

```
print(clf.score(X_test, y_test))
```

Algoritmo de Classificação: SVM

Parâmetro C do SVM



Algoritmo de Clusterização: K-Means

K-means

- ▶ O K-means é um algoritmo do tipo **não-supervisionado**, ou seja, que não trabalha com dados rotulados.
- ▶ Seu objetivo é encontrar **similaridades** entre os dados e **agrupá-los** conforme o **nº de clusters** passado pelo argumento **k**.
- ▶ Seu processo é composto por **4 etapas**.

Algoritmo de Clusterização: K-Means

1 - Inicialização

- ▶ O algoritmo gera de forma **aleatória** K centróides dentre dados (pontos) existentes.
- ▶ **Centróides** são os **pontos centrais dos clusters**.



Algoritmo de Clusterização: K-Means

2 - Atribuição ao cluster

- ▶ É computada a **distância** entre cada **ponto** e cada centróide.
- ▶ Cada **ponto** é **atribuído** ao **cluster** cuja distância ao seu centróide seja a **menor**.
- ▶ Cálculo de distância: Euclidiana.



Algoritmo de Clusterização: K-Means

3 - Movimentação de centróides

- ▶ **Recálculo** dos centróides.
- ▶ Faz-se a **média dos pontos de cada cluster** e o ponto médio será o novo centróide.



Algoritmo de Clusterização: K-Means

4 - Otimização

- ▶ As fases 2 e 3 são **repetidas** até o cluster se tornar **estático** ou **algum critério de parada** tenha sido atingido.
- ▶ Por fim, o K-means chega ao **fim da sua execução** dividindo os dados no **número de clusters especificado** pelo argumento k .



Execução do Algoritmo K-Means

```
from sklearn import datasets
```

Execução do Algoritmo K-Means

```
from sklearn import datasets
```

```
iris= datasets.load_iris()
```

Execução do Algoritmo K-Means

```
from sklearn import datasets
```

```
iris= datasets.load_iris()
```

```
X=iris.data
```

Execução do Algoritmo K-Means

```
from sklearn.cluster import KMeans
```


Execução do Algoritmo K-Means

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters = 3, init = 'random')
```

Execução do Algoritmo K-Means

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters = 3, init = 'random')
```

```
kmeans.fit(X)
```

Execução do Algoritmo K-Means

```
print(kmeans.cluster_centers_)
```

Execução do Algoritmo K-Means

```
print(kmeans.cluster_centers_)
```

```
distance = kmeans.fit_transform(X)
```

Execução do Algoritmo K-Means

```
print(kmeans.cluster_centers_)
```

```
distance = kmeans.fit_transform(X)
```

```
print(distance)
```

Execução do Algoritmo K-Means

```
labels = kmeans.labels_
```

Execução do Algoritmo K-Means

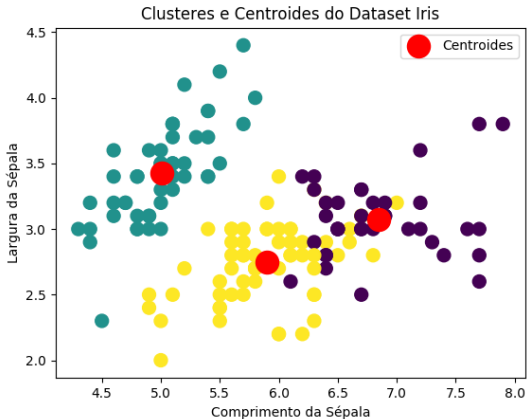
```
labels = kmeans.labels_
```

```
print(labels)
```

Execução do Algoritmo K-Means - Visualização de Dados

```
import matplotlib.pyplot as plt
plt.scatter(X[:, 0], X[:,1], s = 100, c = kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[0, 0],
kmeans.cluster_centers_[0, 1], s = 300, c = 'red',label = 'Centroides')
plt.title('Clusteres e Centroides do Dataset Iris')
plt.xlabel('Comprimento da Sépala')
plt.ylabel('Largura da Sépala')
plt.legend()
plt.show()
```


Execução do Algoritmo K-Means - Visualização de Dados



Sumário

Introdução

Conceitos Iniciais

Extração de Features

Aprendizagem

Demonstração

Extra

- ▶ Existem diversas formas de se aplicar ML, **diferentes linguagens e bibliotecas**.
- ▶ Optamos por trabalhar com a linguagem **Python** através da biblioteca **SciKit Learn**, que é uma poderosa biblioteca que possui **algoritmos de ML** implementados.
- ▶ Além disso, utilizamos a biblioteca **Matplotlib** para visualizar **gráficos 2D**.

Instalação Bibliotecas no Ubuntu

No terminal do Ubuntu digite:

- ▶ `sudo apt install python-pip`
- ▶ `pip install -U scikit-learn`
- ▶ `pip install matplotlib`
- ▶ `python nomedoarquivo.py`



Agradecimentos

Agradecimentos

- ▶ Ao IFSULDEMINAS pelo convite.
- ▶ A todos presentes!

Contatos

- ▶ Slides e códigos disponibilizados no GitHub:
<https://github.com/samirasilva/MinicursoIntroML2020>
- ▶ E-mail: samirapgti@gmail.com

Mais informações

- ▶ Curso da Udacity: “Introdução ao Aprendizado de Máquina”
 - ▶ <https://br.udacity.com/course/intro-to-machine-learning-ud120>

THANK YOU!

