

Técnicas Tradicionais de Classificação de Imagens

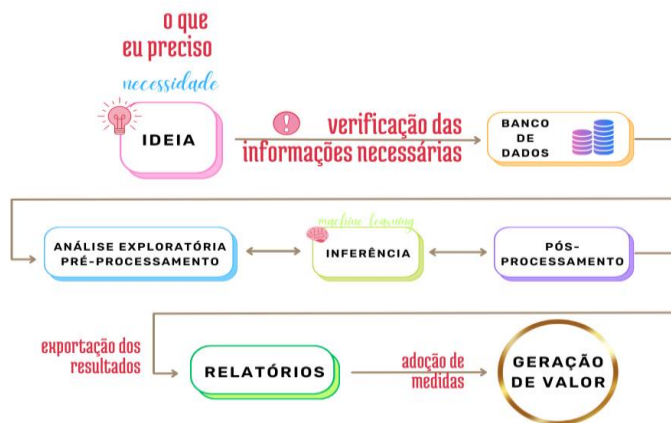
Classificação: KNN e
Deploy: API



Recapitulação

Esquema Básico

Projeto de Visão Computacional



- Desbalanceamento
- Visualização
- Medidas resumo: características
- Metadados de captura
- Metadados técnicos
- Redução de Dimensionalidade (tópico visto em maior profundidade na aula passada com o Leonardo)
- Redimensionamento x Patches x Crop
- Análise de Variância
- Data Augmentation (ainda a ser muito estudado e praticado)
- Histogram Matching vs Histogram Equalization

Esquema Básico

Treinamento/Inferência

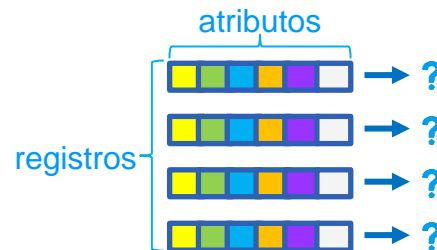
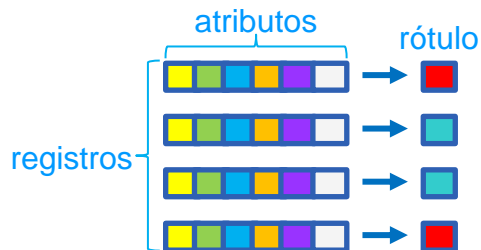
Machine Learning

Supervisionado

Não Supervisionado

Classificação
Regressão
Previsão de séries temporais

Agrupamento
Associação



Esquema Básico

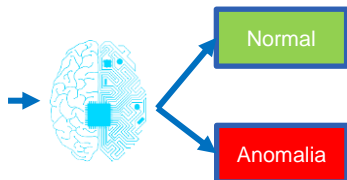
Aprendizado Supervisionado

Classificação

Rótulo é categórico.



Imagens de um equipamento

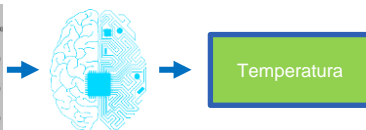


Regressão

Rótulo é contínuo.



Imagens de um equipamento

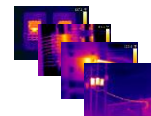


Previsão de Séries

Rótulo é contínuo e dependente do tempo.

Temperatura
Ou estado

Dados
históricos



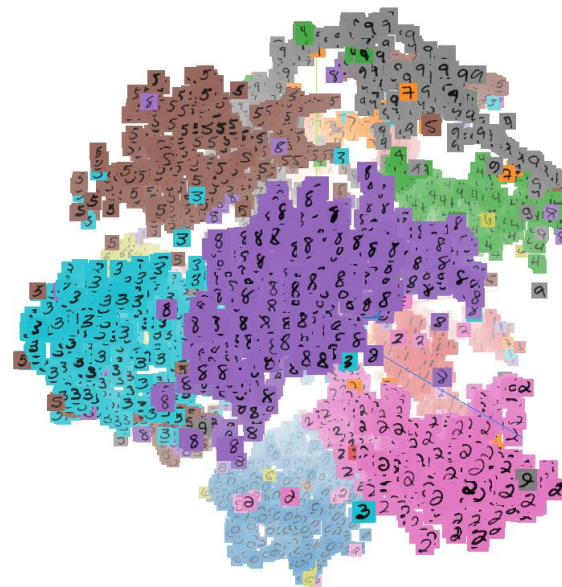
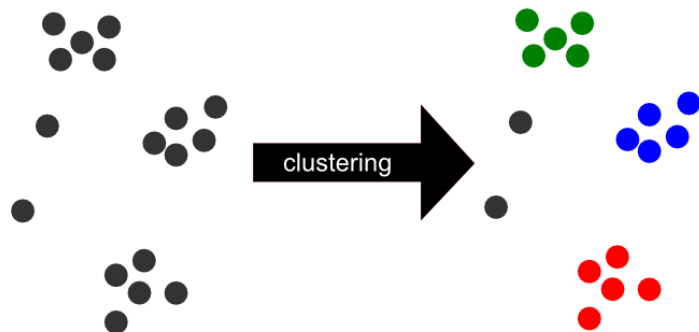
Tempo

Esquema Básico

Aprendizado Não Supervisionado

Agrupamento

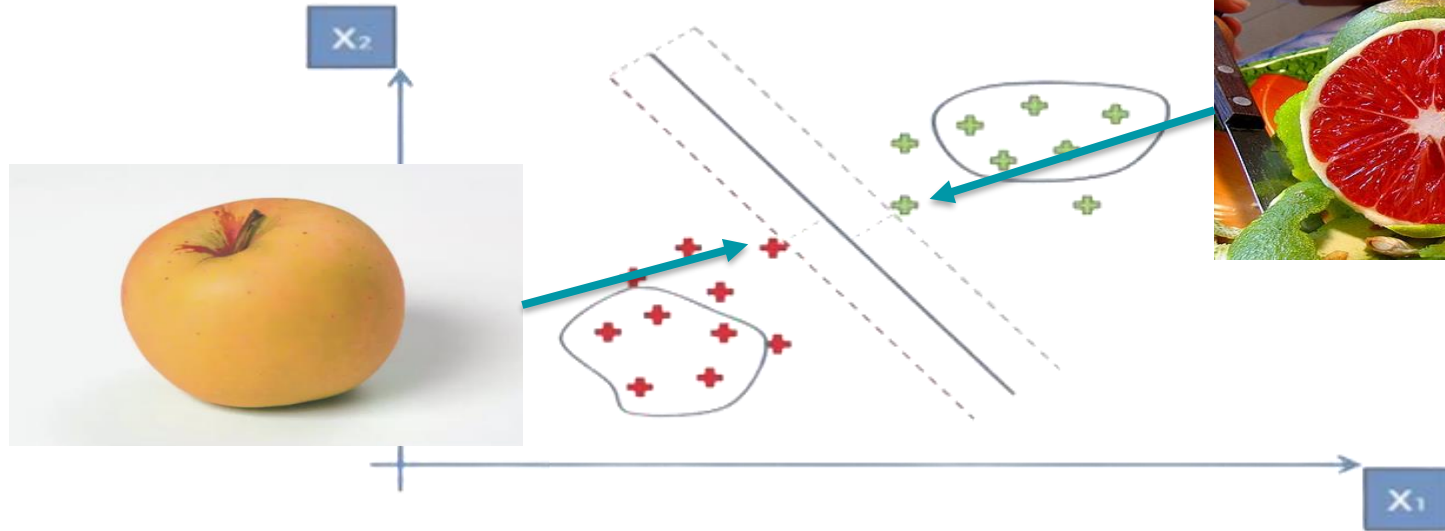
Descoberta de semelhanças e grupos entre registros.



SVM – Support Vector Machine

Classificação

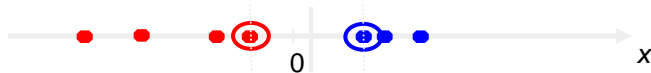
SVM



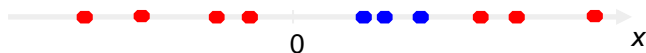
Classificação

SVM

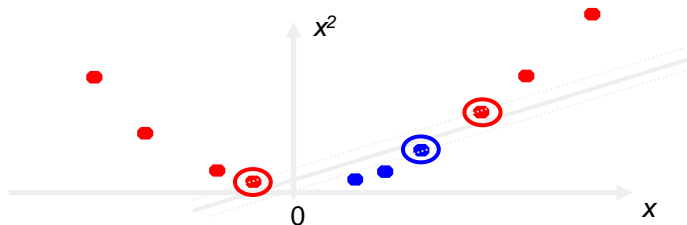
- Dados que são linearmente separáveis (mesmo com algum ruído) funcionam bem:



- Mas e se a base de dados for mais complexa que isso?



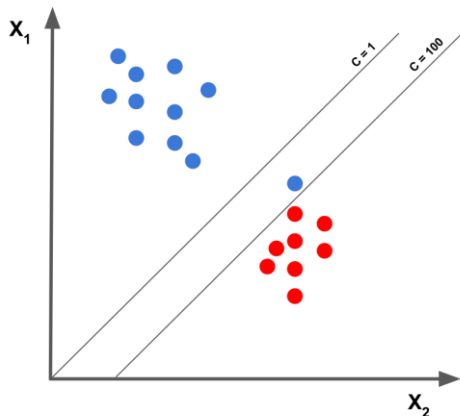
- E se mapearmos os dados para uma dimensão maior?



Classificação

SVM - Regularização

- **C**: custo das violações
- Balanceamento entre uma fronteira de decisão suave e uma fronteira que classifique corretamente todos os pontos.

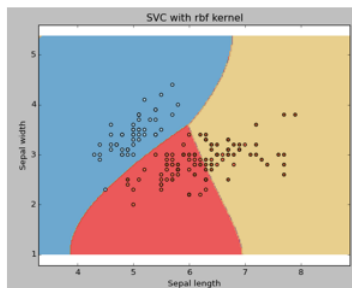


Gamma: raio de influência.

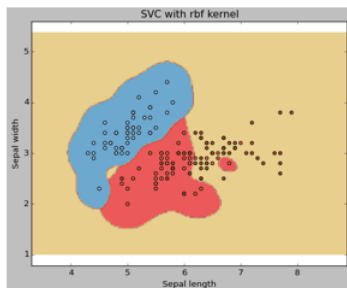
Valores baixos: influência longe → separação mais suave

Valores altos: influência perto → separador que valoriza classificação correta

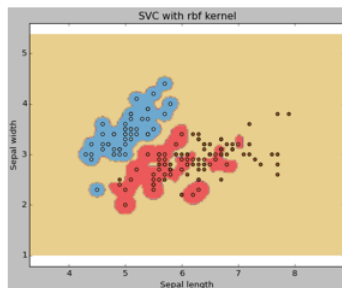
gamma = 0



gamma = 10



gamma = 100



Hiperparâmetros

Valores baixos: separação mais suave

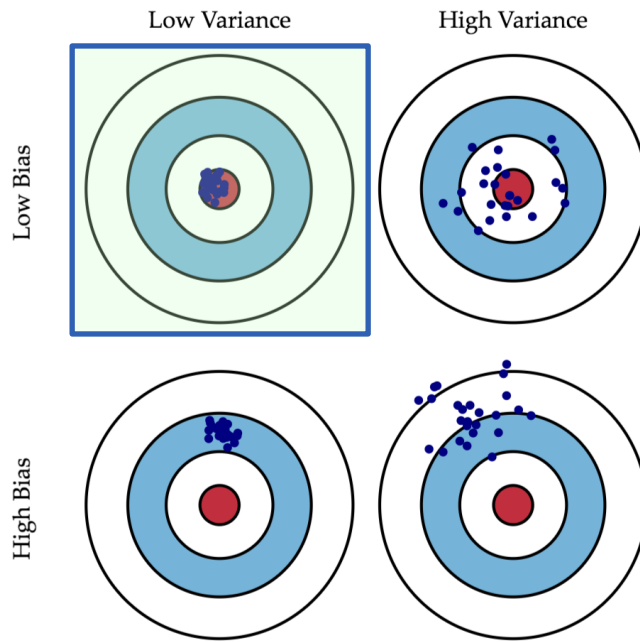
Valores altos: classificação correta

Classificação

Compromisso Viés Variância

Viés (bias): incapacidade do modelo de ML de capturar o verdadeiro relacionamento entre os dados.

Variância: diferença no resultado do modelo de ML para diferentes datasets.



Para modelos de *Machine Learning* em geral !

Classificação – Estudo de Caso

SVM

- Dígitos de 0 a 9 escritos à mão



Curiosidades:

- Yann LeCun foi um dos criadores do dataset
- Motivação:
 - benchmark para reconhecimento de dígitos
 - processamento automático de cheques

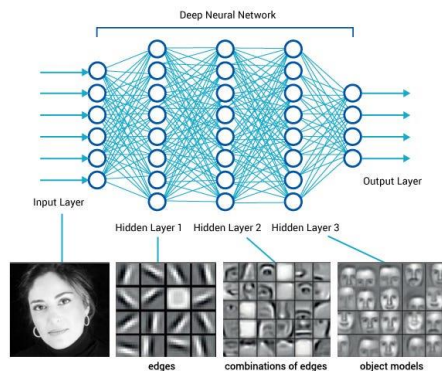
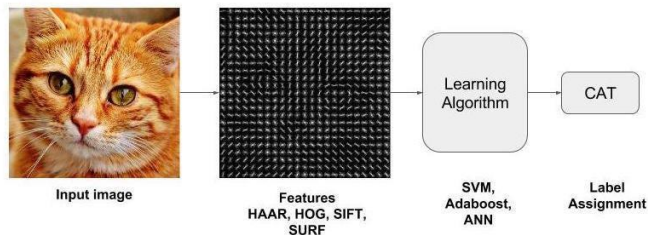
Classificação – Estudo de Caso

SVM

- Importância de se fazer Data Augmentation:
Generalização do Modelo!

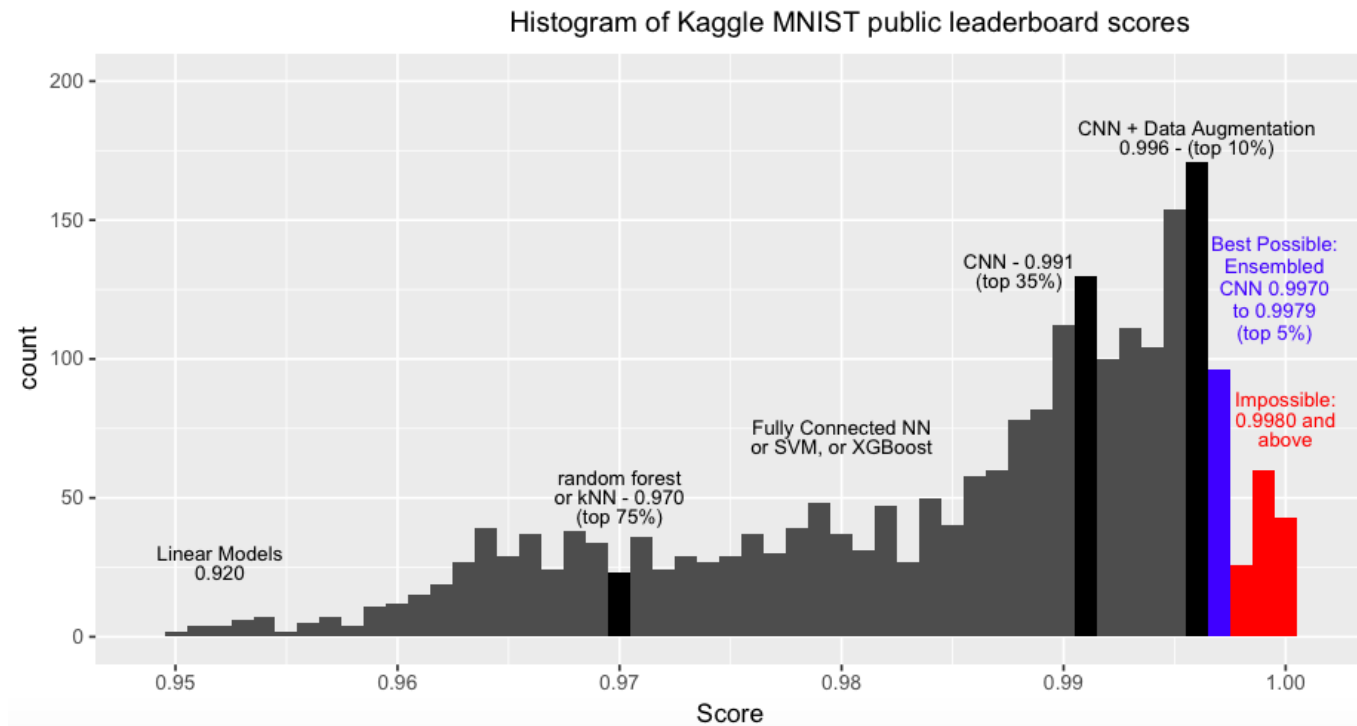


- Usamos os próprios pixels como input, mas não seria melhor, intuitivamente pensando no algoritmo, passarmos features das imagens?



Classificação – Estudo de Caso

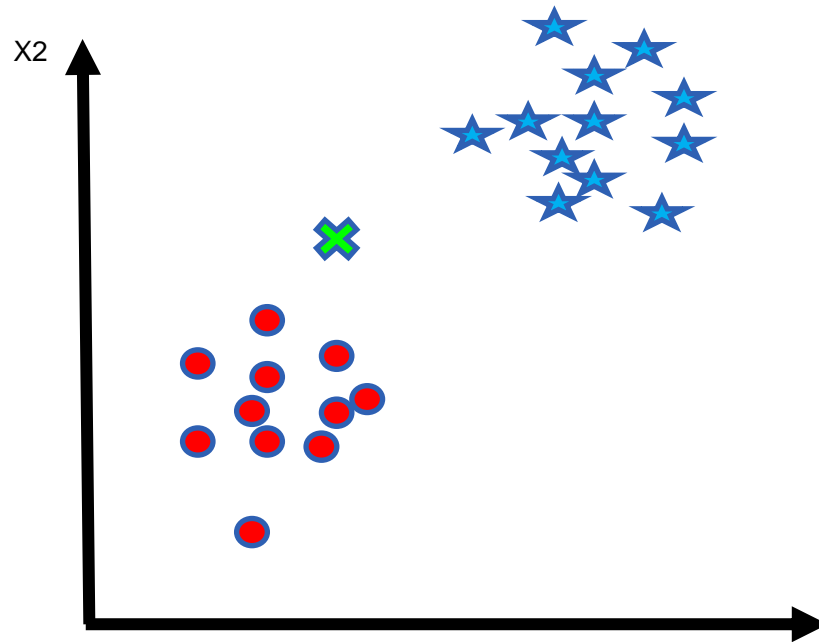
SVM



KNN

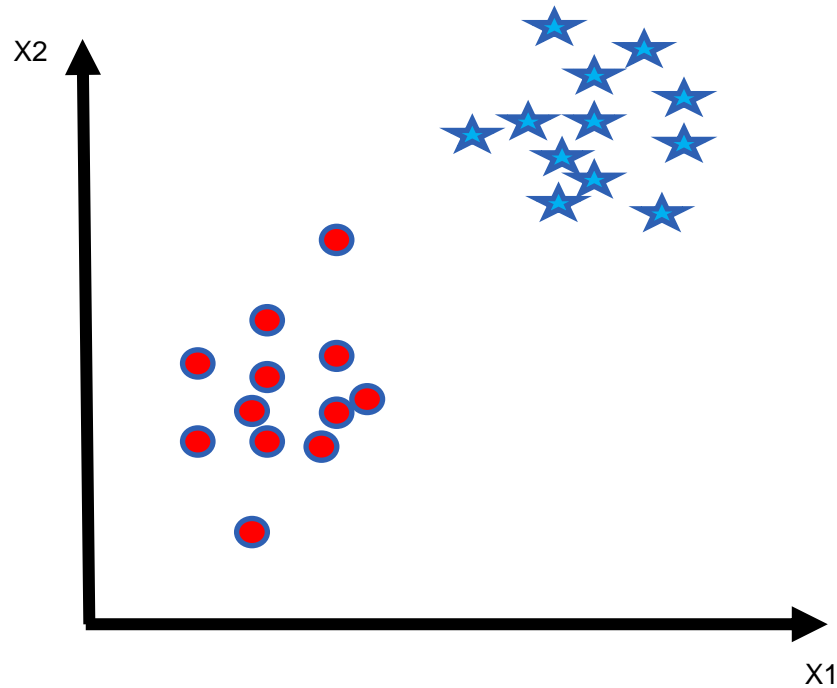
Classificação

KNN



Classificação

KNN



Classificação

KNN

Passos do KNN

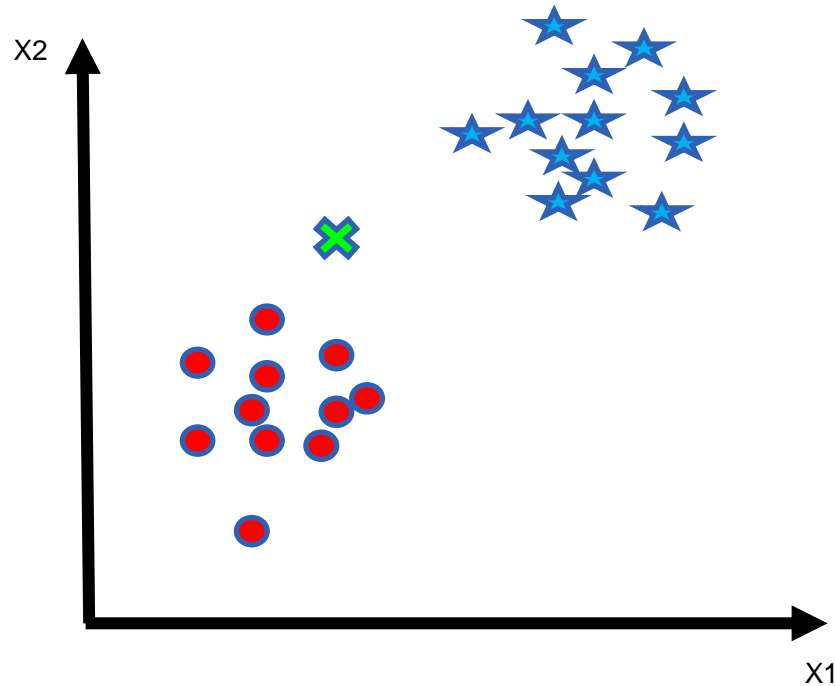
1. Determinar o valor de K , ou número de vizinhos
2. Calcular a distância entre cada par de registros
3. Determinar quais são os K registros (vizinhos) mais próximos do novo registro
4. Dentre esses K vizinhos, contar o número de vizinhos em cada classe
5. O novo registro vai ser da classe majoritária entre os vizinhos mais próximos

Classificação

KNN

Passo 1: Determinar o valor de K, ou número de vizinhos

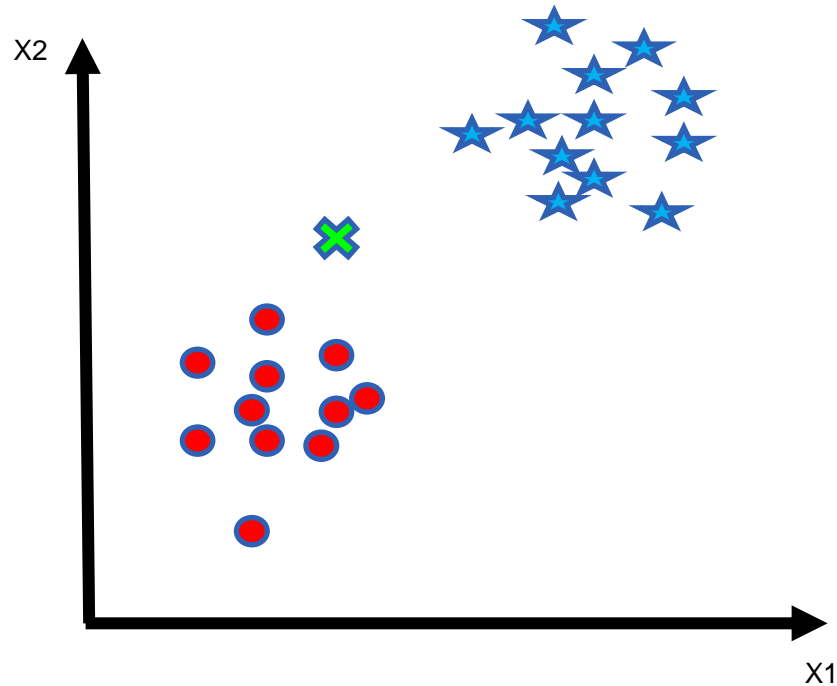
K = 5



Classificação

KNN

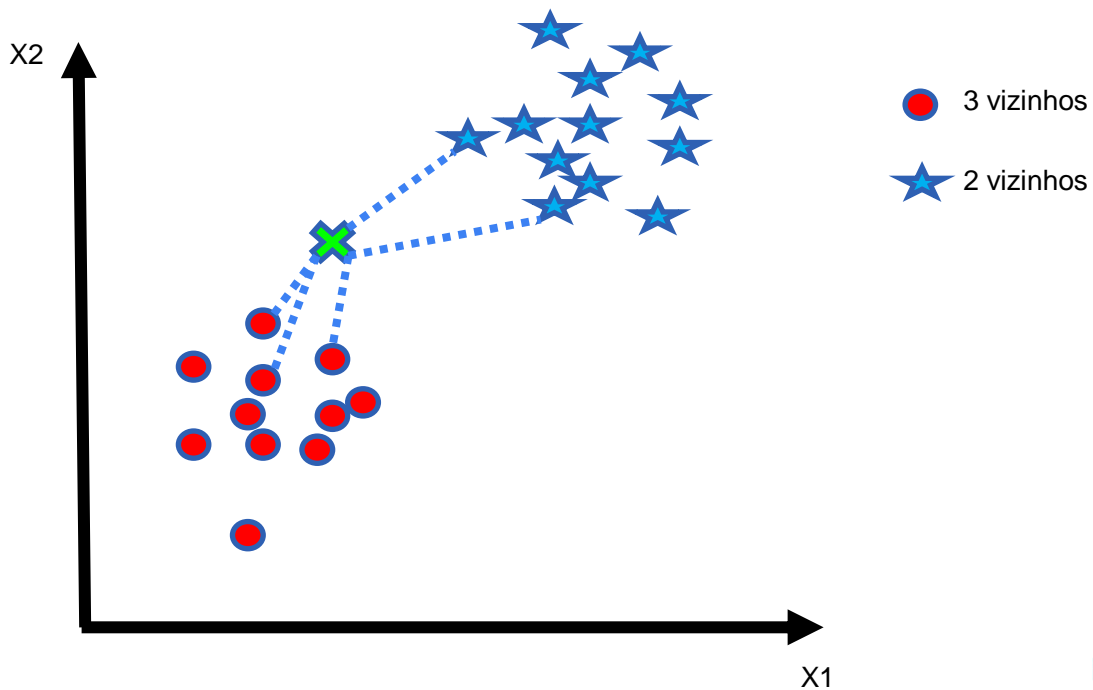
Passo 2: Calcular a distância entre cada par de registros



Classificação

KNN

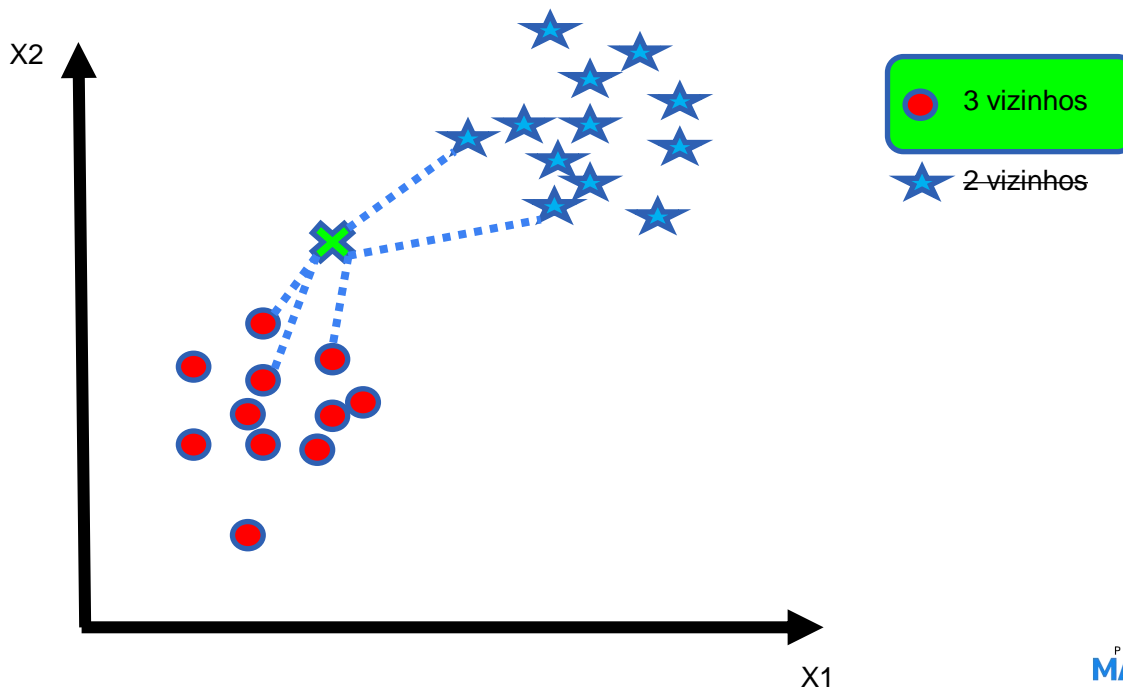
Passo 4: Dentre os 5 vizinhos, contar o número de vizinhos em cada classe



Classificação

KNN

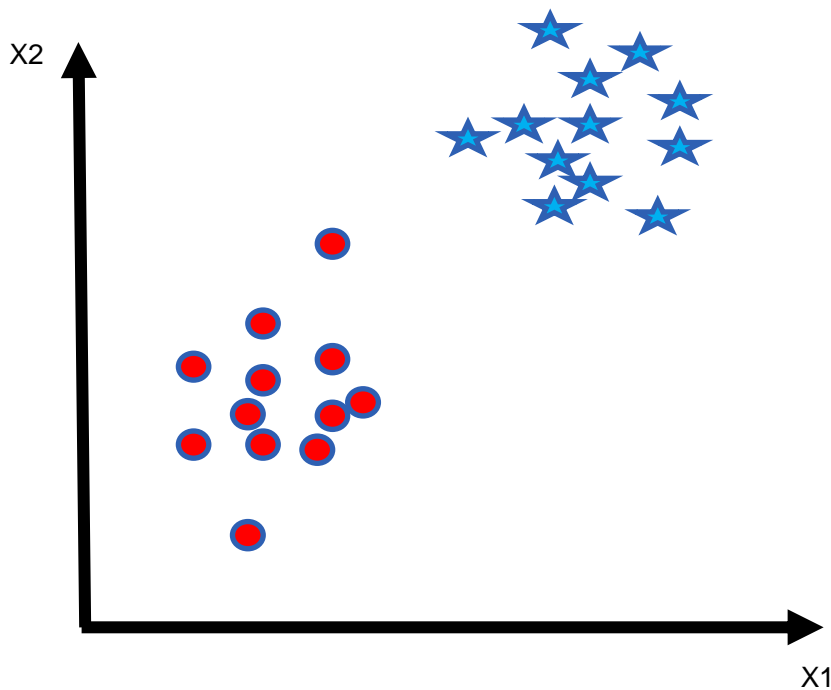
Passo 5: O novo registro vai ser da classe majoritária entre os vizinhos mais próximos



Classificação

KNN

Passo 5: O novo registro vai ser da classe majoritária entre os vizinhos mais próximos



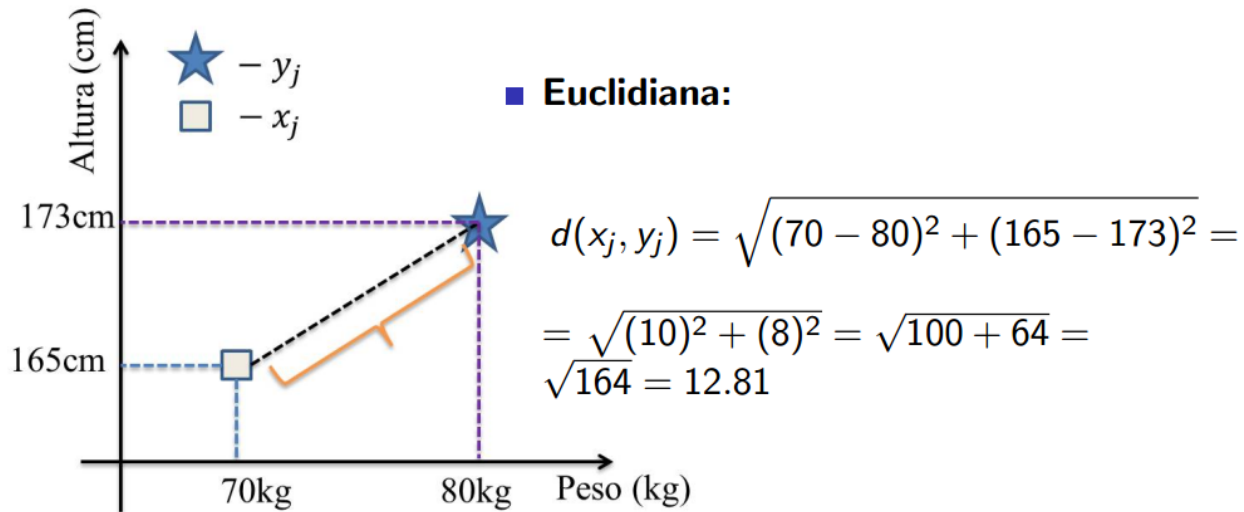
Classificação

KNN

- **A classe do novo padrão é igual ao da K maioria mais próxima.**
- **Pendências:**
 - Qual tipo de distância usar?
 - Qual valor de K?
 - Como Desempatar?

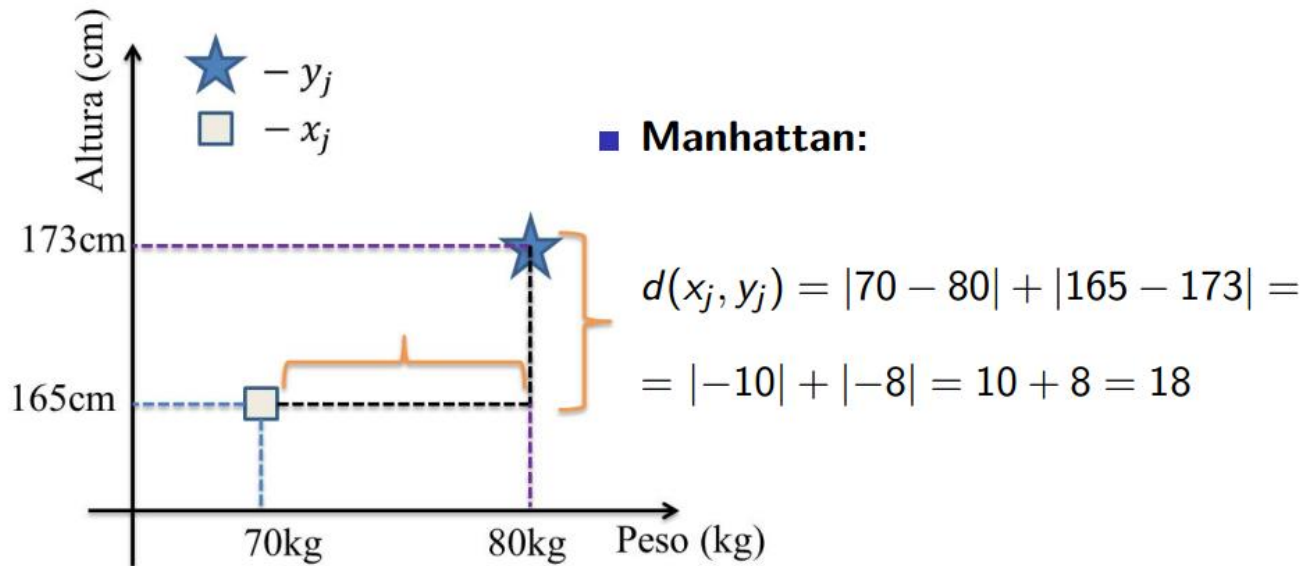
Classificação

KNN



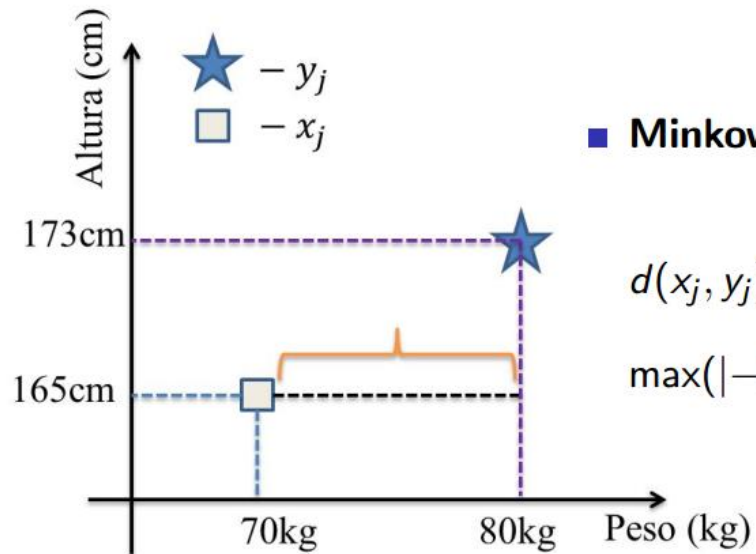
Classificação

KNN



Classificação

KNN



■ Minkowski:

$$d(x_j, y_j) = \max(|70-80|, |165-173|) =$$
$$\max(|-10|, |-8|) = \max(10, 8) = 10$$

Classificação

KNN

- A classe do novo padrão é igual ao da K maioria mais Próxima.
- Pendências:
 - Qual tipo de distância usar?
 - Qual valor de K?
 - Como Desempatar?

Escolha experimental

Classificação

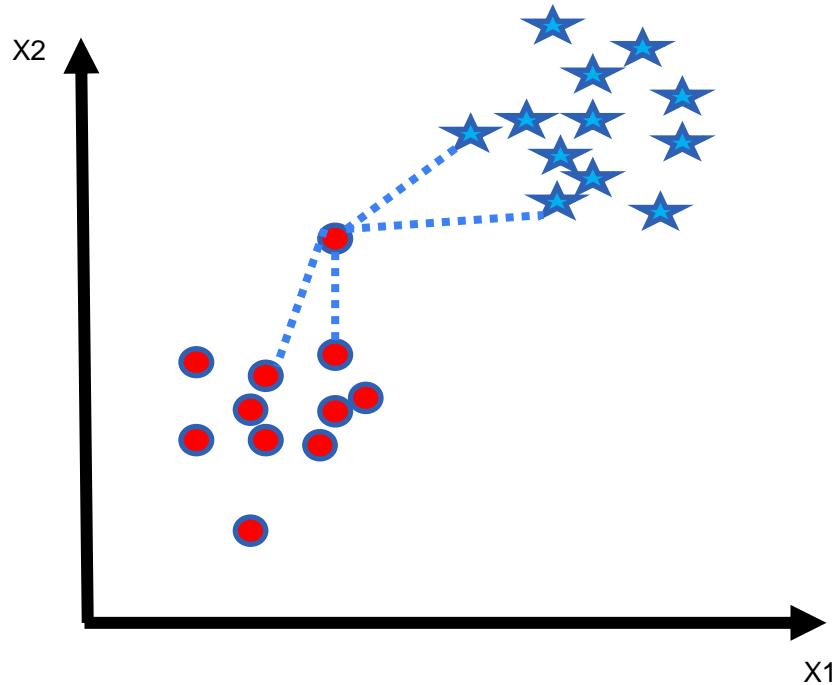
KNN

- A classe do novo padrão é igual ao da K maioria mais Próxima.
- Pendências:
 - Qual tipo de distância usar?
 - Qual valor de K?
 - Como Desempatar?

Escolha aleatória
Escolha aleatória ponderada
Classe mais próxima

Classificação

KNN



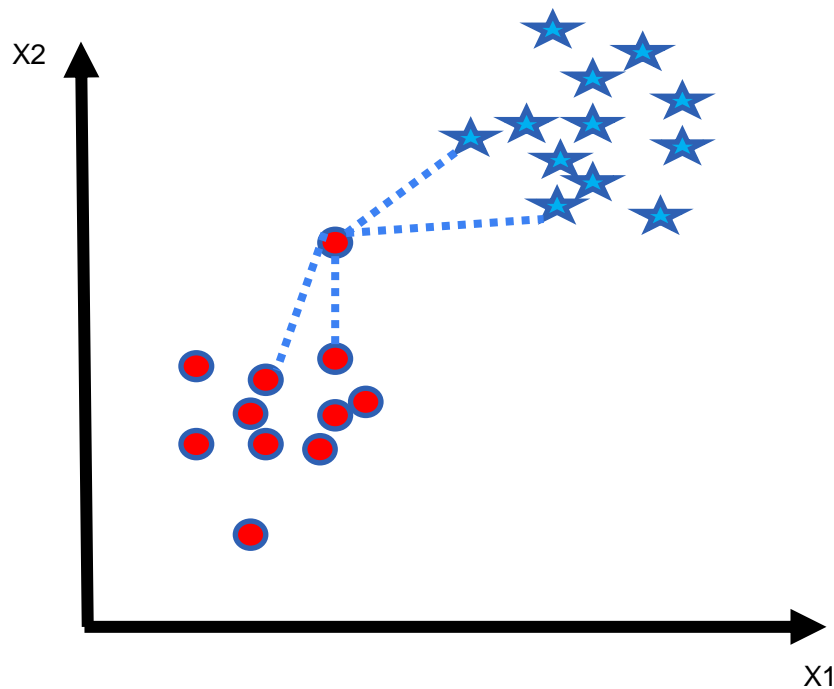
Supondo $K = 4$

Escolha Aleatória:

“Jogue uma moeda honesta”: caso saia cara, escolha a classe vermelha, caso saia coroa, escolha a classe azul.

Classificação

KNN



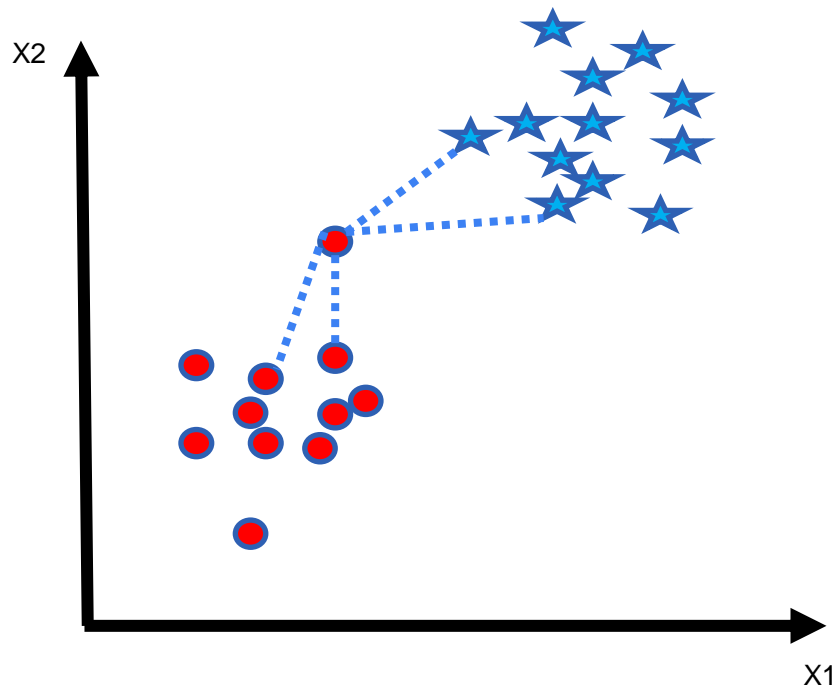
Supondo $K = 4$

Escolha Aleatória Ponderada:

“Jogue uma moeda desonesta”: Dê mais chance à classe que está relacionada a classe que possua mais padrões.

Classificação

KNN



Supondo $K = 4$

Classe mais próxima:

Selecione a classe cuja distância é menor.

Estudo de Caso

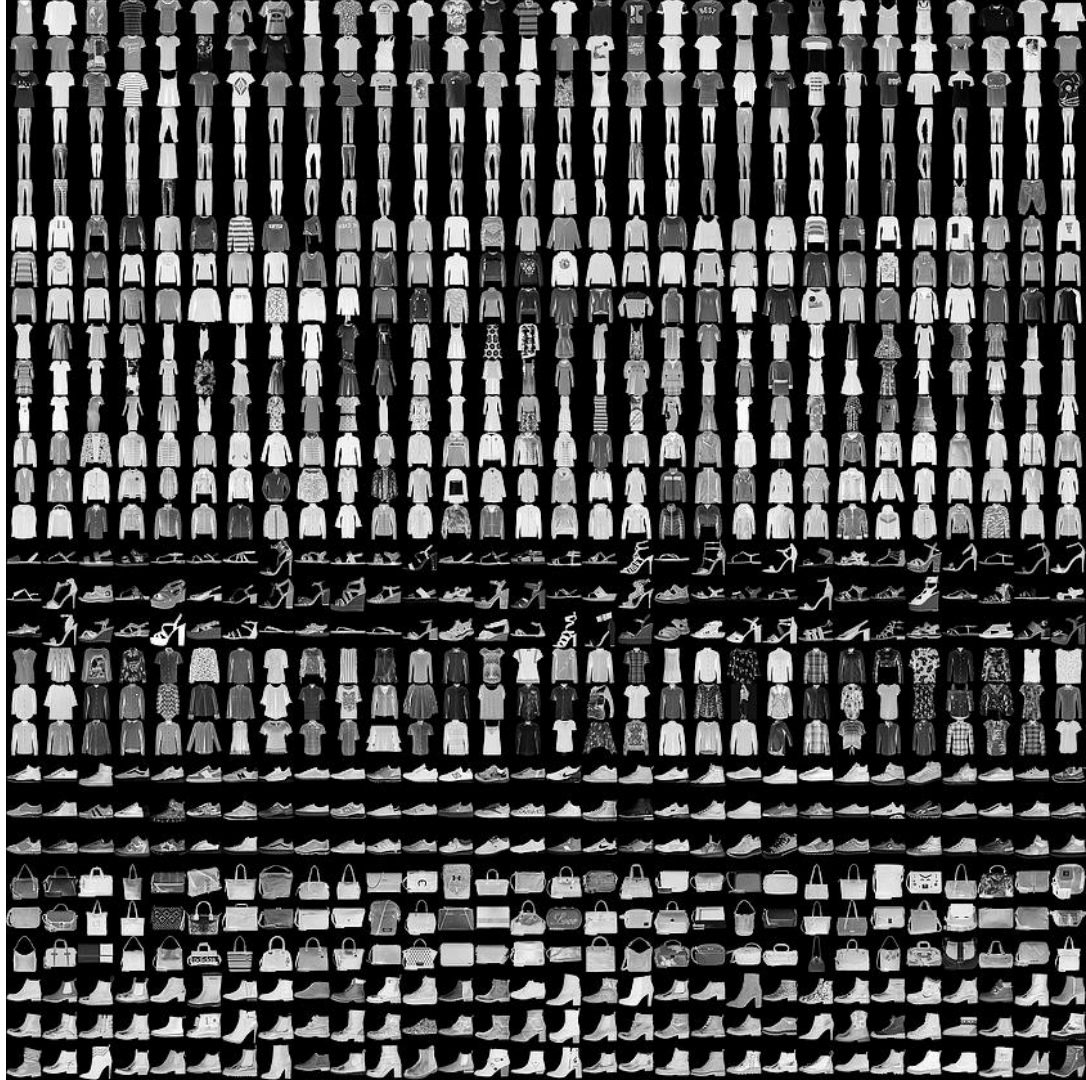
Classificação – Estudo de Caso

KNN

- Fashion-MNIST
 - 60k imagens de treino
 - 10k imagens de teste
 - 10 classes
 - 28x28
 - 8 bits
 - Grayscale

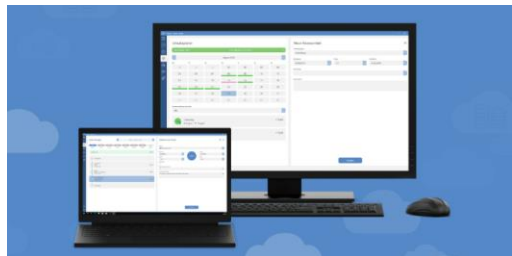
Curiosidades:

- Introdução de uma base de imagens 28 x28 (8 bits) grayscale com um pouco mais de complexidade que o dataset original Mnist.



Deploy com API

Deploy API



App desktop ou web

Executável por linha de comando

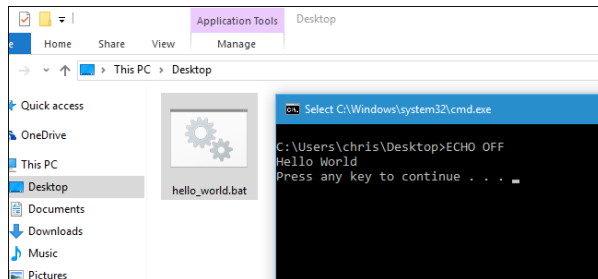
```
C:\Windows\system32\cmd.exe

codename -help
usage: codename [-command] [-path] [-option] [-value] [-option] [-value]]
[command]:
Config      Create or modify a project file with the specified options.
Generate    Generates the app specified in the project file.
Refresh     Refreshes metadata and generated the app. Data models and
            controllers are re-created.

Option(s):
AppName     Specifies a user-friendly application name.
ClassLibrary Enables batch editing in the app.
Copyright   Enables CMC (Content Management System) in the database.
DBConn      Specifies a database connection string for the app.
DBConnection Enables user and role management in the database.
DBProvider  Specifies data provider (MsSql, Oracle, SqlDatabase, etc.)
DBSessionId Enables session state management in the database.
DiscoveryPath Specifies a relationship discovery path for data models.
Framework  Specifies the version of Microsoft .NET Framework for the app.
Language    Specifies C# or VisualBasic for the app implementation.
Multiple    Enables multiple selection in the app.
Namespace   Specifies a namespace for the app source code.
NewDatabase Generates database objects defined in the app.
PageHeader  Specifies a header display in all pages of the app.
PageType    Specifies HTML or ASP page implementation.
Preview      Enables launching of default web browser to preview the app.
UIInterface Specifies touchless or classic for the app user interface.
Reports     Enables reporting based on Microsoft Report Viewer.
SortDates   Enables smart display of dates.
Theme       Specifies the name of the app theme.

Use -OptionName <Value> to change project options.
Option names are case-insensitive. Inter-semicolon spaces around option values
with "space" character. Unless "true" can be omitted for boolean options.

>
```



Bat script

API (agnóstica a linguagem)



Via código

Estamos em ambiente de produção.

1. Carregaremos o modelo treinado
2. Carregaremos novos dados a serem inferidos
3. Faremos as inferências
4. FIM

```
[ ] # 1. Carregaremos o modelo treinado
import pickle
filename = 'model.sav'
model = pickle.load(open(filename, 'rb'))

[ ] # 2. Carregaremos novos dados a serem inferidos
import pandas as pd
new_data = pd.read_csv('Dataset_spine_unknown.csv')
new_data.head()

[ ] # 3. Faremos as inferências
inferences = model.predict(new_data)
print(inferences)
```

FIM

Deploy API

- Queremos criar uma API (rodando localmente) usando o framework FastAPI com documentação e utilização através do OpenAPI (Swagger).
- Mais pra frente faremos Deploy no Heroku ou na Oracle!

VC MASTER - PUC-Rio

0.1.0 OAS 3.1

/openapi.json

default

POST /predict Predict

Parameters

Cancel

Reset

No parameters

Request body *required*

multipart/form-data

file *required*

string(\$binary)

Escolher arquivo

Bag.png

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:5000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@Bag.png;type=image/png'
```

Request URL

http://127.0.0.1:5000/predict

Server response

Code	Details
200	<div>Response body<div><pre>{ "Prediction": ["Bag"] }</pre></div><div>Response headers<div><pre>content-length: 22 content-type: application/json date: Fri, 05 Jul 2024 15:15:45 GMT server: uvicorn</pre></div></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Obrigada!

Prof. Manoela Kohler



prof.Manoela@ica.ele.puc-rio.br



www.linkedin.com/in/manoelakohler