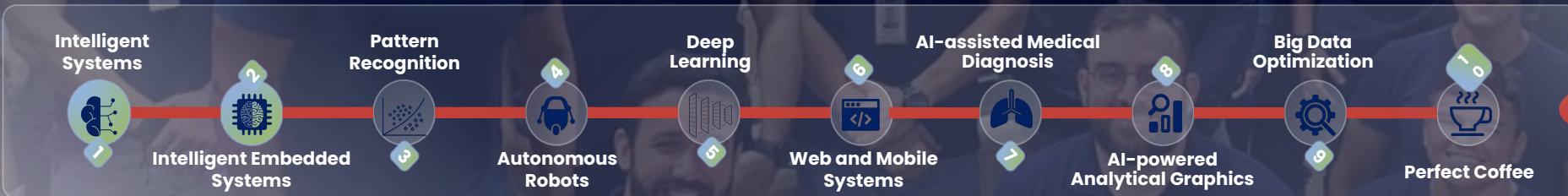




# LABORATORY FOR PROCESSING IMAGES, SIGNALS AND COMPUTER SCIENCE

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ



[lapisco.ifce.edu.br](http://lapisco.ifce.edu.br)



[@lapisco.ifce](https://www.instagram.com/lapisco.ifce)



[company/lapisco-ifce](https://www.linkedin.com/company/lapisco-ifce)



[lapiscoifce](https://www.youtube.com/lapiscoifce)



[lapiscoifce](https://www.facebook.com/lapiscoifce)




[@lapiscoifce](https://twitter.com/lapiscoifce)

## Data Science - Trainee

# Otimização de Hiperparâmetros

Teoria, *Randomized Search* e *Grid Search*

# Otimização de Hiperparâmetros

- 
- **Parâmetro x Hiperparâmetro**
  - **Otimização de Hiperparâmetros**
  - **Randomized Search**
  - **Grid Search**
  - **Código**

# Parâmetros x Hiperparâmetros

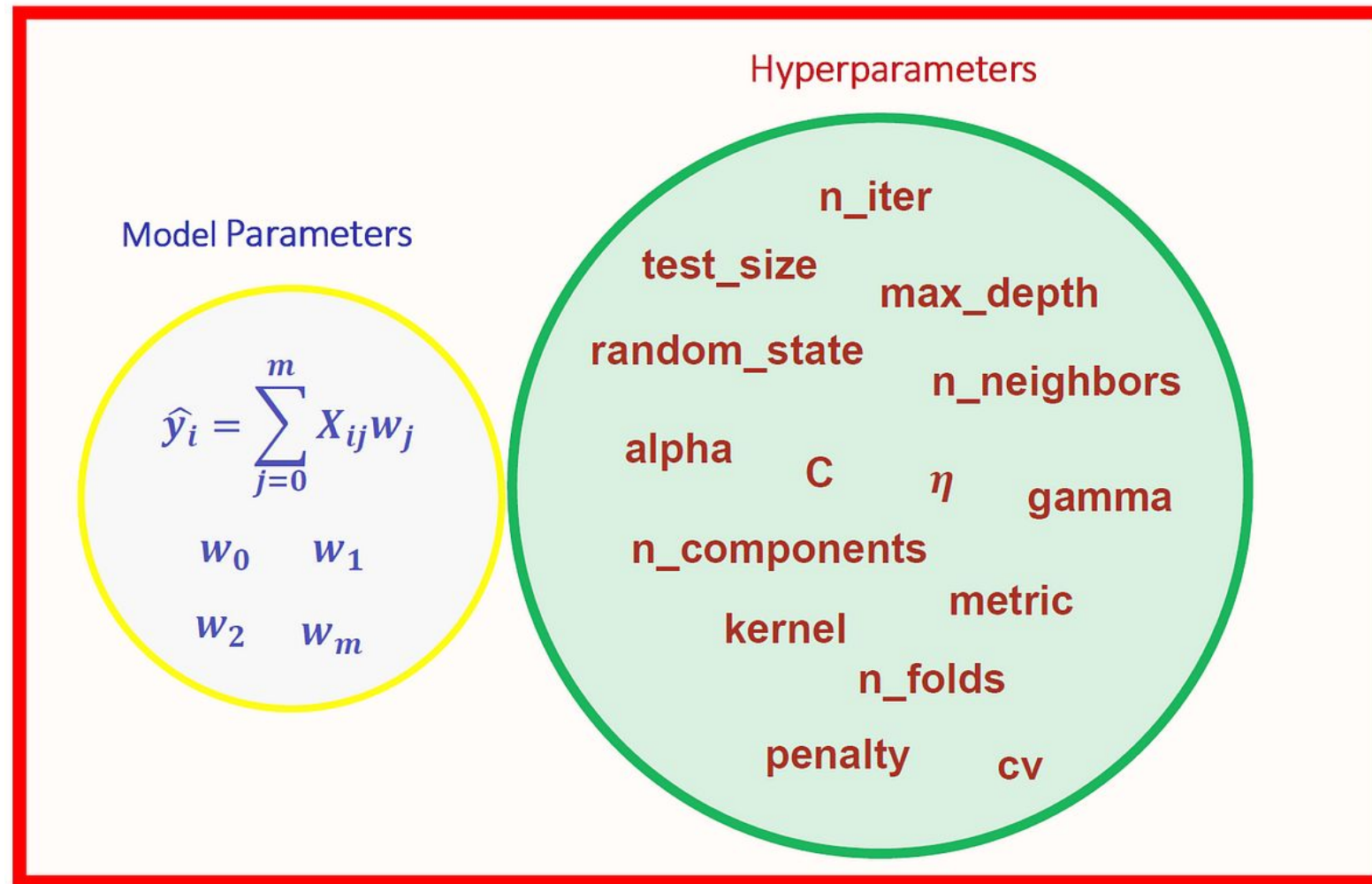
## Parâmetros

Os parâmetros são aqueles que são ajustados no próprio processo de aprendizagem. Ex.: Pesos de uma rede neural, coeficientes de uma regressão linear.

## Hiperparâmetros

São variáveis definidas antes do treinamento, que modificam diretamente o comportamento e a forma como o algoritmo irá aprender. Ex.: Número de neurônios de uma rede neural, número de vizinhos do KNN ou a métrica de avaliação de uma regressão.

# Parâmetros x Hiperparâmetros





# Otimização de Hiperparâmetros

## O que é

É a abordagem de pesquisar objetivamente diferentes valores para hiperparâmetros do modelo e escolher um subconjunto que resulte em um modelo que alcance o melhor desempenho em um determinado conjunto de dados.

## Objetivo

Obter um modelo com a melhor performance para o problema proposto, seja classificação ou regressão, utilizando os hiperparâmetros otimizados.

# Grid Search

## O que é

O cientista de dados configura uma grade de valores com os hiperparâmetros do modelo, e para cada combinação ele treina um modelo e pontua o mesmo nos dados de teste. Aqui cada combinação dos valores é testada e isso pode ter um alto custo computacional.

## Exemplo

```
mlp_clf_wgs = MLPClassifier()  
params_mlp = {'max_iter': [2000, 2500],  
              'activation': ('identity', 'logistic', 'tanh', 'relu'),  
              'solver': ('lbfgs', 'sgd', 'adam'),  
              'learning_rate': ('constant', 'invscaling', 'adaptive')}  
mlp_GS = GridSearchCV(mlp_clf_wgs, params_mlp, cv=5)  
mlp_GS.fit(X_transf, y)
```

# Randomized Search

## O que é

Neste caso é configurado também uma grade de valores de hiperparâmetros, porém, diferente do método passado, neste é selecionado uma combinação aleatória para treinar o modelo e pontuar nos dados de teste.

## Exemplo

```
svm_clf_linear_wgs = SVC(kernel='linear', random_state=42)
params_svm_linear = {'C':[0.1, 1, 10], 'shrinking':(True, False),
                    'probability': (True, False)}
svm_linear_GS = RandomizedSearchCV(svm_clf_linear_wgs, params_svm_linear, verbose=4)
svm_linear_GS.fit(X_train, y_train)
```



# Código - Jupyter Notebook

Implementando o *Randomized Search* e o *Grid Search* usando a biblioteca **sklearn**



# LABORATORY FOR PROCESSING IMAGES, SIGNALS AND COMPUTER SCIENCE

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ

