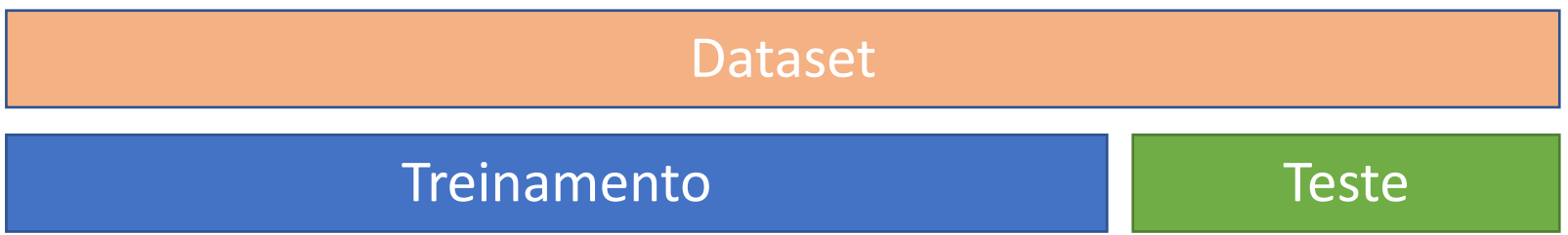
Introdução à Inteligência Artificial
Aula 3

Professor: Alex Pereira

Procedimento simplificado de avaliação

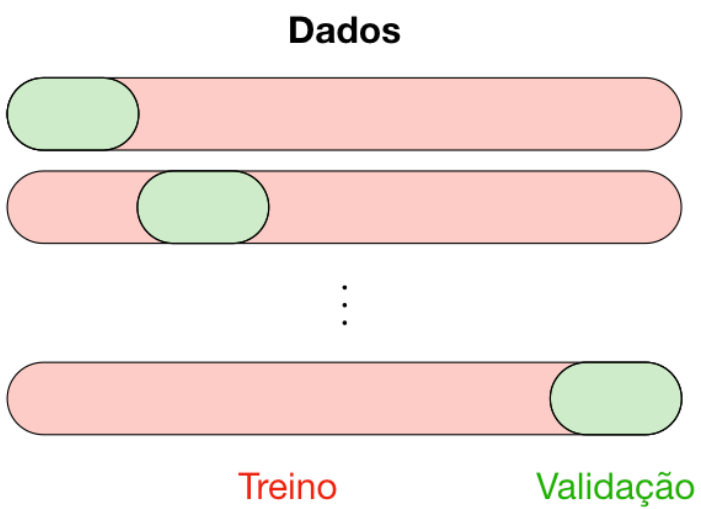
- Particiona-se o Dataset em Treinamento e Teste
  - Por exemplo: 80%/20% ou 70%/30%
    - ✓ Não há valor ideal. Depende do problema. Empírico.
      - Também conhecido como Método Holdout
- Garante-se que o modelo aprende e **generaliza para dados não vistos**
  - Avaliar o seu modelo sobre o dado de treinamento não é uma boa prática
    - ✓ Equivalente a avaliar o aprendizado escolar com uma prova igual a lista de exercícios



E se houver poucos dados rotulados ?

- Validação cruzada (**Cross-validation**) ou **K-Fold**

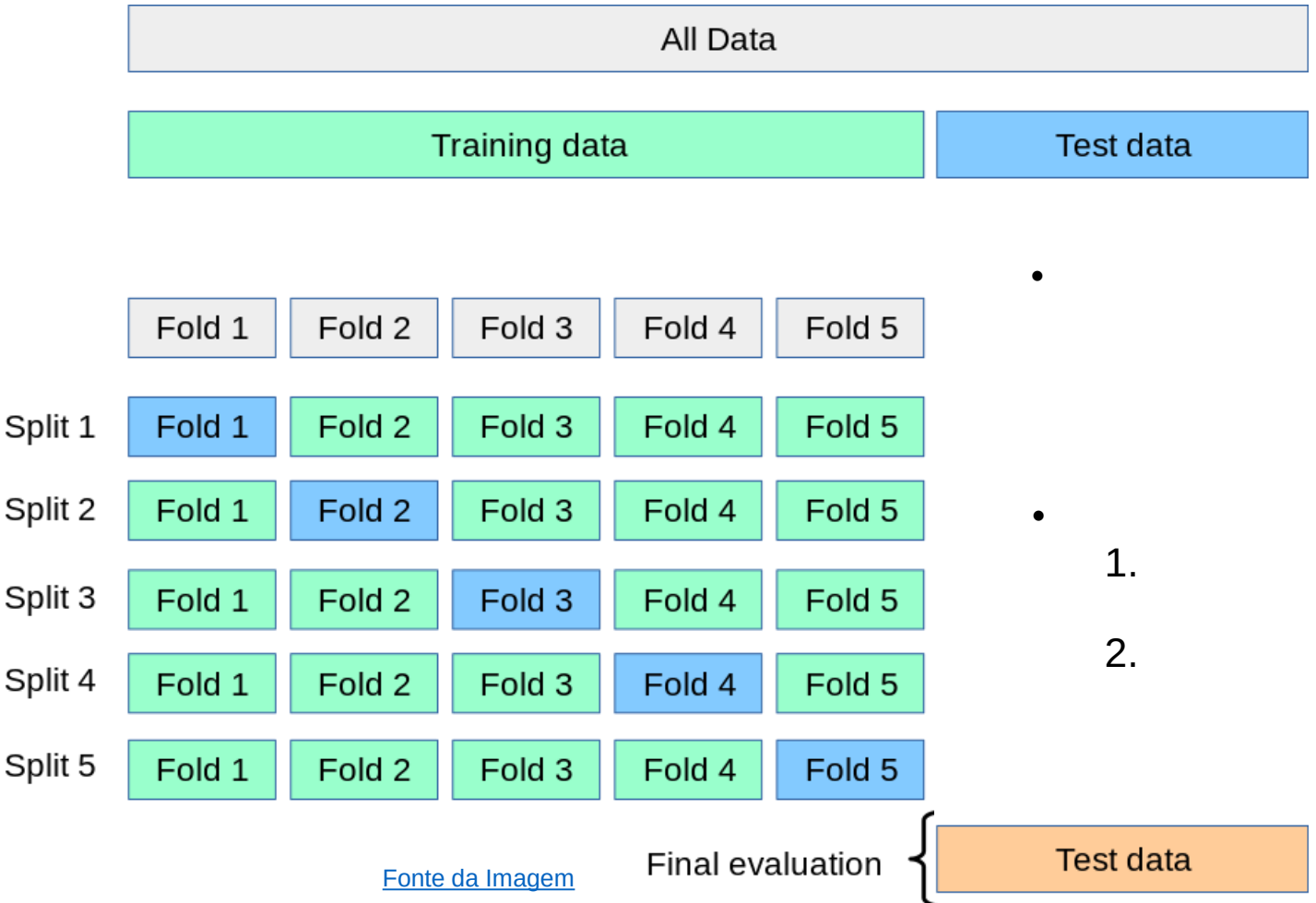
Às vezes chamado de Teste também



Parte	Dados	Erro de validação	Erro de validação cruzada
1		ϵ_1	
2		ϵ_2	$\frac{\epsilon_1 + \ldots + \epsilon_k}{k}$
⋮	⋮	⋮	
k		ϵ_k	

Fonte da Imagem

Exemplo de K-Fold (k=5)



-
-
  1.
  2.

Fonte da Imagem

Implementando a Validação Cruzada

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=10000,  random_state=42
)
```

```
scores = cross_validate(mlp, X, y, scoring=scoring, cv=5,
return_train_score=False)
```

Link para o Notebook no Colab

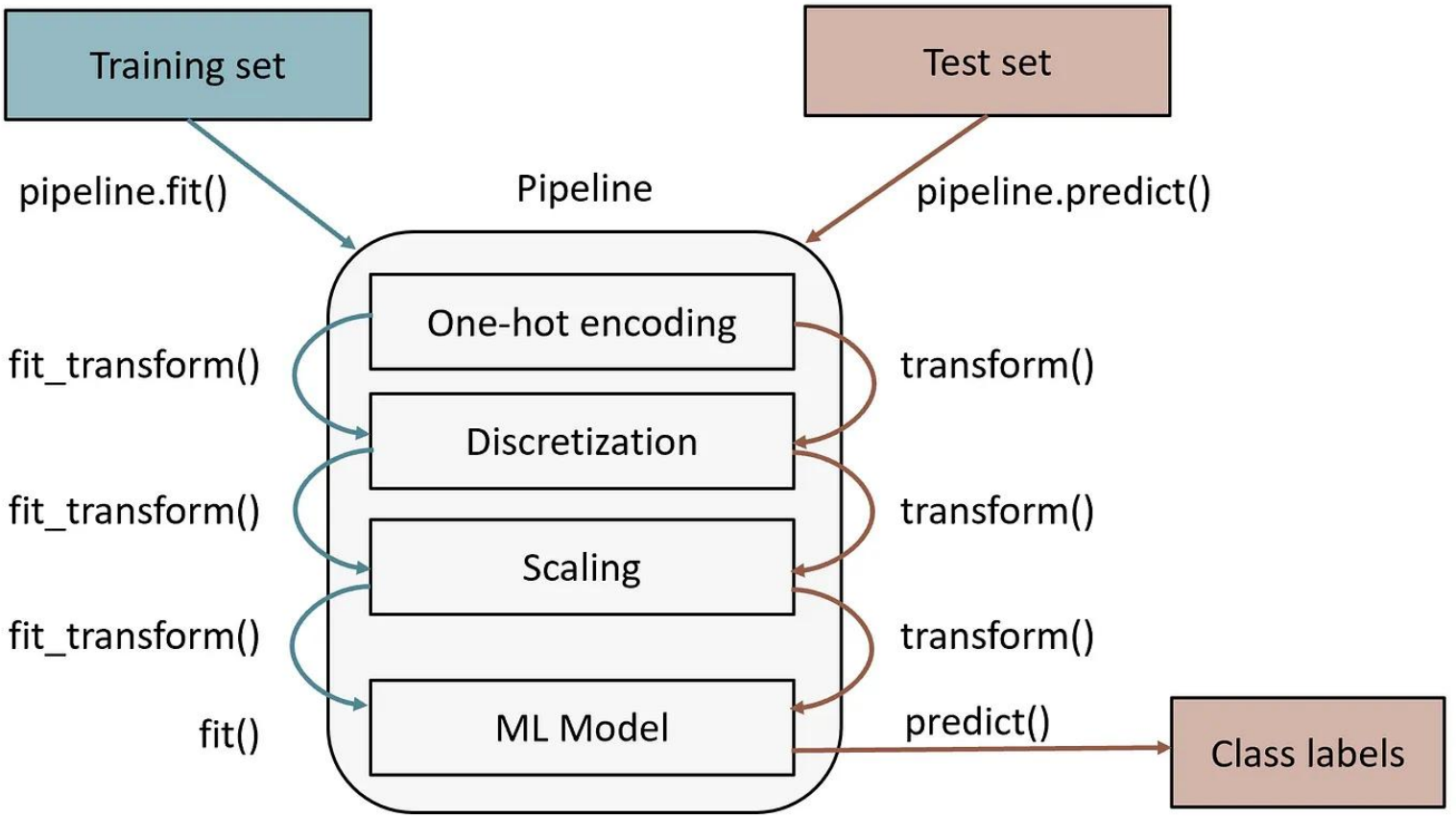
Ou em mais detalhes no outro Notebook

Pipeline

Da biblioteca scikit-learn

Automatizar o tratamento dos dados

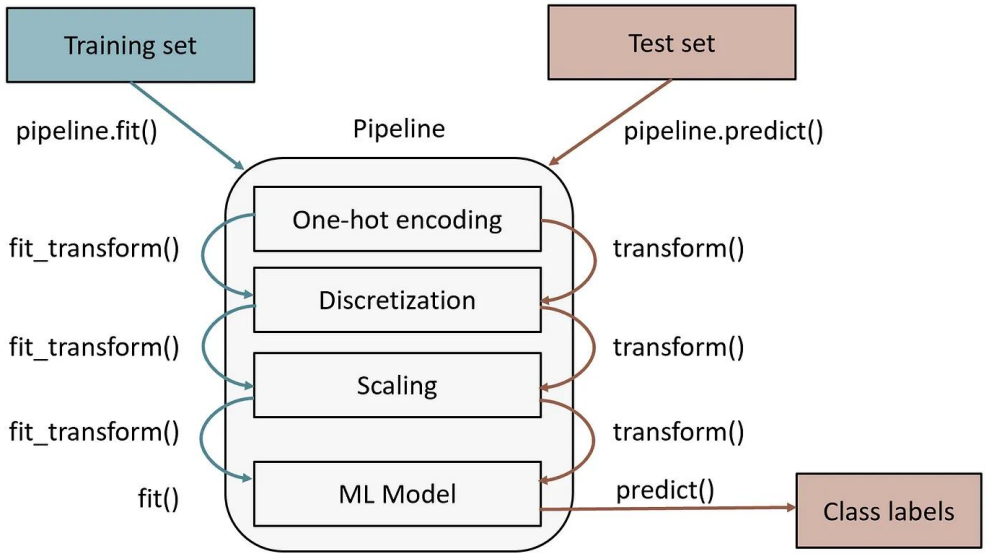
Mantendo consistência entre o treino e o teste



Fonte da Imagem

Automatizar o tratamento dos dados

Mantendo consistência entre o treino e o teste



- pipeline.fit()
  - Executa todos os fit_transform e o fit do treinamento do modelo
- pipeline.predict()
  - Executa todos os transform e o predict do modelo treinado

Exemplo de Pipeline

```
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('mlp', MLPClassifier(hidden_layer_sizes=(100,), max_iter=500))   ])
# Ou use: make_pipeline(StandardScaler(), MLPClassifier( ... ))

X, y = load_iris(return_X_y=True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

pipeline.fit(X_train, y_train)

y_pred = pipeline.predict(X_test)
print("Predicted labels:", y_pred)

accuracy = pipeline.score(X_test, y_test)
print(f"Test accuracy: {accuracy:.2f}")
```

Quais colunas foram transformadas no pipeline anterior?

- E se houvessem variáveis categóricas e numéricas?
  - Use o ColumnTransformer

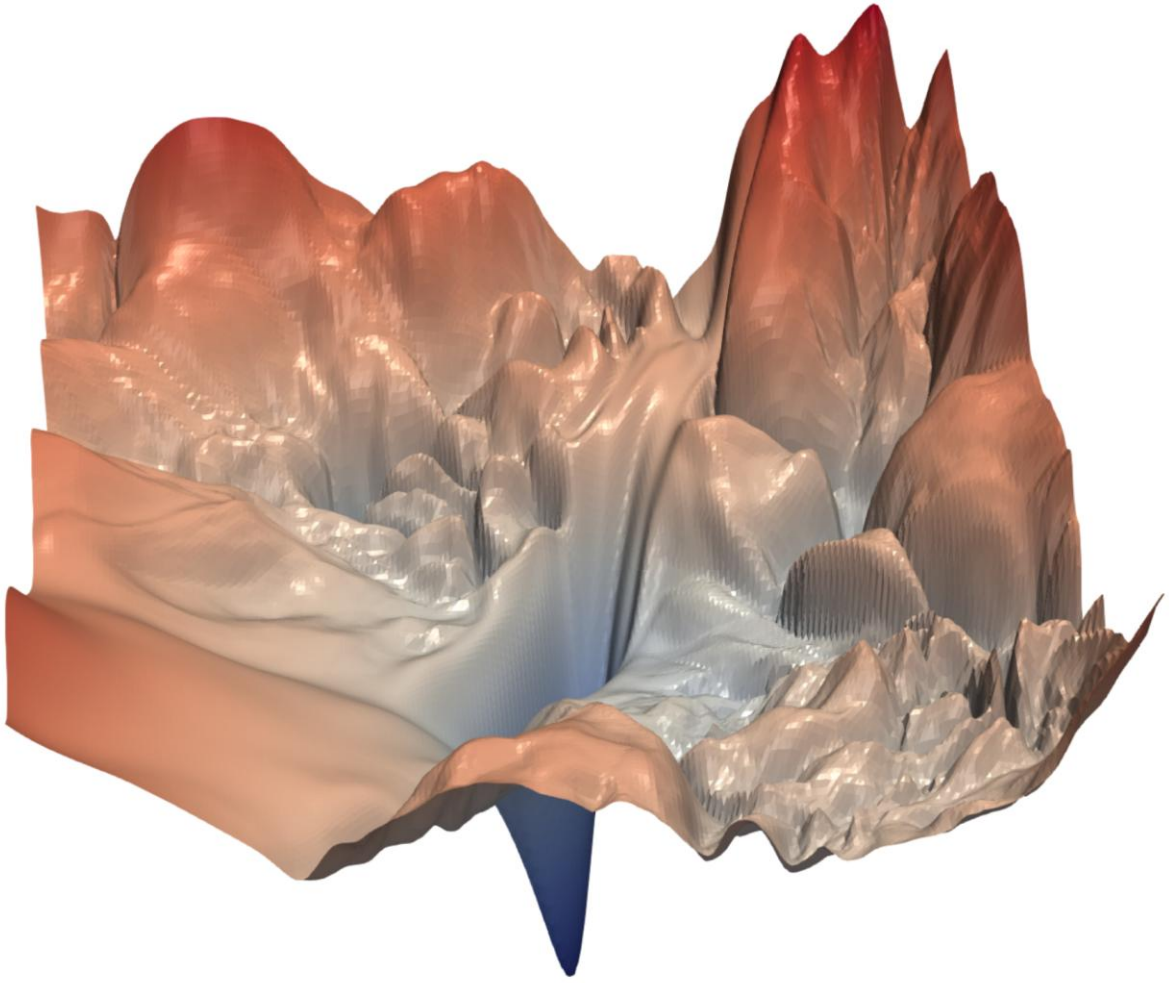
```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), ['idade', 'renda']),
        ('cat', OneHotEncoder(), ['sexo', 'ocupacao'])
    ])

full_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('mlp', MLPClassifier())
])
```

Um astrônomo, um físico e um matemático, estão viajando de trem

- Ao passar por uma paisagem rural, eles avistam uma ovelha negra do lado de fora da janela.
- Astrônomo
  - Olha, todas as ovelhas nesse país são negras!
- Físico
  - Não, não é isso. Pelo menos uma das ovelhas nesse país é negra.
- Matemático
  - Pelo menos um lado de uma ovelha nesse país é negro.

Superfície da função Custo de uma Rede Neural



Otimização de Hiperparâmetros

- O treinamento de redes neurais requer a minimização de uma função de perda não convexa de alta dimensão
  - uma tarefa que é difícil na teoria, mas às vezes fácil na prática.
- Apesar da dificuldade (NP-Hard) do treinamento de loss functions
  - métodos de gradiente simples frequentemente encontram minimizadores globais
    - ✓ configurações de parâmetros com perda de treinamento zero ou quase zero
- Esse bom comportamento não é universal
  - a treinabilidade de redes neurais é altamente dependente de escolhas de
    - ✓ design de arquitetura de rede, de otimizador, inicialização de variáveis entre outras.
    - ✓ o efeito de cada uma dessas escolhas na estrutura da superfície de perda subjacente não é claro.

Solução do Matemático

E não a do astrônomo nem do físico

- Automatizar os testes de muitas combinações de hiper parâmetros
  - Numa quantidade **bem menor** do que a combinação de todas as variações de todos os hiper parâmetros.
- Soluções de busca num espaço multi-dimensional
  - Extratificação regular (Grid Search)
  - Algoritmo Genético
  - Aleatório (Random Search)
  - Probabilístico

Implementação no Sci-kitlearn

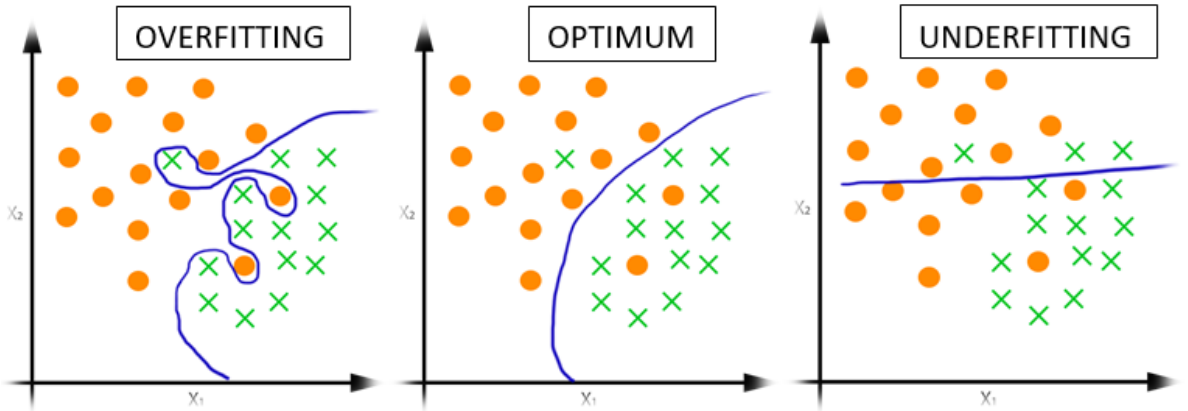
```
parameters = {
    'mlp__hidden_layer_sizes': [(50,), (100,), (50, 50)],
    'mlp__activation': ['relu', 'tanh'],
    'mlp__learning_rate_init': [0.001, 0.01]
}

grid_search = GridSearchCV(pipeline, parameters, cv=5)
grid_search.fit(X_train, y_train)

print(f"Best parameters: {grid_search.best_params_}")
```

Overfitting

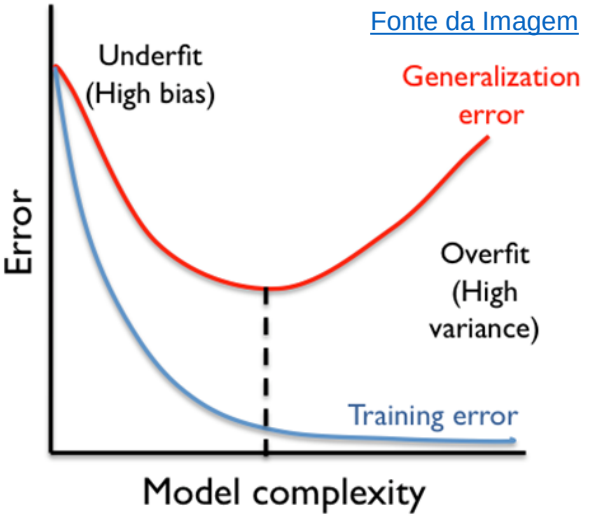
- Significa construir um modelo que ele incorpore
  - o efeito de ruídos presentes no seu conjunto de treinamento



Fonte da Imagem

Fonte da Imagem

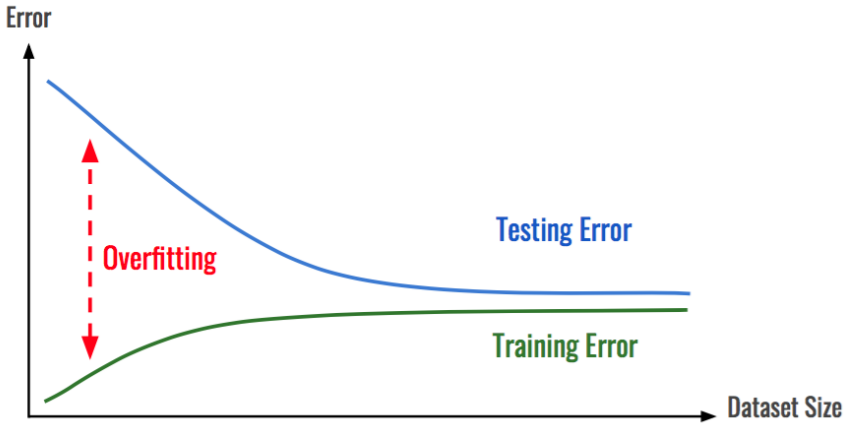
Overfitting: Como evitar?



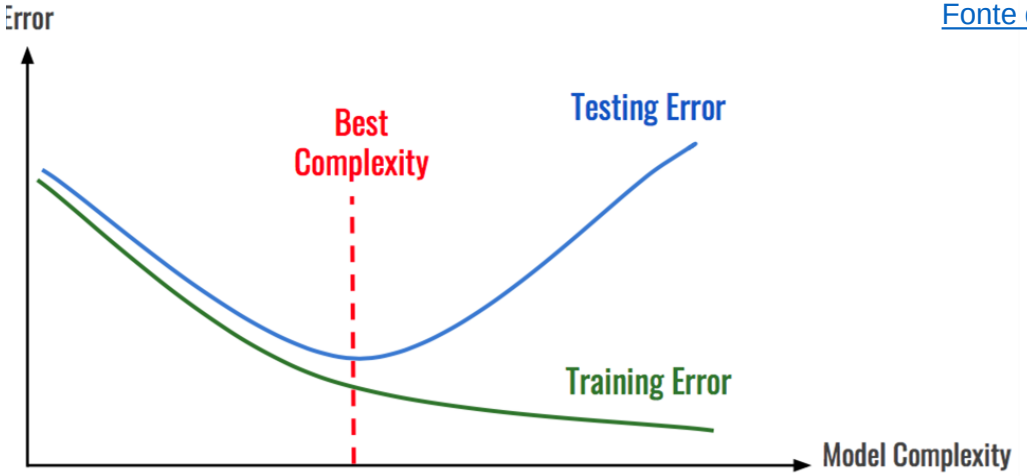
- Particionar os dados em
  - Treinamento e
  - Teste/validação
    - ✓ 70/30? 80/20? Ajuste empírico
      - Existem outras estratégias mais elaboradas (Cross validation).
- Procedimento:
  - Treinar o modelo no conjunto de dados de Treinamento e medir os erros
    - ✓ Aqui os erros são chamados de Erros de Treinamento
  - Validar (medir a acurácia) no conjunto de dados de Teste/validação
    - ✓ Aqui os erros são chamados de Erros de Generalização
      - Que são os erros cometidos ao tentar classificar
        - exemplos novos (não vistos até então)
  - Interromper o treinamento quando o erro de generalização
    - ✓ começar a aumentar

Overfitting: Como evitar?

- Coletar mais dados



- Simplificar o modelo



Fonte da Imagem

Early Stop

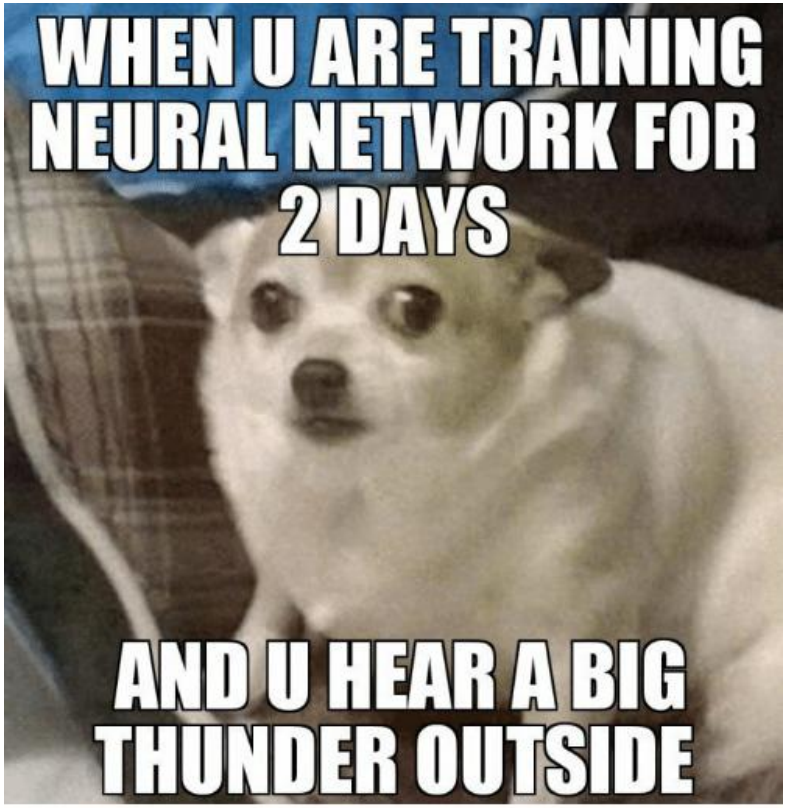
- Monitorar o desempenho do modelo em um conjunto de validação e
  - interromper o treinamento quando o desempenho nesse conjunto de validação para de melhorar
- Isso ajuda a evitar que o overfitting do modelo
  - o que poderia levar a um baixo desempenho de generalização
    - ✓ Ou seja, em dados novos e não vistos.

```
MLPClassifier(
        hidden_layer_sizes=(50,),
        early_stopping=True
    )
```

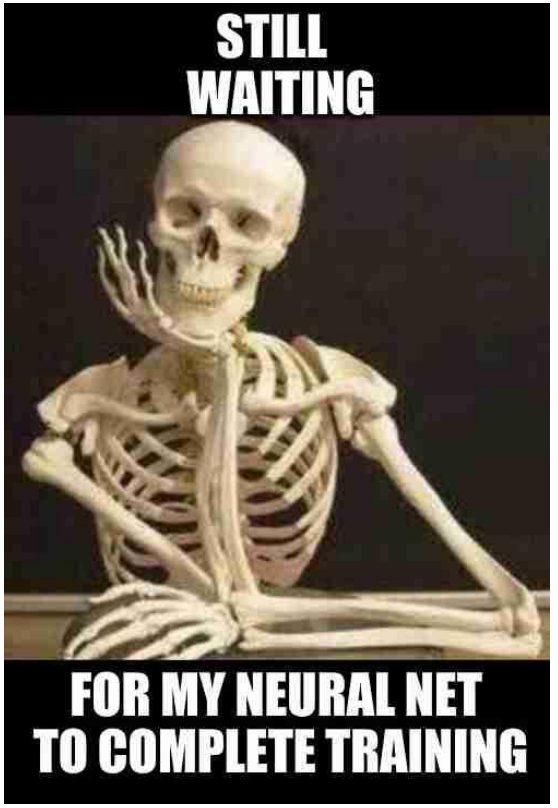
Resumo das Boas Práticas de Experimentos de ML

- Validação Cruzada
  - O que é, que tipo de problema resolve?
- Early stop
  - O que é, que tipo de problema resolve?
- Otimização de Hiperparâmetros
  - O que é, que tipo de problema resolve?
- Pipelines e ColumnTransformers
  - O que é, que tipo de problema resolve?

Se entender a piada é porque já está falando a língua dos nerds

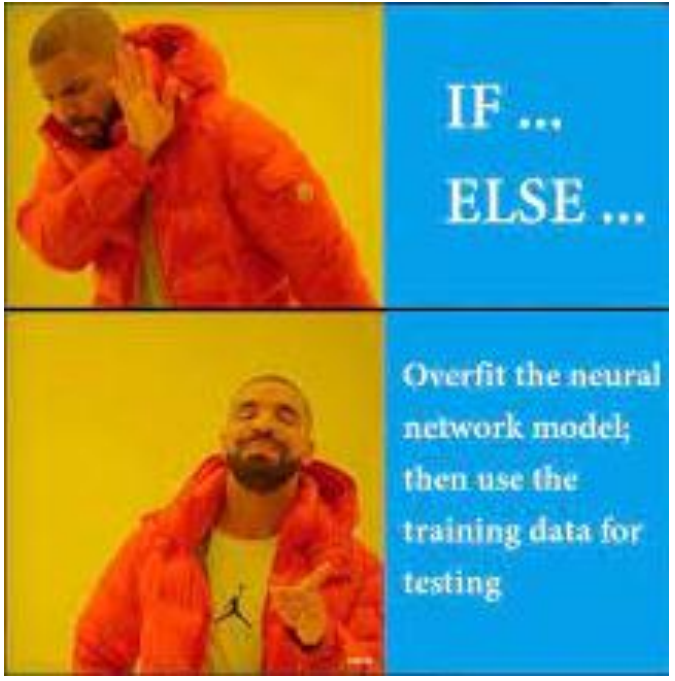


Fonte da Imagem



Fonte da Imagem

Se entender a piada é porque já está falando a língua dos nerds



Fonte da Imagem

Fonte da Imagem