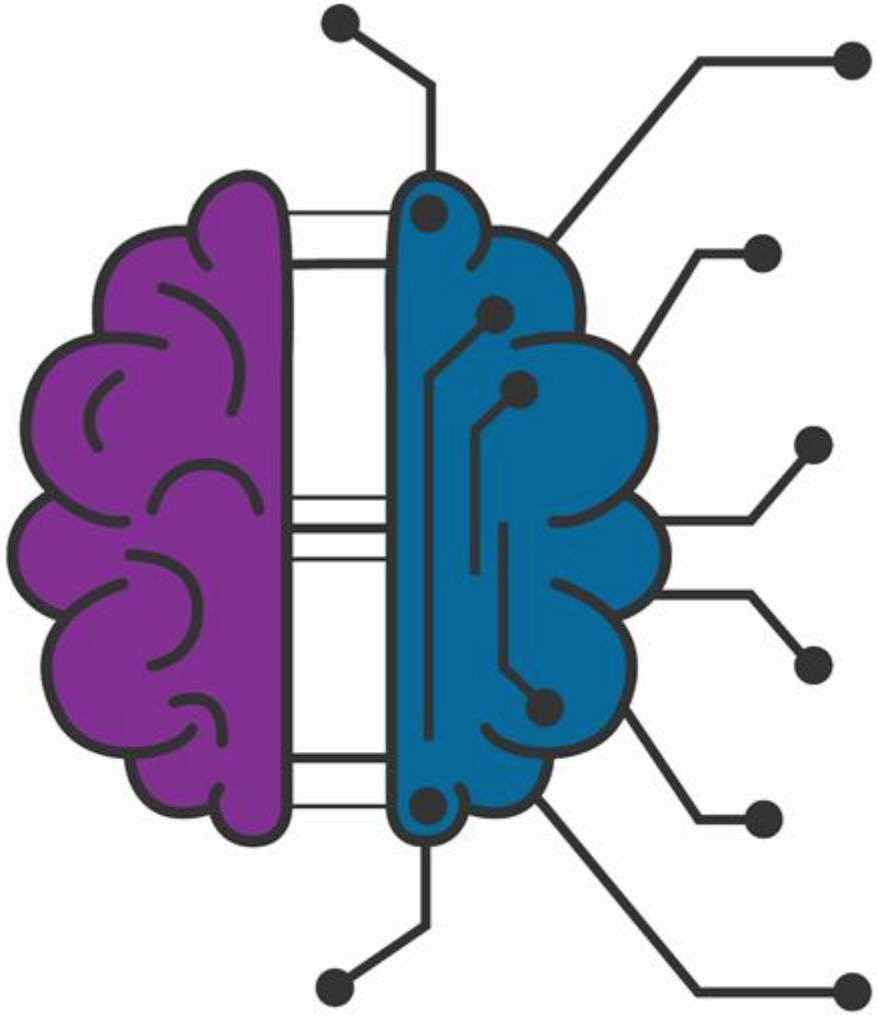
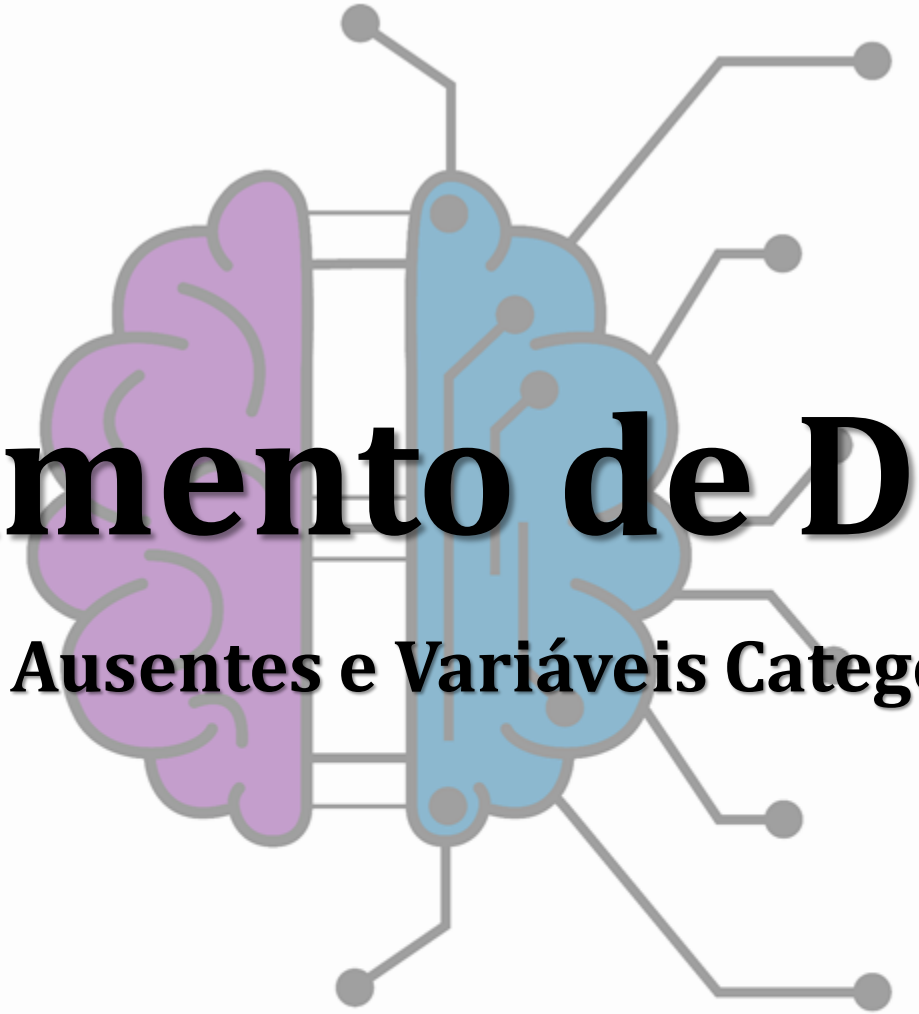


ARTIEEEFICIAIS

IEEE UFBA

WIE UFBA

TECH IEEE



Tratamento de Dados

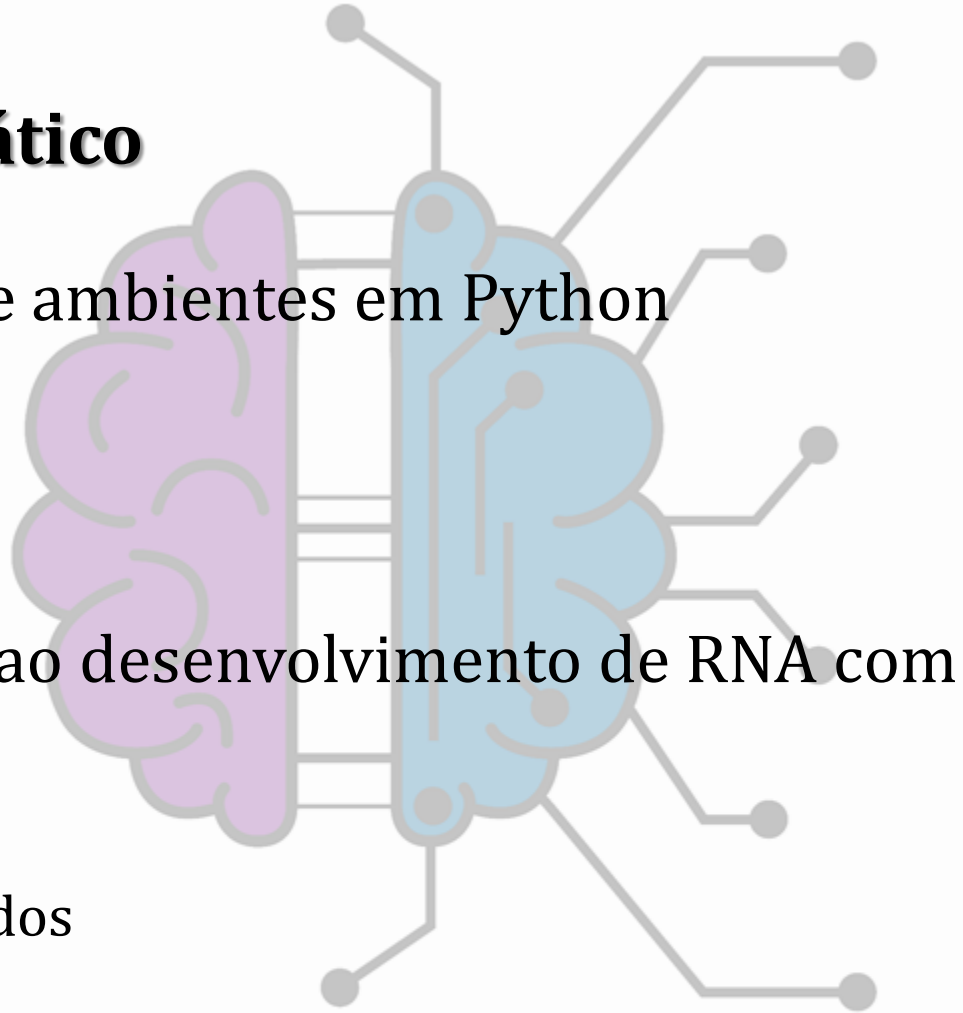
Valores Ausentes e Variáveis Categóricas

Carine Gottschall

Lucas Alves

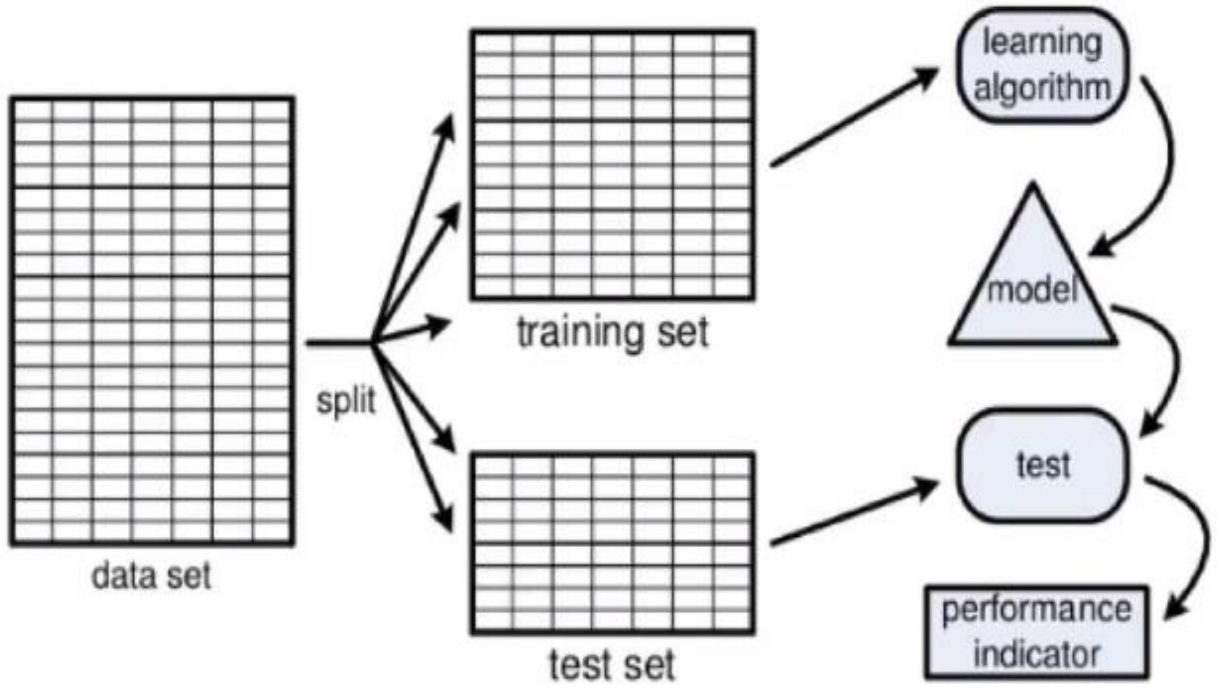
Conteúdo Programático

1. Gestão de pacotes e ambientes em Python
    1. Anaconda
    2. Jupyter Notebook
    3. Google Colab
2. Pacotes essenciais ao desenvolvimento de RNA com Python
    1. Numpy
    2. Pandas
    3. Tratamento de Dados
3. Machine Learning
    1. Regressão Linear
    2. Classificação
    3. Clustering (K-means)



ARTIEEEFICIAIS

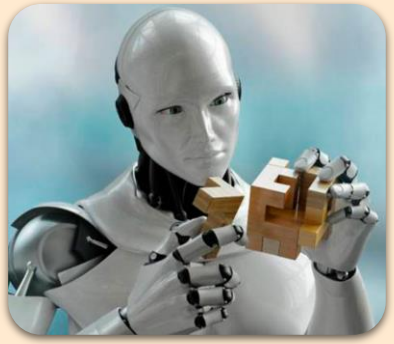
Dataset



Let's let math do this !!

How do we start?

Processo de Aprendizagem



Humano

- Assiste aula, ler livros, pesquisa, estuda, conversa com colegas
- Fazer exercício, trabalhos, provas parciais
- Prova final

Máquina

- Treinamento do Algoritmo
- Validação
- Teste

Dataset

Subset de Dados

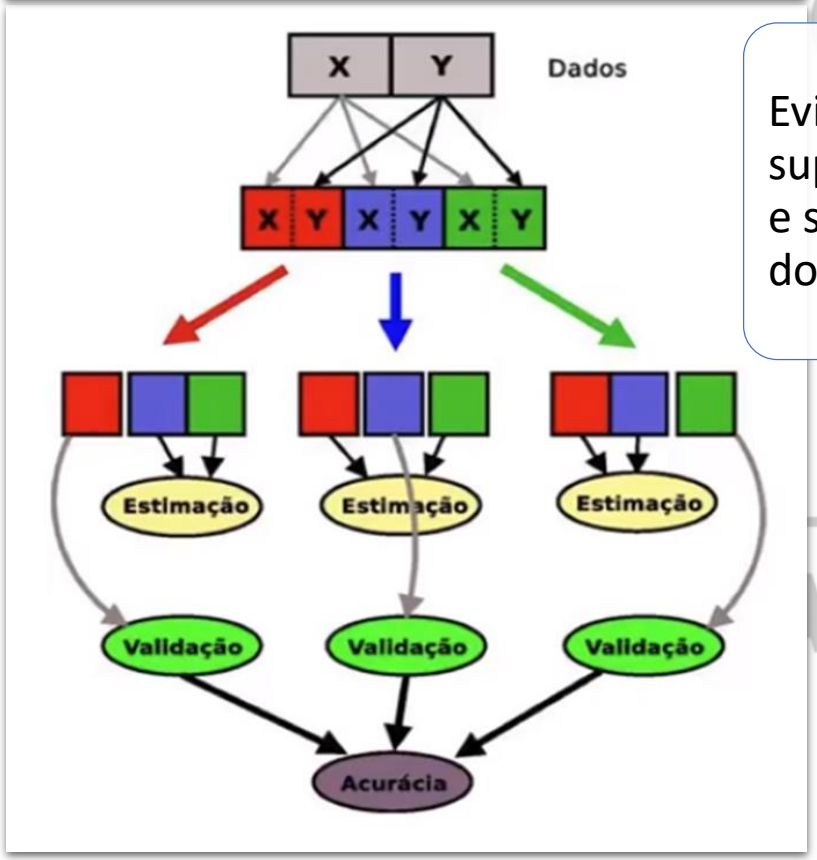
75% a 70% - dados de treino
25% a 30% - dados de teste

ou

75% a 70% - dados de treino
20% - dados de validação
10% - dados de teste

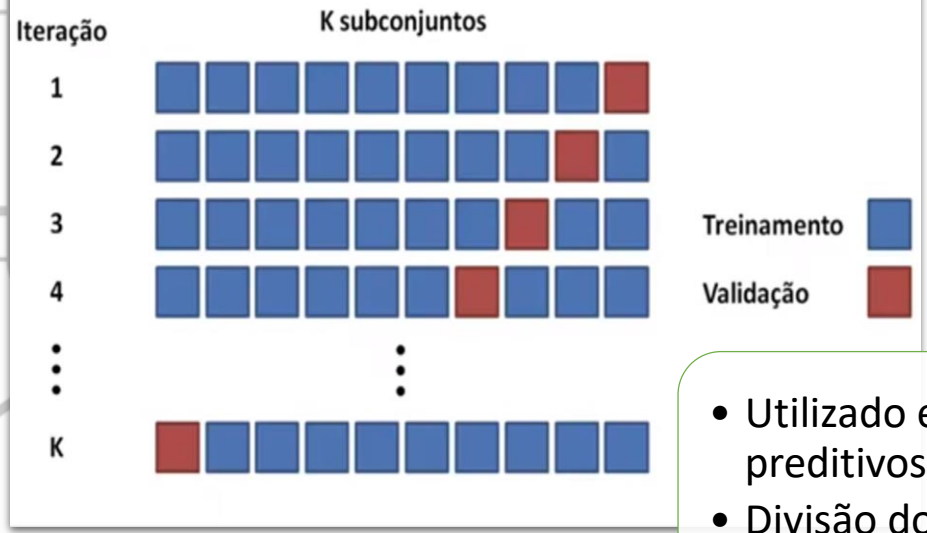
Dataset

Realizar a separação de forma aleatória



Evita a superestimação e subestimação do modelo

Cross Validation



- Utilizado em modelos preditivos
- Divisão do conjunto de dados de validação e treino sendo que para este último são separadas amostras mutuamente exclusivas

Valores ausentes

Existem várias maneiras pelas quais os dados podem acabar com valores ausentes. Por exemplo:

- Uma casa com 2 quartos não incluirá um valor para o tamanho de um terceiro quarto.
- Um respondente da pesquisa pode optar por não compartilhar sua renda.

ARTIEEEFICIAIS

Valores ausentes

- Algoritmos de ML não são capazes de lidar com valores ausentes (*missing data*).
- Um valor ausente é identificado nos campos da sua estrutura de dados como `NaN`.
- Para seu modelo rodar sem problemas, você tem que limpar os dados (*data cleaning*) em uma etapa anterior.

ARTIEEEFICIAIS

Identificando valores ausentes

- A primeira coisa que você tem que saber é a quantidade e proporção dos *missing values*. **E é possível utilizar o Pandas para isto!**

Para identificar valores ausentes, por colunas, você pode usar:

Para retornar um resumo estatístico das variáveis numéricas

- df.describe()

Para dar um resumo de valores não-nulos encontrados

- df.info()

Para retornar a soma dos valores nulos encontrados

- df.isnull().sum()

Excluir?

Completar?

Ignorar?

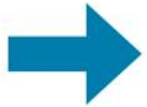
Qual a melhor abordagem?

Saber o que fazer com dados ausentes (*missing data*) vai impactar diretamente na qualidade e desempenho do seu modelo de *Machine Learning*

Tratando valores ausentes

1) Descarte colunas com valores ausentes

Bed	Bath
1.0	1.0
2.0	1.0
3.0	2.0
NaN	2.0



Bath
1.0
1.0
2.0
2.0

A menos que a maioria dos valores nas colunas descartadas esteja ausente, o modelo perde acesso a muitas informações (potencialmente úteis!) com essa abordagem. Como um exemplo extremo, considere um conjunto de dados com 10.000 linhas, em que uma coluna importante está faltando uma única entrada. Essa abordagem eliminaria a coluna completamente!

Tratando valores ausentes

2) Imputação

A imputação preenche os valores ausentes com algum número. Por exemplo, podemos preencher o valor médio ao longo de cada coluna.

Bed	Bath
1.0	1.0
2.0	1.0
3.0	2.0
NaN	2.0

Bed	Bath
1.0	1.0
2.0	1.0
3.0	2.0
2.0	2.0

O valor imputado não será exatamente correto na maioria dos casos, mas geralmente leva a modelos mais precisos do que você obteria ao largar completamente a coluna.

Tratando valores ausentes

3) Uma extensão à imputação

Os valores imputados podem estar sistematicamente acima ou abaixo dos valores reais (que não foram coletados no conjunto de dados). Ou linhas com valores ausentes podem ser exclusivas de alguma outra maneira. Nesse caso, seu modelo faria melhores previsões considerando quais valores estavam originalmente ausentes.

Bed	Bath
1.0	1.0
2.0	1.0
3.0	2.0
NaN	2.0

Bed	Bath	Bed_was_missing
1.0	1.0	FALSE
2.0	1.0	FALSE
3.0	2.0	FALSE
2.0	2.0	TRUE

Nesta abordagem, imputamos os valores ausentes, como antes. Além disso, para cada coluna com entradas ausentes no conjunto de dados original, **adicionamos uma nova coluna que mostra a localização das entradas imputadas.** Em alguns casos, isso melhorará significativamente os resultados. Em outros casos, isso não ajuda em nada.

Hora da prática

ARTIEEEFICIAIS

Variáveis Categóricas

Considere uma pesquisa que pergunta com que frequência você toma café da manhã e oferece quatro opções: "Nunca", "Raramente", "Na maioria dos dias" ou "Todos os dias". Nesse caso, os dados são categóricos, porque as respostas caem em um conjunto fixo de categorias. Se as pessoas respondessem a uma pesquisa sobre qual marca de carro possuíam, as respostas cairiam em categorias como "Honda", "Toyota" e "Ford". Nesse caso, os dados também são categóricos.

- Você receberá um erro se tentar conectar essas variáveis à maioria dos modelos de aprendizado de máquina no Python sem pré-processá-las. primeiro.

Tratando variáveis categóricas

1) Descarte de variáveis categóricas

A abordagem mais fácil para lidar com variáveis categóricas é simplesmente removê-las do conjunto de dados. Essa abordagem só funcionará bem se as colunas não contiverem informações úteis.

ARTIEEEFICIAIS

Tratando variáveis categóricas

2) Codificação de etiqueta

A codificação de etiqueta atribui cada valor exclusivo a um número inteiro diferente.

Breakfast
Every day
Never
Rarely
Most days
Never

Breakfast
3
0
1
2
0

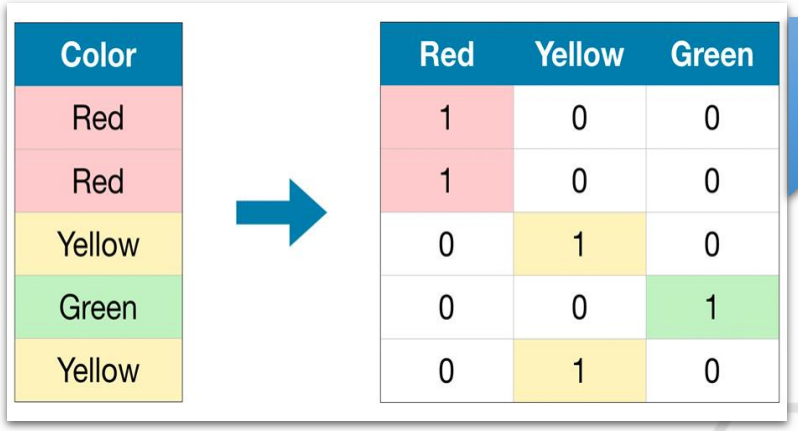
Essa abordagem assume uma ordem das categorias: "Nunca" (0) < "Raramente" (1) < "Na maioria dos dias" (2) < "Todos os dias" (3).

Essa suposição faz sentido neste exemplo, porque há uma classificação indiscutível nas categorias. Nem todas as variáveis categóricas têm uma ordem clara nos valores, mas nos referimos àquelas que funcionam como variáveis ordinais. Para modelos baseados em árvore (como árvores de decisão e florestas aleatórias), você pode esperar que a codificação de rótulo funcione bem com variáveis ordinais.

Tratando variáveis categóricas

3) Codificação One-Hot

A codificação one-hot cria novas colunas indicando a presença (ou ausência) de cada valor possível nos dados originais.



Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

No conjunto de dados original, "Cor" é uma variável categórica com três categorias: "Vermelho", "Amarelo" e "Verde". A codificação one-hot correspondente contém uma coluna para cada valor possível e uma linha para cada linha no conjunto de dados original. Onde quer que o valor original fosse "Vermelho", colocamos 1 na coluna "Vermelho"; se o valor original era "Amarelo", colocamos 1 na coluna "Amarelo" e assim por diante.

Tratando variáveis categóricas

3) Codificação One-Hot

Color
Red
Red
Yellow
Green
Yellow

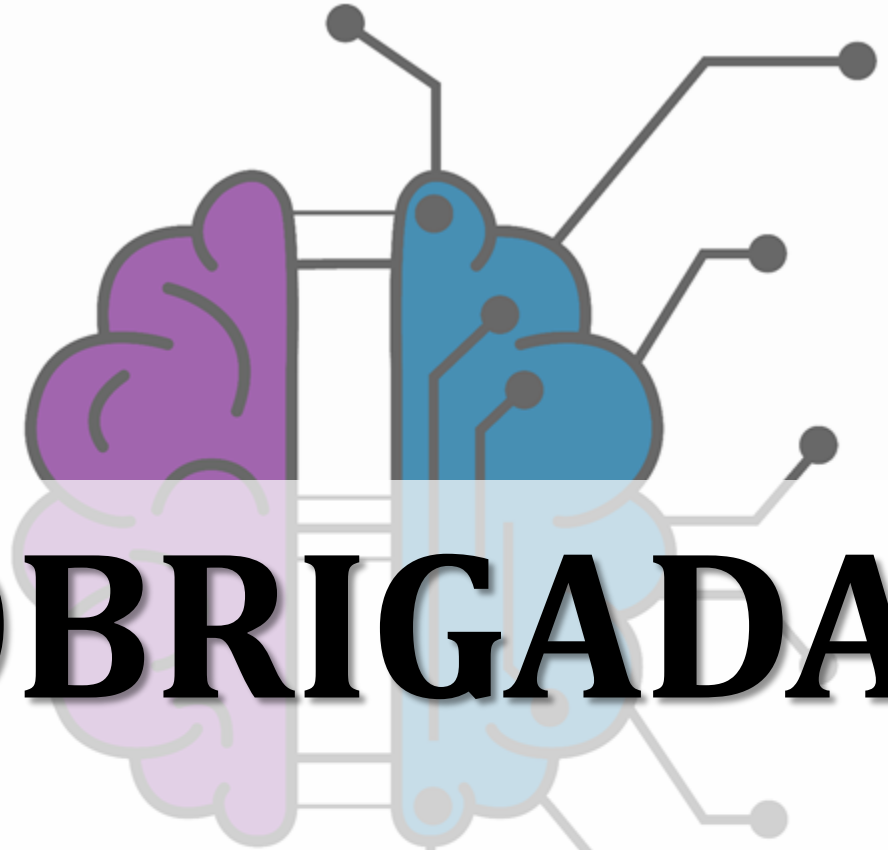
Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

Ao contrário da codificação de etiqueta, a codificação one-hot não assume uma ordem das categorias. Portanto, você pode esperar que essa abordagem funcione particularmente bem se não houver uma ordem clara nos dados categóricos (por exemplo, "Vermelho" não é nem mais nem menos que "Amarelo"). Nos referimos às variáveis categóricas sem uma classificação intrínseca como variáveis nominais.

A codificação one-hot geralmente não funciona bem se a variável categórica assume um grande número de valores (ou seja, você geralmente não a usa para variáveis que levam mais de 15 valores diferentes).

Hora da prática

ARTIEEEFICIAIS



OBRIGADA!

Repositório GitHub:
https://github.com/Skyzenho/ArtIEEEficiais

ARTIEEEFICIAIS