# Codificação

April 2, 2025

## 1 Codificação de Rótulos em Python

Em projetos de aprendizado de máquina, geralmente lidamos com conjuntos de dados com diferentes colunas categóricas onde algumas colunas têm seus apresentam variáveis categóricas ordinais.

Por exemplo: uma coluna com a variável renda que apresenta valores baixo, médio e alto. Esses valores podem ser substituidos pelos números 1, 2 e 3, onde: - 1 representa o valor baixo; - 2 representa o valor médio; - 3 representa o valor alto.

Por meio desse tipo de codificação, tentamos preservar o significado do elemento onde pesos maiores são atribuídos aos elementos com maior prioridade.

#### 1.0.1 Codificação de rótulo

É uma técnica usada para converter colunas categóricas em numéricas para que elas possam ser ajustadas por modelos de aprendizado de máquina, que trabalham unicamente com valores numéricos.

É uma etapa importante de pré-processamento em um projeto de aprendizado de máquina.

### 1.1 O que é Codificação de Rótulos?

Imagine que você tem uma lista de frutas: maçã, banana e laranja. Um computador não entende palavras, ele só entende números. A codificação de rótulos é como traduzir essas palavras para números que o computador possa entender.

### 1.2 Por que usar Codificação de Rótulos?

Muitos algoritmos de machine learning (aprendizado de máquina) funcionam melhor com números do que com palavras. Ao transformar categorias (como "alto", "médio" e "baixo") em números, podemos usar esses dados para treinar modelos de machine learning.

#### 1.2.1 Exemplo Prático:

Vamos usar o exemplo da coluna "Altura" com os valores "Alto", "Médio" e "Baixo". Para codificar essa coluna:

Criamos um dicionário:

- Alto = 0
- Médio = 1
- Baixo = 2

Substituímos as palavras pelos números:

- Onde antes tínhamos "Alto", agora temos 0.
- Onde antes tínhamos "Médio", agora temos 1.
- Onde antes tínhamos "Baixo", agora temos 2.

### 1.2.2 Por que usar 0, 1 e 2?

A escolha dos números é arbitrária. Poderia ser 10, 20 e 30, ou qualquer outro conjunto de números. O importante é que cada categoria tenha um número único.

Tabela Comparativa:

| Altura (texto) | Altura (número) |
|---|---|
| Alto | 0 |
| Médio | 1 |
| Baixo | 2 |

A codificação de rótulos é uma técnica simples e poderosa para transformar dados categóricos em numéricos, permitindo que modelos de machine learning processem e aprendam com esses dados.

## 1.3 Exemplo de codificação de rótulo

Aplicaremos codificação de rótulos no conjunto de dados da íris na coluna de destino que é Species.

Ela contém três espécies:

- Setosa;
- Versicolor;
- Virginica.

[4]:
```
# bibliotecas
import numpy as np
import pandas as pd

# base de dados
df = pd.read_csv('iris.csv')

df['species'].unique()
```

[4]:
```
array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)
```

Após aplicar a codificação de rótulos da classe LabelEncoder(), nosso valor categórico será substituído pelo valor numérico.

[5]:
```
# Importando a biblioteca de codificação
from sklearn import preprocessing

# o objeto label_encoder conhece
# como copreender os rótulos das palavras
```

```
label_encoder = preprocessing.LabelEncoder()

# Codificando rótulos na coluna 'espécies'
df['species']= label_encoder.fit_transform(df['species'])

df['species'].unique()
```

[5]: array([0, 1, 2])

### 1.4 O Problema da Codificação de Rótulos

A codificação de rótulos é uma técnica útil para transformar dados textuais (como "México", "Paris", "Dubai") em números que os computadores podem entender. No entanto, essa técnica pode levar a um problema: o modelo pode interpretar a ordem dos números como uma ordem de importância.

### 1.5 Por que isso é um problema?

Atribuição arbitrária: Os números atribuídos às categorias são arbitrários. Por exemplo, "México" recebeu 0, "Paris" recebeu 1 e "Dubai" recebeu 2.

Isso **não** significa que "Dubai" seja mais importante que as outras cidades.

- **Interpretação incorreta**: O modelo de aprendizado de máquina pode interpretar erroneamente que, como 2 é maior que 1 e 0, "Dubai" tem uma maior influência ou peso na previsão.
- **Viés nos resultados**: Essa interpretação incorreta pode levar a resultados enviesados e imprecisos.

#### 1.5.1 Exemplo:

Imagine que estamos tentando prever o destino de férias de uma pessoa. Se usarmos a codificação de rótulos como descrito acima, o modelo pode ser mais propenso a prever "Dubai" como destino, simplesmente porque o número 2 (associado a "Dubai") é maior que os outros. No entanto, na realidade, não há nenhuma razão para que "Dubai" seja mais provável de ser escolhido do que as outras cidades.

A codificação de rótulos é uma ferramenta útil, mas é importante lembrar que os números atribuídos às categorias não carregam nenhum significado intrínseco além de serem rótulos únicos. Para evitar problemas de interpretação e viés, é fundamental usar essa técnica com cuidado e considerar outras técnicas de codificação, como o one-hot encoding, que não introduzem uma ordem artificial entre as categorias.

#### 1.5.2 Outras técnicas de codificação:

**One-hot encoding**: Cria uma nova coluna para cada categoria e atribui 1 para a categoria correspondente e 0 para as demais. Essa técnica preserva a natureza categórica dos dados sem introduzir uma ordem artificial.

### 1.5.3 Em quais situações a codificação de rótulos pode ser problemática?

**Quando a ordem das categorias é importante**: Se a ordem das categorias tiver algum significado (por exemplo, níveis de escolaridade), a codificação de rótulos pode distorcer essa informação.

**Quando há muitas categorias**: Com um grande número de categorias, a codificação de rótulos pode levar a um aumento significativo na dimensionalidade dos dados, o que pode afetar o desempenho do modelo. Conclusão:

A escolha da técnica de codificação adequada depende do tipo de dados, do problema a ser resolvido e do modelo de aprendizado de máquina que será utilizado. É importante avaliar as vantagens e desvantagens de cada técnica antes de tomar uma decisão.

# 2 O que é One-Hot Encoding e por que usamos?

Imagine que você está construindo um modelo de computador para prever se uma pessoa vai gostar de um determinado filme. Você tem informações sobre essa pessoa, como idade, cidade, gênero, etc. Mas como o computador entende palavras como "Masculino" e "Feminino"? É aí que entra o One-Hot Encoding.

## 2.1 Por que não usar números diretamente?

Se simplesmente substituirmos "Masculino" por 0 e "Feminino" por 1, o modelo pode pensar que ser "Feminino" é melhor do que ser "Masculino", já que 1 é maior que 0. Isso não faz sentido, pois ambos os gêneros são igualmente importantes.

## 2.2 Como funciona o One-Hot Encoding?

O One-Hot Encoding cria novas colunas para cada categoria da variável. Por exemplo, para a variável "Gênero", teríamos duas novas colunas: "É Masculino" e "É Feminino". Para cada pessoa, apenas uma dessas colunas terá o valor 1 (indicando que a pessoa pertence a aquela categoria), e as outras terão o valor 0.

Exemplo:

| Gênero | é Masculino | é Feminino |
|---|---|---|
| Masculino | 1 | 0 |
| Femino | 0 | 1 |

### 2.2.1 Vantagens:

1. **Permite usar dados categóricos em modelos**: Muitos modelos de aprendizado de máquina só funcionam com números.
2. **Melhora o desempenho**: Ao fornecer mais informações ao modelo, ele pode fazer previsões mais precisas.
3. **Evita problemas de ordem**: Se tivermos categorias como "pequeno", "médio" e "grande", o modelo não vai pensar que "médio" é automaticamente melhor que "pequeno".

### 2.2.2 Desvantagens:

1. **Aumenta a quantidade de dados**: Criar novas colunas pode tornar o modelo mais complexo e lento.
2. **Cria dados esparsos**: A maioria dos valores será 0, o que pode dificultar o aprendizado do modelo.
3. **Pode causar overfitting(sobre treinamento)**: Se tivermos muitas categorias e poucos dados, o modelo pode se ajustar demais aos dados de treinamento e não generalizar bem para novos dados.

## 2.3 Quando usar One-Hot Encoding?

O One-Hot Encoding é útil quando as categorias **não têm uma ordem natura**l (como "Masculino" e "Feminino") e quando queremos que o modelo trate todas as categorias como igualmente importantes.

Existem outras técnicas para lidar com dados categóricos, como a codificação ordinal (quando as categorias têm uma ordem natural) e a codificação binária. A escolha da técnica ideal depende do problema específico e dos dados disponíveis.

Um exemplo de One-Hot Encoding

Imagine uma tabela com informações de preços de frutas e seus valores categóricos, conforme segue abaixo:

| Fruta | Valor categórico da fruta | Preço |
|---|---|---|
| maçã | 1 | 5 |
| manga | 2 | 10 |
| maçã | 1 | 15 |
| laranja | 3 | 20 |

Após aplicar o One-Hot Encoding:

| maçã | manga | laranja | preço |
|---|---|---|---|
| 1 | 0 | 0 | 5 |
| 0 | 1 | 0 | 10 |
| 1 | 0 | 0 | 15 |
| 0 | 0 | 1 | 20 |

[6]:
```
data = {'Employee id': [10, 20, 15, 25, 30, 45, 78, 56, 12, 7, 8, 57, 14, 27,␣
↪35],
        'Gender': ['M', 'F', 'F', 'M', 'F','M', 'F', 'F', 'M', 'F','M', 'F',␣
↪'F', 'M', 'F'],
        'Remarks': ['Good', 'Nice', 'Good', 'Great', 'Nice','Good', 'Nice',␣
↪'Good', 'Great', 'Nice','Good', 'Nice', 'Good', 'Great', 'Nice'],
       }
```

[7]:
```
df = pd.DataFrame(data)
```

[8]:
```
df
```

[8]:

| | Employee id | Gender | Remarks |
|---|---|---|---|
| 0 | 10 | M | Good |
| 1 | 20 | F | Nice |
| 2 | 15 | F | Good |
| 3 | 25 | M | Great |
| 4 | 30 | F | Nice |
| 5 | 45 | M | Good |
| 6 | 78 | F | Nice |
| 7 | 56 | F | Good |
| 8 | 12 | M | Great |
| 9 | 7 | F | Nice |
| 10 | 8 | M | Good |
| 11 | 57 | F | Nice |
| 12 | 14 | F | Good |
| 13 | 27 | M | Great |
| 14 | 35 | F | Nice |

### 2.3.1 Valores únicos nas Colunas Categóricas

Podemos usar a função unique() da biblioteca pandas para obter os valores único da coluna da base de dados.

[9]:
```
print(df['Gender'].unique())
print(df['Remarks'].unique())
```

```
['M' 'F']
['Good' 'Nice' 'Great']
```

[10]:
```
df['Remarks'].value_counts()
```

[10]:
```
Good     6
Nice     6
Great    3
Name: Remarks, dtype: int64
```

[11]:
```
df['Gender'].value_counts()
```

[11]:
```
F    9
M    6
Name: Gender, dtype: int64
```

Podemos usar a função pd.get_dummies() do pandas para codificar as colunas categóricas.

[12]:
```
one_hot_encoded_data = pd.get_dummies(df, columns = ['Remarks', 'Gender'])
print(one_hot_encoded_data)
```

| | Employee id | Remarks_Good | Remarks_Great | Remarks_Nice | Gender_F | Gender_M |
|---|---|---|---|---|---|---|
| 0 | 10 | 1 | 0 | 0 | 0 | 1 |
| 1 | 20 | 0 | 0 | 1 | 1 | 0 |
| 2 | 15 | 1 | 0 | 0 | 1 | 0 |
| 3 | 25 | 0 | 1 | 0 | 0 | 1 |
| 4 | 30 | 0 | 0 | 1 | 1 | 0 |
| 5 | 45 | 1 | 0 | 0 | 0 | 1 |
| 6 | 78 | 0 | 0 | 1 | 1 | 0 |
| 7 | 56 | 1 | 0 | 0 | 1 | 0 |
| 8 | 12 | 0 | 1 | 0 | 0 | 1 |
| 9 | 7 | 0 | 0 | 1 | 1 | 0 |
| 10 | 8 | 1 | 0 | 0 | 0 | 1 |
| 11 | 57 | 0 | 0 | 1 | 1 | 0 |
| 12 | 14 | 1 | 0 | 0 | 1 | 0 |
| 13 | 27 | 0 | 1 | 0 | 0 | 1 |
| 14 | 35 | 0 | 0 | 1 | 1 | 0 |

Podemos observar que temos 3 colunas Remarks e 2 Gender nos dados.

No entanto, você pode usar apenas colunas n-1 para definir parâmetros se ele tiver n rótulos exclusivos. Por exemplo, se mantivermos apenas a coluna Gender_Female e removermos a coluna Gender_Male , também podemos transmitir todas as informações, pois quando o rótulo é 1, significa feminino e quando o rótulo é 0, significa masculino. Dessa forma, podemos codificar os dados categóricos e reduzir o número de parâmetros também.

Uma codificação quente usando a biblioteca Sci-kit Learn Scikit-learn(sklearn) é uma biblioteca popular de machine learning em Python que fornece inúmeras ferramentas para pré-processamento de dados. Ela fornece uma função OneHotEncoder que usamos para codificar variáveis categóricas e numéricas em vetores binários.