



VER PLANOS

PROGRAMAÇÃO \_

DATA SCIENCE \_

DEVOPS \_

MOBILE \_

FRONT-END \_

INTELIGÊNCIA ARTIFICIAL \_

UX & DESIGN \_

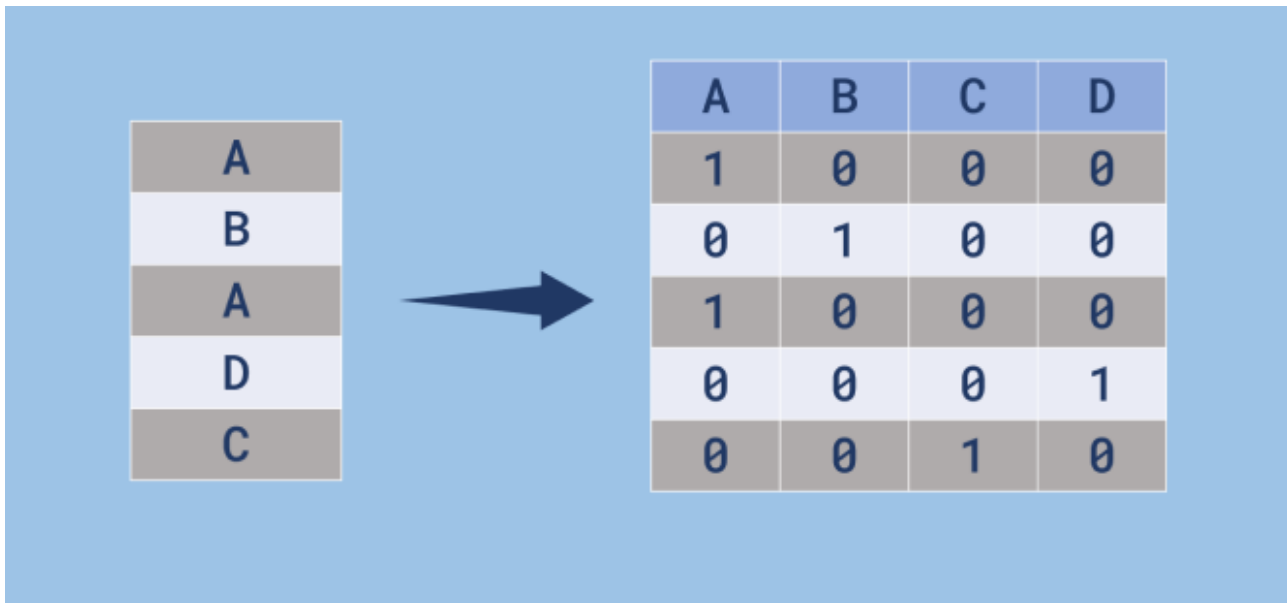
INOVAÇÃO & GESTÃO \_

Artigos > **Data Science**

Oi! Posso indicar os melhores artigos para tirar suas dúvidas!



## método escolher?



João Vitor de Miranda

15/08/2022

COMPARTILHE



Esse artigo faz parte da

**Formação Machine Learning na prática: fundamentos e aplicações**

## Introdução

No momento de realizar o treinamento de modelos de machine learning, nos deparamos com um grande problema ao lidar com variáveis categóricas. As categorias não podem ser utilizadas em forma de texto, uma vez que os algoritmos compreendem apenas valores numéricos. Também não podemos simplesmente atribuir um valor numérico para cada categoria, uma vez que esse processo, conhecido como **ponderação arbitrária**, pode criar uma ordenação e pesos para as categorias que não refletem a realidade.

O processo correto de transformação das variáveis categóricas é feito a partir da criação de novas colunas a partir das categorias. Cada uma delas se torna

**Confira neste artigo:**

- [Introdução](#)
- [get\\_dummies\(\)](#)
- [OneHotEncoder\(\)](#)

Vamos considerar a tabela contendo uma variável com 3 características:

Variável
Característica 1
Característica 2
Característica 3

Ao aplicar o procedimento de "one-hot" nessa tabela, teremos o seguinte resultado:

Característica 1	Característica 2	Característica 3
1	0	0
0	1	0
0	0	1

Para realizar esse processo através da [linguagem Python](#), podemos usar o método [get\\_dummies\(\)](#) da biblioteca [Pandas](#) ou o método [OneHotEncoder\(\)](#) da biblioteca [Sklearn](#). Vamos entender a diferença entre as duas funções.

[VER PLANOS](#)

3 aulas gratuitas para você criar um sistema de agentes que pensam por você.

**GARANTA A SUA VAGA**

## get\_dummies()

O método `get_dummies()` é simples de utilizar e faz a transformação de forma direta das variáveis categóricas.

```
import pandas as pd
dados = pd.DataFrame({'variavel': ['caracteristica 1', 'caracteristica 2', 'caracteristica 3']})
pd.get_dummies(dados, columns = ['variavel'])
```

	variavel_caracteristica 1	variavel_caracteristica 2	variavel_caracteristica 3
0	1	0	0
1	0	1	0
2	0	0	1

Esses dados podem ser passados como dados de entrada de um modelo de machine learning para realizar uma previsão. Porém, imagine que precisamos aplicar o processo para dados novos, contendo uma característica 2, já conhecida, e uma característica 4, que não foi utilizada na construção do modelo. O método `get_dummies()` não conseguirá gerar todas as colunas necessárias para a previsão. Ele vai considerar como válida uma característica não vista anteriormente e que não será entendida pelo modelo treinado.

[VER PLANOS](#)

	variavel_caracteristica 2	variavel_caracteristica 4
0	1	0
1	0	1

## OneHotEncoder()

Já o método `OneHotEncoder()` funciona como outros modelos disponíveis na biblioteca Sklearn. Nele, é necessário instanciar um objeto e depois ajustar aos dados com um método `fit()`. Dessa forma, esse objeto armazena os passos necessários para realizar a transformação dos dados.

```
from sklearn.compose import make_column_transformer
from sklearn.preprocessing import OneHotEncoder
colunas_categoricas = ['variavel']
one_hot_enc = make_column_transformer(
    (OneHotEncoder(handle_unknown = 'ignore'),
     colunas_categoricas),
    remainder='passthrough')
dados = one_hot_enc.fit_transform(dados)
dados = pd.DataFrame(dados, columns=one_hot_enc.get_feature_names_out(dados))
```

	onehotencoder_variavel_caracteristica 1	onehotencoder_variavel_caracteristica 2
0	1	0

[VER PLANOS](#)

1	0
2	0

Ao aplicar em novos dados, o método `OneHotEncoder()` irá construir todas as colunas que foram geradas no treinamento atribuindo valor 0 ou 1, dependendo da presença ou ausência da característica, respectivamente. Logo, esse método é ideal para utilizar em modelos de machine learning.

```
dados_novos = pd.DataFrame({'variavel':['caracteristica 2', 'carac  
dados_novos = one_hot_enc.transform(dados_novos)  
dados_novos = pd.DataFrame(dados_novos, columns=one_hot_enc.get_fe  
dados_novos
```

	onehotencoder_variavel_caracteristica 1	onehotencoder_variavel_caract
0	0	
1	0	

Assim, a característica 4 contida nos novos dados não se torna uma nova coluna, apenas são atribuídos os valores 0 para cada uma das características utilizadas no momento do treinamento do modelo. Além disso, o processamento que foi criado pelo `OneHotEncoder()` pode ser armazenado em arquivos pickle, assim como outros modelos de machine learning, para ser utilizado fora do ambiente onde foi criado, permitindo o uso em outros projetos.

[VER PLANOS](#)

Para importar o modelo em outros projetos, basta usar a função

`pd.read_pickle()` :

```
modelo_one_hot = pd.read_pickle('modelo_onehotenc.pkl')
```

Desse modo, podemos observar que o `get_dummies()` é fácil de utilizar, mas é preferível para atividades de análises de dados, enquanto o `OneHotEncoder()` é mais recomendado para aplicar em modelos de machine learning.

imersão **dev\_**  
AGENTES DE IA

## Saia dos prompts básicos e **CRIE AGENTES DE IA INTELIGENTES**

- 3 aulas online na plataforma Alura
- Certificado Alura para o seu currículo
- Comunidade exclusiva no Discord

**APRENDA GRATUITAMENTE**

alura | com Google Gemini

[VER PLANOS](#)

### João Vitor de Miranda

Bacharel em Matemática e pós graduado em Data Science e Analytics. Com conhecimento em Matemática, Estatística, Excel, Python, R e SQL/NoSQL.

## Leia também



**Executores do Airflow: tipos e funções**

_bill	tip	smoker	day	time
3.99	1.01	No	Sun	Dinn
0.34	1.66	No	Sun	Dinn
1.01	3.50	No	Sun	Dinn
3.68	3.31	No	Sun	Dinn
4.59	3.61	No	Sun	Dinn

**Como remover linhas e colunas no Pandas**



**Polí  
Pos  
reci  
de c**



Veja outros artigos sobre  
[Data Science](#)



[VER PLANOS](#)

## Quer mergulhar em tecnologia e aprendizagem?

Receba conteúdos, dicas, notícias, inovações e tendências sobre o mercado tech diretamente na sua caixa de entrada.

Email\*

**ENVIAR**

### Nossas redes e apps



#### Institucional

[Sobre nós](#)[Carreiras Alura](#)[Para Empresas](#)[Para Sua Escola](#)[Política de Privacidade](#)

#### A Alura

[Como Funciona](#)[Formações](#)[Plataforma](#)[Depoimentos](#)[Instrutores\(as\)](#)

[VER PLANOS](#)[Documentos Institucionais](#)[IA Conference 2025](#)[Status](#)[Cursos imersivos](#)[Certificações](#)[Uma empresa do grupo Alun](#)

## Conteúdos

## Fale Conosco

[Alura Cases](#)[Email e telefone](#)[Imersões](#)[Perguntas frequentes](#)[Artigos](#)[Podcasts](#)[Artigos de educação corporativa](#)[Imersão Dev Agentes de IA Google](#)

## Novidades e Lançamentos

[ENVIAR](#)

## CURSOS

### Cursos de Programação

[Lógica](#) | [Python](#) | [PHP](#) | [Java](#) | [.NET](#) | [Node JS](#) | [C](#) | [Computação](#) | [Jogos](#) | [IoT](#)

### Cursos de Front-end

[VER PLANOS](#)

### **Cursos de Inteligência Artificial**

IA para Programação | IA para Dados

### **Cursos de DevOps**

AWS | Azure | Docker | Segurança | IaC | Linux

### **Cursos de UX & Design**

Usabilidade e UX | Vídeo e Motion | 3D

### **Cursos de Mobile**

Flutter | iOS e Swift | Android, Kotlin | Jogos

### **Cursos de Inovação & Gestão**

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas

## **CURSOS UNIVERSITÁRIOS FIAP**

Graduação | Pós-graduação | MBA