

Trabalho Prático 1 - Ambientação

- **Nome:** Matheus Freitas Martins
- **Matrícula:** ES111281

O propósito deste trabalho é aprofundar o conhecimento em plataformas relevantes para análise e processamento de dados. Para alcançar esse objetivo, será configurada uma instância de máquina virtual na Azure, na qual serão instaladas as dependências requeridas para participar de uma competição de aprendizado de máquina no Kaggle.

Nesta competição, o desafio é desenvolver um modelo capaz de prever se um passageiro da Spaceship Titanic foi transportado para uma dimensão alternativa ou não, usando informações pessoais do passageiro. O resultado esperado é a criação de um modelo que possa classificar corretamente os passageiros em duas categorias: aqueles que foram transportados para outra dimensão e aqueles que não foram.

Configurando o Ambiente

Criando a máquina virtual

Conforme sugerido pela documentação, foi utilizado a plataforma Azure para criar a máquina virtual <https://portal.azure.com/#home>

Caminho: *Virtuais > Criar > Máquina Virtual do Azure*

Inicialmente, as seguintes configurações foram escolhidas para a máquina:

- Nome: **CCF726**
- Região: **(US) East US**
- Imagem (S.O): **Ubuntu Server 20.04 LTS - X64 Gen2**

Criar uma máquina virtual

Detalhes da instância

Nome da máquina virtual * ⓘ
CCF726 ✓

Região * ⓘ
(US) East US ▼

Opções de disponibilidade ⓘ
Nenhuma redundância infraestrutura necessária ▼

Tipo de segurança ⓘ
Computadores virtuais de inicialização confiável ▼
[Configurar os recursos de segurança](#)

Imagem * ⓘ
Ubuntu Server 20.04 LTS - x64 Gen2 ▼
[Ver todas as imagens](#) | [Configurar a geração de VM](#)

Arquitetura de VM ⓘ
☐ Arm64
☒ x64

Executar com desconto de Spot do Azure ⓘ
☐

Tamanho * ⓘ
Standard_D2s_v3 - 2 vcpus, 8 GiB memória (US\$ 70,08/mês) ▼
[Ver todos os tamanhos](#)

Em relação a conta de administrador, foi escolhido autenticar-se por meio de uma senha.

- **Login:** mtsftsmts

Conta de administrador

Tipo de Autenticação ⓘ

- ☐ Chave pública de SSH
- ☒ Senha

Nome de usuário * ⓘ

mtsftsmts ✓

Senha * ⓘ

..... ✓

Confirmar senha * ⓘ

..... ✓

Regras de portas de entrada

Selecione quais portas de rede da máquina virtual podem ser acessadas pela internet pública. Você pode especificar um acesso à rede mais limitado ou granular na guia Rede.

Portas de entrada públicas * ⓘ

- ☐ Nenhum
- ☒ Permitir portas selecionadas

Selecione as portas de entrada *

SSH (22) ✓



Isso permitirá que todos os endereços IP acessem sua máquina virtual.
Isso é recomendado somente para testes. Use os controles Avançados na guia Rede para criar regras para limitar o tráfego de entrada a endereços IP conhecidos.

Uma vez definido essas pequenas configurações, prosseguiu-se clicando no botão "**Revisar + Criar**".

Incluindo regra de inbound

Além das configurações básicas definidas anteriormente, é necessário incluir uma regra de inbound que vai habilitar acessar o Jupyter Notebook. Para isso, foi escolhido a porta **8888** e protocolo **TCP**.

Caminho: *Configurações > Rede > Adicionar regra da porta de entrada*



Origem ⓘ

Any

Intervalos de porta de origem * ⓘ

*

Destino ⓘ

Any

Serviço ⓘ

Custom

Intervalos de porta de destino * ⓘ

8888

Protocolo

☐ Any

☒ TCP

☐ UDP

☐ ICMP

Ação

☒ Permitir

☐ Negar

Prioridade * ⓘ

310

Nome

AllowAnyCustom8888Inbound

Descrição

Salvar

Cancelar



Enviar comentários

Configurando IP estático

Além disso, para facilitar as futuras conexões foi atribuído a rede um ip estático.

Caminho: Configurações > Configurações de IP

[Página inicial](#) > [ccf726949 | Configurações de IP](#) >

ipconfig1 ...

ccf726949



Salvar



Descartar

Configurações de endereço IP público

Endereço IP público

Desassociar

Associar

Endereço IP público *

CCF726-ip (4.246.190.99)

[Criar um](#)

Configurações de endereço IP privado

Rede virtual/sub-rede

[CCF726-vnet/default](#)

Atribuição

Dinâmico

Estático

Endereço IP *

10.2.0.4

Configurando dependências para a máquina virtual criada

Uma vez configurada, é necessário iniciar a máquina no Azure, clicando em "**Iniciar**".

Conectar

Iniciar

Reiniciar

Parar

Capturar

Excluir

Atualizar

Abrir no celular

Comentários

CLI / PS

^ Fundamentos

Exibição JSON

Grupo de recursos (mover) : CCF726_group_03131238

Status : Parada (desalocada)

Local : East US

Assinatura (mover) : Azure for Students

ID da Assinatura : 6023a69c-b659-474e-bccb-0d8d96cc84bb

Marcações (editar) : Clique aqui para adicionar marcações

Sistema operacional : Linux

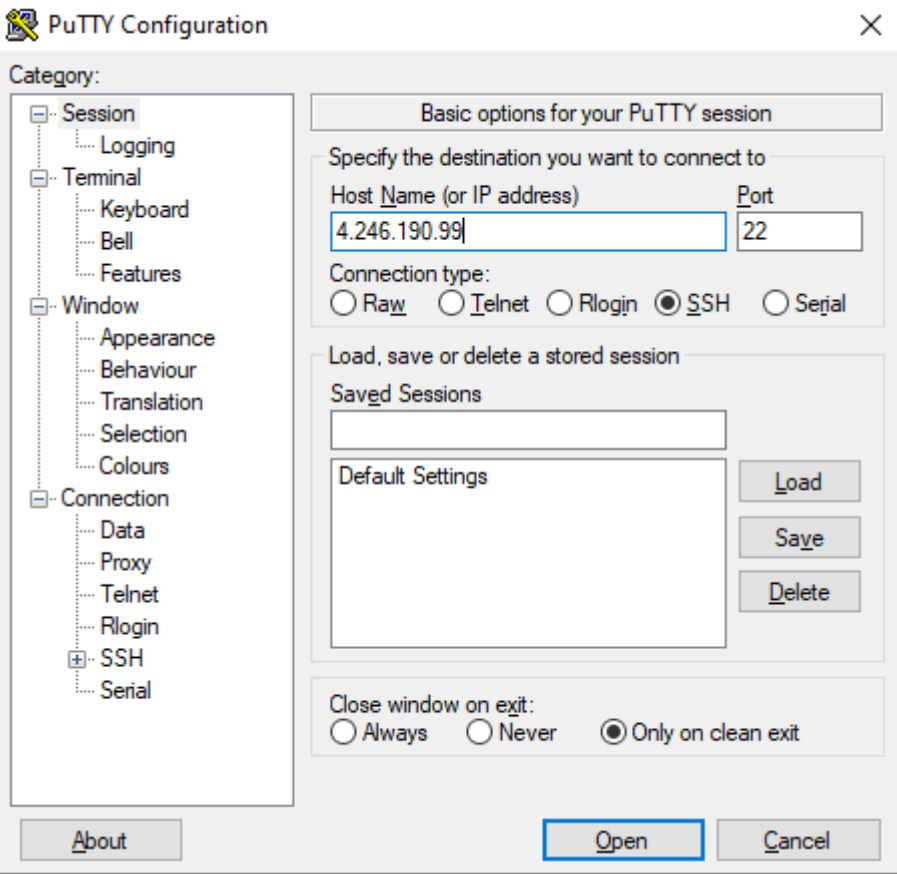
Tamanho : Standard D2s v3 (2 vcpus, 8 GiB de memória)

Endereço IP público : 4.246.190.99

Rede virtual/sub-rede : CCF726-vnet/default

Nome DNS : Não configurado

Abrir o programa Putty e inserir no campo Host Name (or IP address) o Endereço IP público da máquina "**4.246.190.99**".



Ao clicar em "**Open**" um terminal será aberto solicitando as credenciais do login.

```
mtsftsmts@CCF726: ~  
login as: mtsftsmts  
mtsftsmts@4.246.190.99's password:  
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1034-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Sun Mar 19 13:11:20 UTC 2023  
  
System load:  0.0      Processes:            134  
Usage of /:   23.0% of 28.89GB   Users logged in:     0  
Memory usage: 3%      IPv4 address for eth0: 10.2.0.4  
Swap usage:   0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
  just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
* Introducing Expanded Security Maintenance for Applications.  
  Receive updates to over 25,000 software packages with your  
  Ubuntu Pro subscription. Free for personal use.  
  
  https://ubuntu.com/azure/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
New release '22.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sun Mar 19 11:50:46 2023 from 45.178.248.51  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
(base) mtsftsmts@CCF726:~$
```

Em relação as dependências, foi escolhido instalar o pacote Anaconda, pois ele simplifica a instalação e o gerenciamento de pacotes, bibliotecas e ambientes para projetos de ciência de dados e aprendizado de máquina. Ele inclui um grande número de bibliotecas populares e úteis para esses campos, como Python, NumPy, pandas, Matplotlib, scikit-learn, TensorFlow, Jupyter Notebook, entre outras.

Após efetuar o login na máquina, os seguintes comandos foram executados:

- `cd /tmp`
- `curl -O https://repo.anaconda.com/archive/Anaconda3-2022.10-Linux-x86_64.sh`
- `bash Anaconda3-2022.10-Linux-x86_64.sh`
- Aceite os termos da licença

Após a instalação, execute o comando: **source ~/.bashrc**, para fazer as alterações no arquivo .bashrc e aplicar as novas configurações à sessão atual do terminal sem precisar fechar e abrir um novo terminal.

Uma vez instalado, é possível visualizar todas as bibliotecas instaladas pelo Anaconda através do comando: **conda list**.

OBS: A imagem abaixo mostra apenas algumas bibliotecas instaladas.

```
(base) mtsftsmts@CCF726:/$ conda list
# packages in environment at /home/mtsftsmts/anaconda3:
#
# Name                                Version                                Build      Channel
_ipyw_jlab_nb_ext_conf                0.1.0                                py39h06a4308_1
_libgcc_mutex                         0.1                                  main
_openmp_mutex                         5.1                                  1_gnu
alabaster                             0.7.12                              pyhd3eb1b0_0
anaconda                              2022.10                              py39_0
anaconda-client                       1.11.0                              py39h06a4308_0
anaconda-navigator                    2.3.1                                py39h06a4308_0
anaconda-project                      0.11.1                              py39h06a4308_0
anyio                                  3.5.0                                py39h06a4308_0
appdirs                               1.4.4                                pyhd3eb1b0_0
argon2-cffi                           21.3.0                              pyhd3eb1b0_0
argon2-cffi-bindings                  21.2.0                              py39h7f8727e_0
arrow                                  1.2.2                                pyhd3eb1b0_0
astroid                                2.11.7                              py39h06a4308_0
astropy                               5.1                                  py39h7deecbd_0
atomicwrites                           1.4.0                                py_0
attrs                                  21.4.0                              pyhd3eb1b0_0
automat                                20.2.0                              py_0
autopep8                              1.6.0                                pyhd3eb1b0_1
babel                                  2.9.1                                pyhd3eb1b0_0
backcall                              0.2.0                                pyhd3eb1b0_0
backports                              1.1                                  pyhd3eb1b0_0
backports.functools_lru_cache          1.6.4                                pyhd3eb1b0_0
```

Podemos visualizar a versão dos pacotes de forma independente, abaixo podemos ver as versões do **Jupyter** e do **Python** instaladas.

```
(base) mtsftsmts@CCF726:~$ jupyter --version
Selected Jupyter core packages...
IPython          : 7.31.1
ipykernel        : 6.15.2
ipywidgets       : 7.6.5
jupyter_client   : 7.3.4
jupyter_core     : 4.11.1
jupyter_server   : 1.18.1
jupyterlab       : 3.4.4
nbclient         : 0.5.13
nbconvert        : 6.4.4
nbformat         : 5.5.0
notebook         : 6.4.12
qtconsole        : 5.3.2
traitlets        : 5.1.1
(base) mtsftsmts@CCF726:~$ python --version
Python 3.9.13
```

Acessando o Jupyter da máquina virtual

Com a máquina virtual em execução, basta abrir o prompt do sistema operacional do computador pessoal e utilizar o seguinte comando para criar a instância para acessar o Jupyter remotamente:

ssh -L 8080:localhost:8888 mtsftsmts@4.246.190.99

O comando: " `ssh -L 8080:localhost:8888 mtsftsmts@4.246.190.99` " realiza uma conexão SSH segura (Secure Shell) com a máquina remota cujo endereço IP é 4.246.190.99 e nome de usuário é mtsftsmts.

A opção -L 8080:localhost:8888 configura um túnel SSH, que encaminha o tráfego da porta local 8080 para a porta 8888 na máquina remota. Essa técnica é conhecida como "port forwarding" (encaminhamento de porta) e é útil quando deseja-se acessar um serviço na máquina remota (nesse caso, um servidor Jupyter Notebook na porta 8888) através de uma porta local do computador.

Com esse comando, é possível acessar o servidor Jupyter Notebook remoto digitando <http://localhost:8080> no navegador do computador local, como se o servidor estivesse sendo executado localmente na porta 8080.

```
CA. mtsftsmts@CCF726: ~
Microsoft Windows [versão 10.0.19045.2728]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Martins>ssh -L 8080:localhost:8888 mtsftsmts@4.246.190.99
mtsftsmts@4.246.190.99's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1034-azure x86_64)
```

Em seguida, utilizar o seguinte comando: **jupyter notebook --no-browser**.

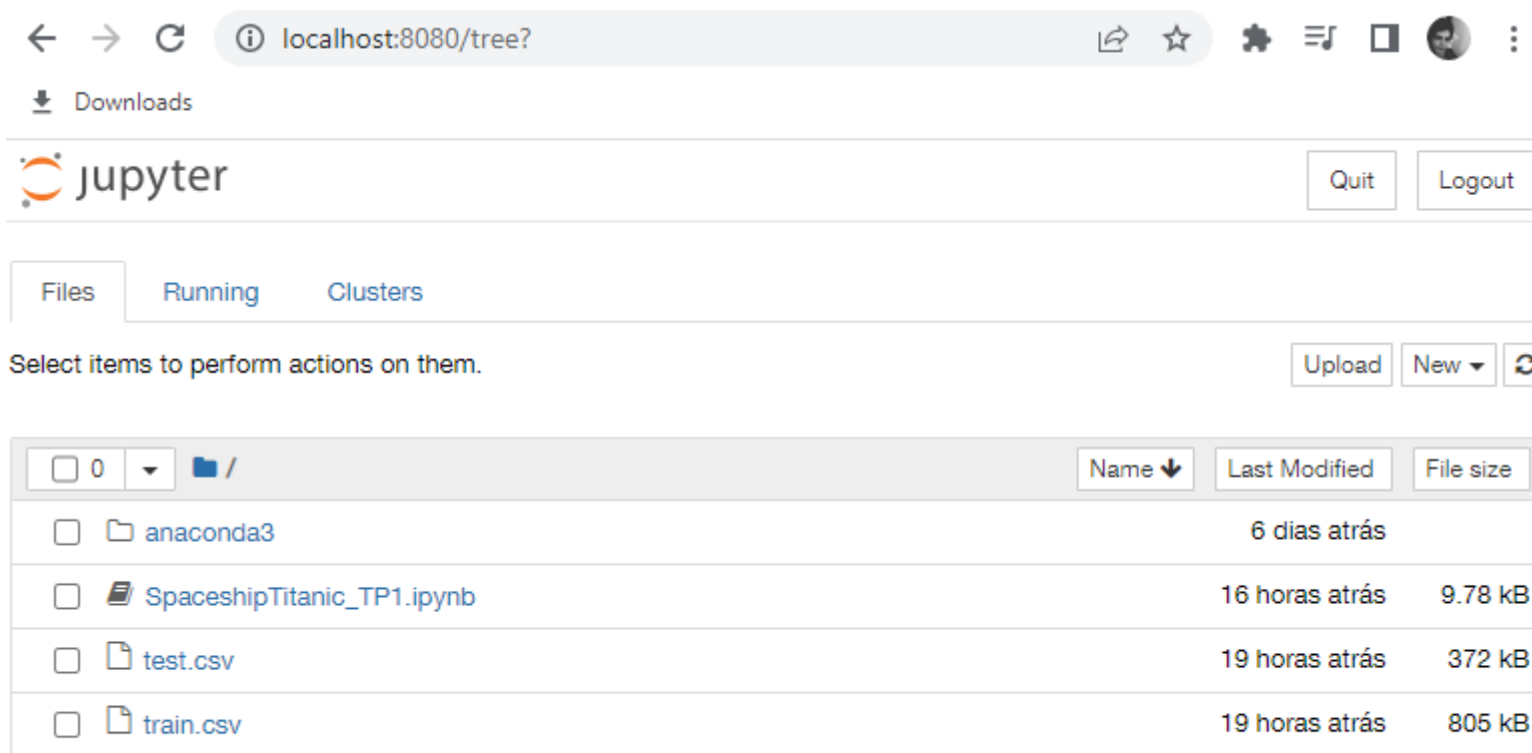

```
(base) mtsftsmts@CCF726:~$ jupyter notebook --no-browser
[I 2023-03-19 14:25:02.255 LabApp] JupyterLab extension loaded from /home/mtsftsmts/anaconda3/lib/python3.9/site-packages/jupyterlab
[I 2023-03-19 14:25:02.255 LabApp] JupyterLab application directory is /home/mtsftsmts/anaconda3/share/jupyter/lab
[I 14:25:02.262 NotebookApp] Serving notebooks from local directory: /home/mtsftsmts
[I 14:25:02.262 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 14:25:02.262 NotebookApp] http://localhost:8888/?token=f301872520c5b9031269238335022e56c6aefa478ef9f568
[I 14:25:02.262 NotebookApp] or http://127.0.0.1:8888/?token=f301872520c5b9031269238335022e56c6aefa478ef9f568
[I 14:25:02.262 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:25:02.271 NotebookApp]

To access the notebook, open this file in a browser:
    file:///home/mtsftsmts/.local/share/jupyter/runtime/nbserver-1486-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=f301872520c5b9031269238335022e56c6aefa478ef9f568
    or http://127.0.0.1:8888/?token=f301872520c5b9031269238335022e56c6aefa478ef9f568
```

Agora, para acessar o Jupyter Notebook, basta digitar no navegador a seguinte URL: [http://localhost:8080/tree?](http://localhost:8080/tree?token=1036cd4eb4546860f131cfb8a01c21423489643ae56efebd)

Se necessário, insira o token gerado para obter acesso ao Jupyter. Por exemplo: <http://localhost:8888/?token=1036cd4eb4546860f131cfb8a01c21423489643ae56efebd>

Após acessar o Jupyter Notebook, você poderá criar um novo arquivo .ipynb e fazer upload de arquivos. Neste exemplo, foi criado o arquivo SpaceshipTitanic_TP1.ipynb e realizada a importação dos arquivos .csv de treinamento e teste, disponíveis em: <https://www.kaggle.com/competitions/spaceship-titanic/data>.



Modelo utilizando Random Forest

Importando bibliotecas

```
In [1]: import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer
```

Carregando os dados

```
In [2]: dados_de_treino = pd.read_csv("train.csv")
dados_de_teste = pd.read_csv("test.csv")
```

Entendendo os dados

Visualizando as cinco primeiras linhas do Dataframe.

```
In [3]: dados_de_treino.head()
```

Out[3]:

	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Name	Transported
0	0001_01	Europa	False	B/0/P	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.0	0.0	Maham Ofracculy	False
1	0002_01	Earth	False	F/0/S	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.0	44.0	Juanna Vines	True
2	0003_01	Europa	False	A/0/S	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.0	49.0	Altark Susent	False
3	0003_02	Europa	False	A/0/S	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.0	193.0	Solam Susent	False
4	0004_01	Earth	False	F/1/S	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.0	2.0	Willy Santantines	True

Resumo geral da estrutura e conteúdo do Dataframe. Exibindo informações básicas como: índice, os nomes das colunas, o número de valores não nulos e o tipo de dados de cada coluna.

In [4]:

```
dados_de_treino.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8693 entries, 0 to 8692
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     8693 non-null   object
1   HomePlanet      8492 non-null   object
2   CryoSleep       8476 non-null   object
3   Cabin           8494 non-null   object
4   Destination     8511 non-null   object
5   Age             8514 non-null   float64
6   VIP             8490 non-null   object
7   RoomService     8512 non-null   float64
8   FoodCourt       8510 non-null   float64
9   ShoppingMall    8485 non-null   float64
10  Spa             8510 non-null   float64
11  VRDeck          8505 non-null   float64
12  Name            8493 non-null   object
13  Transported     8693 non-null   bool
dtypes: bool(1), float64(6), object(7)
memory usage: 891.5+ KB
```

Contando a quantidade de valores ausentes em cada coluna do DataFrame.

In [5]:

```
dados_de_treino.isnull().sum()
```

Out[5]:

```
PassengerId      0
HomePlanet       201
CryoSleep        217
Cabin            199
Destination      182
Age              179
VIP              203
RoomService      181
FoodCourt        183
ShoppingMall     208
Spa              183
VRDeck           188
Name             200
Transported       0
dtype: int64
```

Analisando a distribuição dos valores no conjunto de dados para identificar valores comuns ou incomuns. Retorna uma série contendo a frequência de ocorrência de cada valor único no DataFrame.

In [6]:

```
dados_de_treino.value_counts()
```



```
Out[6]: PassengerId  HomePlanet  CryoSleep  Cabin      Destination      Age  VIP      RoomService  FoodCourt  ShoppingMall  Spa      VRDeck  Name
Transported
0001_01      Europa      False      B/0/P      TRAPPIST-1e      39.0  False  0.0          0.0        0.0          0.0      0.0      Maham Of
racculy      False      1
6162_01      Earth      False      F/1181/S  55 Cancr i e      22.0  False  0.0          0.0        1.0          575.0    0.0      Bonyan H
ineyley      False      1
6175_01      Earth      False      G/1000/P  TRAPPIST-1e      18.0  False  628.0        0.0        0.0          31.0     150.0    The1 Pit
tler         False      1
6174_02      Earth      True       G/999/P   PSO J318.5-22    4.0   False  0.0          0.0        0.0          0.0      0.0      Cherry F
isheparks    True       1
6174_01      Earth      False      F/1274/P  55 Cancr i e      24.0  False  0.0          479.0     116.0        1.0      37.0     Jord Mcb
riddle       False      1

..
3195_02      Earth      False      G/505/S   PSO J318.5-22    60.0  False  0.0          31.0      6.0          223.0    356.0    Fredy Li
tthews       False      1
3195_01      Earth      False      G/505/S   TRAPPIST-1e      1.0   False  0.0          0.0        0.0          0.0      0.0      Rald Lit
thews        False      1
3191_01      Mars      True       F/603/S   TRAPPIST-1e      68.0  False  0.0          0.0        0.0          0.0      0.0      Fex Sin
True         1
3189_01      Mars      True       D/102/P   55 Cancr i e      40.0  False  0.0          0.0        0.0          0.0      0.0      Weers Ch
e            True       1
9280_02      Europa      False      E/608/S   TRAPPIST-1e      44.0  False  126.0        4688.0     0.0          0.0      12.0     Propsh H
ontichre     True       1
Length: 6606, dtype: int64
```

Resumo estatístico das colunas numéricas do DataFrame para compreender a distribuição e a tendência central dos dados numéricos. Contendo estatísticas como a contagem, média, desvio padrão, mínimo, quartis e máximo para cada coluna numérica.

```
In [7]: dados_de_treino.describe()
```

Out[7]:

	Age	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck
count	8514.000000	8512.000000	8510.000000	8485.000000	8510.000000	8505.000000
mean	28.827930	224.687617	458.077203	173.729169	311.138778	304.854791
std	14.489021	666.717663	1611.489240	604.696458	1136.705535	1145.717189
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	19.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	27.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	38.000000	47.000000	76.000000	27.000000	59.000000	46.000000
max	79.000000	14327.000000	29813.000000	23492.000000	22408.000000	24133.000000

```
In [8]: dados_de_teste.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4277 entries, 0 to 4276
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId      4277 non-null   object
1   HomePlanet       4190 non-null   object
2   CryoSleep        4184 non-null   object
3   Cabin            4177 non-null   object
4   Destination      4185 non-null   object
5   Age              4186 non-null   float64
6   VIP              4184 non-null   object
7   RoomService      4195 non-null   float64
8   FoodCourt        4171 non-null   float64
9   ShoppingMall     4179 non-null   float64
10  Spa              4176 non-null   float64
11  VRDeck           4197 non-null   float64
12  Name             4183 non-null   object
dtypes: float64(6), object(7)
memory usage: 434.5+ KB
```

```
In [9]: dados_de_teste.head()
```

Out[9]:

	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Name
0	0013_01	Earth	True	G/3/S	TRAPPIST-1e	27.0	False	0.0	0.0	0.0	0.0	0.0	Nelly Carsoning
1	0018_01	Earth	False	F/4/S	TRAPPIST-1e	19.0	False	0.0	9.0	0.0	2823.0	0.0	Lerome Peckers
2	0019_01	Europa	True	C/0/S	55 Cancr i e	31.0	False	0.0	0.0	0.0	0.0	0.0	Sabih Unhearfus
3	0021_01	Europa	False	C/1/S	TRAPPIST-1e	38.0	False	0.0	6652.0	0.0	181.0	585.0	Meratz Caltilter
4	0023_01	Earth	False	F/5/S	TRAPPIST-1e	20.0	False	10.0	0.0	635.0	0.0	0.0	Brence Harperez

```
In [10]: dados_de_teste.isnull().sum()
```

```
Out[10]: PassengerId      0
HomePlanet    87
CryoSleep     93
Cabin        100
Destination   92
Age           91
VIP           93
RoomService   82
FoodCourt    106
ShoppingMall  98
Spa           101
VRDeck        80
Name          94
dtype: int64
```

Selecionando colunas numéricas

Primeiramente, é preciso estabelecer uma lista contendo os nomes das colunas numéricas que serão empregadas como features no modelo. As respectivas colunas são:

- **CryoSleep**: indica se o passageiro escolheu ser colocado em animação suspensa durante a viagem.
- **Age**: idade do passageiro.
- **VIP**: indica se o passageiro pagou pelo serviço VIP durante a viagem.
- **RoomService**: valor cobrado pelo serviço de quarto.
- **FoodCourt**: valor cobrado pelo uso do refeitório.
- **ShoppingMall**: valor cobrado pelo uso do shopping a bordo.
- **Spa**: valor cobrado pelo uso do spa.
- **VRDeck**: valor cobrado pelo uso do deck de realidade virtual.

```
In [11]: colunas_de_features = ['CryoSleep', 'Age', 'VIP', 'RoomService', 'FoodCourt', 'ShoppingMall', 'Spa', 'VRDeck']
# Criando a variável 'X', que conterá as features dos dados de treino.
X = dados_de_treino[colunas_de_features]
# Criando a variável 'y', que conterá o target dos dados de treino.
# OBS: O alvo é a coluna "Transported", que indica se o passageiro foi transportado para outra dimensão.
y = dados_de_treino["Transported"]
```

Dividindo os dados em treino e validação

- **train_test_split**: É uma função da biblioteca scikit-learn que divide os dados em conjuntos de treino e validação. A função recebe como argumentos os dados das características (X) e o alvo (y), além de outros parâmetros opcionais.
- **test_size**: É um parâmetro opcional da função train_test_split que indica a proporção dos dados que serão reservados para o conjunto de validação. Neste caso, test_size=0.2 significa que 20% dos dados serão usados para validação e os 80% restantes para treino.
- **random_state**: É um parâmetro opcional da função train_test_split que controla a aleatoriedade da divisão dos dados. Ao definir um valor fixo, como random_state=42, garantimos que a divisão seja sempre a mesma, o que facilita a reprodução dos resultados e a comparação entre diferentes experimentos. Se não for especificado, a divisão pode ser diferente a cada execução do código.
- **X_treino, X_validacao, y_treino, y_validacao**: São as variáveis que receberão os dados divididos. A função train_test_split retorna quatro valores, que são as características e o alvo dos conjuntos de treino e validação, respectivamente.

As seguintes variáveis representam:

- X_treino: características do conjunto de treino
- X_validacao: características do conjunto de validação
- y_treino: alvo do conjunto de treino
- y_validacao: alvo do conjunto de validação

```
In [12]: # Dividindo os dados em treino e validação
X_treino, X_validacao, y_treino, y_validacao = train_test_split(X, y, test_size=0.2, random_state=42)
```

Tratando valores ausentes

- **SimpleImputer**: É uma classe da biblioteca scikit-learn que fornece uma estratégia básica para o preenchimento de valores ausentes. A classe aceita um parâmetro chamado strategy, que define a estratégia de preenchimento dos valores ausentes.
- **strategy='mean'**: Neste caso, está sendo utilizado a estratégia 'mean', que preenche os valores ausentes com a média dos valores presentes na coluna.
- **X_treino_preenchido** = preenchedor.fit_transform(X_treino): A função fit_transform ajusta o imputer aos dados de treino (calculando a média de cada coluna) e, em seguida, aplica a transformação nos dados, preenchendo os valores ausentes com as médias calculadas. O resultado dessa transformação, um conjunto de dados de treino com os valores ausentes preenchidos, é atribuído à variável X_treino_preenchido.
- **X_validacao_preenchido** = preenchedor.transform(X_validacao): A função transform aplica a transformação do imputer, que foi ajustado aos dados de treino, aos dados de validação. Isso significa que os valores ausentes no conjunto de validação são preenchidos com as médias calculadas a partir

do conjunto de treino. O resultado dessa transformação, um conjunto de dados de validação com os valores ausentes preenchidos, é atribuído à variável X_validacao_preenchido.

```
In [13]: # Tratando valores ausentes (NaN) nos conjuntos de treino e validação
preenchedor = SimpleImputer(strategy='mean')
X_treino_preenchido = preenchedor.fit_transform(X_treino)
X_validacao_preenchido = preenchedor.transform(X_validacao)
```

Definindo X_test

A variável X_test, conterá as features dos dados de teste. Selecionando apenas as colunas numéricas (definidas na lista colunas_de_features) do DataFrame dados_de_teste. Isso é necessário para fazer as previsões finais do modelo.

```
In [14]: # Definindo X_test (features) dos dados de teste
X_test = dados_de_teste[colunas_de_features]
```

Aplicando o preenchedor (imputer) aos dados de teste para tratar os valores ausentes (NaN) que possam estar presentes. Assim como fizemos para os dados de treino e validação, precisamos garantir que os dados de teste também não contenham valores ausentes antes de usá-lo.

```
In [15]: # Aplicando preenchedor (imputer) aos dados de teste
X_test_preenchido = preenchedor.transform(X_test)
```

Treinando o modelo de Random Forest

- **RandomForestClassifier:** É uma classe da biblioteca scikit-learn que implementa um algoritmo de aprendizado de máquina baseado em árvores de decisão chamado Random Forest.
- **random_state=42:** É um parâmetro opcional da classe RandomForestClassifier que controla a aleatoriedade do processo de construção das árvores de decisão no modelo. Ao definir um valor fixo, como random_state=42, garantimos que o modelo seja sempre o mesmo, o que facilita a reprodução dos resultados e a comparação entre diferentes experimentos. Se não for especificado, o modelo pode ser diferente a cada execução do código.
- **classificador** = armazena o modelo de Random Forest que será treinado com os dados.

```
In [16]: classificador = RandomForestClassifier(random_state=42)

# treina o modelo de Random Forest usando os dados de treino. A função recebe como argumentos as features (X_treino_preenchido) e o target
classificador.fit(X_treino_preenchido, y_treino)

Out[16]: RandomForestClassifier(random_state=42)
```

Previsões nos dados de validação e calculando a acurácia

```
In [17]: previsoes = classificador.predict(X_validacao_preenchido)
accuracy = accuracy_score(y_validacao, previsoes)
print(f"Acurácia: {accuracy:.2f}")
```

Acurácia: 0.76

Arquivo de submissão

```
In [18]: # Previsões finais nos dados de teste.
previsoes_de_teste = classificador.predict(X_test_preenchido)
```

```
In [19]: # Criando o arquivo de submissão
submissao = pd.DataFrame({"PassengerId": dados_de_teste["PassengerId"], "Transported": previsoes_de_teste})
submissao.to_csv("submissao_spaceshiptitanic.csv", index=False)
```

Arquivo de submissão gerado com sucesso.

Files

Running

Clusters

Select items to perform actions on them.

Upload

New

<input type="checkbox"/> 0		Name	Last Modified	File size
<input type="checkbox"/>	📁	anaconda3	6 dias atrás	
<input type="checkbox"/>	📄	SpaceshipTitanic_TP1_ambientacao.ipynb	Running 3 minutos atrás	405 kB
<input type="checkbox"/>	📄	submissao_spaceshiptitanic.csv	4 minutos atrás	57.6 kB
<input type="checkbox"/>	📄	test.csv	um dia atrás	372 kB
<input type="checkbox"/>	📄	train.csv	um dia atrás	805 kB

Submissão no Kaggle


Meu usuário no Kaggle: <https://www.kaggle.com/mtsftsmts>

Submissão do arquivo .csv na plataforma Kaggle:

Getting Started Prediction Competition

Spaceship Titanic

Predict which passengers are transported to an alternate dimension

 Kaggle · 2,298 teams · Ongoing

Overview

Data

Code

Discussion

Leaderboard

Rules

Team

Submissions

Submit Predictions

...


Submissions

All

Successful

Errors

Recent

Submission and Description	Public Score
<div><div></div><div>submissao_spaceshiptitanic.csv Complete · now</div></div>	0.78793