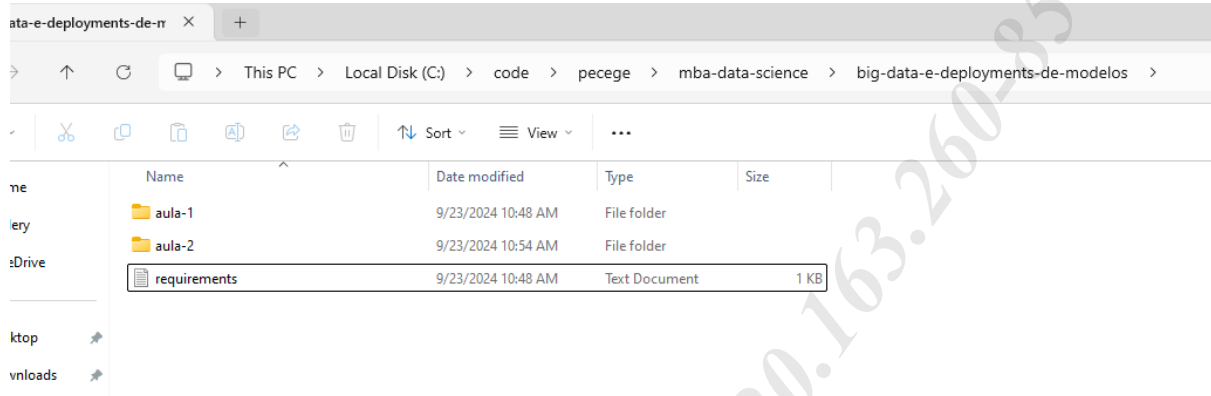


Prof. Helder Prado Santos

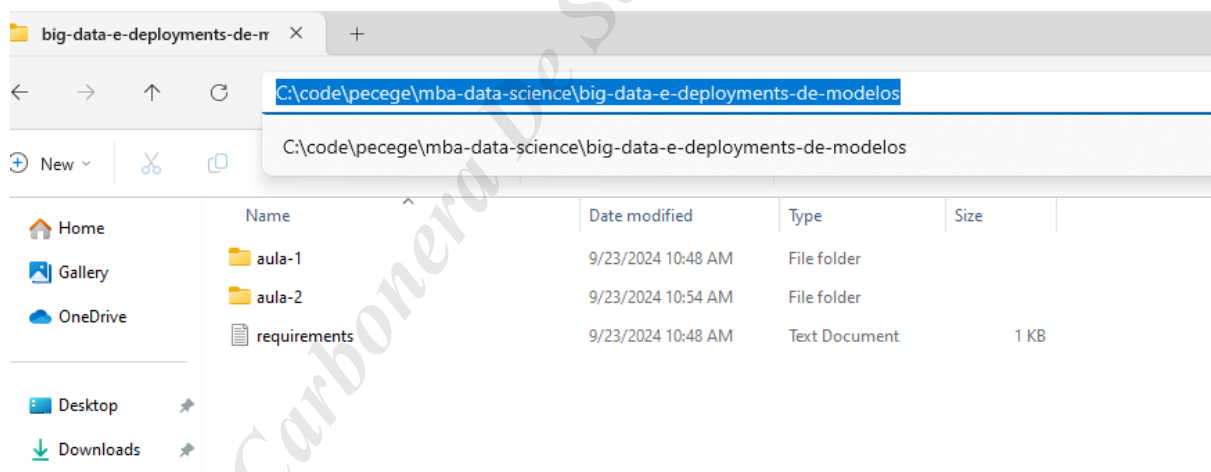
## Passo 1: Instalar o Ambiente Virtual Python

Extraia os arquivos da plataforma acadêmica no local desejado.



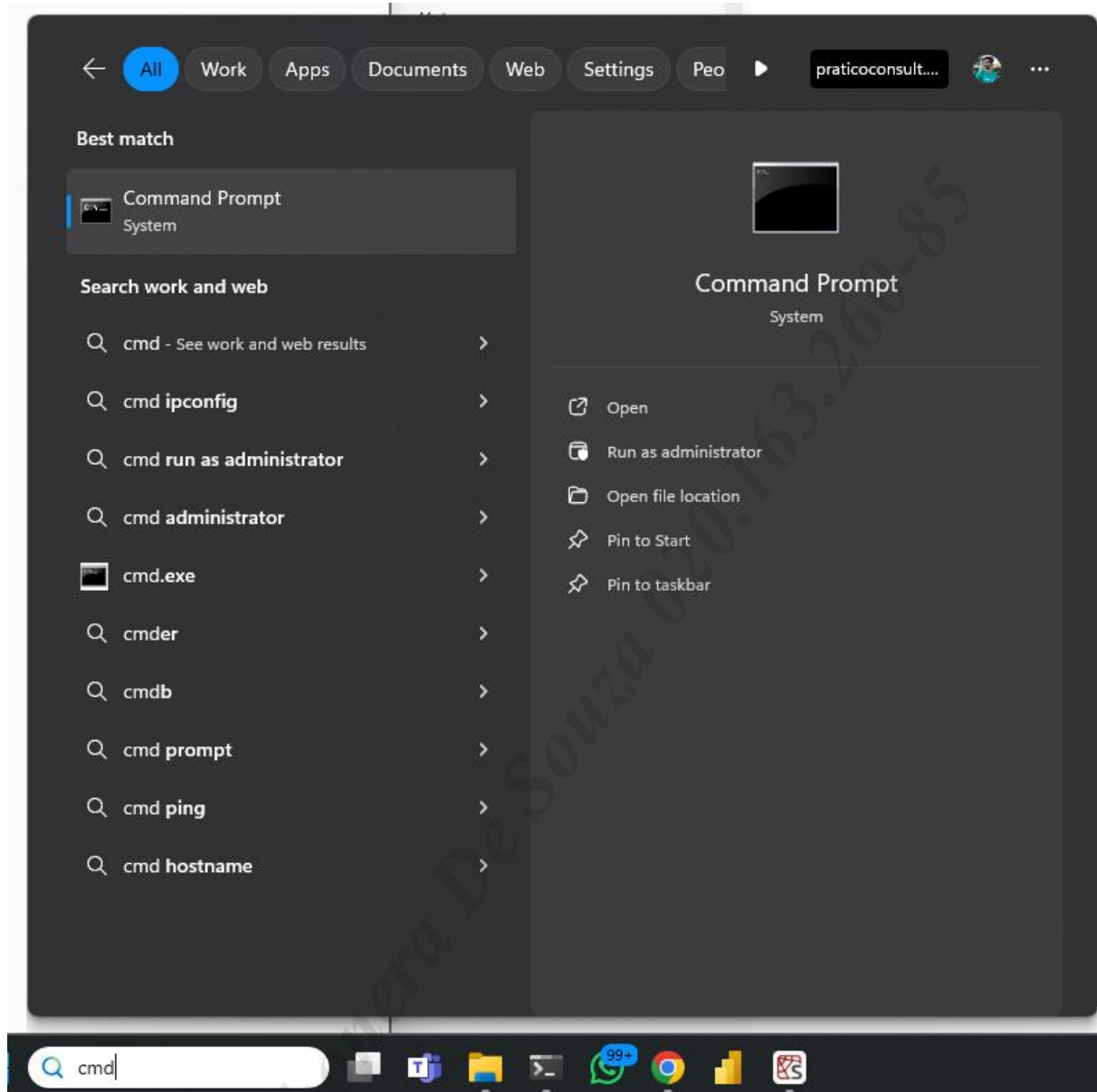
Copie o caminho da pasta onde os arquivos foram extraídos:

- **Windows:** Clique no campo do caminho no Explorer e copie o endereço.
- **Linux/macOS:** Abra o terminal, navegue até a pasta usando o comando `cd` ou copie o caminho completo da pasta.

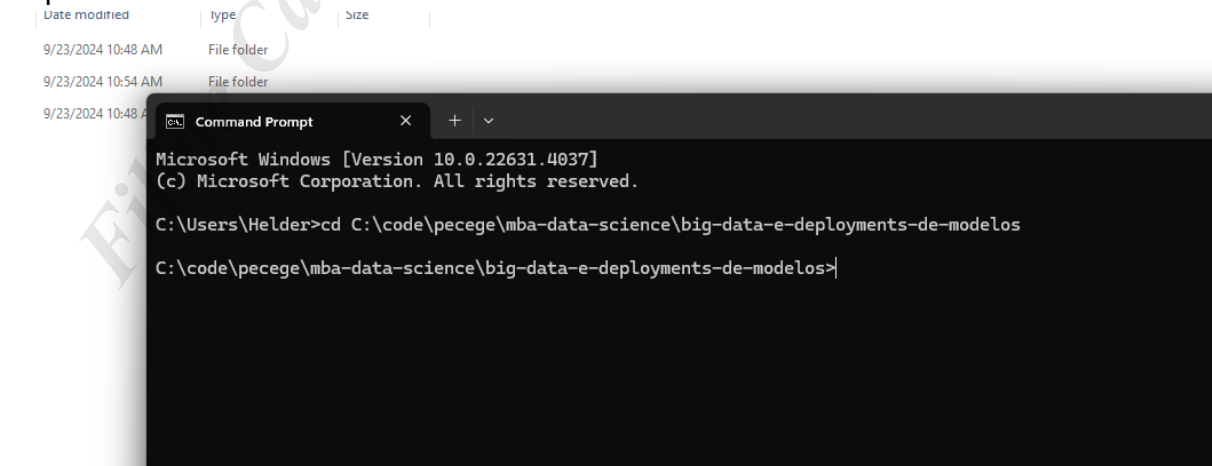


Abra o terminal:

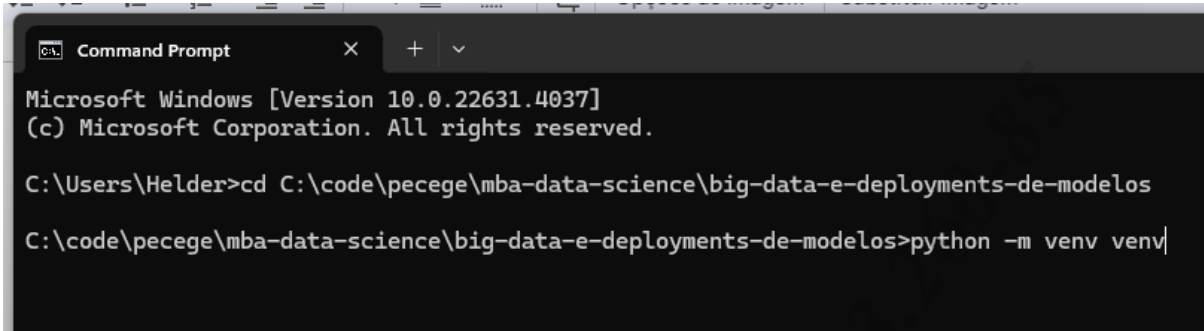
- **Windows:** Pressione `Win + R`, digite `cmd` e aperte `Enter`.
- **Linux/macOS:** Abra o terminal a partir do menu ou atalho.



Dentro do terminal, digite “cd” e cole o caminho copiado do diretório da pasta com os arquivos extraídos



Digite “python -m venv venv” e aperte “enter”. Será criada uma nova pasta chamada “venv” e o processo de criação de ambiente virtual estará concluído. Caso o comando não funcione, digite “python3 -m venv venv” e aperte “enter”.



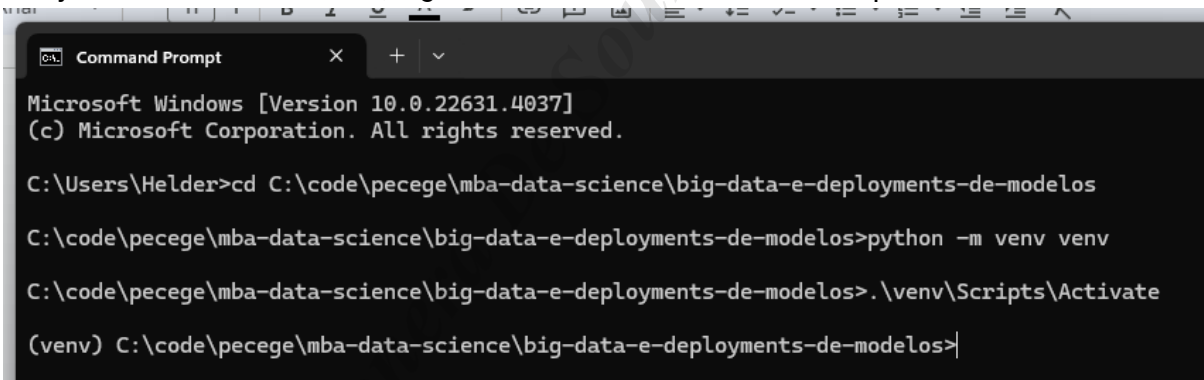
```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Helder>cd C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos

C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>python -m venv venv
```

## Passo 2: Acessar o ambiente virtual

Dentro da pasta dos arquivos extraídos, onde agora tem a pasta chamada “venv”, digite o seguinte comando “.venv\Scripts\Activate” e aperte “enter”. Aparecerá o nome “venv” entre parênteses ao lado do caminho do diretório. Se o nome “(venv)” aparecer, quer dizer que você acessou com sucesso o ambiente virtual. Caso você esteja no Linux ou no macOS, digite “source venv/bin/activate” e aperte “enter”.



```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Helder>cd C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos

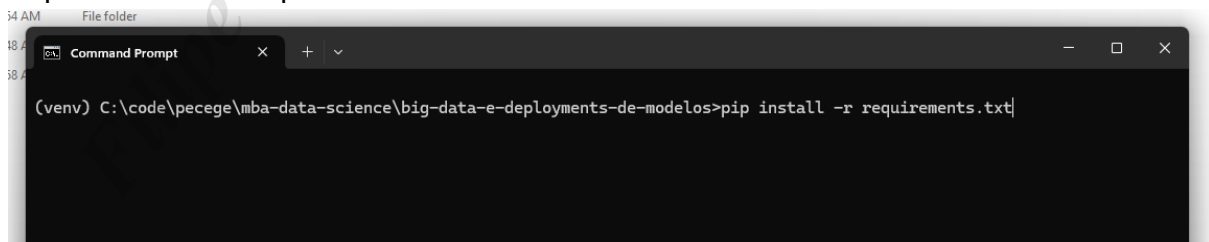
C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>python -m venv venv

C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>.\venv\Scripts\Activate

(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>
```

## Passo 3: Instalação dos pacotes necessários

Ainda dentro do ambiente virtual, digite o seguinte comando “pip install -r requirements.txt” e aperte “enter”.



```
(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>pip install -r requirements.txt
```

A instalação dos pacotes será iniciada.

```
Command Prompt - pip insta x + v
(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>pip install -r requirements.txt
Collecting spyder (from -r requirements.txt (line 1))
  Using cached spyder-6.0.0-py3-none-any.whl.metadata (23 kB)
Collecting mlflow (from -r requirements.txt (line 2))
  Using cached mlflow-2.16.2-py3-none-any.whl.metadata (29 kB)
Collecting pyspark (from -r requirements.txt (line 3))
  Downloading pyspark-3.5.2.tar.gz (317.3 MB)
  36.8/317.3 MB 11.7 MB/s eta 0:00:24
```

A instalação dos pacotes pode levar algum tempo. Aguarde até que o processo seja concluído, após o qual você poderá utilizar o terminal normalmente novamente.

```
Command Prompt - pip insta x + v
Using cached pyasn1-0.6.1-py3-none-any.whl (83 kB)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (pyproject.toml) ... done
  Created wheel for pyspark: filename=pyspark-3.5.2-py2.py3-none-any.whl size=317812406 sha256=d4de855957a0cd9d7102ba712f3738796962d6338f90aa0513c695306e0afc70
  Stored in directory: c:\users\helder\appdata\local\pip\cache\wheels\cf\c0\b9\f147f4220fd1d9277d0981b88b35b26f03ad910fffd60013a6
Successfully built pyspark
Installing collected packages: webencodings, wcwidth, text-unidecode, sortedcontainers, snowballstemmer, pywin32, pyuca, pytz, PyQtWebEngine-Qt5, PyQt5-Qt5, py4j, pure-eval, ptyprocess, pickleshare, fastjsonschema, aniso8601, zipp, wrapt, w hatthepatch, watchdog, waitress, urllib3, ujson, tzdata, typing-extensions, types-python-dateutil, traitlets, tornado, t omlkit, tomli, tinycss2, threadpoolctl, textdistance, tabulate, sqlparse, sphinxcontrib-serializinghtml, sphinxcontrib-q thelp, sphinxcontrib-jsmath, sphinxcontrib-htmlhelp, sphinxcontrib-devhelp, sphinxcontrib-applehelp, soupsieve, smmap, s ix, setuptools, rtree, rpsd-py, pyzmq, pyyaml, pywin32-ctypes, python-slugify, pyspark, PyQt5-sip, pyparsing, pylint-ven v, pyjwt, pygments, pyflakes, pydocstyle, pycparser, pycodestyle, pyasn1, psutil, protobuf, prompt-toolkit, pluggy, plat formdirs, pillow, pexpect, pathspec, parso, pandocfilters, packaging, numpy, nest-asyncio, mypy-extensions, multidict, m ore-itertools, mistune, mdurl, mccabe, MarkupSafe, markdown, kiwisolver, jupyterlab-pygments, joblib, jellyfish, jaraco. context, itsdangerous, isort, intervaltree, inflection, imagesize, idna, greenlet, graphql-core, frozenlist, fonttools, executing, docutils, docstring-to-markdown, dill, diff-match-patch, defusedxml, decorator, debugpy, cycpler, colorama, cl oudpickle, charset-normalizer, chardet, certifi, cachetools, blinker, babel, attrs, atomicwrites, astroid, alabaster, ai ohappyeyeballs, yarl, Werkzeug, three-merge, sqlalchemy, scipy, rsa, requests, referencing, qtpy, qstylizer, pytoolconfi g, python-lsp-jsonrpc, python-dateutil, pyqt5, pylint, pyasn1-modules, pyarrow, patsy, matplotlib-inline, markdown-it-py , Mako, jupyter-core, Jinja2, jedi, jaraco.functools, jaraco.classes, importlib-metadata, graphql-relay, gitdb, flake8, Deprecated, contourpy, comm, click, cffi, bleach, binaryornot, beautifulsoup4, autopep8, asttokens, aiosignal, yapf, sup erqt, stack-data, sphinx, scikit-learn, rich, qtawesome, qdarkstyle, python-lsp-server, pyqtwebengine, pynacl, pandas, o pentelemetry-api, matplotlib, keyring, jupyter-client, jsonschema-specifications, graphene, google-auth, gitpython, Flas k, docker, cryptography, black, arrow, alembic, aiohttp, statsmodels, rope, python-lsp-black, pyls-spyder, opentelemetry -semantic-conventions, numpydoc, jsonschema, ipython, databricks-sdk, cookiecutter, asyncssh, pygithub, opentelemetry-sd k, nbformat, ipykernel, spyder-kernels, qtconsole, nbclient, mlflow-skinny, nbconvert, mlflow, spyder
```

## Passo 4: Acessar o Spyder dentro do ambiente virtual criado

Após a instalação dos pacotes, ainda dentro do ambiente virtual, digite o seguinte comando “spyder” e aperte “enter”. O Spyder irá inicializar em segundo plano e você poderá utilizar novamente o terminal.



```
Command Prompt

le-auth-2.35.0 graphene-3.3 graphql-core-3.2.4 graphql-relay-3.2.0 greenlet-3.1.1 idna-3.10 imagesize-1.4.1 importlib-me
tadata-8.4.0 inflection-0.5.1 intervaltree-3.1.0 ipykernel-6.29.5 ipython-8.27.0 isort-5.13.2 itsdangerous-2.2.0 jaraco.
classes-3.4.0 jaraco.context-6.0.1 jaraco.functools-4.0.2 jedi-0.19.1 jellyfish-1.1.0 joblib-1.4.2 jsonschema-4.23.0 jso
n-schema-specifications-2023.12.1 jupyter-client-8.6.3 jupyter-core-5.7.2 jupyterlab-pygments-0.3.0 keyring-25.4.1 kiwis
lver-1.4.7 markdown-3.7 markdown-it-py-3.0.0 matplotlib-3.9.2 matplotlib-inline-0.1.7 mccabe-0.7.0 mdurl-0.1.2 mistune-3
.0.2 mlflow-2.16.2 mlflow-skinny-2.16.2 more-itertools-10.5.0 multidict-6.1.0 mypy-extensions-1.0.0 nbclient-0.10.0 nbco
nvert-7.16.4 nbformat-5.10.4 nest-asyncio-1.6.0 numpy-2.1.1 numpydoc-1.8.0 opentelemetry-api-1.27.0 opentelemetry-sdk-1.
27.0 opentelemetry-semantic-conventions-0.48b0 packaging-24.1 pandas-2.2.3 pandocfilters-1.5.1 parso-0.8.4 pathspec-0.12
.1 patsy-0.5.6 pexpect-4.9.0 pickleshare-0.7.5 pillow-10.4.0 platformdirs-4.3.6 pluggy-1.5.0 prompt-toolkit-3.0.47 proto
buf-5.28.2 psutil-6.0.0 ptyprocess-0.7.0 pure-eval-0.2.3 py4j-0.10.9.7 pyarrow-17.0.0 pyasn1-0.6.1 pyasn1-modules-0.4.1
pycodestyle-2.12.1 pycparser-2.22 pydocstyle-6.3.0 pyflakes-3.2.0 pygithub-2.4.0 pygments-2.18.0 pyjwt-2.9.0 pylint-3.3.
0 pylint-venv-3.0.3 pyls-spyder-0.4.0 pynacl-1.5.0 pyparsing-3.1.4 pyqt5-5.15.11 pyqtwebengine-5.15.7 pyspark-3.5.2 pyth
on-dateutil-2.9.0.post0 python-lsp-black-2.0.0 python-lsp-jsonrpc-1.1.2 python-lsp-server-1.12.0 python-slugify-8.0.4 py
toolconfig-1.3.1 pytz-2024.2 pyuca-1.2 pywin32-306 pywin32-ctypes-0.2.3 pyyaml-6.0.2 pyzmq-26.2.0 qdarkstyle-3.2.3 qstyl
izer-0.2.3 qtawesome-1.3.1 qtconsole-5.6.0 qtpy-2.4.1 referencing-0.35.1 requests-2.32.3 rich-13.8.1 rope-1.13.0 rpds-py
-0.20.0 rsa-4.9 rtree-1.3.0 scikit-learn-1.5.2 scipy-1.14.1 setuptools-75.1.0 six-1.16.0 smmap-5.0.1 snowballstemmer-2.2
.0 sortedcontainers-2.4.0 soupsieve-2.6 sphinx-8.0.2 sphinxcontrib-applehelp-2.0.0 sphinxcontrib-devhelp-2.0.0 sphinxcon
trib-htmlhelp-2.1.0 sphinxcontrib-jsmath-1.0.1 sphinxcontrib-qthelp-2.0.0 sphinxcontrib-serializinghtml-2.0.0 spyder-6.0
.0 spyder-kernels-3.0.0 sqlalchemy-2.0.35 sqlparse-0.5.1 stack-data-0.6.3 statsmodels-0.14.3 superqt-0.6.7 tabulate-0.9
.0 text-unidecode-1.3 textdistance-4.6.3 threadpoolctl-3.5.0 three-merge-0.1.1 tinycss2-1.3.0 tomli-2.0.1 tomllkit-0.13.2
tornado-6.4.1 traitlets-5.14.3 types-python-dateutil-2.9.0.20240906 typing-extensions-4.12.2 tzdata-2024.1 ujson-5.10.0
urllib3-2.2.3 waitress-3.0.0 watchdog-5.0.2 wcwidth-0.2.13 webencodings-0.5.1 whatthepatch-1.0.6 wrapt-1.16.0 yapf-0.40.
2 yarl-1.11.1 zipp-3.20.2

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>spyder

(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>
```

## Passo 5: Acessar a interface gráfica do Mlflow

Ainda dentro do ambiente virtual, digite o seguinte comando “mlflow ui” e aperte “enter”. O Mlflow irá inicializar e você poderá acessar a sua interface utilizando o seu browser de preferência, no endereço: <http://localhost:5000>.

```
Command Prompt - mlflow u

Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Helder>cd C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos

C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>.venv\Scripts\Activate

(venv) C:\code\pecege\mba-data-science\big-data-e-deployments-de-modelos>mlflow ui
INFO:waitress:erving on http://127.0.0.1:5000
```

A página acessada deverá parecer com a imagem abaixo.

localhost:5000/#/experiments/0?searchFilter=&orderByKey=attributes.start\_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelVersionFilter=

mlflow 2.16.2 Experiments Models

## Experiments

- Search Experiments
- ☒ Default
  - ☐ Regressão Linear Simples - tempodist

## Default

[Provide Feedback](#) [Add Description](#)

Runs

Evaluation

Experimental

Traces

Experimental

📄

📈

Q metrics.rmse &lt; 1 and params.model = "tree"

🔍

Time cre

☐

Run Name

Created

Dataset