

Cap01. Introdução

A maior parte das bibliotecas pode ser facilmente instalada usando pip ou conda:

A biblioteca fastai necessita usar: `pip install --no-deps fastai`.

A biblioteca umap: `pip install umap-learn`,

A biblioteca janitor: `pip install pyjanitor`

A biblioteca autosklearn: `pip install auto-sklearn`.

Em geral, utilizo o Jupyter para fazer análises. Você pode usar outras ferramentas de notebook também. Note que algumas, como o Google Colab, têm várias bibliotecas pré-instaladas (embora suas versões possam estar desatualizadas).

Há duas opções principais para instalar bibliotecas em Python. Uma delas é usar o pip (um acrônimo para Pip Installs Python, isto é, Pip instala Python), uma ferramenta que acompanha Python. A outra opção é usar o Anaconda (<https://anaconda.org>). Ambas serão apresentadas a seguir.

Bibliotecas usadas

```
pip install autosklearn,  
catboost, category_encoders, dtreeviz,  
eli5, fancyimpute, fastai, featuretools,  
glmnet_py, graphviz, hdbscan, imblearn,  
janitor, lime, matplotlib, missingno, mlxtend,  
numpy, pandas, pdpbox, phate, pydotplus, rfpimp, scikitplot,  
scipy, seaborn, shap, sklearn, statsmodels, tpot, treeinterpreter,  
umap, xgbfir, xgboost, yellowbrick
```

```
import  
pip install autosklearn,  
catboost, category_encoders, dtreeviz,
```

```
eli5, fancyimpute, fastai, featuretools,  
glmnet_py, graphviz, hdbscan, imblearn,  
janitor, lime, matplotlib, missingno, mlxtend,  
numpy, pandas, pdpbox, phate, pydotplus, rfpimp, scikitplot,  
scipy, seaborn, shap, sklearn, statsmodels, tpot, treeinterpreter,  
umap, xgbfir, xgboost, yellowbrick
```

```
for lib in [  
    autosklearn,  
    catboost,  
    category_encoders,  
    dtreeviz,  
    eli5,  
    fancyimpute,  
    fastai,  
    featuretools,  
    glmnet_py,  
    graphviz,  
    hdbscan,  
    imblearn,  
    lime,  
    janitor,  
    matplotlib,  
    missingno,  
    mlxtend,  
    numpy,  
    pandas,  
    pandas_profiling,  
    pdpbox,  
    phate,  
    pydotplus,  
    rfpimp,  
    scikitplot,  
    scipy,  
    seaborn,  
    shap,  
    sklearn,  
    statsmodels,  
    tpot,  
    treeinterpreter,  
    umap,  
    xgbfir,  
    xgboost,  
    yellowbrick,  
]:  
  
    try:  
        print(lib.__name__, lib.__version__)  
  
    except:  
        print("Missing", lib.__name__)
```

Há duas opções principais para instalar bibliotecas em Python. Uma delas é usar o pip (um acrônimo para Pip Installs Python, isto é, Pip instala Python), uma ferramenta que acompanha Python. A outra opção é usar o Anaconda (<https://anaconda.org>). Ambas serão apresentadas a seguir.

Instalação com pip

Antes de usar o pip, criaremos um ambiente sandbox no qual instalaremos nossas bibliotecas. É um ambiente virtual que chamaremos de env:

```
python -m venv env
```

No Macintosh e no Linux, utilizar python; no Windows, use python3. Se o Windows não reconhecer esse comando no prompt, talvez seja necessário fazer uma reinstalação ou uma correção em sua instalação a fim de garantir que a opção “Add Python to my PATH” (Adicionar Python ao meu PATH) esteja marcada.

Em seguida, ative o ambiente para que, quando instalar suas bibliotecas, elas sejam colocadas no ambiente sandbox, e não na instalação global de Python. Como muitas dessas bibliotecas mudam e são atualizadas, é melhor isolar as versões por projeto, para que você saiba que seu código executará.

Eis o modo como ativamos o ambiente virtual no Linux e no Macintosh:

```
source env/bin/activate
```

Você perceberá que o prompt será atualizado, mostrando que estamos usando o ambiente virtual:

```
which python  
env/bin/python
```

No Windows, ative o ambiente executando o seguinte comando:

```
env\Scripts\activate.bat
```

Novamente, você perceberá que o prompt será atualizado, mostrando que estamos usando o ambiente virtual:

```
where python  
env\Scripts\python.exe
```

Você poderá instalar pacotes usando o pip em qualquer plataforma.

```
pip install pandas
```

Alguns dos nomes de pacotes são diferentes dos nomes das bibliotecas. Você pode procurar os pacotes usando:

```
pip search nomebiblioteca
```

Depois que tiver seus pacotes instalados, será possível criar um arquivo com todas as versões dos pacotes usando pip:

```
pip freeze > requirements.txt
```

Com esse arquivo requirements.txt, você poderá instalar facilmente os pacotes em um novo ambiente virtual:

(other_env) \$ pip install -r requirements.txt

Instalação com **conda**

A ferramenta conda acompanha o Anaconda, e ela nos permite criar ambientes e instalar pacotes.

Para criar um ambiente chamado env, execute o seguinte:

```
conda create --name env python=3.8
```

Para ativar esse ambiente, execute:

```
conda activate env
```

Esse comando atualizará o prompt tanto em sistemas Unix como Windows. Agora você pode procurar pacotes usando:

```
conda search nomedabiblioteca
```

Para instalar um pacote, por exemplo o Pandas, execute:

```
conda install pandas
```

Para criar um arquivo contendo os requisitos de pacotes, execute:

```
conda env export > environment.yml
```

Para instalar esses requisitos em um novo ambiente, execute:

```
conda create -f environment.yml
```

Alerta

Algumas das bibliotecas mencionadas neste livro não estão disponíveis no repositório do Anaconda para instalação. Não se preocupe. **Você pode usar o pip dentro de um ambiente conda** (não é necessário criar outro ambiente virtual) para instalar essas bibliotecas.