MINERAÇÃO DE DADOS EM PYTHON

ICMC-USP

Victor Alexandre Padilha
André Carlos Ponce de Leon Ferreira de Carvalho
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

9 de agosto de 2017

Capítulo 1

Instalação do Python e bibliotecas

A linguagem de programação Python possui bibliotecas que ajudam a escrever códigos voltados para aplicações em diferentes áreas de conhecimento. Uma dessas áreas é a mineração de dados. Esta apostila tem por objetivo ilustrar como bibliotecas da linguagem Python podem ser utilizadas para a realização de experimentos de mineração de dados.

Todos os exemplos a serem apresentados neste material utilizarão a linguagem de programação Python na versão 3 e algumas de suas principais bibliotecas desenvolvidas para dar suporte a análise de dados, aprendizado de máquina e mineração de dados. Portanto, esta primeira parte apresenta um breve tutorial sobre como instalar essas ferramentas em diferentes sistemas operacionais, Linux e Windows.

1.1. Instalação em Linux

Diversas distribuições atuais do sistema operacional Linux disponibilizam, como parte de seu conjunto padrão de pacotes, um ambiente para programação na linguagem Python 3. Para conferir, basta digitar os seguintes comandos no terminal:

```
$ which python3
```

e

```
$ which pip3
```

os quais devem retornar algo como `/usr/bin/python3` e `/usr/bin/pip3`, respectivamente. Caso algum erro seja informado, deverão ser instalados os pacotes apropriados para que seja possível utilizar a linguagem. Para isso, será utilizado o gerenciador de pacotes disponível na distribuição usada. Os dois gerenciadores mais comumente utilizados são o apt-get (Debian, Ubuntu e derivados) e o yum (RedHat, CentOS e derivados). Portanto, para a instalação em sistemas derivados deles, basta digitar algum dos seguintes comandos no terminal:

```
$ sudo apt-get install python3 python3-dev python3-pip
```

ou

```
$ sudo yum install python3 python3-dev python3-pip
```

Após digitar esses comandos, toda vez que você quiser executar algum *script*, basta digitar o comando `python3 script.py`.

Adicionalmente, nos exemplos apresentados neste material e para os trabalhos aplicados distribuídos no decorrer da disciplina, serão necessárias quatro bibliotecas amplamente utilizadas para relizar experimentos de mineração e ciência de dados:

- NumPy[1],
- SciPy[2]
- scikit-learn[3]
- matplotlib[4]

Para a instalação dessas bibliotecas, basta digitar no terminal o comando a seguir:

```
$ pip3 install numpy scipy sklearn matplotlib pandas --user
```

1.2. Instalação em Windows

Como primeiro passo, a versão mais atualizada e estável do Python 3 deve ser baixada em `https://www.python.org/downloads/windows/`. Para a instalação do gerenciador de pacotes pip, o *script* get-pip.py[5] deve ser baixado e executado no terminal do Windows como `python3 get-pip.py`. Finalmente, utilizaremos o seguinte comando para a instalação das bibliotecas NumPy, SciPy, scikit-learn e matplotlib[6]:

```
$ pip3 install numpy scipy sklearn matplotlib pandas
```

1.3. Referências e tutoriais

Esta primeira parte deste material teve como objetivo descrever os passos necessários para a instalação das ferramentas que serão necessárias no decorrer da disciplina de Mineração de Dados Biológicos. Nos próximos capítulos, diversos exemplos de códigos serão apresentados para a exploração de conjunto de dados, visualização de dados, pré-processamento de bases de dados, construção de modelos preditivos, construção de modelos descritivos, além de outros temas que serão cobertos na disciplina. Não será necessário qualquer conhecimento prévio acerca das bibliotecas citadas nas seções anteriores. Entretanto, alguns bons tutoriais introdutórios estão disponíveis nos *websites* de cada biblioteca para que o leitor possa tanto consultar suas funcionalidades como aprender a usar melhor os seus recursos.

[1] `http://www.numpy.org/`
[2] `http://www.scipy.org/`
[3] `http://http://scikit-learn.org/stable/`
[4] `http://matplotlib.org/`
[5] `https://bootstrap.pypa.io/get-pip.py`
[6] Para o ambiente Windows, se os comandos `python3` e `pip3` não funcionarem, deve-se testar com os comandos `python` e `pip`, respectivamente.