

Introdução

Linguagem de Programação AplicadaSemana 1 / Parte 1

Prof. Alex Kutzke

26 de março 2022

Linguagem de Programação Aplicada

Objetivos da Disciplina

- Segundo a ementa:
 - “Programação em Python. Aplicações em Inteligência Artificial”;
- Nosso objetivo será:
 - Ser capaz de utilizar a linguagem Python, e seus recursos, como meio para a solução de problemas computacionais relacionados à IA;
- O que faremos, então:
 - Conhecer Python e seus recursos;
 - **Programar em Python**;
 - Trabalhar com problemas aplicados à IA.

Material e Recursos

- Material no moodle;
- Repositório GIT com o material da disciplina:
 - <https://gitlab.com/iaa003-alexkutzke/material>
- Material do curso é baseado nos seguintes livros:
 - GRUS, Joel - Data Science do Zero: Primeiras Regras com Python, Editora Alta Books, 1a Edição, 2016;
 - McKinney, Wes - Python para Análise de Dados: Tratamento de Dados com Pandas, Numpy e IPython, Editora Novatec, 1a Edição, 2019;

Avaliação

- Trabalhos práticos de programação em Python.

Por que Python para IA e Análise de Dados?

Python

- Linguagem interpretada;
- Criada por volta de 1991 por Guido van Rossum;
- Se tornou muito popular:
 - Muito para desenvolvimento web com o framework *Django*;
- Não é apenas uma linguagem de *scripting*:
 - É utilizada para criação de programas robustos;
- Legível e fácil de aprender

Python como Ferramenta para IA

- Comunidade muito ativa:
 - Tanto de programadores gerais;
 - Quanto programadores de análise de dados e processamento científico;
- Nos últimos 10 anos, suas bibliotecas de processamento científico deram um salto qualitativo;
- Hoje, Python é considerada uma das linguagens mais importantes em ciência de dados e IA.
- É comparável a outras linguagens “científicas” como: R, MATLAB, SAS, Strata e outras:
 - Porém, ao meu ver, é **muito** mais legível.

O problema das “duas linguagens”

- Em geral, na solução de problemas de IA, cria-se protótipos em uma linguagem de *scripting*, como R ou MATLAB;
- Na sequência, após testes, esses protótipos são reescritos em uma segunda linguagem (C, C++, Java, ...) para, então, se tornarem softwares aplicáveis;
- Python possui a vantagem de resolver o problema com apenas uma linguagem:
 - Seus programas são robustos e portáteis;
 - Há uma fácil e eficiente integração com códigos em C, C++ e FORTRAN;

Onde o Python perde?

- Aplicações que necessitam de desempenho “máximo”:
 - Análise de grande quantidade de dados em tempo real;
 - Python é uma linguagem interpretada e isso sempre gera custos;
- Aplicações que necessitam de comportamento multithreaded:
 - Rotinas altamente concorrentes, limitadas por CPU (*CPU-bound*);
 - GIL (*global Interpreter Lock*);

- É possível contornar o problema, mas dificultam a implementação e o uso de objetos Python.

Quais bibliotecas utilizaremos?

- Numpy: processamento numérico, principalmente vetorial;
- pandas: manipulação de dados eficiente e facilitada;
- matplotlib: visualização de dados;
- scikit-learn: kit de ferramentas de propósito geral para aprendizado de máquina;
- E algumas outras ...

Ambiente de Desenvolvimento

Anaconda

- Para facilitar, utilizaremos a Distribuição Python chamada Anaconda:
 - <https://www.anaconda.com/>
 - Uma espécie de plataforma para Data Science
 - Instala um ambiente python especial (separado) com as bibliotecas mais relevantes e outros recursos;
 - É obrigatório?
 - * Não? Eu particularmente uso Vim;
 - * Porém, facilitará bastante nossa vida na disciplina;

Anaconda permite ambientes virtuais

- Alternativa para venv e Virtualenv:

```
# create a Python 3.6 environment named "dsfs"
conda create -n dsfs python=3.6
```

Python 2 ou Python 3 ?

- Utilizaremos a versão 3 da linguagem Python;
- Essa versão (3.x) foi lançada em 2008:
 - Há mais de 10 anos!
- Por não possuir compatibilidade com a versão 2, por muito tempo, bibliotecas só funcionavam na versão anterior;
- Porém, atualmente, praticamente todos os códigos já foram atualizados;
- Além disso, a versão 2.x terá seu encerramento em 2020.

IDEs e Editores de Texto

- É possível editar programas Python em diferentes IDEs e editores de texto;
- Os mais famosos:

- PyDev: IDE incluído na plataforma Eclipse;
- PyCharm: um dos mais utilizados. Possui compatibilidade com Anaconda;
- Python Tools para Visual Studio;
- Spyder: atualmente acompanha o Anaconda;
- Além de editores de texto:
 - Sublime Text;
 - Atom;
 - Vim!

Ferramentas Interessantes

- Duas ferramentas que acompanham o Anaconda são bastante interessantes:
 - IPython;
 - Jupyter Notebooks.

IPython

- Prompt e interpretador de comando Python com mais recursos:

```
$ ipython
Python 3.6.0 | packaged by conda-forge | (default, Jan 13 2017, 23:17:12)
Type "copyright", "credits" or "license" for more information.
```

```
IPython 5.1.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: %run hello_world.py
Hello world
```

```
In [2]:
```

Jupyter Notebooks

- Aplicação web open-source;
- Permite criar e compartilhar documentos com códigos executáveis, equações, visualizações e textos explicativos;
- Como relatórios “vivos” para ciência de dados e IA;

```
$ jupyter notebook
[I 15:20:52.739 NotebookApp] Serving notebooks from local directory:
/home/wesm/code/pydata-book
[I 15:20:52.739 NotebookApp] 0 active kernels
[I 15:20:52.739 NotebookApp] The Jupyter Notebook is running at:
```

```
http://localhost:8888/  
[I 15:20:52.740 NotebookApp] Use Control-C to stop this server and shut down  
all kernels (twice to skip confirmation).  
Created new window in existing browser session.
```

Referências

- GRUS, Joel - Data Science do Zero: Primeiras Regras com Python, Editora Alta Books, 1a Edição, 2016;
- McKinney, Wes - Python para Análise de Dados: Tratamento de Dados com Pandas, Numpy e IPython, Editora Novatec, 1a Edição, 2019;