***** Introduction to Python for ML *****

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2020/2021

DHBW
Duale Hochschule
Baden-Württemberg

Agenda for this Unit

Introduction	**3**
What is Python?	3
Installation and Components	5
Writing and running Python Scripts	7
Data Types	**9**
Basic Data Types	9
Conversion between Types	11

Introduction

What is Python?

- Python was developed by Guido van Rossum in the early 1990s.
- It is an interpreted, high-level, general-purpose programming language which is the state-of-the-art for data science and machine learning.
- **White space has a meaning!** (important difference to other languages like Java or C++)
  - Semantic code blocks (e. g. conditions, loops, ...) have to be indented.
  - You may use tabs or spaces for indentation, but the interpreter **does not allow you to mix these two!**
- Python supports many programming paradigms: We can write **procedural**, **functional** or **object-oriented** programs.

- **Caveat:** Python has two **non-compatible** branches:
  - `Python 2.7` ⟸ This one is deprecated and no longer maintained
  - `Python 3.x` ⟸ Use this one!

Installation and Components

Python

- Python is pre-installed on Mac and Linux. You may check this by executing `python -V` in the command line.
- For Windows (and Mac), download the newst version from ⇒ www.python.org/downloads/.
- Don't forget to check the '`Add Python 3.x to PATH`' option!

Jupyter

- It is a browser-based Python environment (either on local or remote machine).
- The code is editable directly in the browser.
- Nice for (interactive) demonstrations, **but hard to debug**.

Anaconda

- A package manager for creating **isolated Python environments** (with different package versions)
- It is good programming practice to create separate environments for different projects in order to avoid version mismatches, package clashes etc.
- It can be downloaded from ⇒ https://www.anaconda.com/download/

I recommend to use the *Anaconda Navigator* which combines all components, including a Python IDE called *spyder*.

Writing and running Python Scripts

```python
print("Hello world!") # print to console
i = 0
if i == 0:
    print("Success")
```

Data Types

Basic Data Types

- Unlike Java, Python is **typed dynamically**, i. e. the data type does not have to be specified when declaring variables.
- Nevertheless, Python works with these types internally (cf. ⇒ table 1).

Type	Description	Example
`int`	Integer (no maximum for integers, only limited by memory size)	0, 1, 2, ...
`float`	Floating point number	3.1415, 2.7182
`str`	String (unicode characters)	'Hello world'
`bool`	Boolean values	`True` or `False`

Table 1: Basic data types in Python

Some examples

```python
# this declares a variable s initialized with value "Hello"
s = 'Hello'

# you may also use double quotes
s = "Hello"

# indexing is possible (starts at 0)
s = "Hello"[1] # gives "e"

# indexing from the back
s = "Hello"[-1] # gives "o"

# dynamic typing: now you can use the same variable to save numbers;
# keep track of what you do with your variables!
s = 0.50**2 * 3.1415
```

Conversion between Types

- Although you don't have to bother with types in Python, you may want to convert a variable to another type.
- E. g. when concatenating a number to a string

```python
# conversion between types
int("42") # gives 42
str(42) # gives "42"

# example:
# this does NOT work (TypeError: can only concatenate str to str)
print("The answer is " + 42)

# use conversion:
print("The answer is " + str(42))
# or alternatively:
print("The answer is", 42)
```

Thank you very much for the attention!

Topic: ***** Introduction to Python for ML *****
Term: Winter term 2020/2021

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?