Introduction to Programming and Data Analysis Modules 1 & 2

Course Introduction

Outline

1. **Course introduction**

2. **Sneak preview of Module 1**

3. **Sneak preview of Module 2**

Let’s play a game on Kahoot!

- Using your smartphone or a second display
- Visit www.kahoot.it
- Type the given code

Our anti-motto: "But it works . . ."

“Can You Learn To Ski Without Lessons?”

https://www.skibro.com/blog/en/can-you-learn-to-ski-without-lessons/

Most of the times you get to the valley.

The problem is how you get there . . .

Course Responsibles

- Andrea Vandin, andrea.vandin@santannapisa.it
  - ★ Tenure-track Assistant Professor in Computer Science at Institute of Economics & EMbeDS @ SSSA
  - ★ PhD in Computer Science @ IMT Lucca, MSc in Computer Science @ Unipi
  - ★ Formerly:
    - ▶ Associate Professor in Computer Science at DTU Denmark
      - Most related teaching activity: responsible for 3 years of course 'Programming in C++ for non-computer scientists', 250 students
      - Had a 1.5 years course on effective university teaching.
- Daniele Licari, daniele.licari@santannapisa.it
  - ★ Master in Data Science @ Unipi, PhD in Computational Chemistry @ Scuola Normale Superiore, MSc in Computer Science @ Unipi
  - ★ EMbeDS Senior Data Scientist @ SSSA
  - ★ Academic & Industrial Experience on Python and ML

Note on the 2-modules structure

2-modules structure: http://bit.ly/IProDASSSA2021

The course is organized in two modules. Intuitively

- M1: Module 1 focuses on programming
  - ★ M1 core: 17 hours
  - ★ M1&M2: 6 hours
- M2: Module 2 focuses on data analysis and machine learning
  - ★ M1&M2: 6 hours
  - ★ M2 core: 13 hours

M1 gives the necessary background for M2

Course References & Material

- Webpages of the course:
  - ★ http://bit.ly/IProDASSSA2021
    - ▶ Slides and examples from the lectures, further material and links
    - ▶ Self-testing coding exercises
    - ▶ Results of gamified recap quizzes
- Suggested books:
  - ★ M. Lutz, Learning Python;
  - ★ W. McKinney, Python for Data Analysis.
  - ★ E.Duchesnay, T.Löfstedt, F.Younes, Statistics and Machine Learning in Python
- Well-done tutorial: https://docs.python.org/3/tutorial/
- Software
  - ★ Python: https://www.python.org/
  - ★ Python editor: JupyterLab https://jupyter.org/
  - ★ Setup your machine: http://bit.ly/IProDASSSA2021

Course Description

Module 1

introduces the fundamental principles of (object-oriented) structured programming with applications to data processing. Using Python, the course starts from basic notions of programming (variables, data types, collections, repetition structures, functions & modules), up to data processing functionalities (loading, manipulation, and visualization of CSV data) with advanced libraries.

Module 2

introduces the typical components of data analysis processes. It first introduces popular Python libraries for data manipulation and visualization (NumPy, Pandas, Seaborn). Then treats a case study on breast cancer classification (data preprocessing, dimensionality reduction, clustering, and classification). We also present KNIME, a popular Python-integrated graphical language for data analysis.

Learning Objectives

Module 1

Participants will acquire an understanding of the issues involved in *good* computer programming, to be able to make informed decisions. Participants will be able to write simple to medium python programs of various nature, including those for reading, manipulating and visualizing data.

Module 2

Participants will acquire an understanding of the issues involved in *good* data analysis, to be able to choose the appropriate data pre-processing task for the considered analysis. Participants will be able to perform simple to medium data analysis and machine learning tasks using Python and KNIME.

Learning Objectives

Module 1

- select and use the correct data types and collections for the problem at hand
- use and describe variables, operations, and control strctures (if, loops)
- create and use functions and classes
- use libraries for I/O, data manipulation, and data visualization
- use principles of structured program design and methods

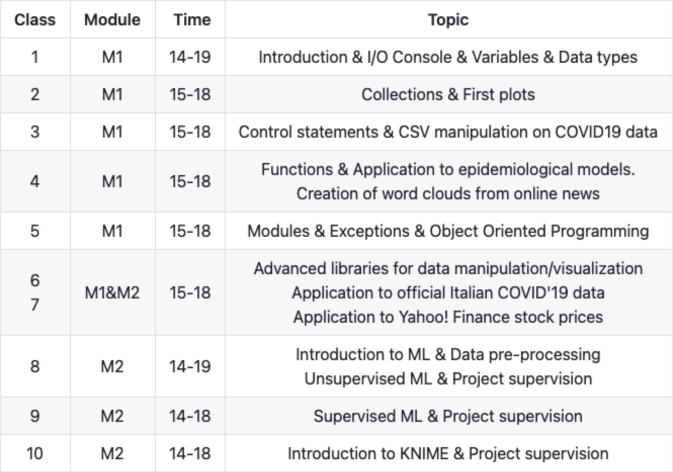
Module 2

- use advanced Python libraries for data processing and visualization
- select and use the correct data pre-processing tasks for the given analysis
- apply data dimensionality reduction techniques
- apply and explain clustering techniques
- create, train, evaluate, and improve classifiers in Python and KNIME

Self-Evaluation

- Each class starts with a recap gamified quiz on kahoot.com
- Regular coding exercises for M1
  - Automatic tests for your code and hints to fix bugs
  - Locally or on the cloud: Google COLAB http://bit.ly/IProDAcolab
    - Zero configuration required
    - Free access to GPUs
    - Easy sharing
  - A fundamental learning tool of this course
- Project for M2
  - You can evaluate your learning process on a simple project
    - Would you have survived the titanic shipwreck?
  - A fundamental learning tool of this course

Lecture Plan



Class	Module	Time	Topic
1	M1	14-19	Introduction & I/O Console & Variables & Data types
2	M1	15-18	Collections & First plots
3	M1	15-18	Control statements & CSV manipulation on COVID19 data
4	M1	15-18	Functions & Application to epidemiological models. Creation of word clouds from online news
5	M1	15-18	Modules & Exceptions & Object Oriented Programming
6 7	M1&M2	15-18	Advanced libraries for data manipulation/visualization Application to official Italian COVID'19 data Application to Yahoo! Finance stock prices
8	M2	14-19	Introduction to ML & Data pre-processing Unsupervised ML & Project supervision
9	M2	14-18	Supervised ML & Project supervision
10	M2	14-18	Introduction to KNIME & Project supervision

Further info

Prerequisities

- No prerequisites for M1
- M2 requires experience in programming in Python

Hands-on course

- You will never learn programming if you don't practice it
- You will never learn data analysis if you don't practice it
- We offer a rich set of e-learning features for this reason

Ideas for an Effective Course

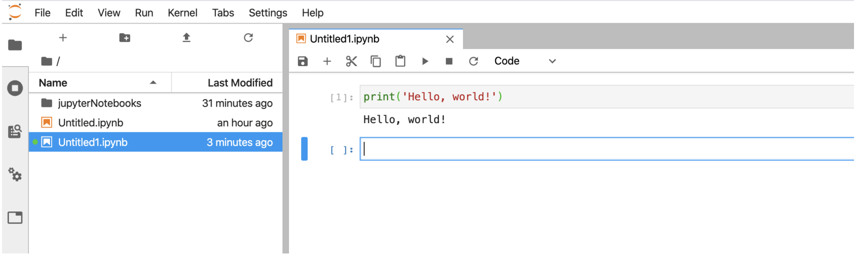
Live Programming & Examples & Exercises

We mostly have blocks of 3 hours.

- First part:
  Intro to class's topics & Live programming
  - ★ No slides!
  - ★ Instead: we develop a few example programs
    - ▶ http://bit.ly/IProDASSSA2021
    - ▶ Please have your laptop ready!
    - ▶ You find code in advance
- Second part:
  - ★ You consolidate your learning working on the exercises or project
    - ▶ With our support
  - ★ We discuss complex examples

Live Programming

Find the JupyterLab notebooks at http://bit.ly/IProDASSSA2021



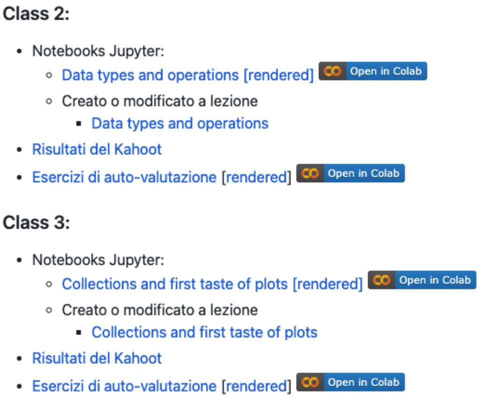
Name	Last Modified
jupyterNotebooks	31 minutes ago
Untitled.ipynb	an hour ago
Untitled1.ipynb	3 minutes ago

```
[1]: print('Hello, world!')

     Hello, world!

[ ]:
```

Exercises on COLAB



Class 2:

- Notebooks Jupyter:
  - Data types and operations [rendered] CO Open in Colab
  - Creato o modificato a lezione
    - Data types and operations
- Risultati del Kahoot
- Esercizi di auto-valutazione [rendered] CO Open in Colab

Class 3:

- Notebooks Jupyter:
  - Collections and first taste of plots [rendered] CO Open in Colab
  - Creato o modificato a lezione
    - Collections and first taste of plots
- Risultati del Kahoot
- Esercizi di auto-valutazione [rendered] CO Open in Colab

- We will upload notebooks with exercises and tests.
- You can solve them on COLAB

Outline

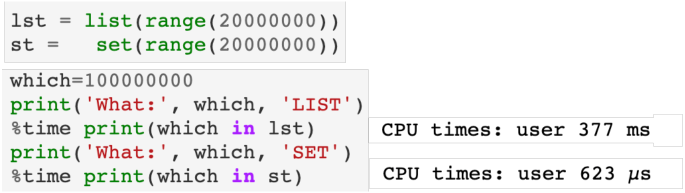
1. Course introduction

2. **Sneak preview of Module 1**

3. Sneak preview of Module 2

Sneak preview of Module 1

Which data structure should I use? It depends!



```
lst = list(range(20000000))
st =   set(range(20000000))
```

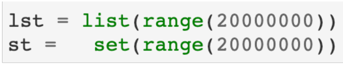
```
which=100000000
print('What:', which, 'LIST')
%time print(which in lst)
print('What:', which, 'SET')
%time print(which in st)
```

```
CPU times: user 377 ms
```

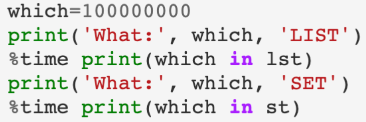
```
CPU times: user 623 µs
```

Sneak preview of Module 1

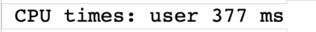
Which data structure should I use? It depends!



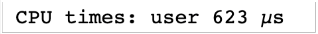
```
lst = list(range(20000000))
st =   set(range(20000000))
```



```
which=100000000
print('What:', which, 'LIST')
%time print(which in lst)
print('What:', which, 'SET')
%time print(which in st)
```

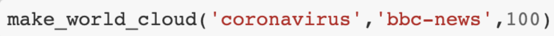


```
CPU times: user 377 ms
```

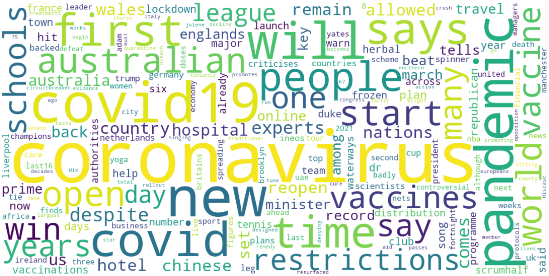


```
CPU times: user 623 µs
```

What are the most used words in news?

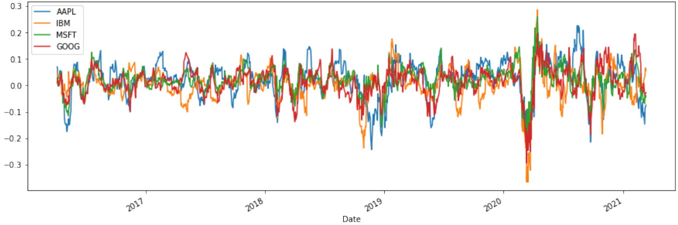


```
make_world_cloud('coronavirus','bbc-news',100)
```



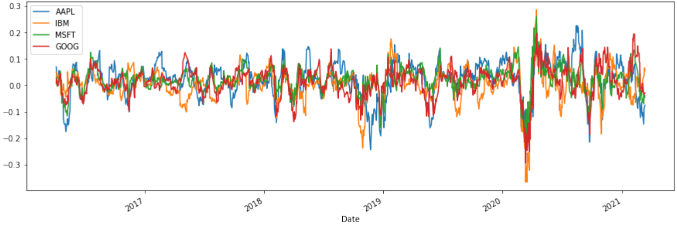
Sneak preview of Module 1

How is the stock market performing?

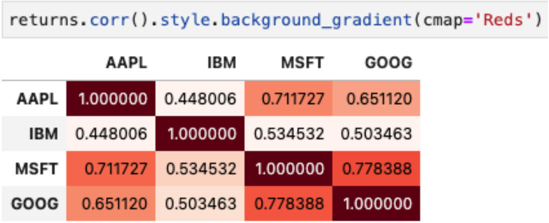


Sneak preview of Module 1

How is the stock market performing?



Do I have correlations in my data?

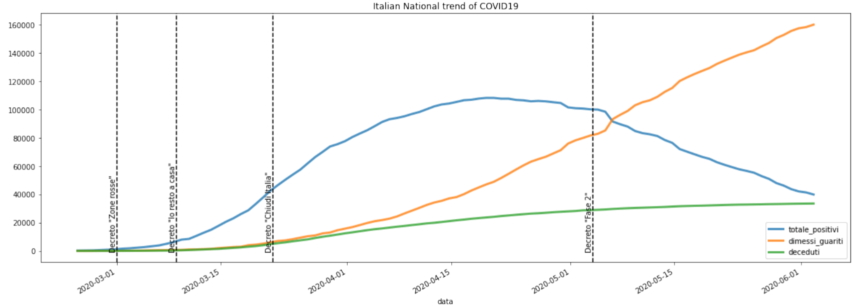


```
returns.corr().style.background_gradient(cmap='Reds')
```

	AAPL	IBM	MSFT	GOOG
AAPL	1.000000	0.448006	0.711727	0.651120
IBM	0.448006	1.000000	0.534532	0.503463
MSFT	0.711727	0.534532	1.000000	0.778388
GOOG	0.651120	0.503463	0.778388	1.000000

Sneak preview of Module 1

The pandemic and the Italian decrees



Outline

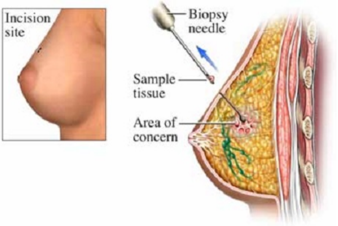
1. Course introduction

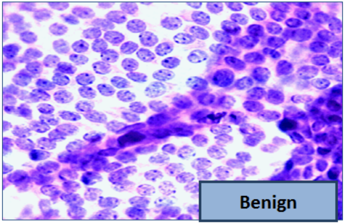
2. Sneak preview of Module 1

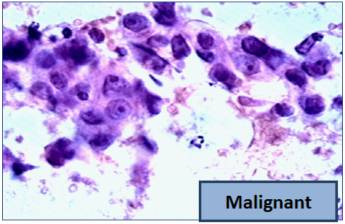
3. **Sneak preview of Module 2**

Sneak preview of Module 2

Starting from the competences developed in the first module, we will study how to apply data analysis techniques from Machine learning







Can we classify them automatically?

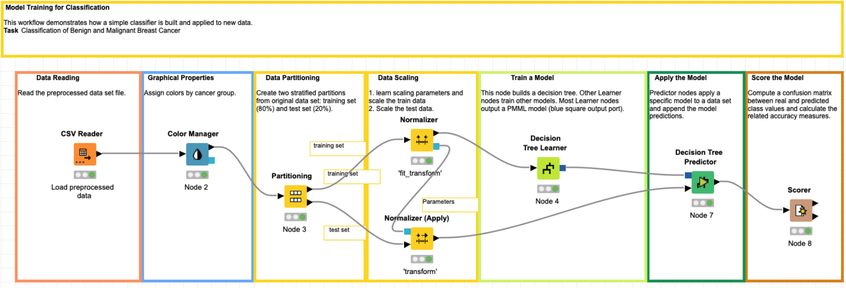
Sneak preview of Module 2

We will go through a classic pipeline for these data analysis tasks

- with emphasis on data pre-processing.

We will use two alternative approaches

- Python: **main focus**
- Knime: a graphical workflow language



Model Training for Classification

This workflow demonstrates how a simple classifier is built and applied to new data.
Task Classification of Benign and Malignant Breast Cancer

Data Reading

Read the preprocessed data set file.

Graphical Properties

Assign colors by cancer group.

Data Partitioning

Create two stratified partitions from original data set: training set (80%) and test set (20%).

Data Scaling

1. learn scaling parameters and scale the train data
2. Scale the test data.

Train a Model

This node builds a decision tree. Other Learner nodes train other models. Most Learner nodes output a PMML model (blue square output port).

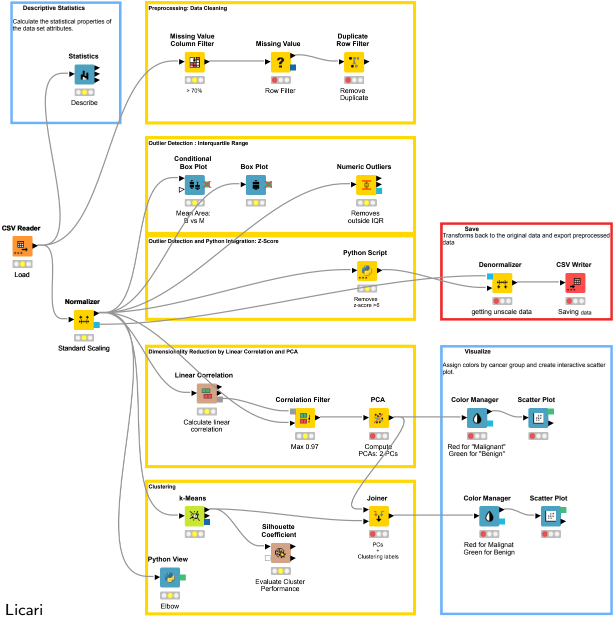
Apply the Model

Predictor nodes apply a specific model to a data set and append the model predictions.

Score the Model

Compute a confusion matrix between real and predicted class values and calculate the related accuracy measures.

Sneak preview of Module 2



Descriptive Statistics

Calculate the statistical properties of the data set attributes.

Preprocessing: Data Cleaning

Outlier Detection : Interquartile Range

Outlier Detection and Python Integration: Z-Score

Save

Transforms back to the original data and export preprocessed data

Dimensionality Reduction by Linear Correlation and PCA

Visualize

Assign colors by cancer group and create interactive scatter plot.

Clustering