1-Introduction

Data Science Practicum II 2021/22, Lesson 1

Marko Tkalčič

Univerza na Primorskem

Table of Contents

Course

- Data Science Practicum II
- Lectures in English
- 6 ECTS
- 180 hours
  - 60 Labs
  - 120 Home Work
- English
- Marko Tkalčič
  - Lecturer and TA
  - marko.tkalcic@famnit.upr.si
  - https://www.famnit.upr.si/sl/zaposleni-in-sodelavci/marko.tkalcic/
  - Office hours by email appointment (I/1, Kettejeva 1, Koper)

Schedule

Wednesday: 15.00-18.00

Goal of the Course

The students will be able to devise a data science project pipeline from importing data to evaluation.

Syllabus

1. Data import,
2. Tidying the data,

loop:

3. Transforming the data,
4. Visualizing the data,
5. Modelling the data,

end of loop

6. Communicate the findings.

Structure

- Python 3.7+ (the kernel)
  - how to check version?
  - cmd:
    - `python3 -V` (capital V)
    - `python3 --version`
- Jupyter Notebook (the IDE)
  - local installation
  - run in the terminal/cmd: `jupyter notebook`
  - https://colab.research.google.com

Structure

- Python 3.7+ (the kernel)
  - how to check version?
  - cmd:
    - `python3 -V` (capital V)
    - `python3 --version`
- Jupyter Notebook (the IDE)
  - local installation
  - run in the terminal/cmd: `jupyter notebook`
  - https://colab.research.google.com

- Short lecture
- Hands-on exercises
- Discussion
- Weekly assignments
- Final Kaggle competition

Table of Contents

Assessment

- 50% weekly assignments
- 50% project work (Kaggle competition)
  - thresholds
  - ranking

Weekly assignments

- small programming assignment
- upload ZIP file of the Jupyter notebook to Moodle
- due date: Tuesday, 23:55

Kaggle Competition

- concrete task will be provided in a couple of weeks
- Kaggle competition
  - I will provide the dataset
  - you will develop a Data Science solution and submit it to **Kaggle**, not Moodle
  - several submissions possible (time-wise)
  - only one submission stream per student
  - no teams
- ranked list based on performance of the submission
- Assessment based on:
  - passing thresholds
  - ranking (all can get a 10)

Organizing data science projects

- general rule: store all relevant files under one directory
  - exception: if you have shared scripts, they can be outside
- folder structure
  - data
  - scripts
  - results
  - doc
  - misc

Organizing data science projects

- data:
  - raw: read only/never change
  - derived: cleaned versions of raw data
- derived: data should be stored chronologically in additional directories, e.g. YYYYMMDD_cleaned
- results:
  - results should be stored chronologically, e.g. YYYYMMDD_results
  - notebook.md:
    - entries should be dated and relatively verbose
    - describe what you did
    - add results/outcomes/images/tables etc.
    - notes, observations, conclusions etc. for future work
    - paste emails, conversations etc.
    - learn MarkDown, e.g. https://daringfireball.net/projects/markdown/syntax

Organizing data science projects

- scripts
  - have a runall script (depends on the language used)
  - comment generously
  - store intermediate results
  - have the scrip(s) restartable, i.e. `if (<output file not exist>) then <perform operation>`
  - script types:
    - runall: one or two such scripts per directory
    - single-use script: for a single use, e.g. for cleaning the data
- version control
  - scripts
  - result data (not figures)

Participation in User Studies

- University staff conducts experiments with users
- If you participate, each such experiment adds 5%
- Does not count into requirements
  - i.e. you cannot accumulate experiment points to compensate for failed exam
  - you need to pass the exam separately

Table of Contents

Moodle

- All the material for the course will be available on-line in Moodle
- Communication:
  - Student -> Prof: marko.tkalcic@famnit.upr.si
  - Prof -> Students: e-classroom

Communication etiquette

I had a couple of bad experiences in previous years.

How to communicate via email?

- no slang
- **neutral attitude**
- structure
  - clear subject
  - greeting
  - description of yourself
  - description of the issue
  - what have you done to address the issue so far
  - what do you expect from me
  - wrap up

Communication etiquette

Good Example

Subject: issue with XML parsing

Dear Dr. Tkalcic,

I am Robert Zimmerman, first year BSc student attending the course Programming Project (student ID 123456789).

While doing the home assignment on XML parsing I came across an issue I was unable to solve. When loading the data file and applying the XXX method, the java compiler yields the error "XXX". I tried using the AAA parser and the BBB parser, always with the same outcome. I searched on stackoverflow.com but couldn't find an appropriate solution.

I would be grateful if I could come to your office during the office hours to discuss the issue.

Thank you in advance.

Kind regards,

Robert Zimmerman

Communication etiquette

Bad example - missing info

From: mare89@hotmail.com
To: marko.tkalcic@gmail.com
Subject:exam

hi, whats the score of my exam

Communication etiquette

Bad example - neutral attitude

Dear Tkalcic Marko,

my name is AAA BBB and I am unable to understand how I can get 0 points for my Programming Project project. I am not willing to accept "0 Does not run, not even from console, instructions insufficient to run it (NullPointerException)" as an excuse to hand me 0 points for all that work. I wrote over 3000 lines of code for this project and included each one of the seven techniques. Sorry if this sounds like an insult for you; but are you kidding me?? Do you really think I invest that much time and work in a program that can't be run?! Do you think I wrote all these lines without opening the program a single time? I opened the program more than a hundred times now and have never gotten that so called "NullPointerException". I am really disappointed and angry to get such a message and hope everything is only a misunderstanding. I would include images of my running program in this mail but I guess this won't make a difference because the proof of my running program is already in the report where I put pictures to describe it. Waiting for your reply.

AAA BBB.

Communication etiquette

Bad example - slang

Dear Marko,

sry for the inconveniece and ty for pointing out my problem so that I am hopefully not going to make the same mistake again.

Best Regards,

AAA BBB

Questions?

Table of Contents

Exercise 1

- create a project in Jupyter
- create a new Python 3 notebook
- give it a name

Exercise 2

- the notebook is composed of **cells**
  - code cells [In]
    - contains code
    - one or multiple lines
    - output areas [Out]
- write in the first cell

```
print("Hello, world!")
```

- run it `Shift+Enter`

Exercise 2

- the notebook is composed of **cells**
  - code cells [In]
    - contains code
    - one or multiple lines
    - output areas [Out]
- write in the first cell

```
print("Hello, world!")
```

- run it `Shift+Enter`

- cell:
  - green (edit mode)
  - blue (command mode)

Exercise 3

- in the next cell write the following

```
a = 5
b = 6
a+b
```

Exercise 3

- in the next cell write the following

```
a = 5
b = 6
a+b
```

```
Out[3]: 11
```

Exercise 3

- in the next cell write the following

```
a = 5
b = 6
a+b
```

```
Out[3]: 11
```

- get the result from the previous cell

```
_*10
```

Exercise 4

- create a **markdown** cell
  - M : change cell to markdown
  - Y : change cell to code
- use markdown to create the following elements in the md cell:
  - H1, H2, H3
  - italic
  - bold
  - link
  - image
  - syntax-highlighted code
  - Latex equation
- run it

Exercise 4

```
# This is H1
some *italic* text
## this is H2
some **bold** text
### this is H3
some text with a [link](http://markotkalcic.com)
an equation:
$$m(X) = {\sum_{i=1}^I x_i \over N}$$
an image
![Caption](../data/derived/surprise_left_right.png)
and some syntax-highlighted code in Python
\```python
print('Hello, world')
\```
and in Java
\```java
public void catchMeIfYouCan(Integer a){
    System.ou.println(a);
}
\```
```

Exercise 4

```
# This is H1
some *italic* text
## this is H2
some **bold** text
### this is H3
some text with a [link](http://markotkalcic.com)
an equation:
$$m(X) = {\sum_{i=1}^I x_i \over N}$$
an image
![Caption](../data/derived/surprise_left_right.png)
and some syntax-highlighted code in Python
\```python
print('Hello, world')
\```
and in Java
\```java
public void catchMeIfYouCan(Integer a){
    System.ou.println(a);
}
\```
```

- double click in edit mode to get back to markdown source
- learn markdown and latex

Exercise 5 - Notebook GUI

- User interface tour

Exercise 6 - Shortcuts

- switch between edit and command mode `Enter <--> Esc`
- Command Mode:
  - F : find and replace
  - Enter : enter edit mode
  - Shift+Enter : run cell, select below
  - Y : change cell to code
  - M : change cell to markdown
  - R : change cell to raw
  - A : insert cell above
  - B : insert cell below
  - Z : undo cell deletion
  - D,D : delete selected cells
  - Ctrl+S : Save and Checkpoint
  - S : Save and Checkpoint
  - I,I : interrupt the kernel
  - 0,0 : restart the kernel (with dialog)

Exercise 7 - Shell Commands

- we can use shell commands
- exclamation mark!
  - `!ls`
  - `!dir`

Exercise 7 - Shell Commands

- we can use shell commands
- exclamation mark!
  - `!ls`
  - `!dir`

- create an empty file called `test.md`
- list the files in the folder

Exercise 7 - Shell Commands

- we can use shell commands
- exclamation mark!
  - `!ls`
  - `!dir`

- create an empty file called `test.md`
- list the files in the folder

```
!touch test.md
!ls

!type nul > test.md (win)
!dir
```

Exercise 8 - Notebook Format

- through your file browser (not in Jupyter) locate the notebook file
- extension `ipynb` (IPYthonNoteBook)
- open it with a text editor and inspect it. What do you observe?

Exercise 8 - Notebook Format

- through your file browser (not in Jupyter) locate the notebook file
- extension `ipynb` (IPYthonNoteBook)
- open it with a text editor and inspect it. What do you observe?

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "scrolled": true
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Hello world\n"
     ]
    }
   ],
   "source": [
    "print(\"Hello world\")"
   ]
  },
```

- export it to Python, HTML, PDF. Observe the outputs.

Exercise 9

- create a project in Jupyter
- create three folders
  - data
  - scripts
  - results
- create a new Python 3 notebook
- give it a name

Exercise 10/1

- copy the `timespent.csv` file into the data folder
- write the following in a new code cell and run it

```
import pandas as pd
ts = pd.read_csv('../data/derived/timespent.csv')
```

Exercise 10/1

- copy the `timespent.csv` file into the data folder
- write the following in a new code cell and run it

```
import pandas as pd
ts = pd.read_csv('../data/derived/timespent.csv')
```

- write and run

```
ts.head()
```

Exercise 10/1

- copy the `timespent.csv` file into the data folder
- write the following in a new code cell and run it

```
import pandas as pd
ts = pd.read_csv('../data/derived/timespent.csv')
```

- write and run

```
ts.head()
```

Out[7]:

	Unnamed: 0	session_id	song1	song2	slider	song1_listening_time	song2_listening_time	listesning_time_delta
0	0	01btp0ejkjh95vpdeb6e4kf7gk49tmin	1.0	165.0	0.0	132.0	0.0	132.0
1	1	01btp0ejkjh95vpdeb6e4kf7gk49tmin	27.0	49.0	-2.0	182.0	149.0	33.0
2	2	01btp0ejkjh95vpdeb6e4kf7gk49tmin	37.0	21.0	2.0	171.0	195.0	-24.0
3	3	01btp0ejkjh95vpdeb6e4kf7gk49tmin	50.0	4.0	0.0	167.0	105.0	62.0
4	4	01btp0ejkjh95vpdeb6e4kf7gk49tmin	52.0	29.0	0.0	195.0	134.0	61.0

Exercise 10/2

- write and run

```
ts.describe()
```

Exercise 10/2

- write and run

```
ts.describe()
```

	Unnamed: 0	song1	song2	slider	song1_listening_time	song2_listening_time	listesning_time_delta
count	719.000000	719.000000	719.000000	719.000000	719.000000	719.000000	719.000000
mean	359.000000	92.823366	91.995828	0.044506	129.770515	126.025035	3.745480
std	207.701709	54.508676	56.265722	1.233569	70.640574	70.284871	47.372177
min	0.000000	1.000000	1.000000	-2.000000	0.000000	0.000000	-218.000000
25%	179.500000	45.000000	37.000000	-1.000000	66.000000	63.500000	-19.000000
50%	359.000000	88.000000	93.000000	0.000000	133.000000	129.000000	2.000000
75%	538.500000	136.500000	141.000000	1.000000	182.000000	176.000000	25.000000
max	718.000000	200.000000	200.000000	2.000000	426.000000	568.000000	233.000000

Exercise 11

- in a new cell write and run

```
ts.to_csv('../results/20181015_01_times.csv', sep='\t', encoding='utf-8')
```

Exercise 11

- in a new cell write and run

```
ts.to_csv('../results/20181015_01_times.csv', sep='\t', encoding='utf-8')
```

- check the results folder

References

Part of the material has been taken from the following sources. The usage of the referenced copyrighted work is in line with fair use since it is for nonprofit educational purposes.