Python for Data Science

Sirakorn Lamyai September 19, 2019

Student, Kasetsart U.

Obtain the material



http://bit.ly/cpe-datascience

About me



Sirakorn Lamyai

- · DAKDL Laboratory, Kasetsart University
- Research Assistant Intern, 2019, Vidyasirimedhi Institute of Science and Technology
- Research Assistant Intern, 2018, Vidyasirimedhi Institute of Science and Technology
- Love drinking tea
- Knows a little about Python

When I say I know a little about Python...

 \cdot I think there's some better methods than I'm using

- I think there's some better methods than I'm using
- · I think I do sometimes make mistakes

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes
- · There are tons of people who know things much more than me

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes
- · There are tons of people who know things much more than me
- I think there's much more for me to learn!

Prerequisite

A basic Python knowledge will do!

Your expectations from this talk

Outline

Data Science

Python

Python environments

Jupyter

Python Data Structures

Pandas

QAs

Data Science

· Obtain data from relevent sources

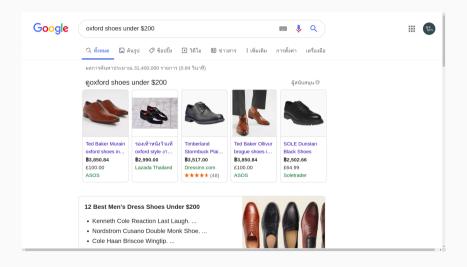
- · Obtain data from relevent sources
- · Scrub, sanitise, and clean the data into machine-understandable formats

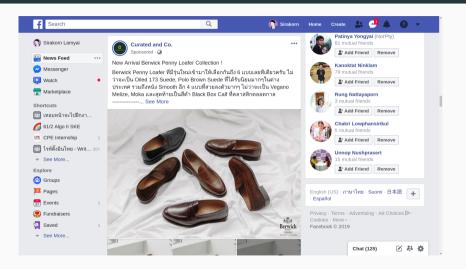
- · Obtain data from relevent sources
- · Scrub, sanitise, and clean the data into machine-understandable formats
- Explore significant and meaningful patterns with statistical methods

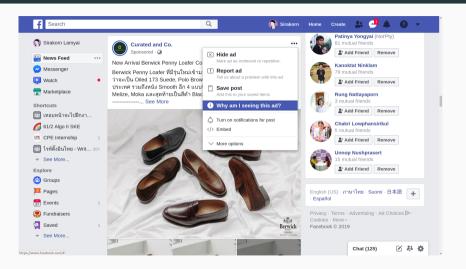
- · Obtain data from relevent sources
- · Scrub, sanitise, and clean the data into machine-understandable formats
- Explore significant and meaningful patterns with statistical methods
- \cdot Model construction for prediction and forecast

- · Obtain data from relevent sources
- · Scrub, sanitise, and clean the data into machine-understandable formats
- Explore significant and meaningful patterns with statistical methods
- Model construction for prediction and forecast
- iNterpret and use the results obtained

- · Obtain data from relevent sources
- · Scrub, sanitise, and clean the data into machine-understandable formats
- Explore significant and meaningful patterns with statistical methods
- Model construction for prediction and forecast
- iNterpret and use the results obtained
- Interate and rethink about your outputs









Data is the new oil

Tools for data analysis

With GUIs

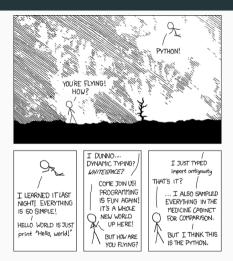
- Spreadsheets
 - Excel
 - Google Spreadsheets
 - · Lotus 1-2-3
- · Modelling and Visualisation
 - RapidMiner Studio
 - Weka
 - · Tableau

As programming languages

- For data insights
 - · R
 - Python
- · For data retrieval
 - · SQL

Python

Python



Courtesy: xkcd (https://xkcd.com/353/)

I loved Python...

- · Read it, understand it
- Multiparadigm
- · Batteris included
- · Lots of great, great libraries!

pip



 PyPA (Python Packaging Authority)'s recommended package installer



- PyPA (Python Packaging Authority)'s recommended package installer
- Obtains packages from PyPI (Python Packaging Index)



- PyPA (Python Packaging Authority)'s recommended package installer
- Obtains packages from PyPI (Python Packaging Index)
- · Many useful packages for us to use!

Anaconda Python Distribution

Anaconda Python Distribution



Cross-platform Python Distribution





- · Cross-platform Python Distribution
- Ships with its own package and environment manager



- · Cross-platform Python Distribution
- Ships with its own package and environment manager
 - Its environment manager capability is not found in Python vanilla installation



- · Cross-platform Python Distribution
- Ships with its own package and environment manager
 - Its environment manager capability is not found in Python vanilla installation
 - Fetches the packages from its own repository, not PyPI



- · Cross-platform Python Distribution
- Ships with its own package and environment manager
 - Its environment manager capability is not found in Python vanilla installation
 - Fetches the packages from its own repository, not PyPI
- Aims for Data Science use



- · Cross-platform Python Distribution
- Ships with its own package and environment manager
 - Its environment manager capability is not found in Python vanilla installation
 - Fetches the packages from its own repository, not PyPI
- · Aims for Data Science use
- Entirely separated Python

Environments 101: \$PATH

Different machines, different Pythons

On my laptop...

```
srakrn@epsilon-ubuntu:~$ which python
/home/srakrn/.pyenv/shims/python
```

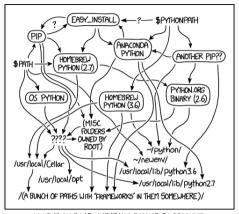
On my https://charles.srakrn.me/ server...

```
srakrn@charles:~$ which python
/usr/bin/python
srakrn@charles:~$ which python3
/usr/bin/python3
```

Installed pip

```
$ pip -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 2.7)
$ pip3 -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 3.6)
```

Perhaps now you understand me...



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Courtesy: xkcd (https://xkcd.com/1987/)

Jupyter





Interactive computing environment





 Thinks of a more dynamic coding environment.



- Thinks of a more dynamic coding environment.
- Inserts snippets of codes alternately with texts, maths, and images.



- Thinks of a more *dynamic* coding environment.
- Inserts snippets of codes alternately with texts, maths, and images.
- A wonderful tool for coding documented code.





Think of an online Jupyter Notebook provided by Google



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
 - In other words, your code are remotely executed

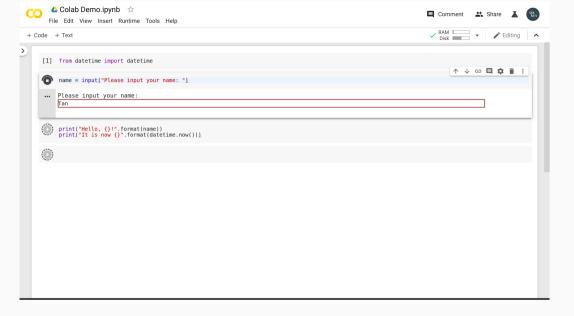


- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
 - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
 - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer
- · Free!

```
https:
//colab.research.google.com/
```



Caveats 1: Execution order



Caveats 1: Execution order

```
[2] a = 10

[1] a = 5

[3] print(a)

□ 10

↑ ↓ ⊕ ■ 章 :
```

You'll do a lot of out-of-order code execution!



```
[1] a = 10

[2] a *= 2

[4] print(a).

□ 60

↑ ↓ ⇔ □ ‡ :
```

You might sometimes remove a cell, and that shows no visible trace without explicit query.



```
[1] a = 5

[2] a = 20

[3] print(a).

□ 10

↑ ↓ ⊕ □ ☆ :
```

Jupyter Notebook offers no cell edited marks, while Colab offers them

Jupyter Notebook offers no cell edited marks, while Colab offers them (note: observe the greyed out cell number)

Caveats 3: Be neat and tidy

Jupyter Notebook and Colab, unlike IDE and code editors, offers a relatively poor **clean code** tools

- Syntax error highlighting
- Autocomplete
- Linting
- · Code formatter

Caveats 3: Be neat and tidy

Sirakorn's Workflow Demo

Python Data Structures

• **Lists** are a compilation of objects.

- **Lists** are a compilation of objects.
- · Can store multiple data types.

- **Lists** are a compilation of objects.
- · Can store multiple data types.
 - · This includes storing lists in a list

- **Lists** are a compilation of objects.
- · Can store multiple data types.
 - This includes storing lists in a list
 - · So-called a **nested list**

- · Lists are a compilation of objects.
- · Can store multiple data types.
 - This includes storing lists in a list
 - So-called a nested list
- · Can be resized.

- · Lists are a compilation of objects.
- Can store multiple data types.
 - This includes storing lists in a list
 - · So-called a **nested list**
- · Can be resized.
 - No need to declare its size on the first declaration.

```
a = [1, 2, 3, 4, 5]
a [0] # Accessing elements
a [1:3] # Slicing
```

Accessing list

```
a = [1, 2, 3, 4, 5]
a [0] # Accessing elements
a [1:3] # Slicing
```

Accessing list

```
a = [1, 2, 3, 4, 5]
a [0] # Accessing elements
a [1:3] # Slicing
```

Accessing list

• Elementwise: accessing one elements at a time)

```
a = [1, 2, 3, 4, 5]
a [0] # Accessing elements
a [1:3] # Slicing
```

Accessing list

- Elementwise: accessing one elements at a time)
- · Slicing: accessing a sublist

List Functions

```
vowels = ["a", "e", "o", "u"]
3 # Get a's length
4 len(a)
5 # Append the new element to the end of a
a.append("v")
7 # Deletes the first occurence of the element from a
8 a.remove("v")
9 # Inserts the item into a list with a specified index
10 a.insert(2, "i")
```

List Functions

```
vowels = [1, 3, 2, 5, 4]

# Get the first index of a specified element
a.index(4)

# Sort a list and store into a new list
sorted_a = sorted(a)

# Sort a list, making changes directly to the old one
a.sort()
```

Dictionary

```
_{1} names = {
     "Cherprang": "Cher",
     "Manipa": "Khamin",
     "Jiradapa": "Pupe"
7 kami nickname = names["Manipa"
```

Dictionary

```
_{1} names = {
     "Cherprang": "Cher",
     "Manipa": "Khamin",
     "Jiradapa": "Pupe"
7 kami nickname = names["Manipa"
```

 Dictionaries store values in a key-pair format.

Dictionary

```
names = {
     "Cherprang": "Cher",
     "Manipa": "Khamin",
     "Jiradapa": "Pupe"
7 kami nickname = names["Manipa"
```

- Dictionaries store values in a key-pair format.
- From the positional index, dictionary takes the key as an index instead.





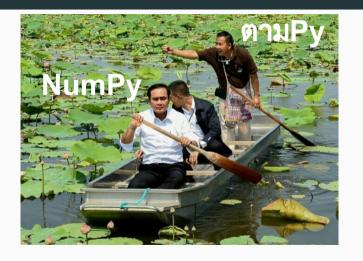
 NumPy is a powerful library for mathematical computation in Python



- NumPy is a powerful library for mathematical computation in Python
- It offers wide range of tools from data structures to advanced functions and operations



- NumPy is a powerful library for mathematical computation in Python
- It offers wide range of tools from data structures to advanced functions and operations
- A very strong ease in mathematical computation, don't reinvent the wheels!



Courtesy: Rebellious Professor,

Let's go to Notebook!

I'm too lazy to cover the contents twice...

Pandas

Pandas

$\mathsf{pandas}_{y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}}$















· Tabular-like structure









- · Tabular-like structure
- High performance









- · Tabular-like structure
- High performance
- Easy to use









- · Tabular-like structure
- High performance
- Easy to use
- · Helps a lot in data preparation









- · Tabular-like structure
- High performance
- Easy to use
- · Helps a lot in data preparation
- Considered as a wrapper for Numpy, although there's a lot more

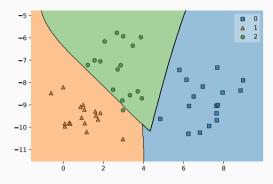
Series and DataFrame



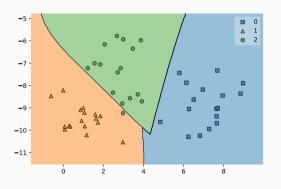
Let's go to Notebook!

Data Visualisation

Data Visualisation

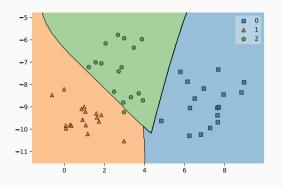


Data Visualisation



Why Visualise?

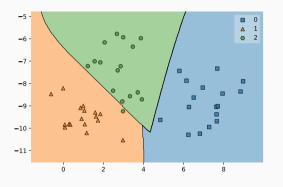
Data Visualisation



Why Visualise?

· Our visual system is so great!

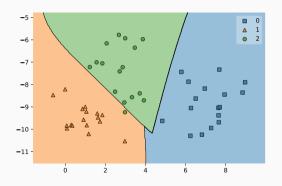
Data Visualisation



Why Visualise?

- · Our visual system is so great!
- A much better way to represent data than statistical values

Data Visualisation



Why Visualise?

- · Our visual system is so great!
- A much better way to represent data than statistical values
- Meaningful plots show meaningful insights without needing to do much

What if we don't visualise?

The Datasaurus Dozen

https://www.autodeskresearch.com/publications/samestats





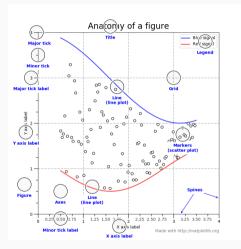
 feature-rich plotting library in python



- feature-rich plotting library in python
- plots multiple types of charts



- feature-rich plotting library in python
- plots multiple types of charts
- extensive support for jupyter notebook



Source: https://matplotlib.org/tutorials/introductory/usage.html

Let's go to Notebook!

QAs

Internships

Applications

Today's and tomorrow's importance of Data Science

Neural Networks?