CHULA ENGINEERING
Foundation toward Innovation

COMPUTER

Chula Big Data and IoT
Center of Excellence
(CUBIC)

DataMind
Data Mining Group
Machine Intelligence and Knowledge Discovery Lab
Chulalongkorn University

Overview

Python for Data Analytics

Peerapon Vateekul, Ph.D.
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University
Peerapon.v@chula.ac.th

Outlines

- Why Learn Python?
- Why Use Python?
- 7 Important Reasons Why You Should Use Python
- Python Packages
- Top 20 Python libraries for data science in 2018
- Introduction to Colaboratory (Google Colab)
- Reference

Why Learn Python?

- Python is a **general-purpose language**, which means it can be used to build just about anything, which will be made easy with the right tools/libraries.
- Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing.
- Many developers have also used Python to build productivity tools, games, and desktop apps, so there are plenty of resources to help you learn how to do those as well.

Reference: http://www.bestprogramminglanguagefor.me/why-learn-python

Why Learn Python? (cont.)

- **Beginner Friendliness**
  - **Easy to Understand**
    - Being a very high level language, Python reads like English, which takes a lot of syntax-learning stress off coding beginners. Python handles a lot of complexity for you, so it is very beginner-friendly in that it allows beginners to focus on learning programming concepts and not have to worry about too much details.
  - **Very Flexible**
    - As a dynamically typed language, Python is really flexible. This means there are no hard rules on how to build features, and you'll have more flexibility solving problems using different methods (though the Python philosophy encourages using the obvious way to solve things). Furthermore, Python is also more forgiving of errors, so you'll still be able to compile and run your program until you hit the problematic part.

Reference: http://www.bestprogramminglanguagefor.me/why-learn-python

Why Learn Python? (cont.)

- **Community**
  - As you step into the programming world, you'll soon understand how vital support is, as the developer community is all about giving and receiving help. The larger a community, the more likely you'd get help and the more people will be building useful tools to ease the process of development.
  - **5th Largest StackOverflow Community**
    - StackOverflow is a programming Q&A site you will no doubt become intimate with as a coding beginner. Python has 85.9k followers, with over 500k Python questions.
  - **3rd Largest Meetup Community**
    - There are 1300+ Python groups on Meetup.com, totaling 608k+ members. Thus, in terms of programming languages, Python is the 3rd largest community.
  - **4th Most-Used Language at GitHub**
    - The more useful projects there are, the more likely someone has already built a function you need and built it well, which will greatly speed up your development process.Over 950 Python projects have over 500 stars.

Reference: http://www.bestprogramminglanguagefor.me/why-learn-python

Why Learn Python? (cont.)

- **Career Opportunities**
  - On Angel List, Python is the 2nd most demanded skill and also the skill with the highest average salary offered.
  - With the rise of big data, Python developers are in demand as data scientists, especially since Python can be easily integrated into web applications to carry out tasks that require machine learning.

Popular sites built with Python

Google Dropbox Pinterest Instagram

Salary information from gooroo.io

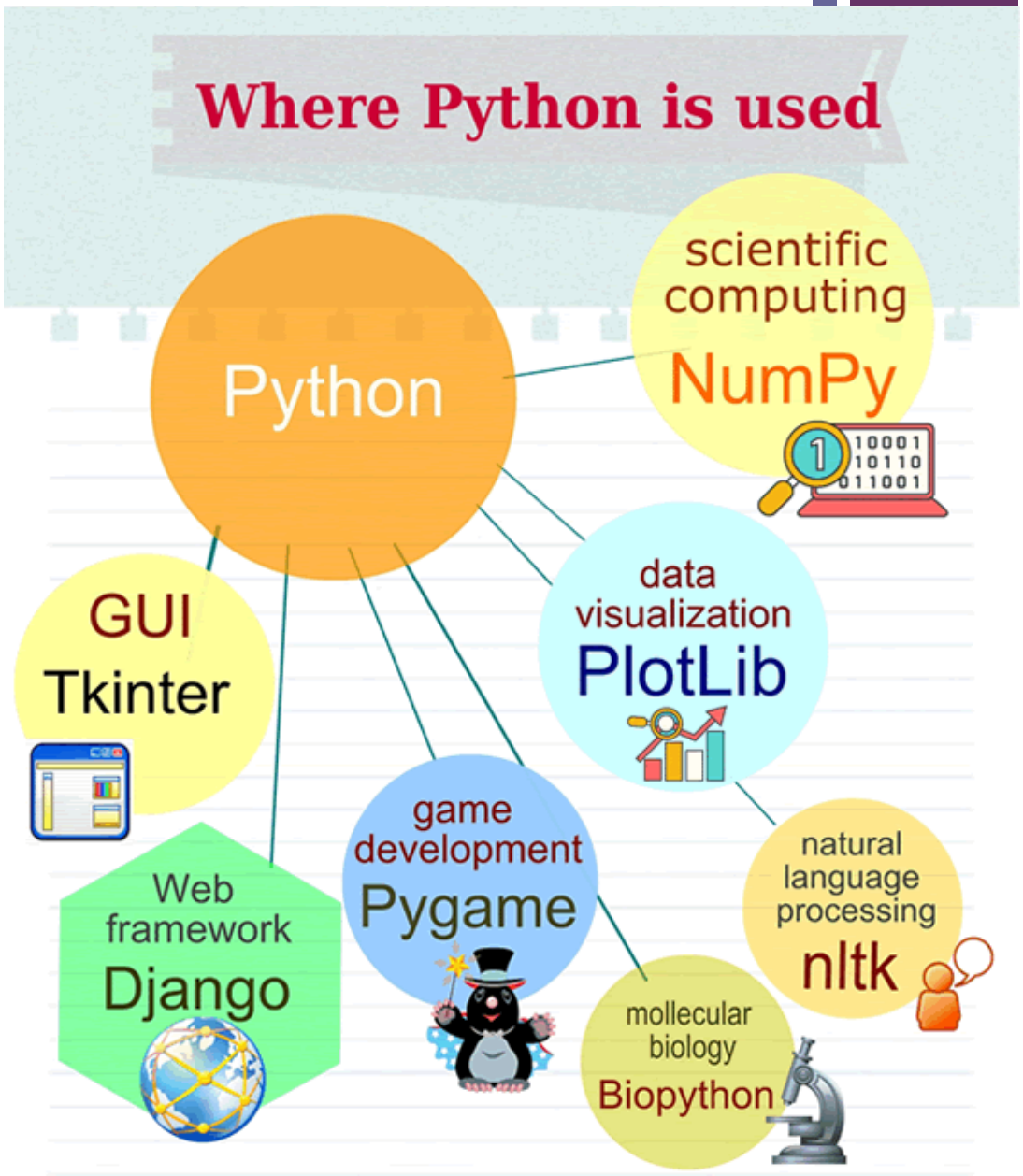
Salary Range
43K - 135K

Average Salary
$94,053

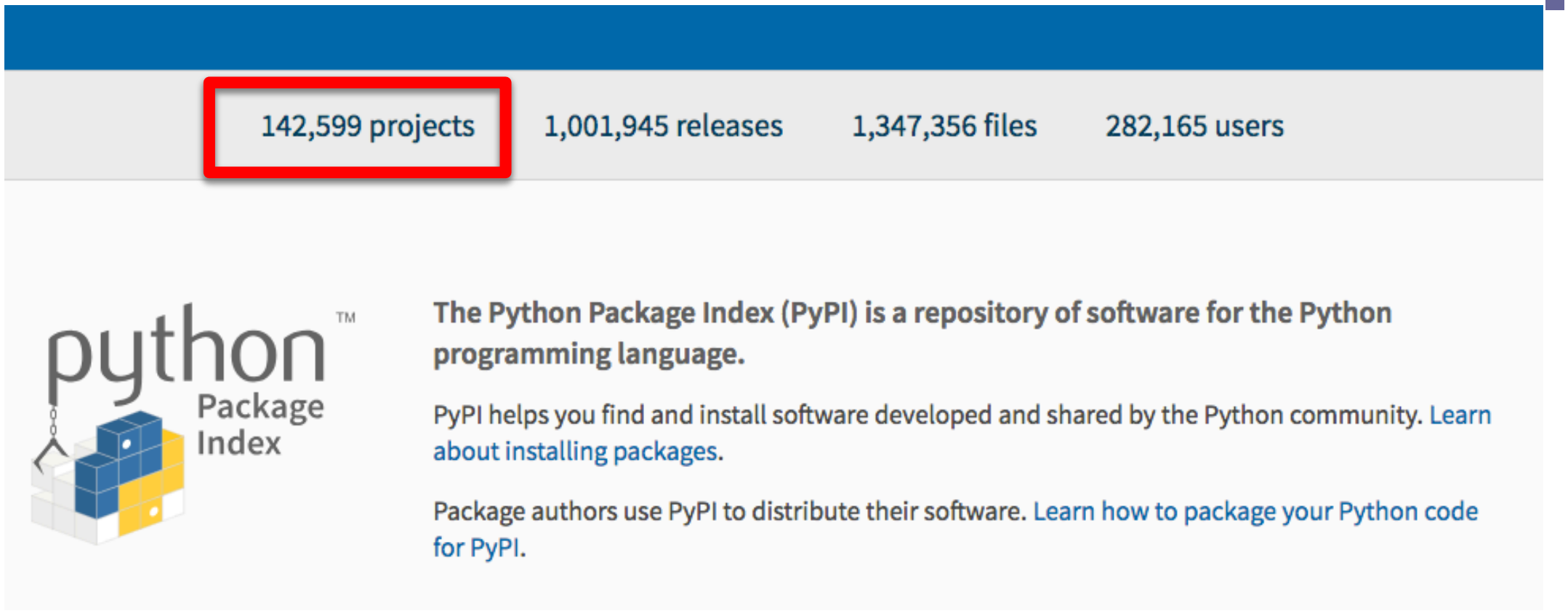
Reference: http://www.bestprogramminglanguagefor.me/why-learn-python

Why Use Python?

- Python's expansive library of open source data analysis tools, web frameworks, and testing instruments make its ecosystem one of the largest out of any programming community.
- Python is an accessible language for new programmers because the community provides many introductory resources. The language is also widely taught in universities and used for working with beginner-friendly devices such as the Raspberry Pi.



Python Packages



Top 20 Python libraries for data science in 2019

- Python continues to take leading positions in solving data science tasks and challenges.
- Our selection actually contains more than 20 libraries, as some of them are alternatives to each other and solve the same problem.
- Therefore we have grouped them as it's difficult to distinguish one particular leader at the moment.

Reference: https://bigdata-madesimple.com/top-20-python-libraries-for-data-science/

Reference: https://bigdata-madesimple.com/top-20-python-libraries-for-data-science/

Top 20 Python libraries for data science in 2019 (cont.)

matplotlib
Bokeh
plotly
Seaborn
pydot
learn
XGBoost
LightGBM
CatBoost
eli5
SciPy
NumPy
pandas
StatsModels
Statistics in Python
TensorFlow
PYTORCH
Keras
dist-keras
elephas
spark-deep-learning
Natural Language ToolKit
spaCy
gensim
Scrapy

Reference: https://bigdata-madesimple.com/top-20-python-libraries-for-data-science/

Top 20 Python libraries for data science in 2019 (cont.)

Core Libraries & Statistics

- 1. NumPy (Commits: 17911, Contributors: 641)

- Traditionally, we start our list with the libraries for scientific applications, and NumPy is one of the principal packages in this area. It is intended for processing large multidimensional arrays and matrices, and an extensive collection of high-level mathematical functions and implemented methods makes it possible to perform various operations with these objects.

- During the year, a large number of improvements have been made to the library. In addition to bug fixes and compatibility issues, the crucial changes regard styling possibilities, namely the printing format of NumPy objects. Also, some functions can now handle files of any encoding that is available in Python.

Top 20 Python libraries for data science in 2019 (cont.)

- 2. SciPy (Commits: 19150, Contributors: 608)

- Another core library for scientific computing is SciPy. It is based on NumPy and therefore extends its capabilities. SciPy main data structure is again a multidimensional array, implemented by Numpy. The package contains tools that help with solving linear algebra, probability theory, integral calculus and many more tasks.

- SciPy faced major build improvements in the form of continuous integration into different operating systems, new functions and methods and, what is especially important - the updated optimizers. Also, many new BLAS and LAPACK functions were wrapped.

Top 20 Python libraries for data science in 2019 (cont.)

- 3. Pandas (Commits: 17144, Contributors: 1165)

- Pandas is a Python library that provides high-level data structures and a vast variety of tools for analysis. The great feature of this package is the ability to translate rather complex operations with data into one or two commands. Pandas contains many built-in methods for grouping, filtering, and combining data, as well as the time-series functionality. All of this is followed by impressive speed indicators.

- There have been a few new releases of the pandas library, including hundreds of new features, enhancements, bug fixes, and API changes. The improvements regard pandas abilities for grouping and sorting data, more suitable output for the *apply* method, and the support in performing custom types operations.

Top 20 Python libraries for data science in 2019 (cont.)

- 4. StatsModels (Commits: 10067, Contributors: 153)

- Statsmodels is a Python module that provides many opportunities for statistical data analysis, such as statistical models estimation, performing statistical tests, etc. With its help, you can implement many machine learning methods and explore different plotting possibilities.

- The library is continuously developing, enriching new and new opportunities. Thus, this year brought time series improvements and new count models, namely GeneralizedPoisson, zero inflated models, and NegativeBinomialP, and new multivariate methods - factor analysis, MANOVA, and repeated measures within ANOVA.

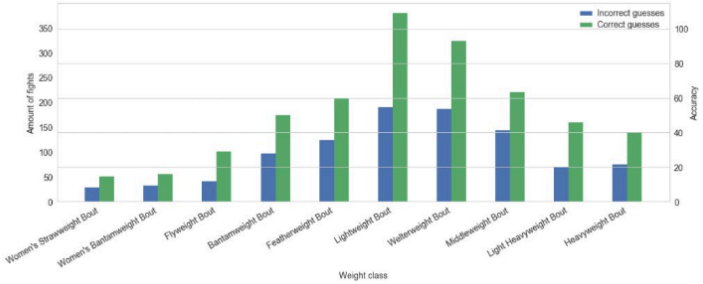
Top 20 Python libraries for data science in 2018 (cont.)

Visualization

- 5. Matplotlib (Commits: 25747, Contributors: 725)

- Matplotlib is a low-level library for creating two-dimensional diagrams and graphs. With its help, you can build diverse charts, from histograms and scatterplots to non-Cartesian coordinates graphs. Moreover, many popular plotting libraries are designed to work in conjunction with matplotlib.

- There have been style changes in colors, sizes, fonts, legends, etc. As an example of an appearance improvements are an automatic alignment of axes legends and among significant colors improvements is a new colorblind-friendly color cycle.



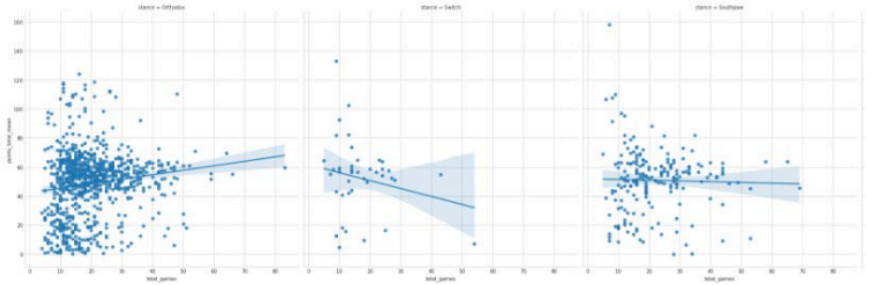
Top 20 Python libraries for data science in 2019 (cont.)

Seaborn

- 6. Seaborn (Commits: 2044, Contributors: 83)

- Seaborn is essentially a higher-level API based on the matplotlib library. It contains more suitable default settings for processing charts. Also, there is a rich gallery of visualizations including some complex types like time series, jointplots, and violin diagrams.

- The seaborn updates mostly cover bug fixes. However, there were improvements in compatibility between FacetGrid or PairGrid and enhanced interactive matplotlib backends, adding parameters and options to visualizations.



Top 20 Python libraries for data science in 2019 (cont.)

- 7. Plotly (Commits: 2906, Contributors: 48)

- Plotly is a popular library that allows you to build sophisticated graphics easily. The package is adapted to work in interactive web applications. Among its remarkable visualizations are contour graphics, ternary plots, and 3D charts.

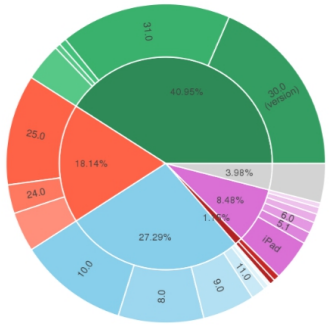
- The continuous enhancements of the library with new graphics and features brought the support for "multiple linked views" as well as animation, and crosstalk integration.

Top 20 Python libraries for data science in 2019 (cont.)

- 8. Bokeh (Commits: 16983, Contributors: 294)

- The Bokeh library creates interactive and scalable visualizations in a browser using JavaScript widgets. The library provides a versatile collection of graphs, styling possibilities, interaction abilities in the form of linking plots, adding widgets, and defining callbacks, and many more useful features.

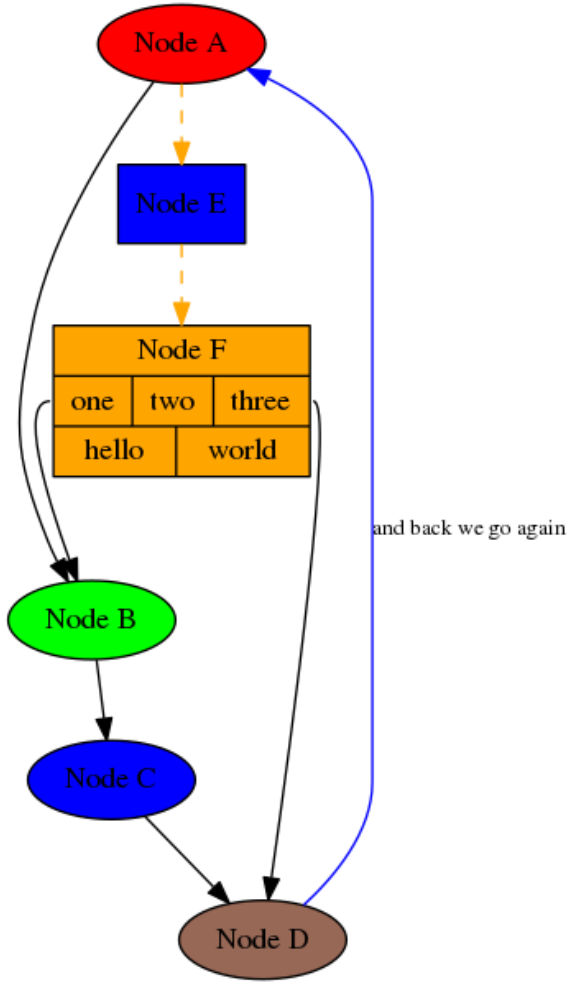
- Bokeh can boast with improved interactive abilities, like a rotation of categorical tick labels, as well as small zoom tool and customized tooltip fields enhancements.



Top 20 Python libraries for data science in 2019 (cont.)

- 9. Pydot (Commits: 169, Contributors: 12)

- Pydot is a library for generating complex oriented and non-oriented graphs. It is an interface to Graphviz, written in pure Python. With its help, it is possible to show the structure of graphs, which are very often needed when building neural networks and decision trees based algorithms.



Top 20 Python libraries for data science in 2019 (cont.)

Machine Learning

- 10. Scikit-learn (Commits: 22753, Contributors: 1084)

- This Python module based on NumPy and SciPy is one of the best libraries for working with data. It provides algorithms for many standard machine learning and data mining tasks such as clustering, regression, classification, dimensionality reduction, and model selection.

- There is a number of enhancements made to the library. The cross validation has been modified, providing an ability to use more than one metric. Several training methods like nearest neighbors and logistic regressions faced some minor improvements. Finally, one of the major updates is the accomplishment of the Glossary of Common Terms and API Elements which acquaints with the terminology and conventions used in Scikit-learn.

- Improve your skills with Data Science School

Top 20 Python libraries for data science in 2019 (cont.)

- 11. XGBoost / LightGBM / CatBoost (Commits: 3277 / 1083 / 1509, Contributors: 280 / 79 / 61)

- Gradient boosting is one of the most popular machine learning algorithms, which lies in building an ensemble of successively refined elementary models, namely decision trees. Therefore, there are special libraries designed for fast and convenient implementation of this method. Namely, we think that XGBoost, LightGBM, and CatBoost deserve special attention. They are all competitors that solve a common problem and are used in almost the same way. These libraries provide highly optimized, scalable and fast implementations of gradient boosting, which makes them extremely popular among data scientists and Kaggle competitors, as many contests were won with the help of these algorithms.

Top 20 Python libraries for data science in 2019 (cont.)

- 12. Eli5 (Commits: 922, Contributors: 6)

- Often the results of machine learning models predictions are not entirely clear, and this is the challenge that eli5 library helps to deal with. It is a package for visualization and debugging machine learning models and tracking the work of an algorithm step by step. It provides support for scikit-learn, XGBoost, LightGBM, lightning, and sklearn-crfsuite libraries and performs the different tasks for each of them.

Top 20 Python libraries for data science in 2019 (cont.)

Deep Learning

- 13. TensorFlow (Commits: 33339, Contributors: 1469)

- TensorFlow is a popular framework for deep and machine learning, developed in Google Brain. It provides abilities to work with artificial neural networks with multiple data sets. Among the most popular TensorFlow applications are object identification, speech recognition, and more. There are also different layer-helpers on top of regular TensorFlow, such as tflearn, tf-slim, skflow, etc.

- This library is quick in new releases, introducing new and new features. Among the latest are fixes in potential security vulnerability and improved TensorFlow and GPU integration, such as you can run an Estimator model on multiple GPUs on one machine.

Top 20 Python libraries for data science in 2019 (cont.)

- 14. PyTorch (Commits: 11306, Contributors: 635)

- PyTorch is a large framework that allows you to perform tensor computations with GPU acceleration, create dynamic computational graphs and automatically calculate gradients. Above this, PyTorch offers a rich API for solving applications related to neural networks.

- The library is based on Torch, which is an open source deep learning library implemented in C with a wrapper in Lua. The Python API was introduced in 2017 and from that point on, the framework is gaining popularity and attracting an increasing number of data scientists.

Top 20 Python libraries for data science in 2019 (cont.)

- 15. Keras (Commits: 4539, Contributors: 671)

- Keras is a high-level library for working with neural networks, running on top of TensorFlow, Theano, and now as a result of the new releases, it is also possible to use CNTK and MxNet as the backends. It simplifies many specific tasks and greatly reduces the amount of monotonous code. However, it may not be suitable for some complicated things.

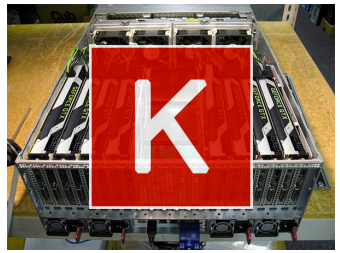
- This library faced performance, usability, documentation, and API improvements. Some of the new features are Conv3DTranspose layer, new MobileNet application, and self-normalizing networks.

Top 20 Python libraries for data science in 2019 (cont.)

Distributed Deep Learning

- 16. Dist-keras / elephas / spark-deep-learning (Commits: 1125 / 170 / 67, Contributors: 5 / 13 / 11)

- Deep learning problems are becoming crucial nowadays since more and more use cases require considerable effort and time. However, processing such an amount of data is much easier with the use of distributed computing systems like Apache Spark which again expands the possibilities for deep learning. Therefore, dist-keras, elephas, and spark-deep-learning are gaining popularity and developing rapidly, and it is very difficult to single out one of the libraries since they are all designed to solve a common task. These packages allow you to train neural networks based on the Keras library directly with the help of Apache Spark. Spark-deep-learning also provides tools to create a pipeline with Python neural networks.



Top 20 Python libraries for data science in 2019 (cont.)

Natural Language Processing

- 17. NLTK (Commits: 13041, Contributors: 236)

- NLTK is a set of libraries, a whole platform for natural language processing. With the help of NLTK, you can process and analyze text in a variety of ways, tokenize and tag it, extract information, etc. NLTK is also used for prototyping and building research systems.

- The enchantments to this library cover minor changes in APIs and compatibility and a new interface to CoreNLP.

Top 20 Python libraries for data science in 2019 (cont.)

- 18. SpaCy (Commits: 8623, Contributors: 215)

- SpaCy is a natural language processing library with excellent examples, API documentation, and demo applications. The library is written in the Cython language which is C extension of Python. It supports almost 30 languages, provides easy deep learning integration and promises robustness and high accuracy. Another great feature of spaCy is an architecture designed for entire documents processing, without breaking the document into phrases.

Top 20 Python libraries for data science in 2019 (cont.)

- 19. Gensim (Commits: 3603, Contributors: 273)

- Gensim is a Python library for robust semantic analysis, topic modeling and vector-space modeling, and is built upon Numpy and Scipy. It provides an implementation of popular NLP algorithms, such as word2vec. Although gensim has its own models.wrappers.fasttext implementation, the fasttext library can also be used for efficient learning of word representations.

Top 20 Python libraries for data science in 2019 (cont.)

Data Scraping

- 20. Scrapy (Commits: 6625, Contributors: 281)

- Scrapy is a library used to create spiders bots that scan website pages and collect structured data. In addition, Scrapy can extract data from the API. The library happens to be very handy due to its extensibility and portability.

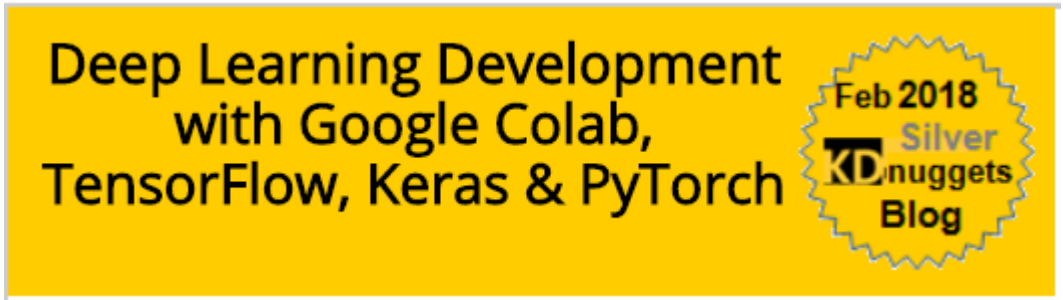
- Among the advances made through the year are several upgrades in proxy servers and improved system of errors notification and problems identification. There are also new possibilities in metadata settings using *scrapy parse*.

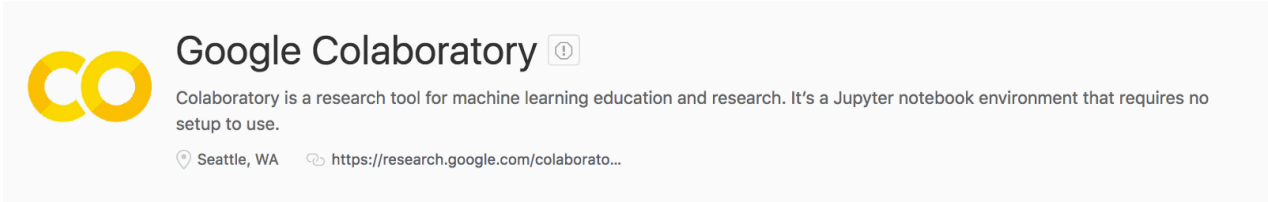
Introduction to Colaboratory (Google Colab)

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

- **What is Google Colab?**
  - Google Colab is a free cloud service and now it supports free GPU!
  - You can;
    - improve your **Python** programming language coding skills.
    - develop deep learning applications using popular libraries such as **Keras**, **TensorFlow**, **PyTorch,** and **OpenCV**.
  - The most important feature that distinguishes Colab from other free cloud services is; **Colab** provides GPU and is totally free.

Python 3.6.7

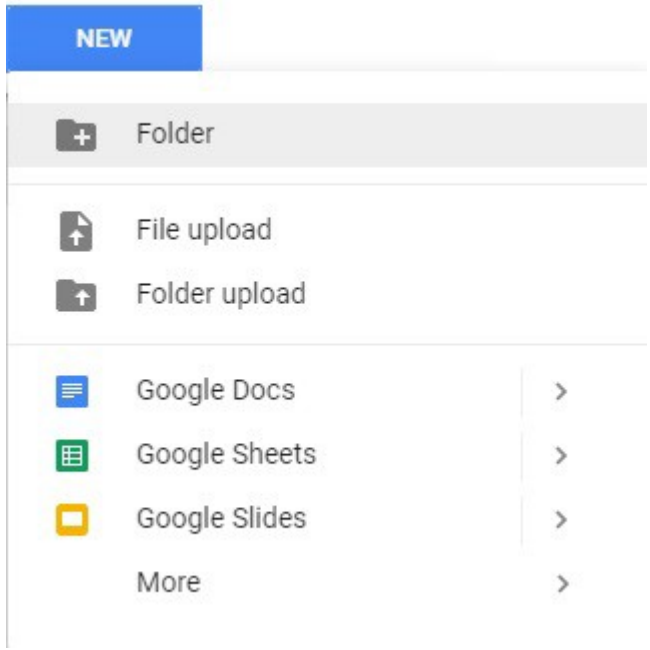




Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.)

- Getting Google Colab Ready to Use
  - Creating Folder on Google Drive

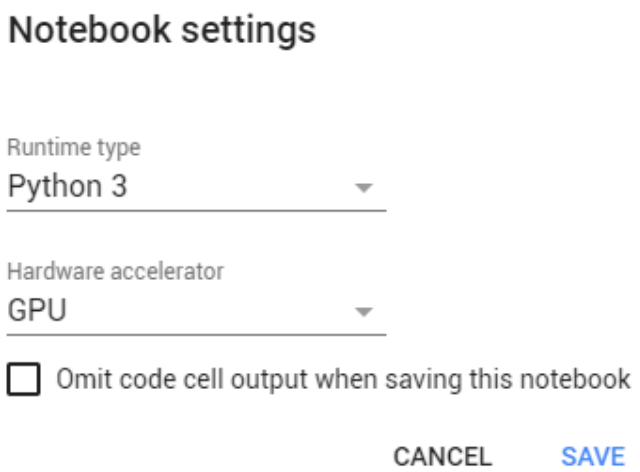


Since Colab is working on your own Google Drive, we first need to specify the folder we'll work. I created a folder named "app" on my Google Drive. Of course, you can use a different name or choose the default Colab Notebooks folder instead of app folder.

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.)

- **Setting Free GPU**
  - It is so simple to alter default hardware **(CPU to GPU or vice versa)**; just follow **Edit > Notebook settings** or **Runtime>Change runtime type** and **select GPU** as **Hardware accelerator**.



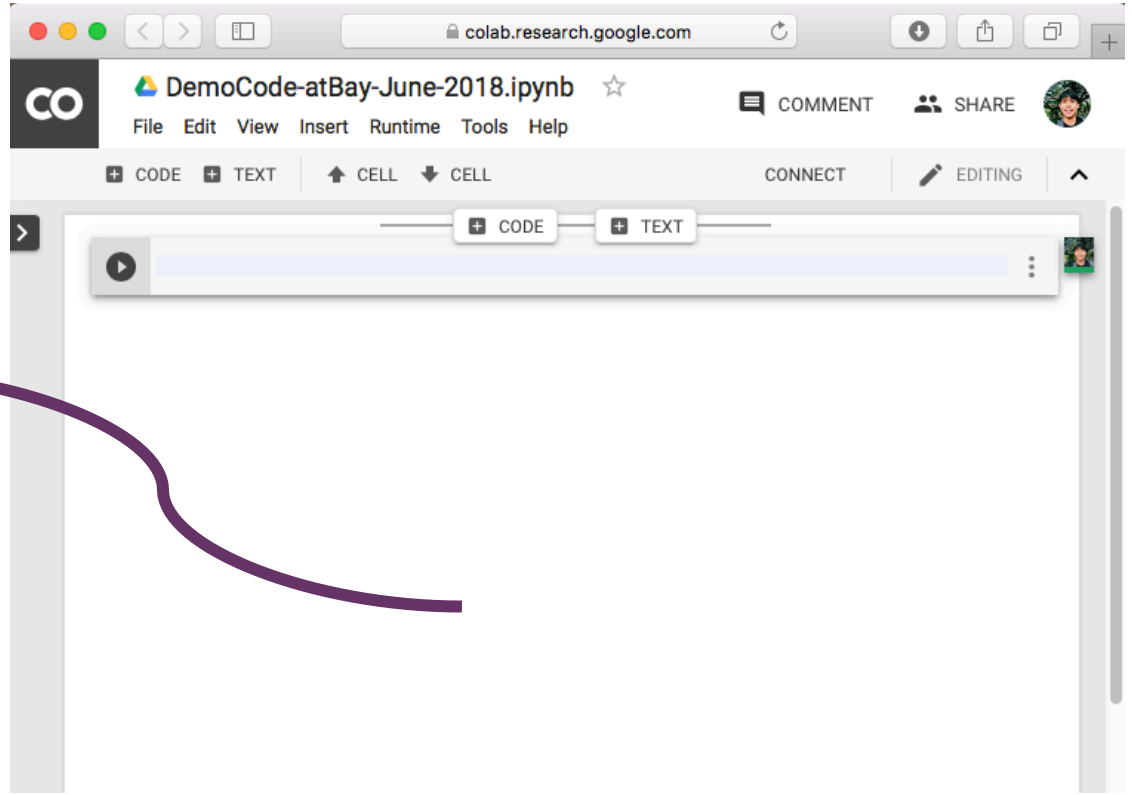
Now you can develop **deep learning** applications with Google Colaboratory -on the **free Tesla K80 GPU**- using Keras, Tensorflow and PyTorch.

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.)

- **Running Basic Python Codes with Google Colab**
- Now we can start using **Google Colab**.

```
[1]  x = 3

[2]  print(type(x)) # Prints "<class 'int'>"

     <type 'int'>

[3]  print(x)       # Prints "3"

     3

     print(x + 1)   # Addition; prints "4"

     4
```



Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- 1. How to Install Libraries?

Keras

```
!pip install -q keras
import keras
```

Other Libraries

`!pip install` or `!apt-get install` to install other libraries.

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- **2. Is GPU Working?**
  - To see if you are currently using the GPU in Colab, you can run the following code in order to cross-check:

```
import tensorflow as tf
tf.test.gpu_device_name()
```

```
import tensorflow as tf
tf.test.gpu_device_name()
```

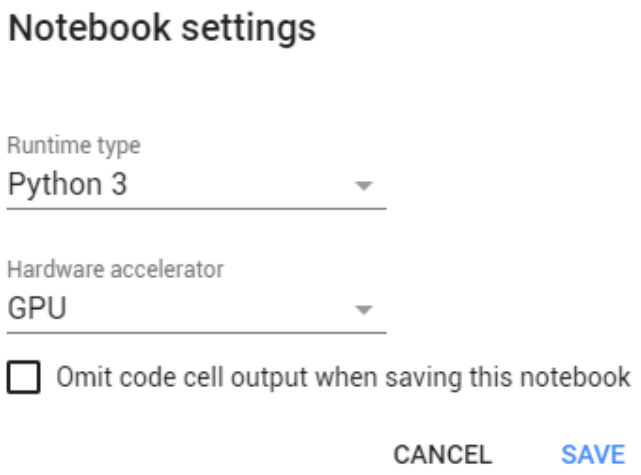
''

only CPU

```
import tensorflow as tf
tf.test.gpu_device_name()
```

'/device:GPU:0'

GPU



Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- **3. Which GPU Am I Using?**

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

- Currently, **Colab only provides Tesla K80.**

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

```
[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 115010925269716724, name: "/device:GPU:0"
 device_type: "GPU"
 locality {
   bus_id: 1
 }
 incarnation: 2835578110136398533
 physical_device_desc: "device: 0, name: Tesla K80, pci bus id: 0000:00:04.0, compute capability: 3.7"]
```

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- 4. What about RAM?

```
!cat /proc/meminfo
```

```
!cat /proc/meminfo

MemTotal:       13342000 kB
MemFree:         4059676 kB
MemAvailable:   11139900 kB
Buffers:          637980 kB
Cached:          6078588 kB
SwapCached:            0 kB
Active:          4728852 kB
Inactive:        3296644 kB
Active(anon):    1468368 kB
Inactive(anon):   121888 kB
```

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- **6. Changing Working Directory**
  - Normally when you run this code:

```
!ls
```

  - You probably see **datalab and drive** folders.
  - Therefore you must add **drive/app** before defining each filename.
  - To get rid of this problem, you can simply change the working directory. (In this tutorial I changed to **app folder**) with this simple code:

```
import os
os.chdir("drive/app")
```

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- 5. What about CPU?

```
!cat /proc/cpuinfo
```

```
!cat /proc/cpuinfo

processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU @ 2.20GHz
stepping        : 0
microcode       : 0x1
cpu MHz         : 2199.998
cache size      : 56320 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm const
bugs            :
bogomips        : 4399.99
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:
```

Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

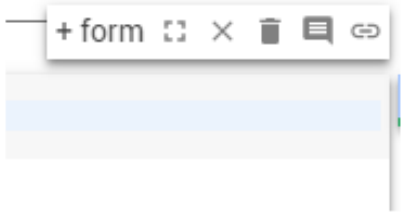
- **7. How to Restart Google Colab?**
  - In order to restart (or reset) your virtual machine, simply run:

```
!kill -9 -1
```

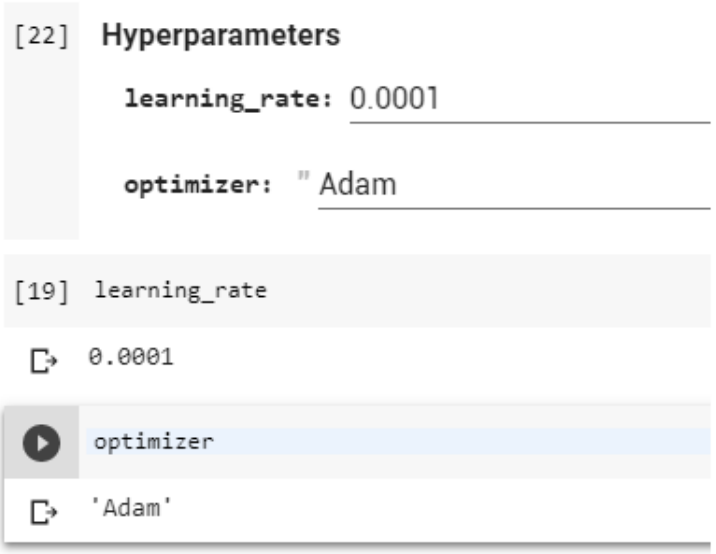
Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- **8. How to Add Form to Google Colab?**
  - In order not to change hyperparameters every time in your code, you can simply add form to Google Colab.



  - For instance, I added form which contain **learning_rate** variable and **optimizer** string.



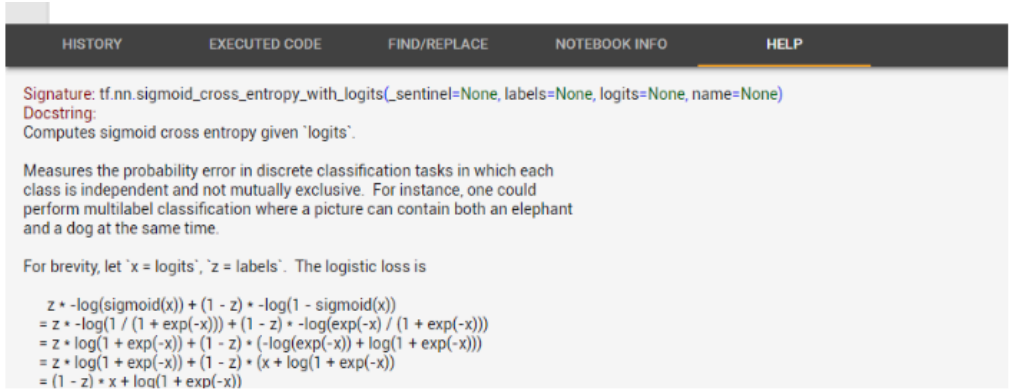
Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Introduction to Colaboratory (Google Colab) (cont.) -- Some Useful Tips

- **9. How to See Function Arguments?**
  - To see function arguments in TensorFlow, Keras etc, simply **add question mark (?)** after function name:

```
import tensorflow as tf
tf.nn.sigmoid_cross_entropy_with_logits?
```

  - Now you can see original documentation without clicking TensorFlow website.

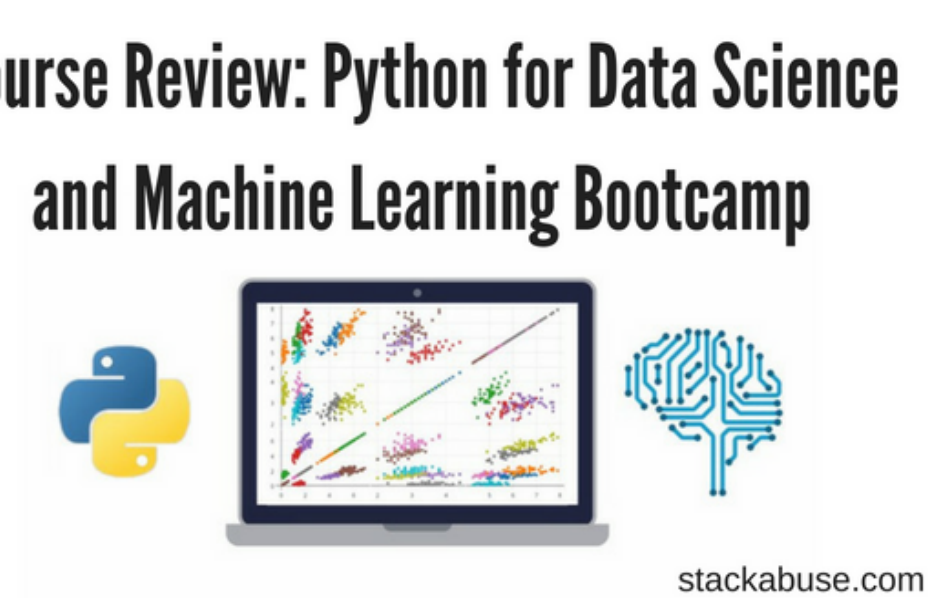


Reference: https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

Reference

- กิตติภณ พละการ, กิตติภพ พละการ, สมชาย ประสิทธิ์จูตระกูล และ สุกรี สินธุภิญโญ, หนังสือสอนเขียนโปรแกรมภาษา **Python** ใช้ประกอบการเรียนวิชา 2110101 **Computer Programming,** ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย https://www.cp.eng.chula.ac.th/~somchai/

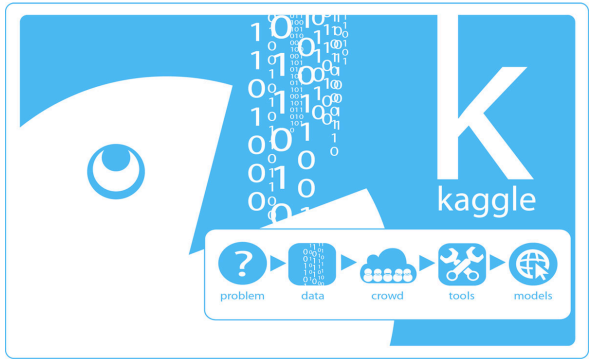
- **Python for Data Science and Machine Learning Bootcamp** https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/



รศ. ดร. สมชาย ประสิทธิ์จูตระกูล

Reference (cont.)

- https://keras.io/
- https://www.kaggle.com/
- https://archive.ics.uci.edu/ml/index.php



+

Any Questions?