

C.A.S.A. in C.V.

Context Aware Security Analytics in
Computer Vision

Paola Barra



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Tools



Git and GitHub



Anaconda ... & friends



Python 3.6

What is Machine Learning?



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Git



- is a command-line tool used for version control;
- runs locally on a computer;
- is completely self-sufficient;
- it does not require the use of any external, cloud hosting service.

Most used Git command-line tools:

- commit
- push
- pull
- revert
- ...



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Git command-line tools



git config	
<code>git config --global user.name <name></code>	Define the author name to be used for all commits by the current user.
<code>git config --global user.email <email></code>	Define the author email to be used for all commits by the current user.
<code>git config --global alias.<alias-name> <git-command></code>	Create shortcut for a Git command. E.g. <code>alias.glog log --graph --oneline</code> will set <code>git glog</code> equivalent to <code>git log --graph --oneline</code> .
<code>git config --system core.editor <editor></code>	Set text editor used by commands for all users on the machine. <code><editor></code> arg should be the command that launches the desired editor (e.g., vi).
<code>git config --global --edit</code>	Open the global configuration file in a text editor for manual editing.
git log	
<code>git log --<limit></code>	Limit number of commits by <code><limit></code> . E.g. <code>git log -5</code> will limit to 5 commits.
<code>git log --oneline</code>	Condense each commit to a single line.
<code>git log -p</code>	Display the full diff of each commit.
<code>git log --stat</code>	Include which files were altered and the relative number of lines that were added or deleted from each of them.
<code>git log --author="<pattern>"</code>	Search for commits by a particular author.
<code>git log --grep="<pattern>"</code>	Search for commits with a commit message that matches <code><pattern></code> .
<code>git log <since>...<until></code>	Show commits that occur between <code><since></code> and <code><until></code> . Args can be a commit ID, branch name, HEAD, or any other kind of revision reference.
<code>git log --<file></code>	Only display commits that have the specified file.
<code>git log --graph --decorate</code>	--graph flag draws a text based graph of commits on left side of commit msgs. --decorate adds names of branches or tags of commits shown.
git diff	
<code>git diff HEAD</code>	Show difference between working directory and last commit.
<code>git diff --cached</code>	Show difference between staged changes and last commit
git reset	
<code>git reset</code>	Reset staging area to match most recent commit, but leave the working directory unchanged.
<code>git reset --hard</code>	Reset staging area and working directory to match most recent commit and overwrites all changes in the working directory.
<code>git reset <commit></code>	Move the current branch tip backward to <code><commit></code> , reset the staging area to match, but leave the working directory alone.
<code>git reset --hard <commit></code>	Same as previous, but resets both the staging area & working directory to match. Deletes uncommitted changes, and all commits after <code><commit></code> .
git rebase	
<code>git rebase -i <base></code>	Interactively rebase current branch onto <code><base></code> . Launches editor to enter commands for how each commit will be transferred to the new base.
git pull	
<code>git pull --rebase <remote></code>	Fetch the remote's copy of current branch and rebases it into the local copy. Uses git rebase instead of merge to integrate the branches.
git push	
<code>git push <remote> --force</code>	Forces the git push even if it results in a non-fast-forward merge. Do not use the --force flag unless you're absolutely sure you know what you're doing.
<code>git push <remote> --all</code>	Push all of your local branches to the specified remote.
<code>git push <remote> --tags</code>	Tags aren't automatically pushed when you push a branch or use the --all flag. The --tags flag sends all of your local tags to the remote repo.



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



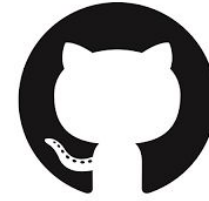
Git command-line tools



Git Basics		Rewriting Git History	
<code>git init <directory></code>	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.	<code>git commit --amend</code>	Replace the last commit with the staged changes and last commit combined. Use with nothing staged to edit the last commit's message.
<code>git clone <repo></code>	Clone repo located at <repo> onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.	<code>git rebase <base></code>	Rebase the current branch onto <base>. <base> can be a commit ID, a branch name, a tag, or a relative reference to HEAD.
<code>git config user.name <name></code>	Define author name to be used for all commits in current repo. Devs commonly use --global flag to set config options for current user.	<code>git reflog</code>	Show a log of changes to the local repository's HEAD. Add --relative-date flag to show date info or --all to show all refs.
<code>git add <directory></code>	Stage all changes in <directory> for the next commit. Replace <directory> with a <file> to change a specific file.	Git Branches	
<code>git commit -m "message"</code>	Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.	<code>git branch</code>	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.
<code>git status</code>	List which files are staged, unstaged, and untracked.	<code>git checkout -b <branch></code>	Create and check out a new branch named <branch>. Drop the -b flag to checkout an existing branch.
<code>git log</code>	Display the entire commit history using the default format. For customization see additional options.	<code>git merge <branch></code>	Merge <branch> into the current branch.
<code>git diff</code>	Show unstaged changes between your index and working directory.	Remote Repositories	
Undoing Changes		<code>git remote add <name> <url></code>	Create a new connection to a remote repo. After adding a remote, you can use <name> as a shortcut for <url> in other commands.
<code>git revert <commit></code>	Create new commit that undoes all of the changes made in <commit>, then apply it to the current branch.	<code>git fetch <remote> <branch></code>	Fetches a specific <branch>, from the repo. Leave off <branch> to fetch all remote refs.
<code>git reset <file></code>	Remove <file> from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.	<code>git pull <remote></code>	Fetch the specified remote's copy of current branch and immediately merge it into the local copy.
<code>git clean -n</code>	Shows which files would be removed from working directory. Use the -f flag in place of the -n flag to execute the clean.	<code>git push <remote> <branch></code>	Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.



GitHub



- is a cloud-based platform built around the Git tool;
- online editing of file;
- is an online service to which developers who use Git can connect and upload or download resources.



UNIVERSITÀ
DEGLI STUDI
DI SALERNO





Git vs. GitHub comparison



GIT

GITHUB

Installed locally

Hosted in the cloud

First released in 2005

Company launched in 2008

Maintained by The Linux Foundation

Purchased in 2018 by Microsoft

Focused on version control and code sharing

Focused on centralized source code hosting

Primarily a command-line tool

Administered through the web

Provides a desktop interface named Git Gui

Desktop interface named GitHub Desktop

No user management features

Built-in user management

Minimal external tool configuration features

Active marketplace for tool integration

Competes with Mercurial, Subversion, IBM, Rational Team Concert and ClearCase

Competes with Atlassian Bitbucket and GitLab

Open source licensed

Includes a free tier and pay-for-use tiers

Git & GitHub



As you can see, Git and GitHub are clearly different in features and purpose

- they compliment each other
- they are not competitors

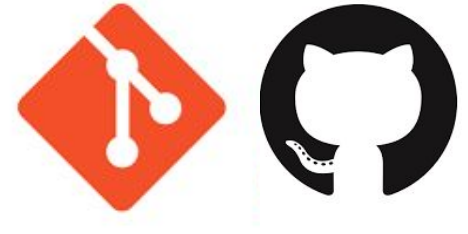
Integrating Git and GitHub together is a powerful combination.



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Let's start



- 1) Set up a GitHub account <https://github.com>
- 2) Download GitHub desktop on you computer
<https://desktop.github.com>
- 3) Create a repository in local and an URL will uniquely identifies it.
- 4) With the repository created, developers can install Git locally and **clone** the repository.
- 5) Developers can make local changes to project files and then **push** the **commit** back to GitHub.
- 6) To get updates developers use **pull** and **fetch** commands.



UNIVERSITÀ
DEGLI STUDI
DI SALERNO





The open-source **Anaconda Distribution** is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with **Conda**
- Develop and train machine learning and deep learning models with **scikit-learn**, **TensorFlow**, and **Theano**
- Analyze data with scalability and performance with **Dask**, **NumPy**, **pandas**, and **Numba**
- Visualize results with **Matplotlib**, **Bokeh**, **Datashader**, and **Holoviews**

Download Anaconda on
www.anaconda.com

Python 3.7 version

Download

64-Bit Graphical Installer (653 MB)

64-Bit Command Line Installer (435 MB)



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Do not install Python 2.7

```
Using TensorFlow backend.
```

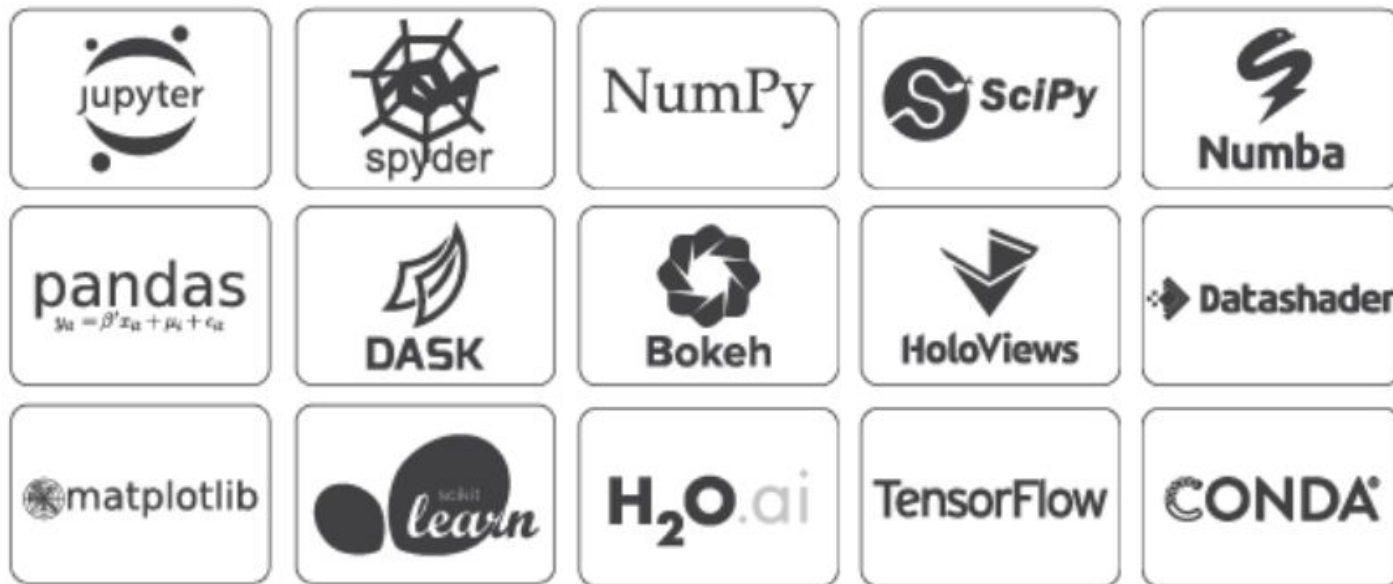
```
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020  
. Please upgrade your Python as Python 2.7 won't be maintained after that d  
ate. A future version of pip will drop support for Python 2.7.
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python2.7/site-pack  
age (1.5.6)
```



UNIVERSITÀ
DEGLI STUDI
DI SALERNO





UNIVERSITÀ
DEGLI STUDI
DI SALERNO



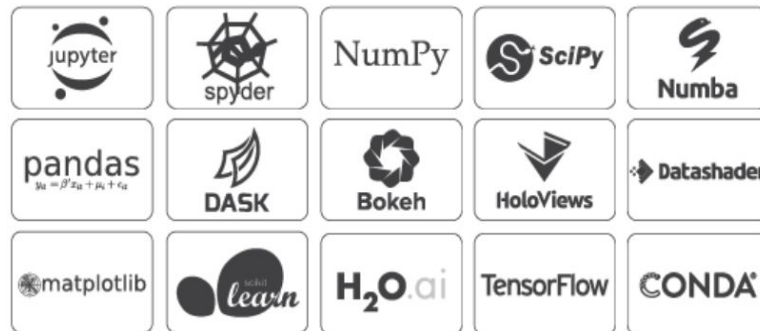
Conda



Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/C++, FORTRAN, and more.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

And a lot of other things....



scikit 1



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



Create a conda environment



#Create a new environment

conda create -n name_env python=3.7

conda activate name_env

conda deactivate

#Enter your new environment

#Exit all the environments

matplotlib



Is the best plotting library out there. Install it:

conda install -c conda-forge matplotlib



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



scikit-learn



Is the best Machine Learning library in the world.

Is important that you are familiar with python and statistics

You can load dataset using **pandas**. Is Built on **Numpy, Scipy & Matplotlib**. There is a huge documentation on scikit-learn.org

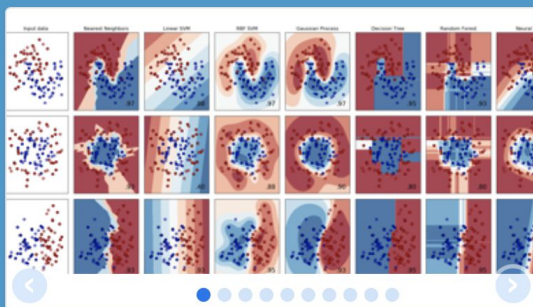
Install it:

conda install -c anaconda scikit-learn



UNIVERSITÀ
DEGLI STUDI
DI SALERNO





scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extract



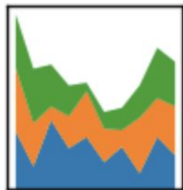
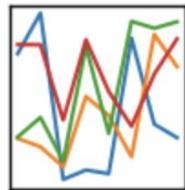
UNIVERSITA
DEGLI STUDI
DI SALERNO



data manipulation

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Pandas is used to manage data. It works with dataframes (strings of numerical data). Here we will see:

- how to load csv file in Pandas
- how to use prebuilt datasets, to help you testing scikit-learn algorithms (you'll never use this kind of dataset but they are useful indeed)

Install it:

conda install -c anaconda pandas



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



6.2. Toy datasets

scikit-learn comes with a few small standard datasets that do not require to download any file from some external website.

They can be loaded using the following functions:

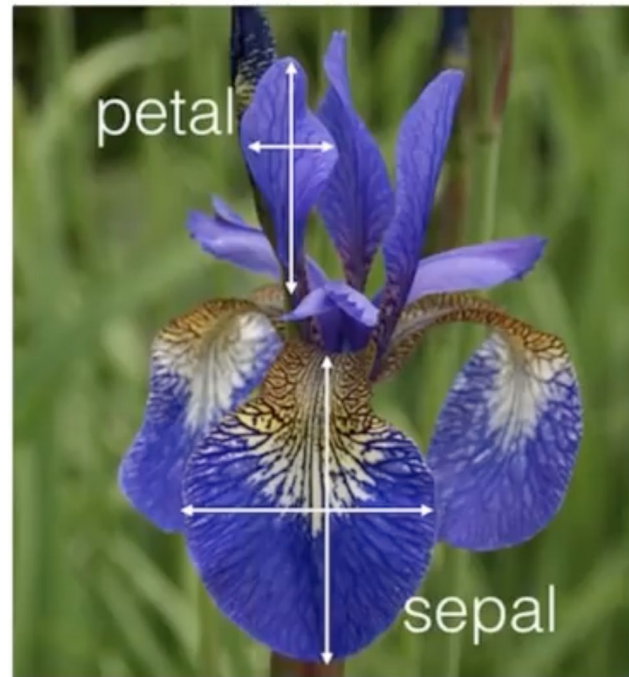
<code>load_boston</code> ([return_X_y])	Load and return the boston house-prices dataset (regression).
<code>load_iris</code> ([return_X_y])	Load and return the iris dataset (classification).
<code>load_diabetes</code> ([return_X_y])	Load and return the diabetes dataset (regression).
<code>load_digits</code> ([n_class, return_X_y])	Load and return the digits dataset (classification).
<code>load_linnerud</code> ([return_X_y])	Load and return the linnerud dataset (multivariate regression).
<code>load_wine</code> ([return_X_y])	Load and return the wine dataset (classification).
<code>load_breast_cancer</code> ([return_X_y])	Load and return the breast cancer wisconsin dataset (classification).

These datasets are useful to quickly illustrate the behavior of the various algorithms implemented in scikit-learn. They are however often too small to be representative of real world machine learning tasks.



IRIS Dataset

- is a famous example dataset
- the dataset consists of 50 samples from three species of Iris (**Setosa**, **Virginica** and **Versicolor**)
- Four features were measured from each samples
(**sepals length**, **sepals width**, **petal length** and **petal width**) in centimeters.



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



IRIS Dataset



Iris Versicolor



Iris Setosa



Iris Virginica



UNIVERSITÀ
DEGLI STUDI
DI SALERNO



