Andrea Marcelli

Ph.D. Student - DAUIN

Practical intro to Machine Learning in Python with Scikit-learn and AutoML strategies

Outline

ML in practice

Tools

Examples with sklearn

About AutoML

ML in practice

Types of learning

Supervised Learning

- Makes machine learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/ future
- Resolves classification & regression problems

Unsupervised Learning

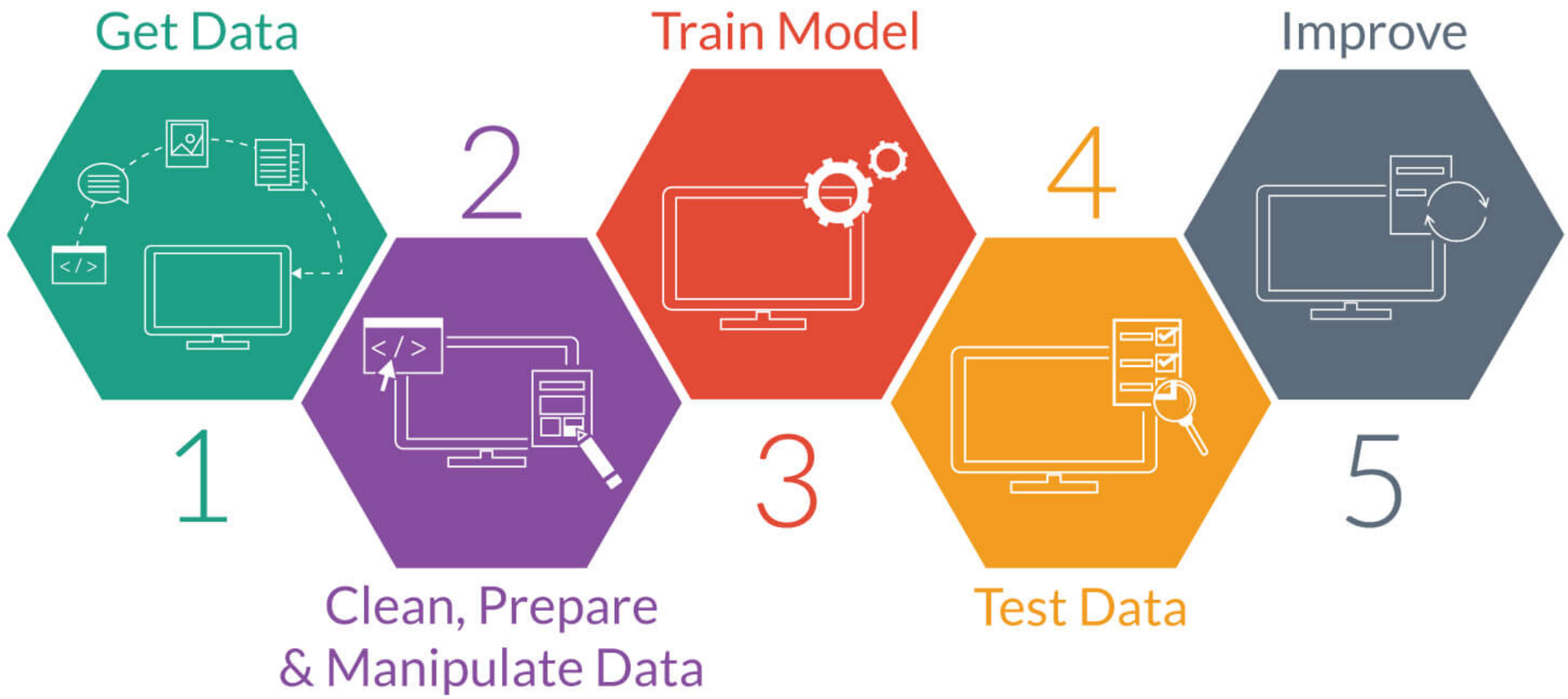
- Machine understands the data (Identifies patterns/ structures)
- Evaluation is qualitative or indirect
- Does not predict / find anything specific

Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning from +ve & -ve reinforcement
- Machine learns how to act in a certain environment
- To maximize rewards

source: https://upxacademy.com/introduction-machine-learning/

Steps to predictive modeling



source: https://upxacademy.com/introduction-machine-learning/

Get the data

Download a dataset or create your own

Web scraping could be necessary

CSV is the most common format

Managing high quantity of data could be challenging
(e.g., data transfer (API limits), storage, preprocessing)

Explore your data

Extract useful knowledge from your data

Visualize your data

Plot all your variables against the target variable being predicted

Compute summary statistics.

Clean, prepare, manipulate data

Convert each column to a fixed type
(e.g., int, float, ascii or unicode strings)

Manage missing data
(e.g., remove incomplete data or assign default values)

Feature selections and normalization
Several ways to encode categorical variables, sequences and text

Feature extraction

Some encodings for categorical data:

Ordinal variables: (e.g., *New York* as 1, *Tehran* as 2 and *New Jersey* as 3)
**beware of the distance meaning*

One hot encoding: each category becomes a binary vector
**can produce very high dimensionality*
**rare values can be collapsed in one category*

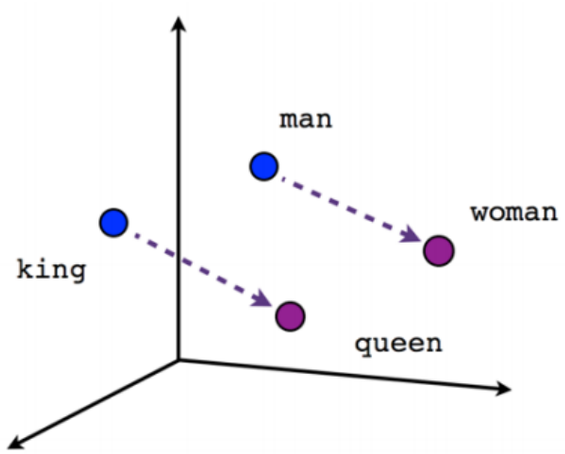
Feature hashing: (e.g., Hash(New York) mod 5 = 3 -> (0,0,1,0,0))
represents categories in a “one hot encoding style” as a sparse matrix but with a much lower dimensions.
**not interpretable*
**hash can generate collision*

Feature extraction - part 2

Encoding from dataset statistics: (e.g., number of occurrences in the dataset, or within the same sample)

Encoding from domain knowledge: (e.g., replace URLs with Alexa rankings)

Extract categories from Word2Vec: categories are in a “one hot encoding style” in a sparse matrix but with a much lower dimensions.
**leverage an unsupervised method*



Feature normalization

If features have very different scales and contain some very large outliers, they can degrade the predictive performance of many machine learning algorithms

example:
StandardScaler removes the mean and scales the data to unit variance.
http://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

Supervised learning phases

Training phase: you present your data from your "gold standard" and train your model, by pairing the input with expected output

Validation phase: look at your models and select the best performing approach using the validation data

Test phase: in order to estimate how well your model has been trained and to estimate model properties

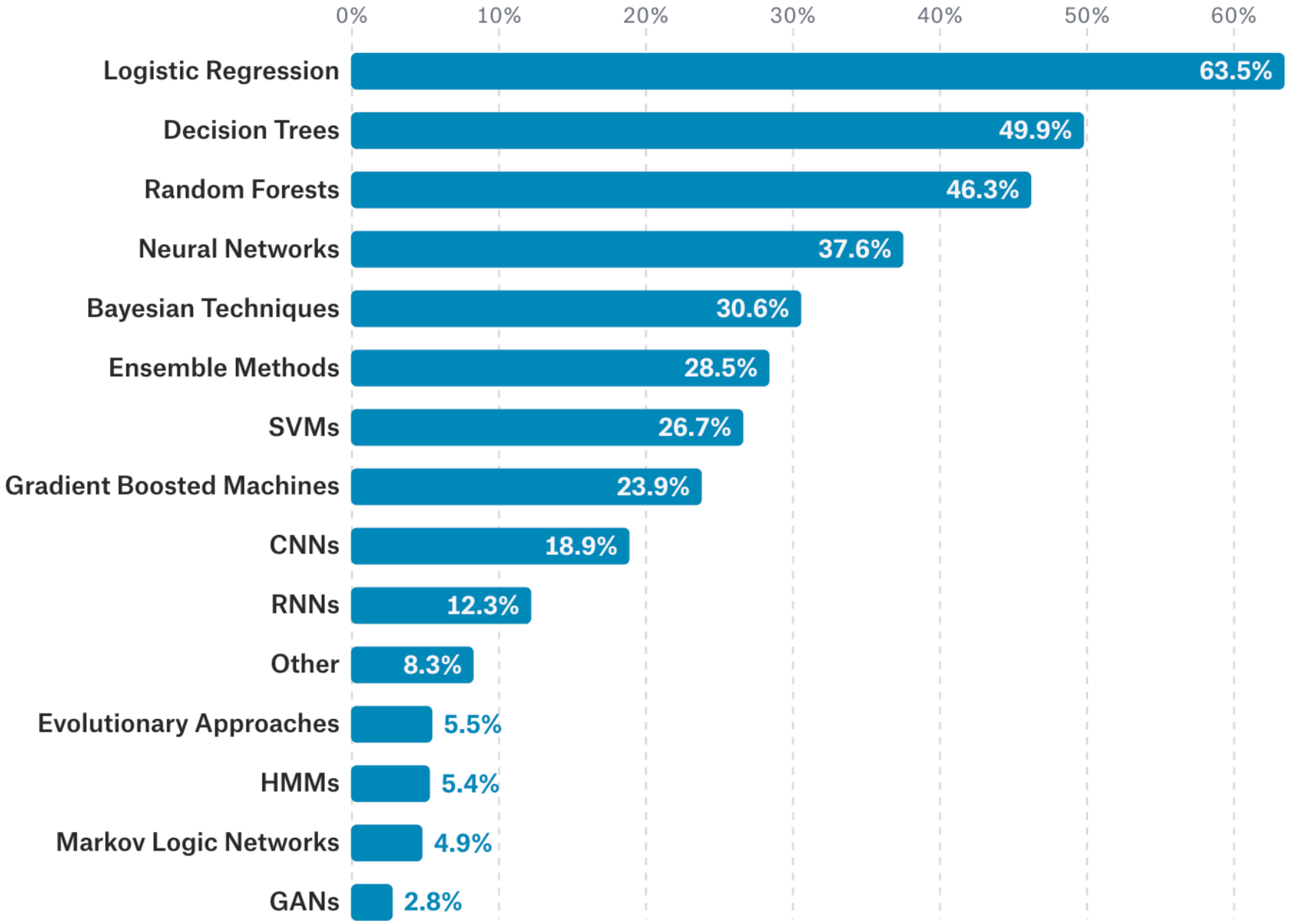
Train the model

Select a model

Initially use default values

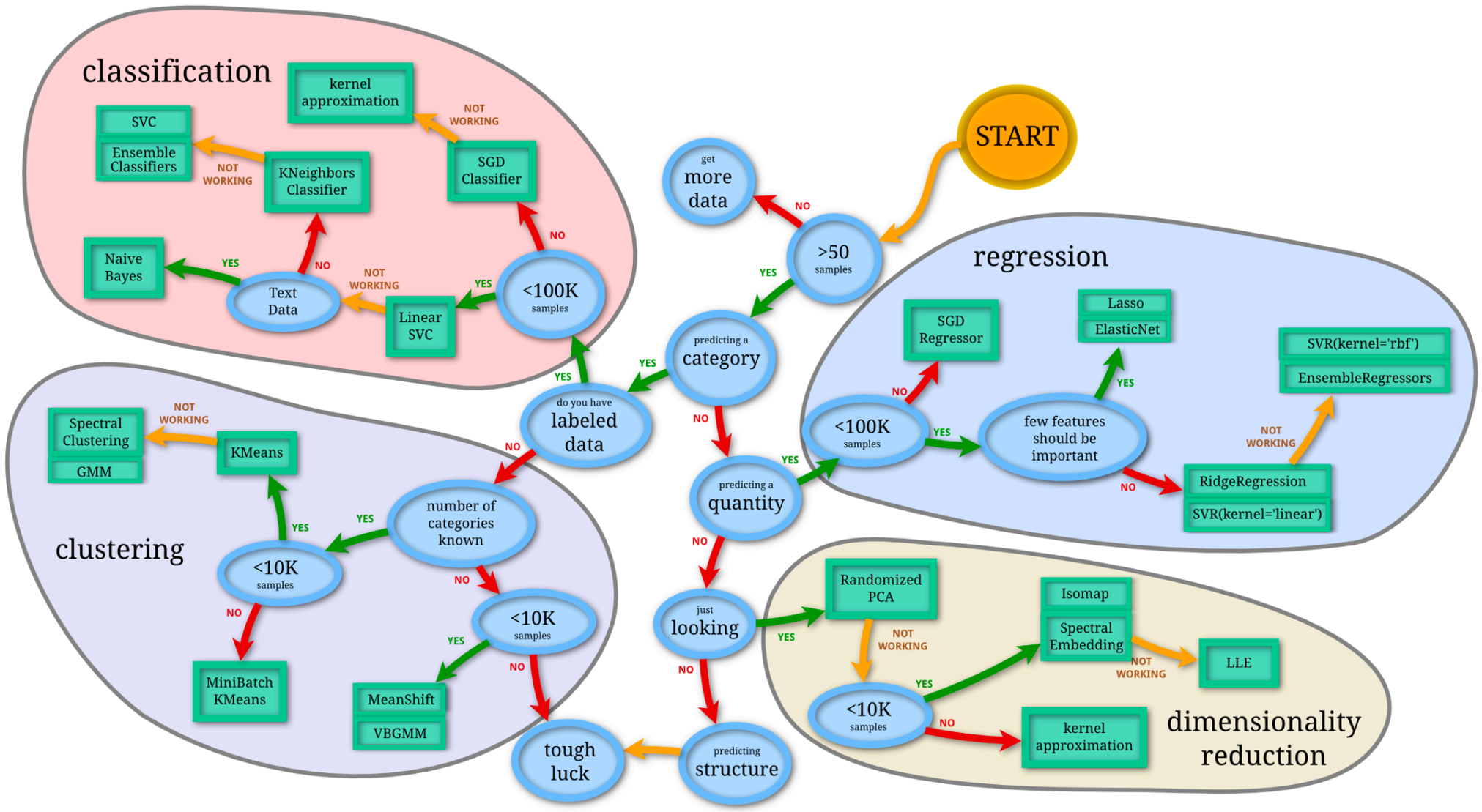
Dimensionality reduction could be applied
(e.g., PCA, auto encoders)

What data science methods are used?



source: https://www.kaggle.com/surveys/2017

Choosing the right estimator



source: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Test data

Use **K-Folds cross-validator**: split data in train/test sets by splitting data into k consecutive folds. Each fold is then used once as a validation while the k - 1 remaining folds form the training set.

Use several **loss**, **score**, and utility functions to measure model performance
(e.g., mean error for numeric predictors, precision, recall, F1 score, ROC curve for classifier)

Be aware of common problems of ML
(e.g., overfitting, course of dimensionality, data leakage)

*Data Leakage is the creation of unexpected additional information in the training data, allowing a model or machine learning algorithm to make unrealistically good predictions.
https://www.kaggle.com/wiki/Leakage

Improve your model

Try several algorithms

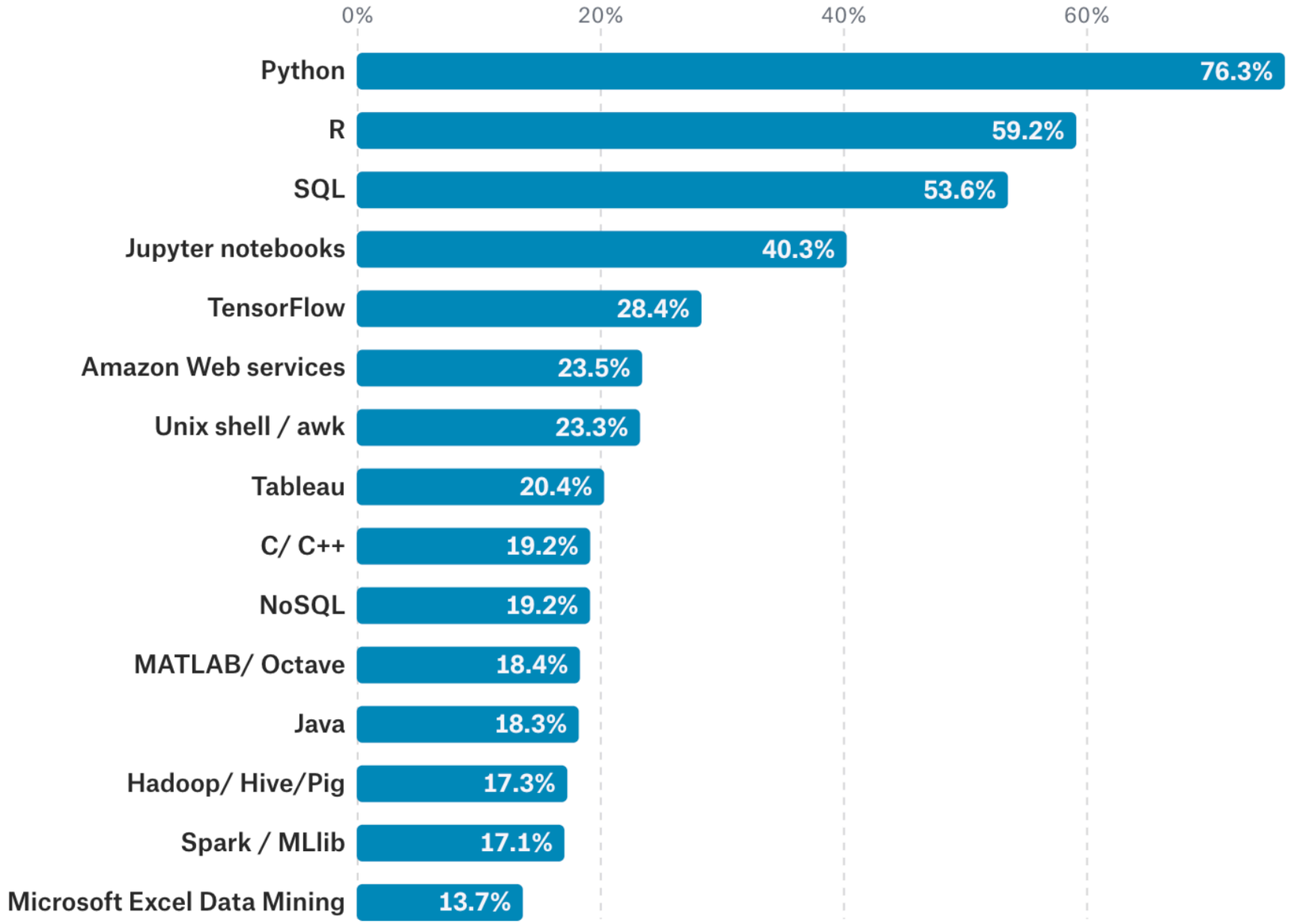
Hyper-parameter Tuning

Try a different distance metric

Try a different set of features

Tools

What tools are used at work?



source: https://www.kaggle.com/surveys/2017

Tools

Python 3, IPython and Jupyter Notebook

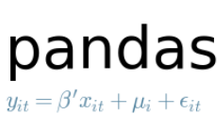
Pandas, SciPy, NumPy, Networkx

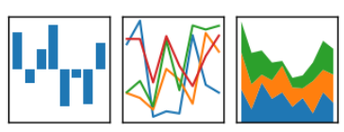
Scrapy, Statsmodel

Matplotlib, Seaborn, Bokeh

Scikit-learn, Keras (TensorFlow or Theano)

NLTK, Gensim





How to install all the packages?

Manual installation with **pip**

or install **Anaconda**

https://docs.anaconda.com/anaconda/packages/py3.6_osx-64

Use case #1

You need to install different version of the same package on your system:

Use python **virtualenv**, an isolated working copy of Python

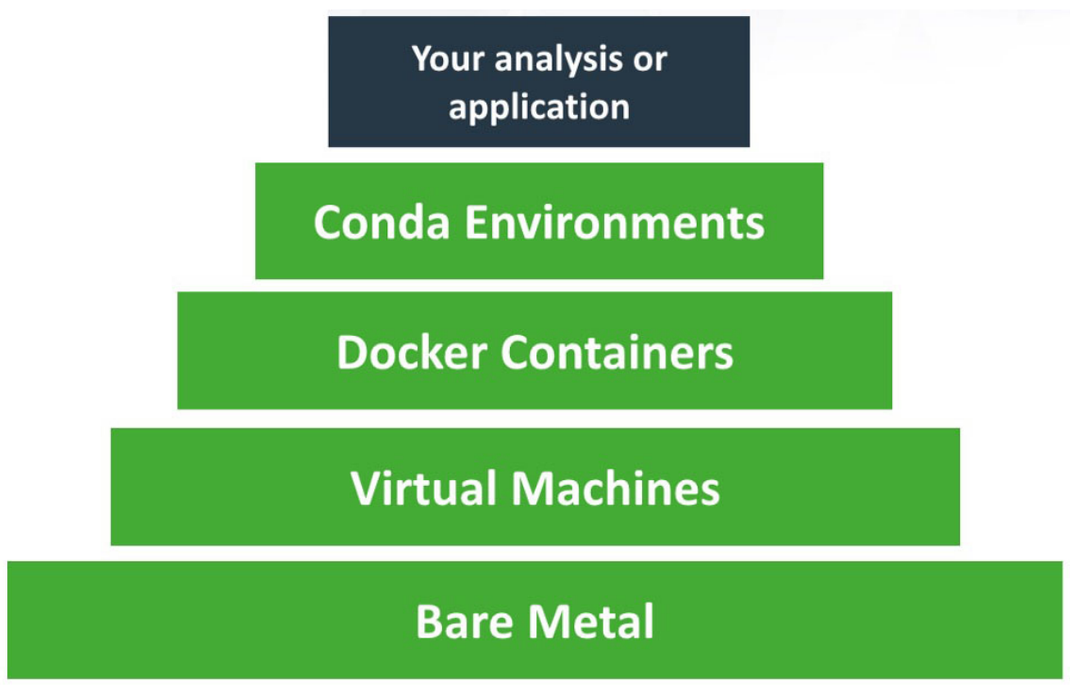
```
$ mkdir vvenv
$ virtualenv vvenv/my_app
$ source vvenv/my_app/bin/activate
(my_app) $ pip install networkx==1.9
(my_app) $ python3 -c "import networkx as nx; print(nx.__version__)”
1.9
(my_app) $ deactivate
```

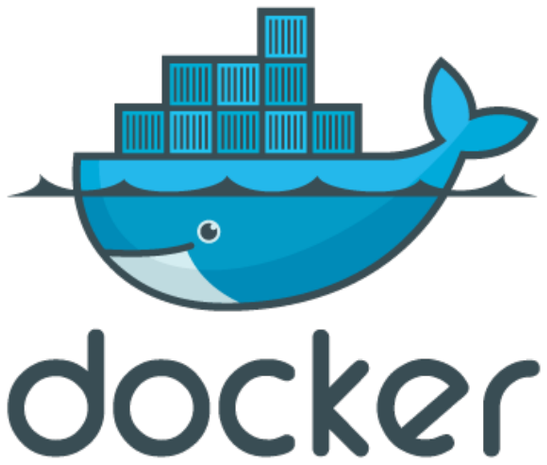
Packages are installed in: `vvenv/my_app/lib/python3.6/site-packages/`

Use case #2

You need to easily reproduce your result on different systems:

Use a **Docker** container
https://hub.docker.com/r/continuumio/anaconda3/





Examples with sklearn

Some examples

Linear regression

https://github.com/justmarkham/scikit-learn-videos/blob/master/06_linear_regression.ipynb

`http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv`

Classification

https://www.kaggle.com/ash316/ml-from-scratch-part-2

`https://www.kaggle.com/uciml/pima-indians-diabetes-database/downloads/diabetes.csv`

TPOT

https://github.com/jimmy-sonny/practical-intro-ml/blob/master/sklean%20LinearRegression%20vs%20TPOT.ipynb

AutoML

AutoML

ML success crucially relies on human experts to perform the following tasks:

Preprocess the data

Select appropriate features

Select an appropriate model family

Optimize model hyperparameters

Postprocess machine learning models

Critically analyze the results obtained.

source: http://www.ml4aad.org/automl/

AutoML

There is a growing community around creating tools that study how to automate the tasks that are part of the machine learning workflow

The scope of AML is ambitious, however, is it really effective?
It depends: most machine learning problems require domain knowledge and human judgement to set up correctly

Tasks like **exploratory data analysis**, **pre-processing of data**, **hyper-parameter tuning**, **model selection** and **putting models into production** can be automated to some some extent with an Automated Machine Learning framework.

source: https://medium.com/airbnb-engineering/automated-machine-learning-a-paradigm-shift-that-accelerates-data-scientist-productivity-airbnb-f1f8a10d61f8

AutoML

More info at:

https://blog.keras.io/the-future-of-deep-learning.html

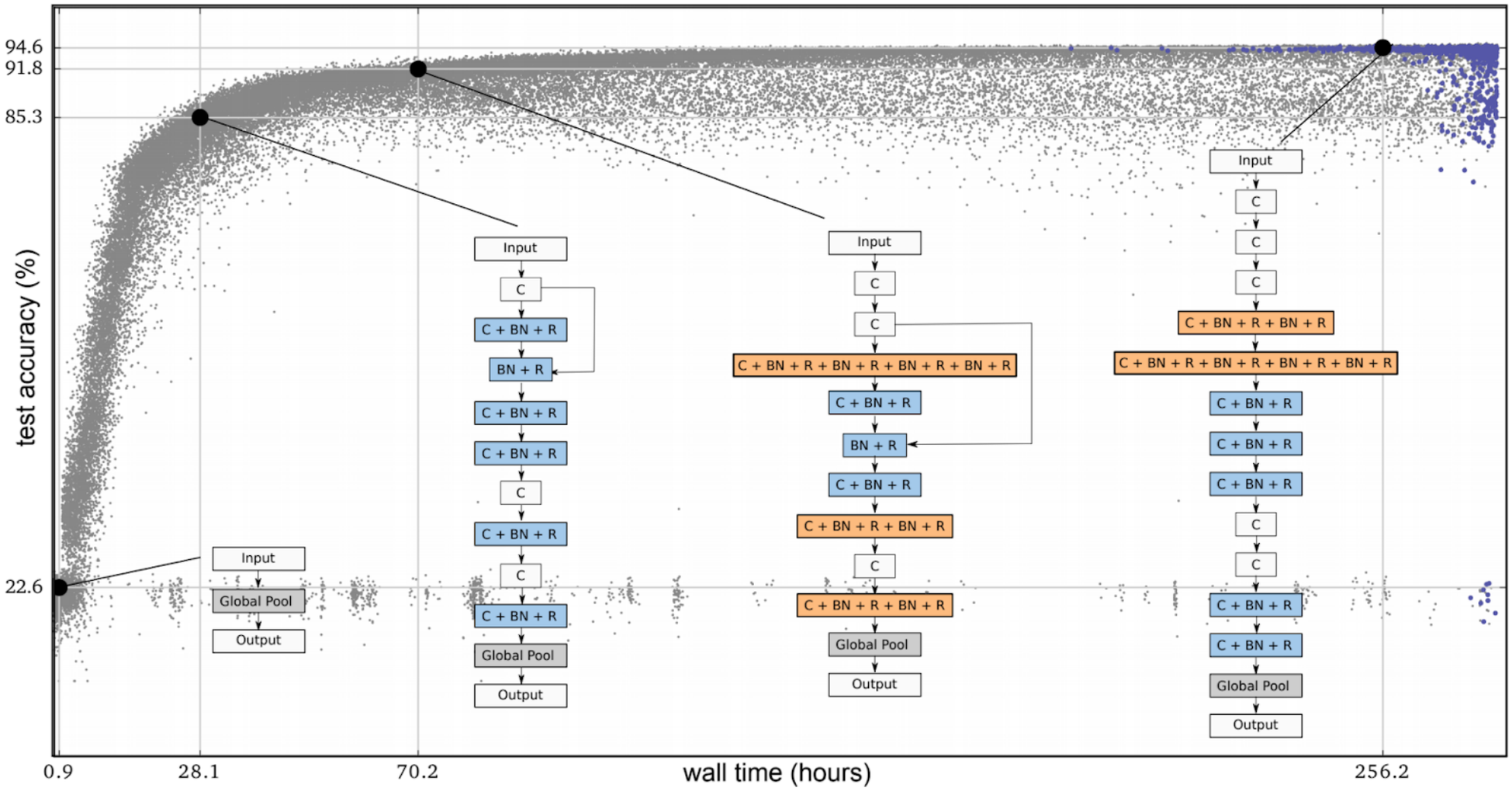
AutoML with sklearn

TPOT
https://github.com/EpistasisLab/tpot

auto-sklearn
https://github.com/automl/auto-sklearn

machineJS
https://github.com/ClimbsRocks/machineJS

AutoML with NN



source: https://research.googleblog.com/2018/03/using-evolutionary-automl-to-discover.html

AutoML with NN

auto-ml
https://github.com/ClimbsRocks/auto_ml

autokeras
https://github.com/jhfjhfj1/autokeras

Andrea Marcelli

Ph.D. Student

andrea.marcelli@polito.it

jimmy-sonny.github.io