

Machine Learning via Optimisation Approach

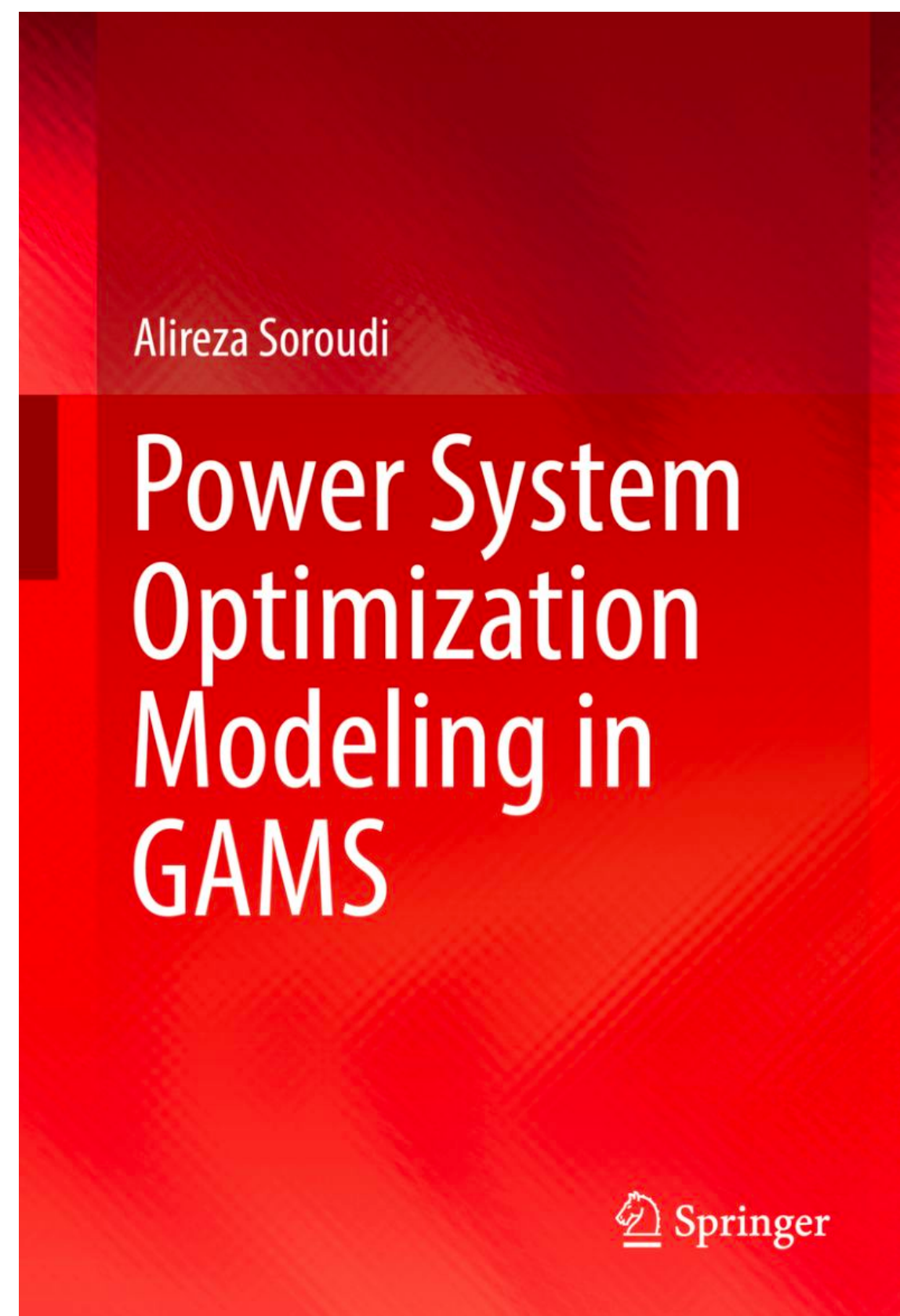
Alireza Soroudi

alireza.soroudi@gmail.com

www.linkedin.com/in/soroudi/

 github.com/OptimizationExpert

How it started

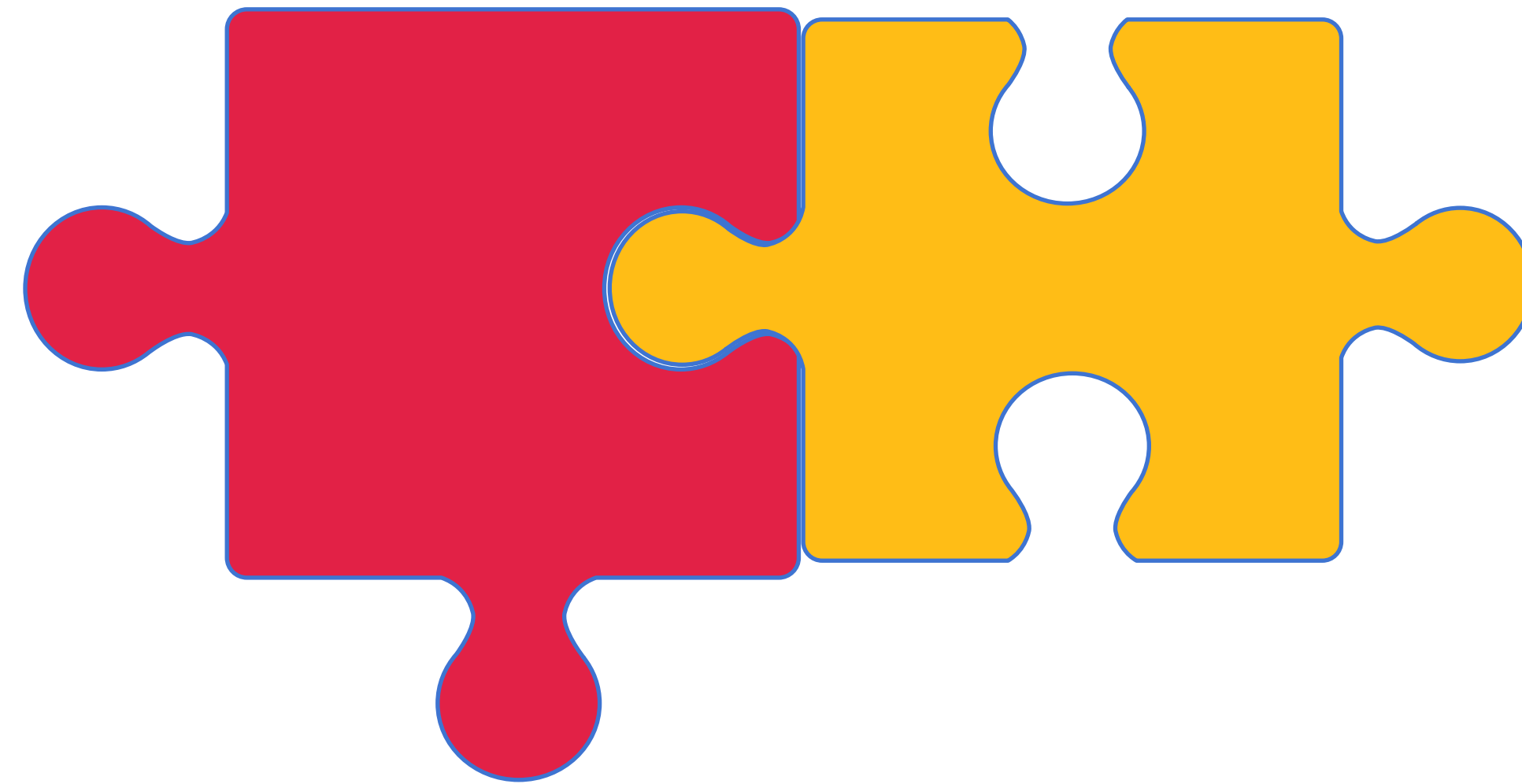


How it's going





Machine Learning - Operations Research



Course plan

- Understand the concept of machine learning method
- Mathematical formulation of ML problem
- Solving the ML problem





Course plan



Session 1

- **Introduction**
- **Pyomo**
- **Some python tips**
- **GoogleColab**
- **Q&A**





What is Machine Learning ?

- **Learn from experience**
- **Identify the invisible patterns**

What is Machine Learning ?

Application

APOSTILLE (Convention de La Haye du 5 octobre 1961)	
1. Country: Pays / Pais:	United Kingdom of Great Britain and Northern Ireland
This public document Le présent acte public / El presente documento público	
2. Has been signed by a été signé par ha sido firmado por	
3. Acting in the capacity of agissant en qualité de quien actúa en calidad de	
4. Bears the seal / stamp of est revêtu du sceau / timbre de y está revestido del sello / timbre de	
Certified Attesté / Certificado	
5. at à / en	London
6. the le / el día	
7. by par / por	Her Majesty's Principal Secretary of State for Foreign and Commonwealth Affairs
8. Number sous no / bajo el numero	APO-
9. Seal / stamp Sceau / timbre Sello / timbre	10. Signature Signature Firma



This Apostille is not to be used in the UK and only confirms the authenticity of the signature, seal or stamp on the attached UK public document. It does not confirm the authenticity of the underlying document. Apostilles attached to documents that have been photocopied and certified in the UK confirm the signature of the UK official who conducted the certification only. It does not authenticate either the signature on the original document or the contents of the original document in any way.

If this document is to be used in a country not party to the Hague Convention of the 5th of October 1961, it should be presented to the consular section of the mission representing that country

To verify this apostille go to www.verifyapostille.service.gov.uk

Apostille (Convention de La Haye du 5 octobre 1961)	
1. Country: Pays / Pais:	United Kingdom of Great Britain and Northern Ireland
This public document Le présent acte public / El presente documento público	
2. Has been signed by a été signé par ha sido firmado por	
3. Acting in the capacity of agissant en qualité de quien actúa en calidad de	
4. Bears the seal / stamp of est revêtu du sceau / timbre de y está revestido del sello / timbre de	
Certified Attesté / Certificado	
5. at à / en	Milton Keynes
6. the le / el día	
7. by par / por	Her Majesty's Principal Secretary of State for Foreign and Commonwealth Affairs
8. Number sous no / bajo el numero	APO-
9. Seal / stamp Sceau / timbre Sello / timbre	10. Signature Signature Firma



This Apostille is not to be used in the UK and only confirms the authenticity of the signature, seal or stamp on the attached UK public document. It does not confirm the authenticity of the underlying document. Apostilles attached to documents that have been photocopied and certified in the UK confirm the signature of the UK official who conducted the certification only. It does not authenticate either the signature on the original document or the contents of the original document in any way.

If this document is to be used in a country not party to the Hague Convention of the 5th of October 1961, it should be presented to the consular section of the mission representing that country

To verify this apostille go to www.verifyapostille.service.gov.uk

What is Machine Learning ?

Application



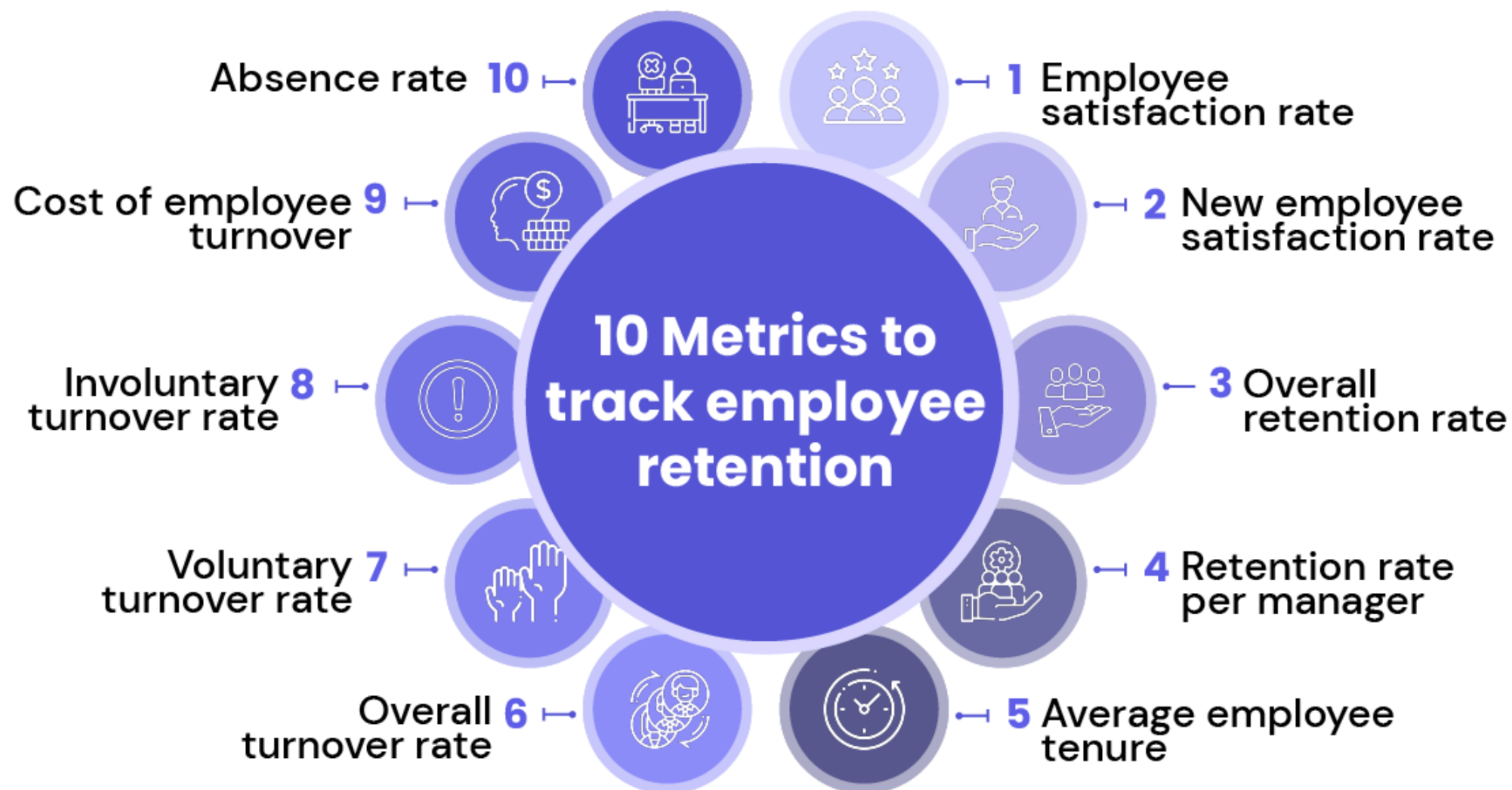
What is Machine Learning ?

Application



What is Machine Learning ?

Application



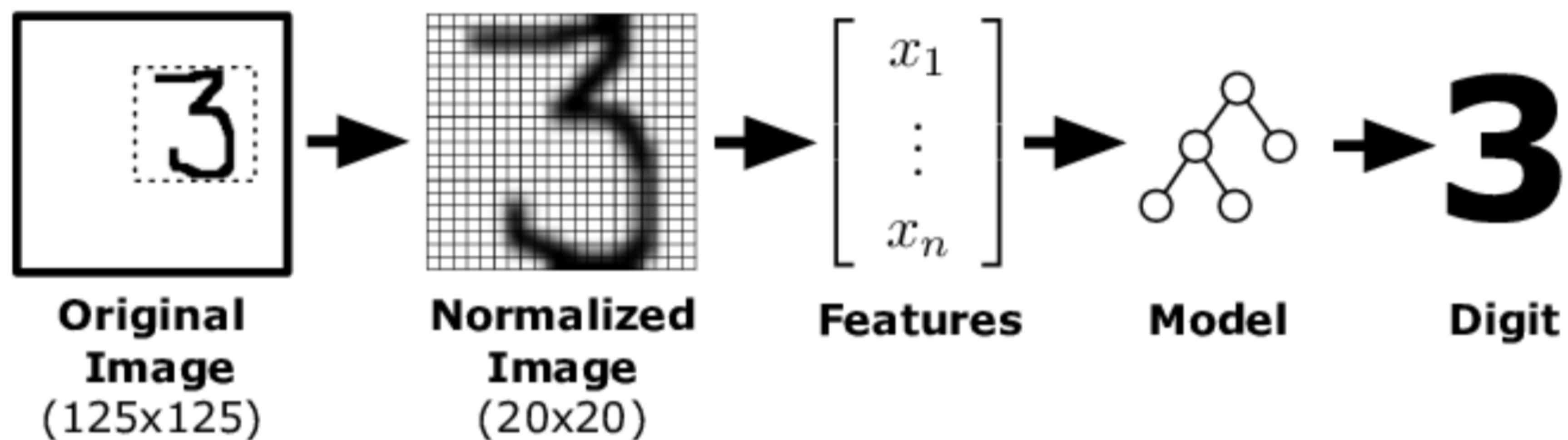
What is Machine Learning ?

Application



What is Machine Learning ?

Application



Machine Learning Packages

Scikit-learn

scikit-learn.org/stable/

scikit-learn
Machine Learning in Python

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

Getting Started Release Highlights for 1.5

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)

Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)

Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

Algorithms: [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)

Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, increased efficiency.

Algorithms: [PCA](#), [feature selection](#), [non-negative matrix factorization](#), and [more...](#)

Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning.

Algorithms: [Grid search](#), [cross validation](#), [metrics](#), and [more...](#)

Examples

Preprocessing

Feature extraction and normalization.

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

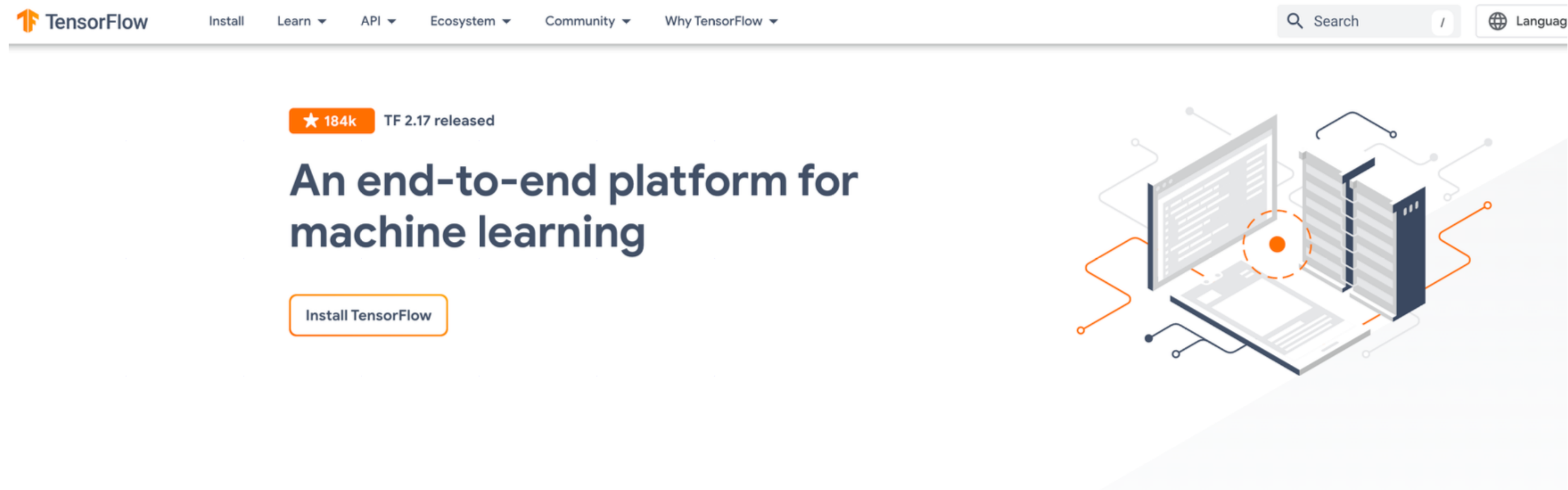
Algorithms: [Preprocessing](#), [feature extraction](#), and [more...](#)

Examples



Machine Learning Packages

Tensorflow





Machine Learning Packages

Keras

Keras

About Keras

Getting started

Developer guides

Keras 3 API documentation

Keras 2 API documentation

Code examples

Computer Vision

Natural Language Processing

Structured Data

Timeseries

Generative Deep Learning

Audio Data

Reinforcement Learning

Graph Data

Quick Keras Recipes

KerasTuner: Hyperparameter Tuning

KerasCV: Computer Vision Workflows

KerasNLP: Natural Language Workflows

► Code examples

Code examples

Our code examples are short (less than 300 lines of code), focused demonstrations of vertical deep learning workflows.

All of our examples are written as Jupyter notebooks and can be run in one click in [Google Colab](#), a hosted notebook environment that requires no setup and runs in the cloud. Google Colab includes GPU and TPU runtimes.

★

 = Good starter example

V3

 = Keras 3 example

Computer Vision

Image classification

★

V3

 Image classification from scratch

★

V3

 Simple MNIST convnet

★

V3

 Image classification via fine-tuning with EfficientNet

V3

 Image classification with Vision Transformer

V3

 Classification using Attention-based Deep Multiple Instance Learning

V3

 Image classification with modern MLP models

V3

 A mobile-friendly Transformer-based model for image classification

V3

 Pneumonia Classification on TPU

V3

 Compact Convolutional Transformers

V3

 Image classification with ConvMixer

V3

 Image classification with EANet (External Attention Transformer)

nail.com

Optimisation Tools





Optimisation Package

Pyomo



```
model.Locations = RangeSet(1,model.N)
model.Customers = RangeSet(1,model.M)
model.d = Param(model.Locations, model.Customers, initialize=lambda n, m, model : random.uniform(0.0,1.0))

model.x = Var(model.Locations, model.Customers, bounds=(0.0,1.0))
model.y = Var(model.Locations, within=Binary)

def rule(model):
    return sum( (model.d[n,m]*model.x[n,m] for n in model.Locations for m in model.Customers))
model.obj = Objective(rule=rule)

def rule(model, m):
    return (sum( (model.x[n,m] for n in model.Locations)), 1.0)
model.single_x = Constraint(model.Customers, rule=rule)

def rule(model, n,m):
    return (None, model.x[n,m] - model.y[n], 0.0)
model.bound_y = Constraint(model.Locations, model.Customers, rule=rule)

def rule(model):
    return (sum( (model.y[n] for n in model.Locations) ) - model.P, 0.0)
model.num_facilities = Constraint(rule=rule)
```

Flexible modeling of optimization problems in Python

What Is Pyomo?

Pyomo is a Python-based, open-source optimization modeling language with a diverse set of optimization capabilities.

[Read More](#)

Installation

The easiest way to install Pyomo is to use pip. Pyomo also needs access to optimization solvers.

[Read more](#)

Docs and Examples

Pyomo documentation and examples are available online. Pyomo is also described in book and journal publications.

[Read more](#)

latest release **v6.7.3**



Different Machine Learning Methods

Supervised Learning (classification)

- KNN, ANN
- Regression
- SVM

Set of labeled examples to learn from:
training data

Develop model from training data
Use model to make predictions about new
data Unsupervised machine learning

- **Unsupervised Learning**
 - Clustering

Unlabelled data, look for patterns or
structure

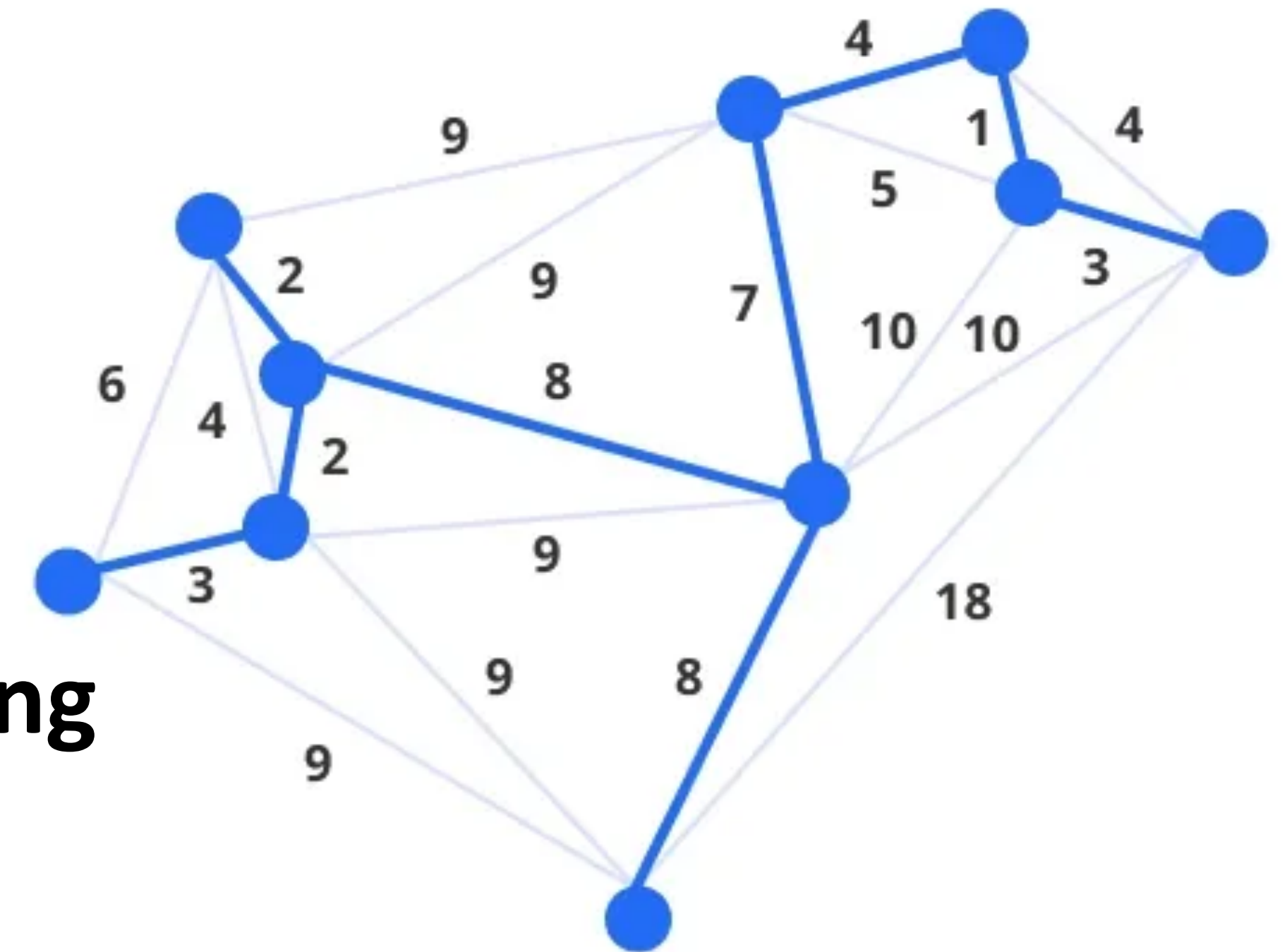
- **PCA**

What makes this course unique ?



Optimisation in Python

- Introduction to optimisation
- Optimisation problem modelling
- Open source/commercial optimisation tools
- Application of optimisation tools in Engineering
- Useful resources for learning optimisation
- Academia/Industry pathway for optimisation experts



Skill Levels



بهبود مدلسازی ریاضی

مدلسازی یک مساله به فرم ریاضی

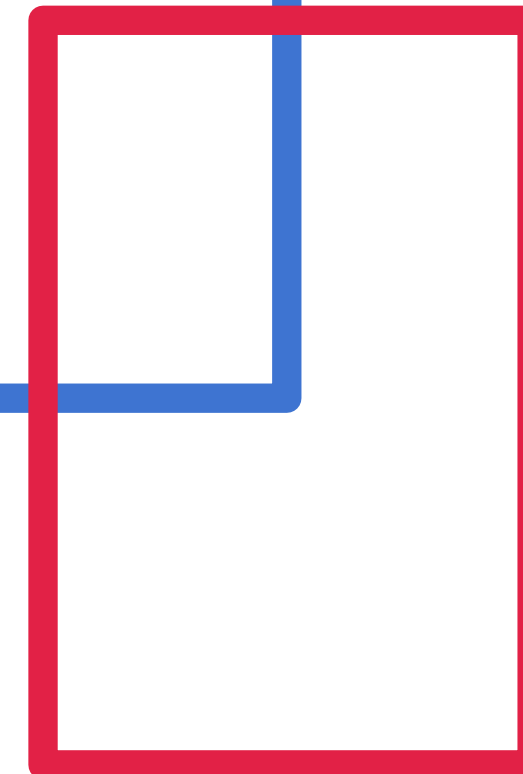
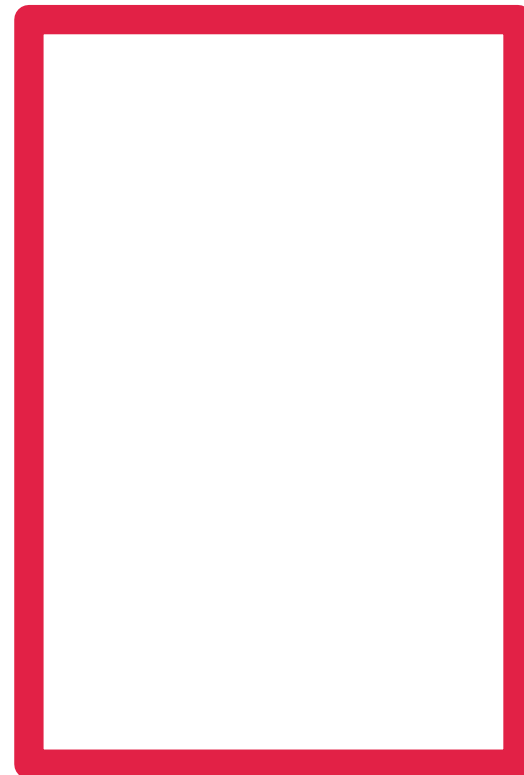
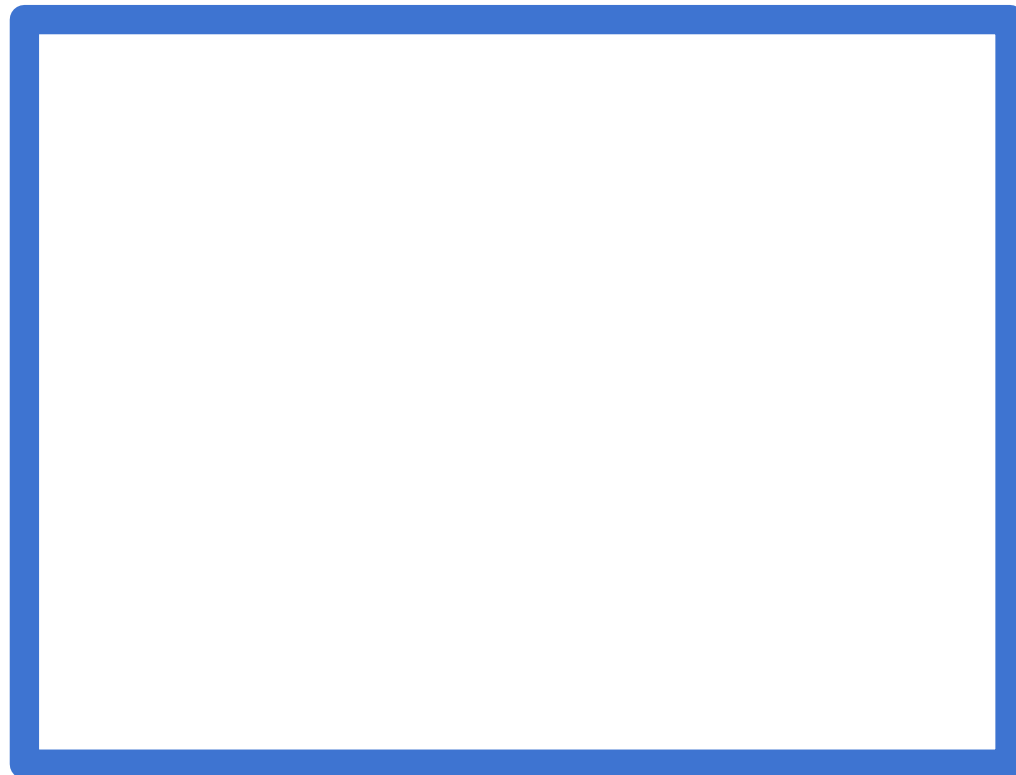
کد کردن یک الگوریتم ریاضی

کد کردن یک مدل ریاضی

فهمیدن یک کد آماده

Example

Are these rectangles overlapping ?



W

(X,Y)

L

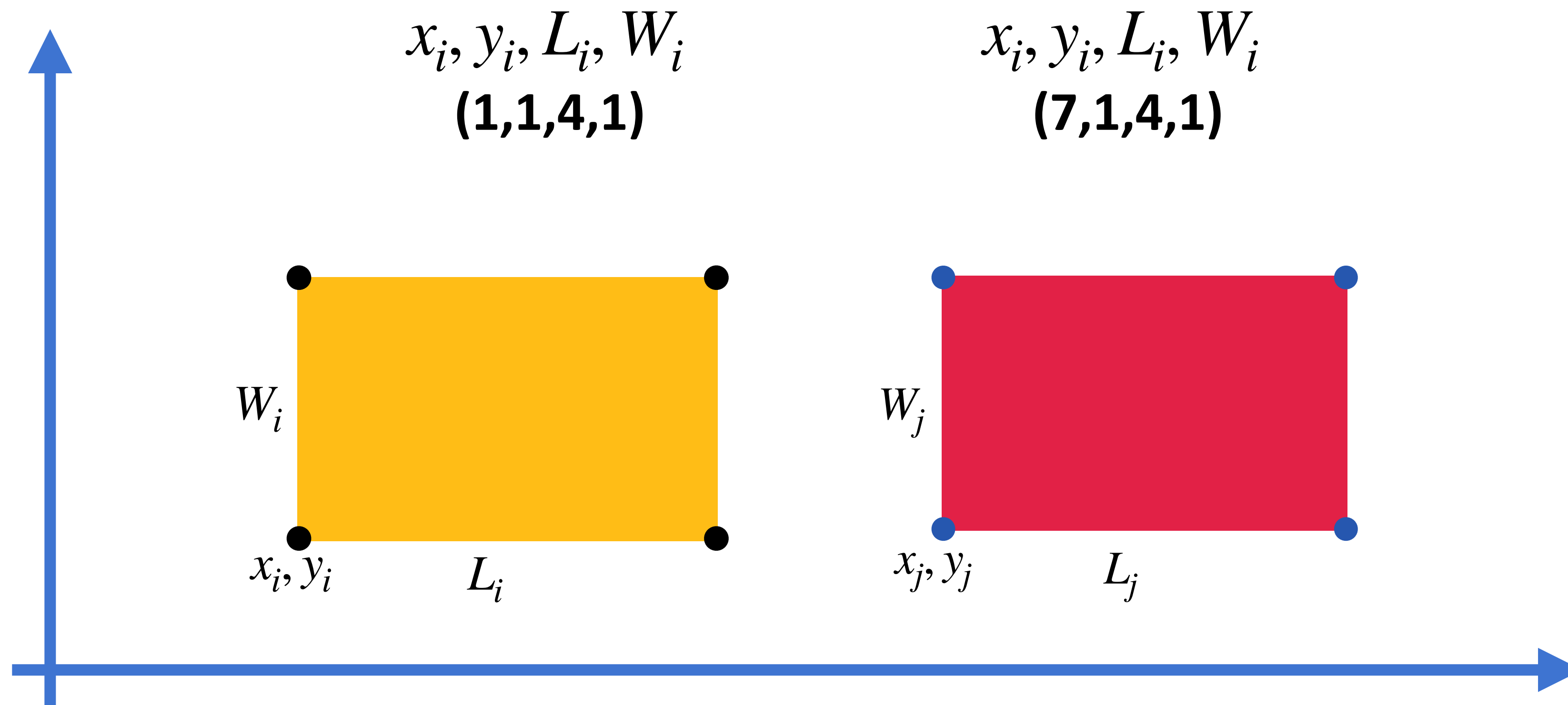
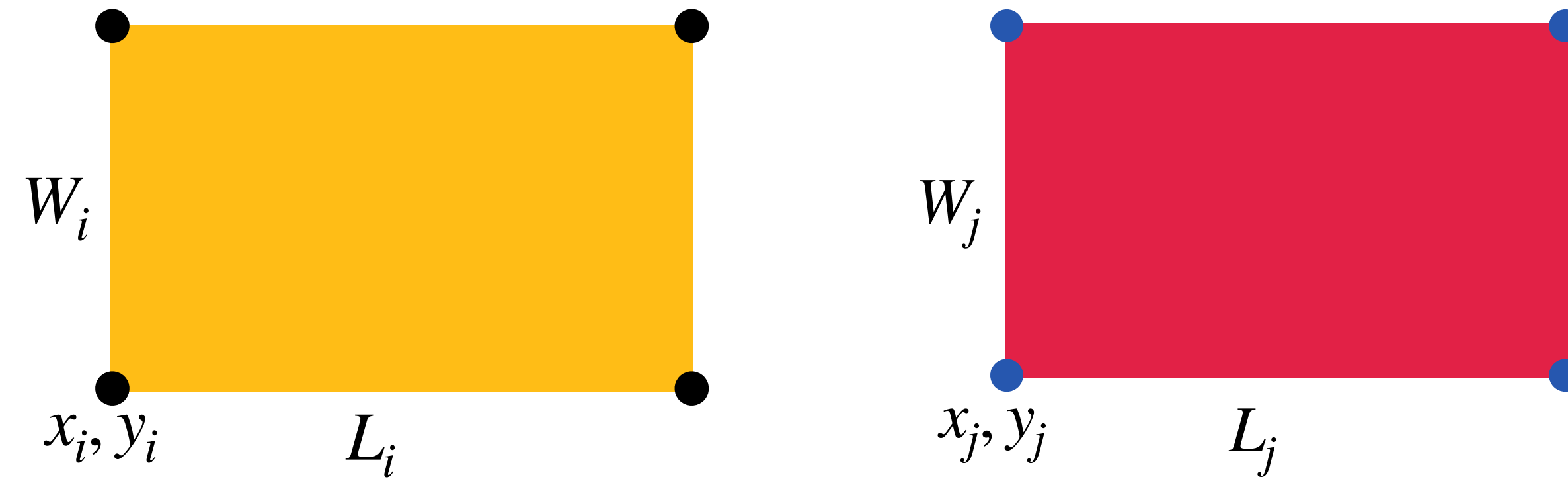


Diagram illustrating a sequence of steps in a spiral pattern, labeled with variables x_i, y_i, L_i, W_i .

The sequence consists of 12 steps, arranged in a spiral pattern:

- Step 1: x_i, y_i, L_i, W_i
- Step 2: x_i, y_i, L_i, W_i
- Step 3: x_i, y_i, L_i, W_i
- Step 4: x_i, y_i, L_i, W_i
- Step 5: x_i, y_i, L_i, W_i
- Step 6: x_i, y_i, L_i, W_i
- Step 7: x_i, y_i, L_i, W_i
- Step 8: x_i, y_i, L_i, W_i
- Step 9: x_i, y_i, L_i, W_i
- Step 10: x_i, y_i, L_i, W_i
- Step 11: x_i, y_i, L_i, W_i
- Step 12: x_i, y_i, L_i, W_i

A blue arrow points from the left towards the sequence.



$$x_i + L_i - M(1 - R_{i,j}) \leq x_j$$

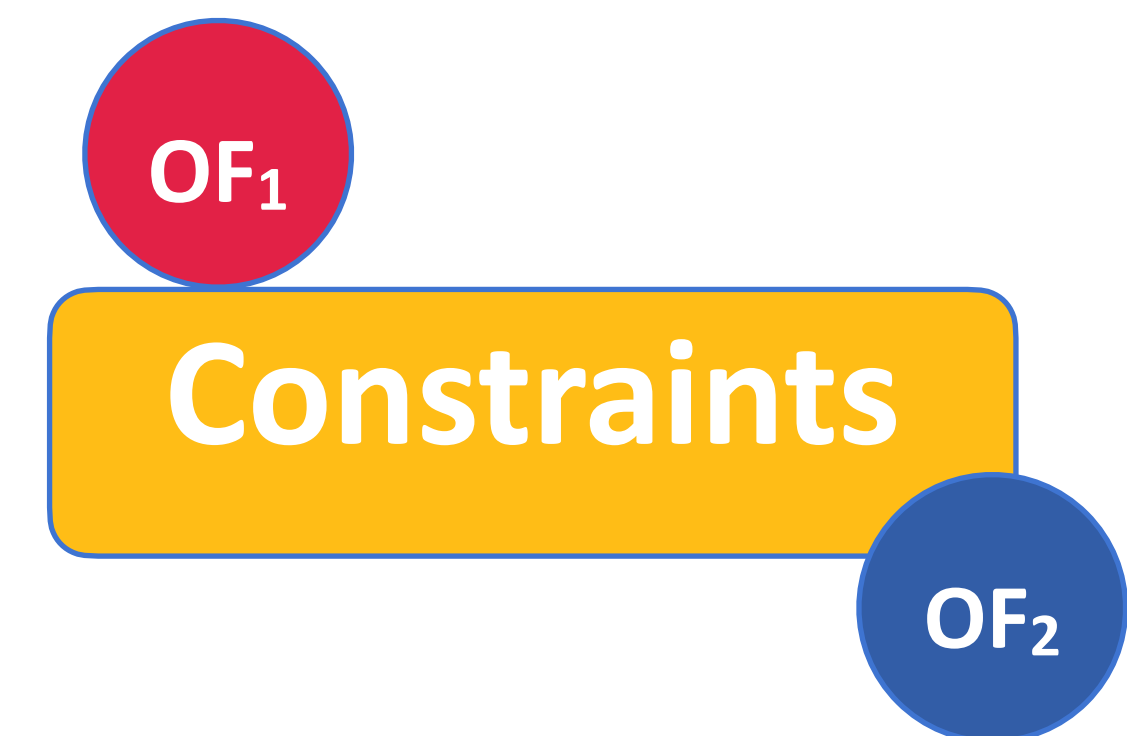
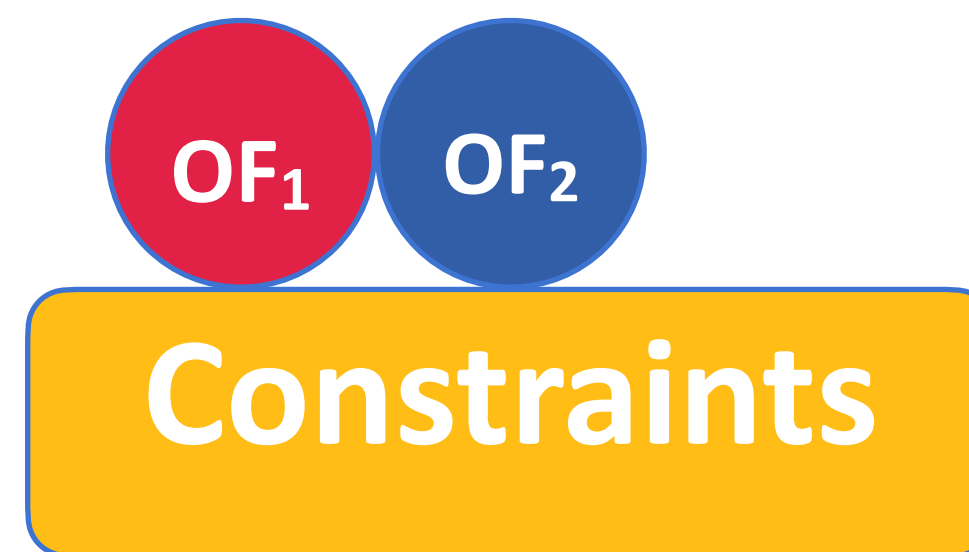
$$x_i + L_i + MR_{i,j} \geq x_j$$

$$x_i \geq 0$$

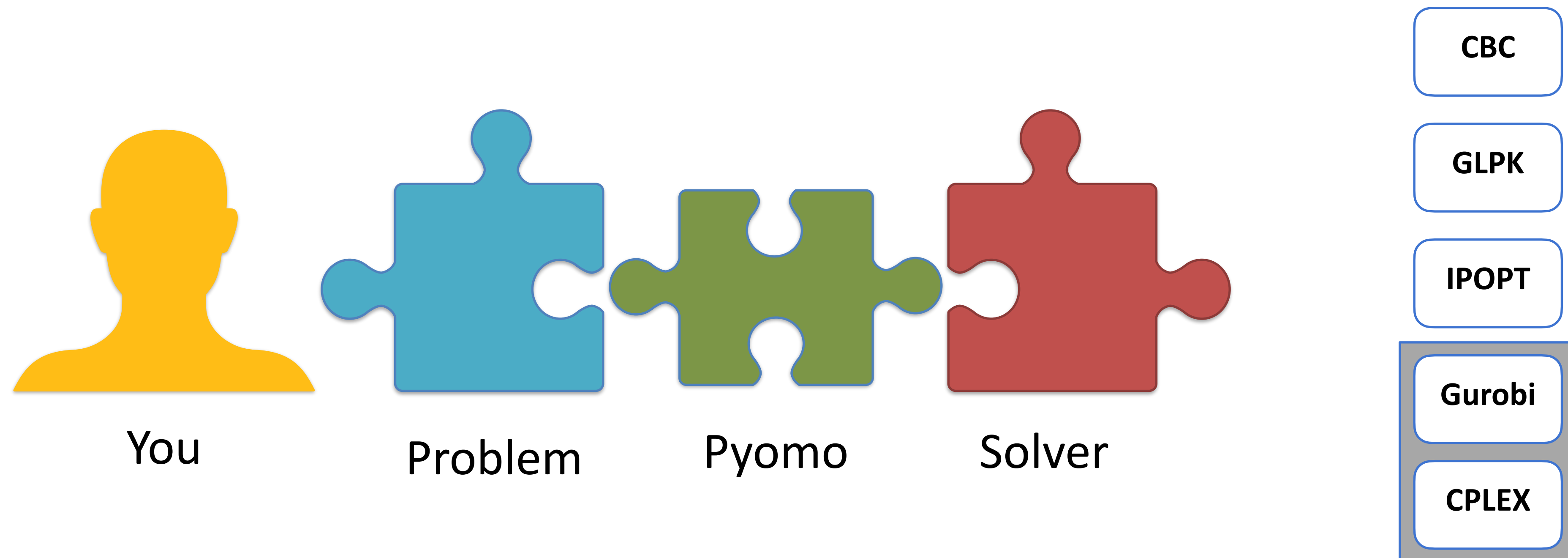
$$R_{i,j} \in \{0,1\}$$

J is in the right
hand side of I

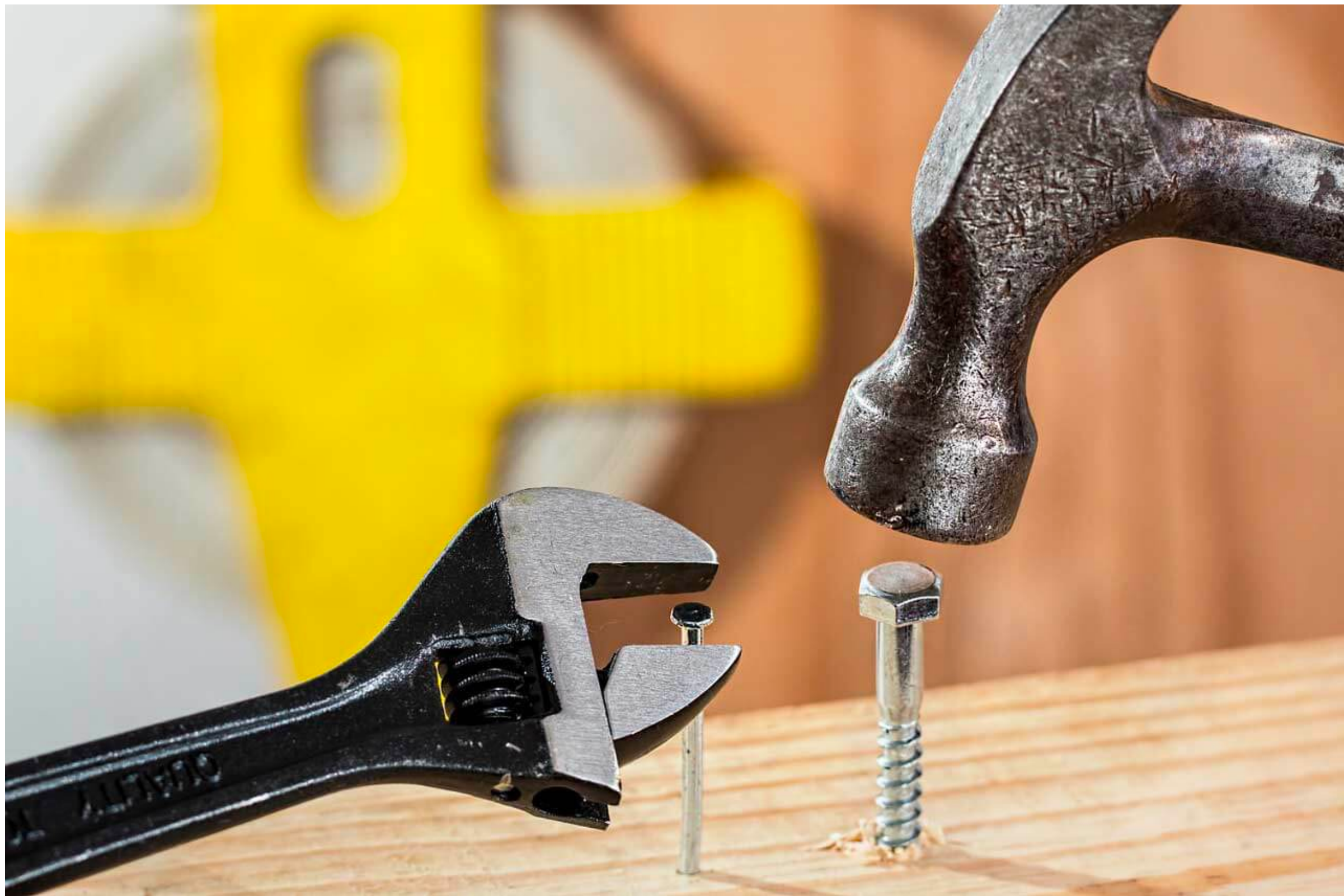
Optimisation Types



Optimisation modelling steps

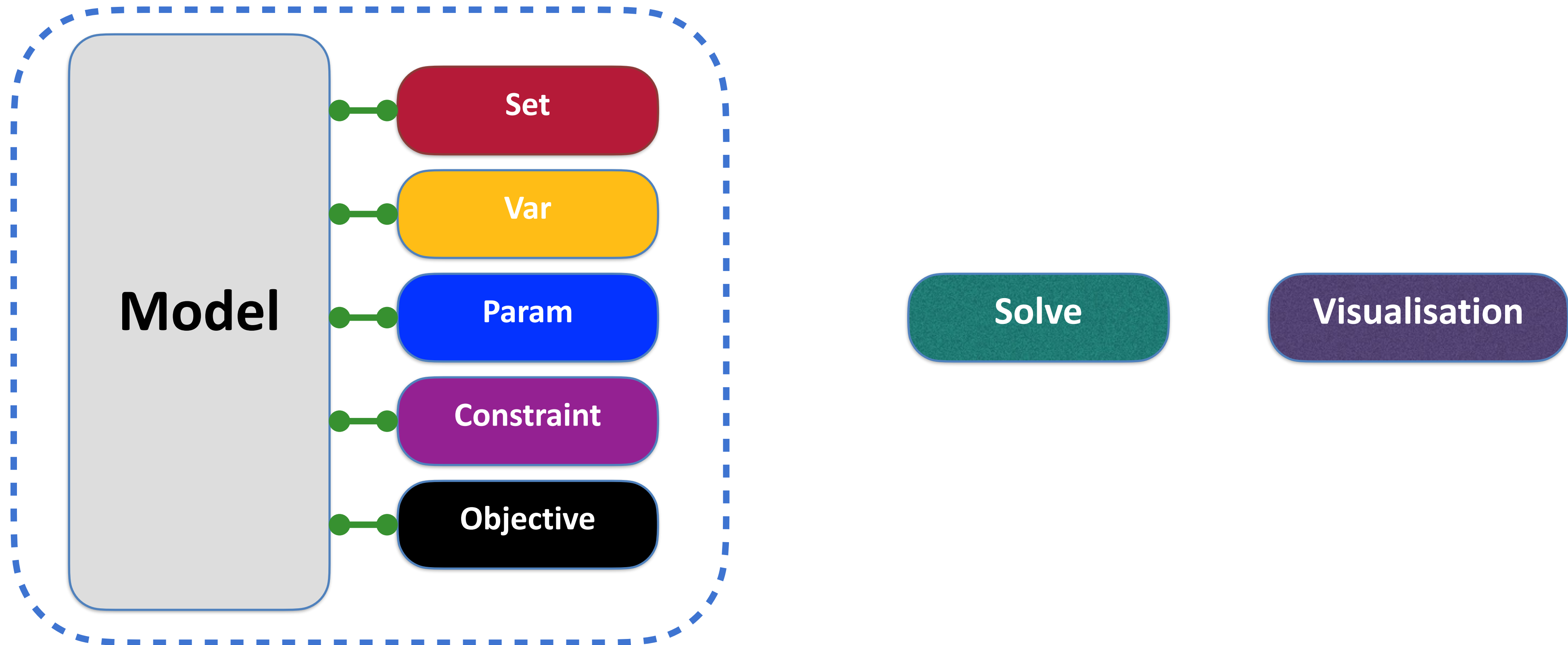


Right Tool for Right Problem

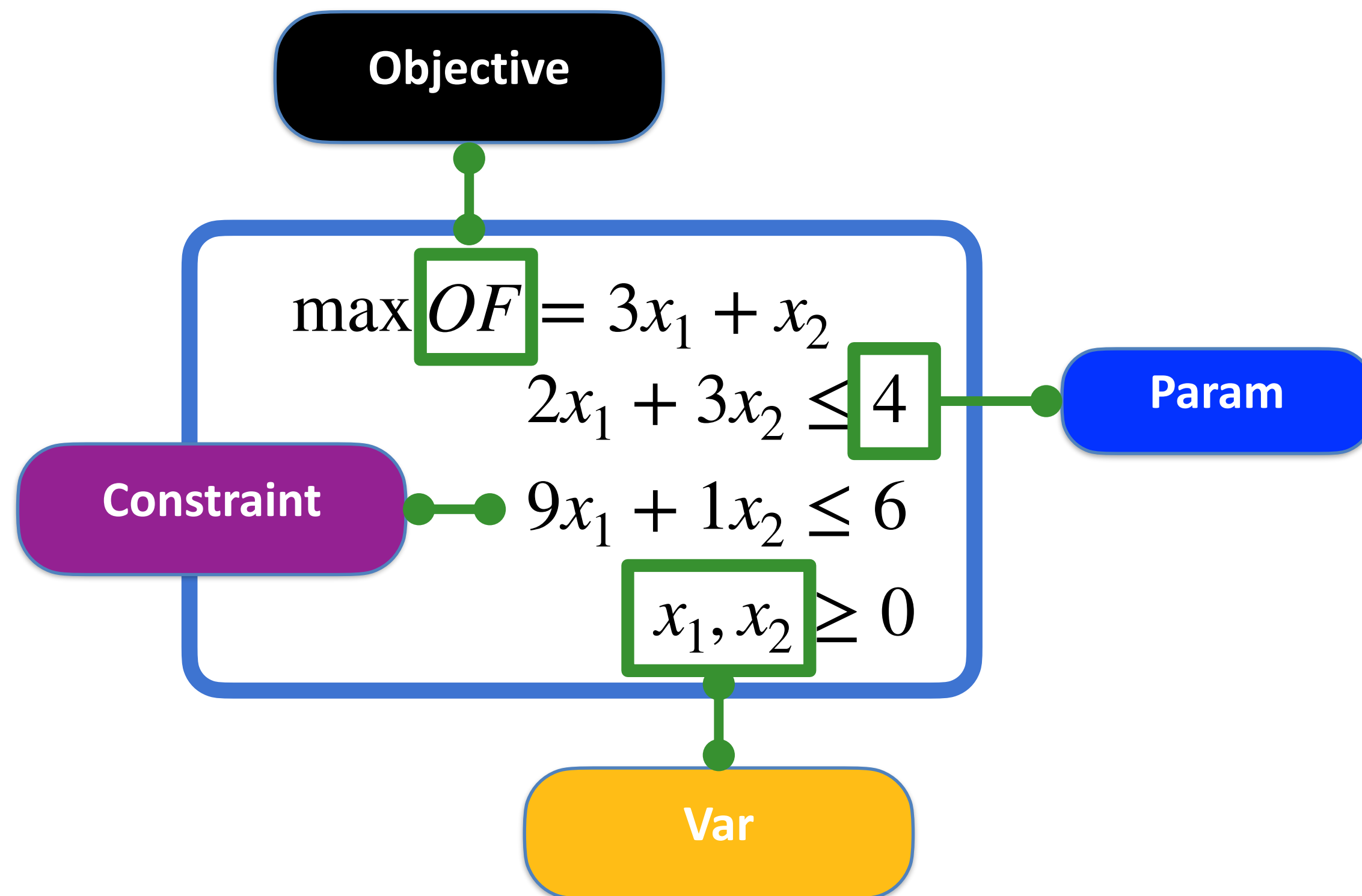


	LP	MIP	MIQP	QCP	NLP
CBC	✓	✓			
GLPK	✓	✓			
IPOPT	✓				✓
Gurobi	✓	✓	✓	✓	
CPLEX	✓	✓	✓	✓	

Pyomo Elements



Pyomo Example



Python code

```
[1]: 1 from pyomo.environ import *

[2]: 1 model = AbstractModel()
      2 model.x1 = Var(initialize=0, within=NonNegativeReals)
      3 model.x2 = Var(initialize=0, within=NonNegativeReals)
      4 model.C1 = Constraint(expr=2*model.x1+3*model.x2<=4)
      5 model.C2 = Constraint(expr=9*model.x1+ model.x2<=6)
      6 model.obj1 = Objective(expr=3*model.x1+model.x2, sense=maximize)

[3]: 1 opt = SolverFactory('glpk')
      2 instance = model.create_instance()

[4]: 1 results = opt.solve(instance) |
      2 print('OF= ',value(instance.obj1))

OF= 2.64
```

Code

Pyomo Example

Minimize $Z = 2x_1 + 3x_2 - x_1^2 - 2x_2^2$
subject to $x_1 + 3x_2 \leq 6$
 $5x_1 + 2x_2 \leq 10$
 $x_i \geq 0, i = 1, 2.$

[Code](#)

Session 2

Graph Colouring

- Find the minimum number of colours for painting this graph
no two connected nodes have the same colours

$$\min_{X_{i,c}, y} OF = y$$

$$\forall i, j, c \quad (X_{i,c} + X_{j,c})L_{i,j} \leq 1$$

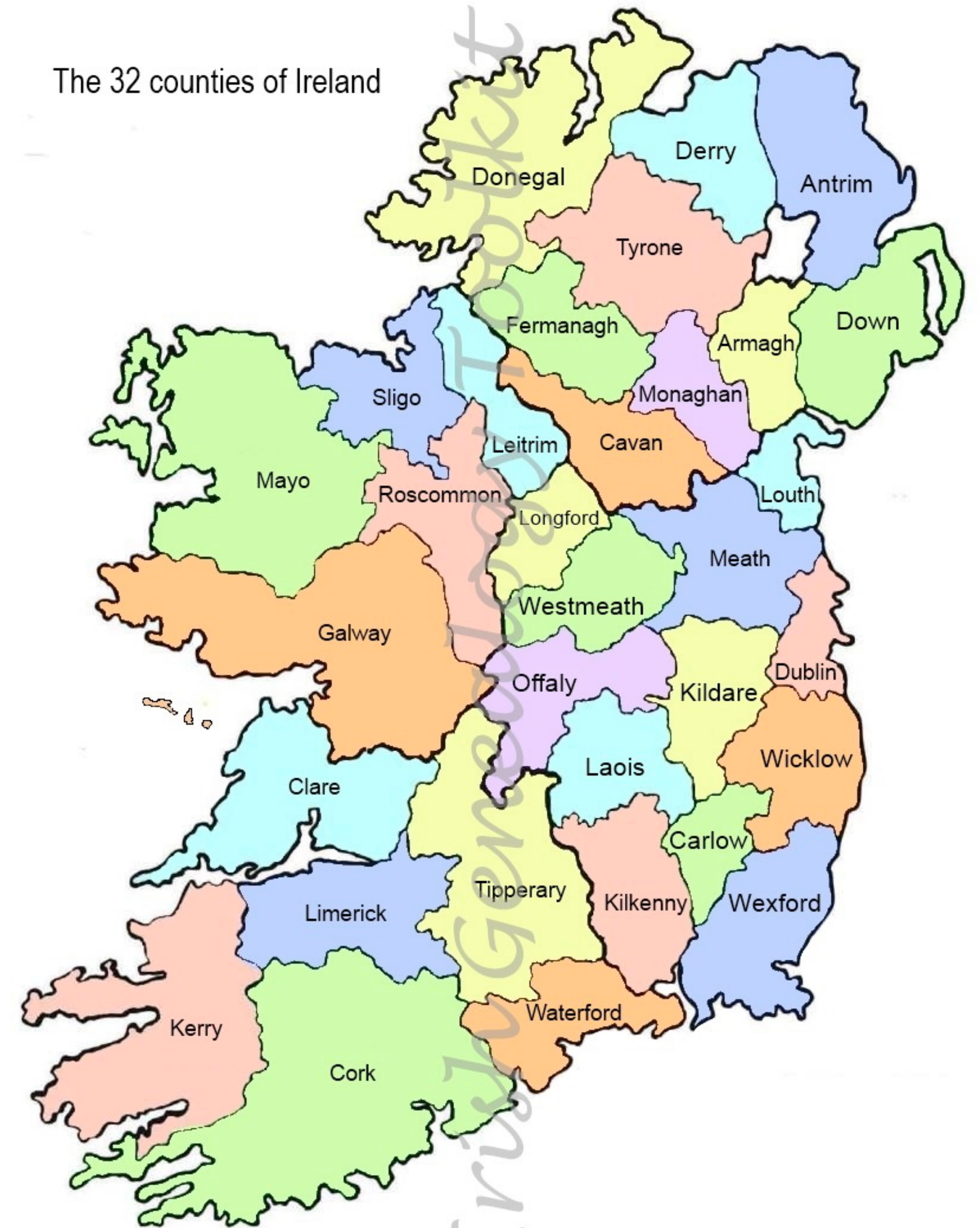
$$\forall i, c \quad y \geq cX_{i,c}$$

$$\forall i \quad \sum_c X_{i,c} = 1$$

Code

type of each variable?

The 32 counties of Ireland



Python Tips

Data storage

- DataFrame
- Array
- List
- Tuple
- Dictionary
- Set
- json
- ...?

Code



Python Visualisation

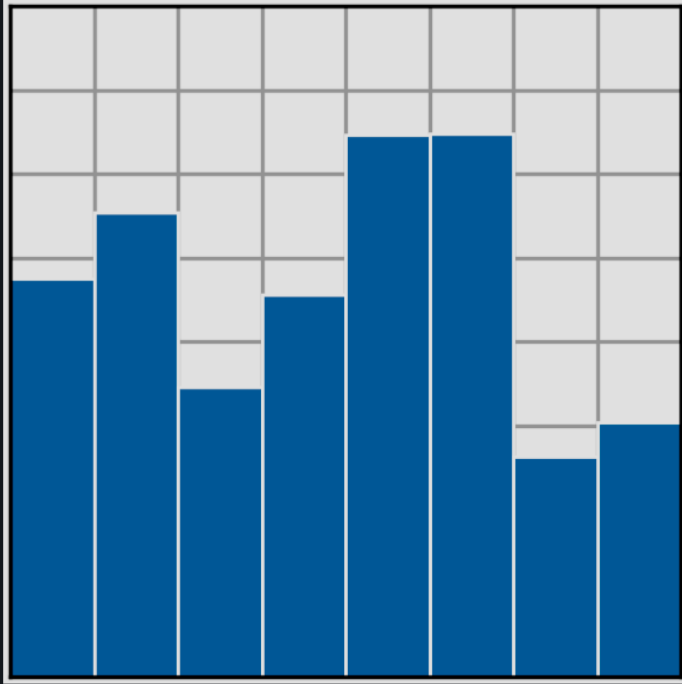
matplotlib.org

☆ | 🌐

matplotlib

Plot types | User guide | Tutorials | Examples | Reference | Contribute | Releases

🔍 | 🌙 | 📄 | 🗨️ | 🐦



bar(x, height) / barh(y, width)

Matplotlib: Visualization with Python

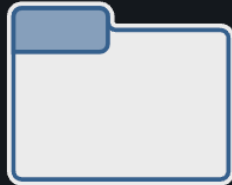
Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create [publication quality plots](#).
- Make [interactive figures](#) that can zoom, pan, update.
- Customize [visual style](#) and [layout](#).
- Export to [many file formats](#).
- Embed in [JupyterLab](#) and [Graphical User Interfaces](#).
- Use a rich array of [third-party packages](#) built on Matplotlib.

Try Matplotlib (on Binder) →



[Getting Started](#)



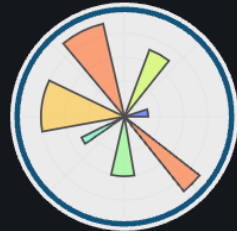
[Examples](#)



[Reference](#)



[Cheat Sheets](#)



[Documentation](#)

News

May 30, 2024

[SOC 2024: Bivariate Colormaps](#) →

Resources

Be sure to check the [Users guide](#) and the [API docs](#). The full text [search](#) is a good way to discover the docs including the many examples.

Python Graphs



Install Tutorial Backends Reference Gallery Developer Releases More ▾

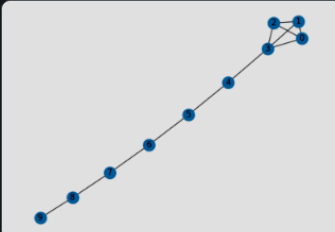
3.4.2 (stable) ▾

[Home](#) > Gallery

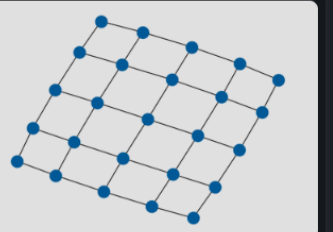
Gallery

General-purpose and introductory examples for NetworkX. The [tutorial](#) introduces conventions and basic graph manipulations.

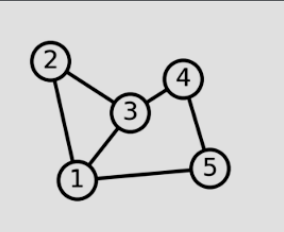
Basic



Properties

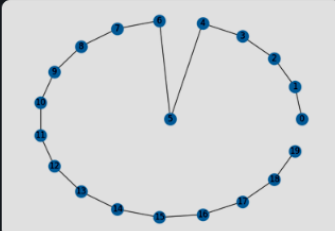


Read and write graphs.

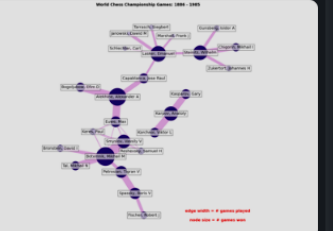


Simple graph

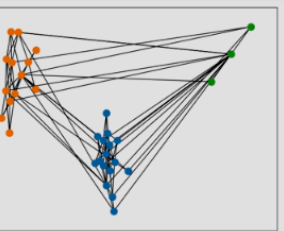
Drawing



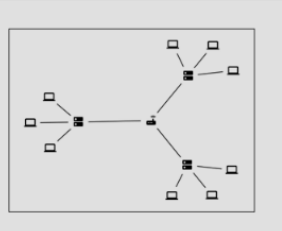
Custom Node Position



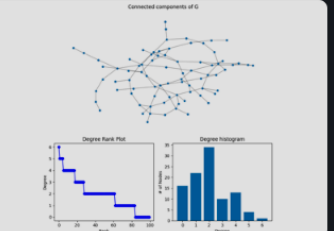
Chess Masters



Cluster Layout



Custom node icons



Degree Analysis

On this page

- Basic
- Drawing
- 3D Drawing
- Graphviz Layout
- Graphviz Drawing
- Graph
- Algorithms
- External libraries
- Geospatial
- Subclass



DataSets + resources

kaggle

+

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Your Work

VIEWED

Traffic_Accidents_EDA...

Key Factors Influencin...

Fina_Machine_project

Regression Models

Loan Approval Classifi...

EDITED

Exercise: Airline Price ...

notebookfe3b77dd46

Search

Datasets

Explore, analyze, and share quality data. [Learn more](#) about data types, creating, and collaborating.

+ New Dataset

Your Work

Search datasets

Filters

All datasets

Computer Science

Education

Classification

Computer Vision

NLP

Data Visualization

Pre-Trained Model

Trending Datasets

See All



Comprehensive Lung Cancer : Dataset

Akshay Choudhary · Updated a day ago

Usability 10.0 · 10 MB

1 File (CSV)

6



Popular Tourist Destinations : and Their Features

Ayush_Panwar · Updated 5 days ago

Usability 8.8 · 17 kB

1 File (CSV)

8



Key Factors Influencing Traffic Accidents

Santiago Torres · Updated 12 days ago

Usability 7.6 · 245 kB

16



Adidas Webstore Shoe Data

Tamás Nédó · Updated 2 hours ago

Usability 10.0 · 2 MB

3 Files (CSV)

5

d-learn.ir/mlopt

39



Read more

<https://towardsdatascience.com/variance-sample-vs-population-3ddbd29e498a>

https://adam-rumpf.github.io/documents/kkt_intro.pdf