Universidad de Guadalajara

Thompson Rivers University Seminar

Introduction to Machine Learning

Dr. Carlos Villaseñor

Content

Day 1 (Dr. Carlos Villaseñor)

- Exploring data with Pandas
- Introduction to machine learning (ML)
- Paradigms of learning
- Prototypical problems

Rest

- Linear Regression
- Regression performance metrics
- Scaling data and Polynomial regression
- Overfit and underfit

Rest

- Nonlinear regression (DT, SVM, KNN, MLP)
- Hyperparameter search

Content

Day 2 (Dr. Carlos Villaseñor)

- Classification problem
- Logistic regression
- Nonlinear classification (DT, SVM, KNN, MLP)

Rest

- Classification metrics
  - Confusion matrix
  - Classification report

Rest

- Non supervised learning
  - Clustering techniques (Kmeans, Spectral clustering, DBSCAN)
  - Silhouette score
  - Dimensionality reduction (PCA, t-SNE)

Content

Day 3 (Dr. Carlos Villaseñor)

- Introduction to neural networks
  - Biological neuron
  - Artificial neuron
  - Perceptron algorithm

Rest

  - Gradient Descent
  - Linear Neurons and Logistic neurons (MSE and BCE)
  - One-layer Network (Softmax and CCE)
  - How to model the last layer of a net

Rest

  - Multi-layer Perceptron
  - Keras/TensorFlow framework

Content

Day 4 (Dr. Javier Gómez)

- Introduction to digital image processing
- Introduction to convolutional neural networks

Day 5 (Dr. José Hernández)

- Practice machine learning with a real problem

Course repository

https://github.com/Dr-Carlos-Villasenor/TRSeminar.git

Guidelines of the course

- The first to days of the course we are going to avoid the mathematical detail of the ML models, instead we are going to focus on the ML practice.
- In the third day, we are going to explore neural networks with some mathematical detail.
- I've included some random facts about Mexico!!.

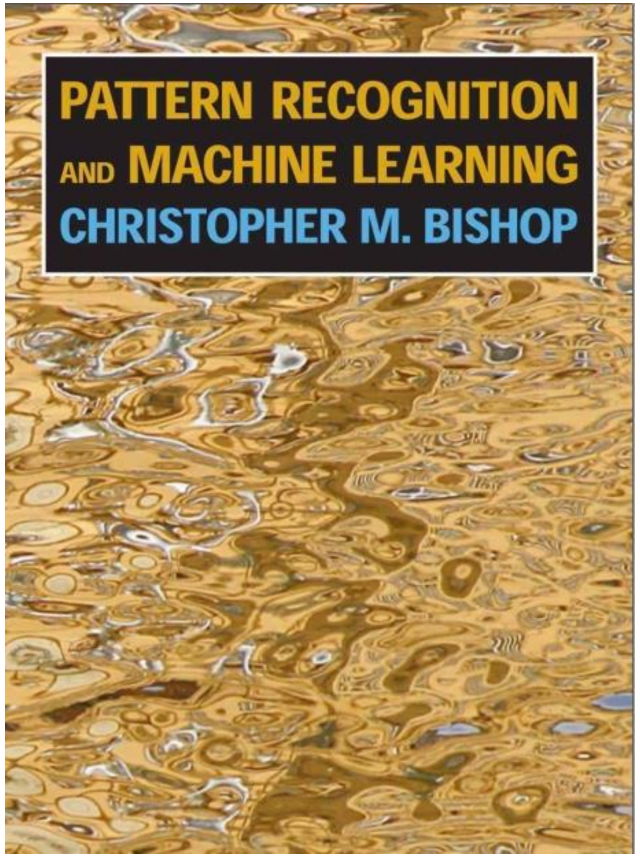
Libraries to learn

https://pandas.pydata.org/

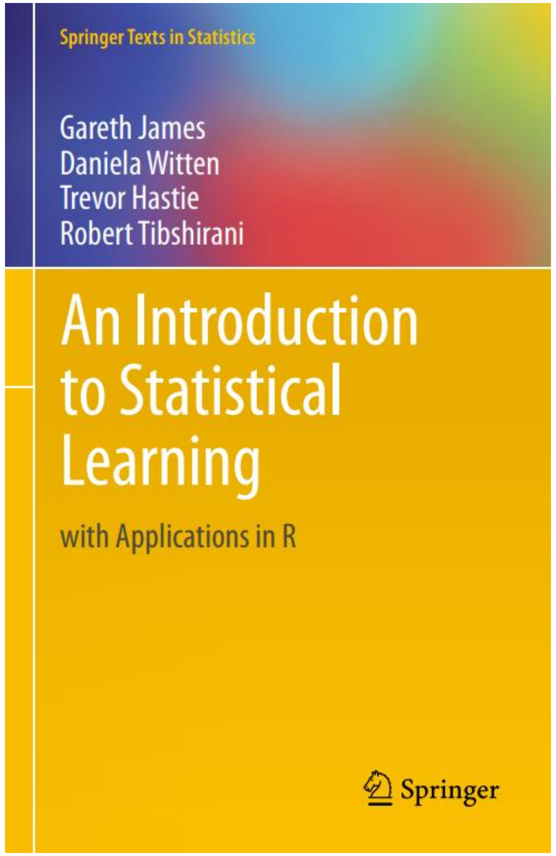
https://scikit-learn.org/

https://www.tensorflow.org/

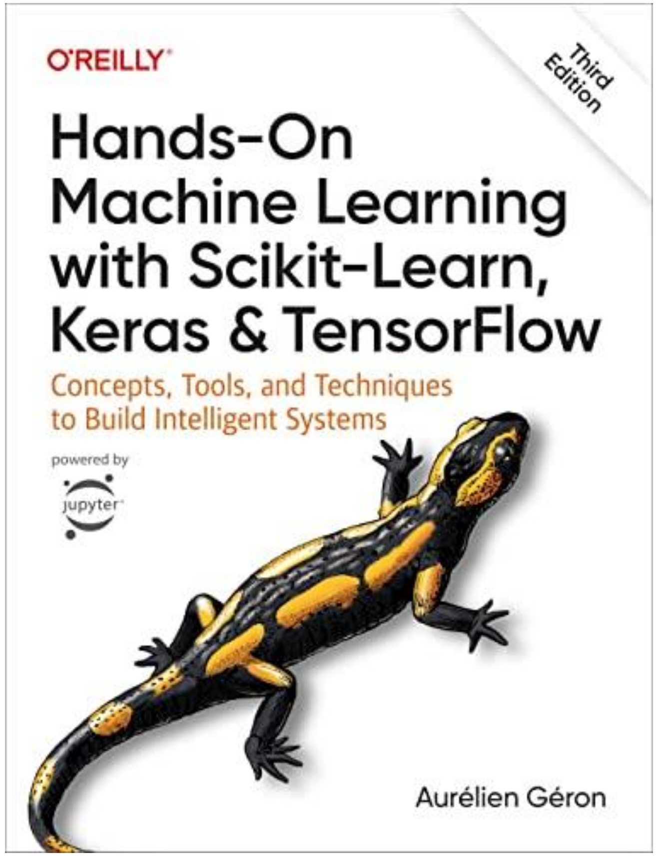
What to read after this lectures?



Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

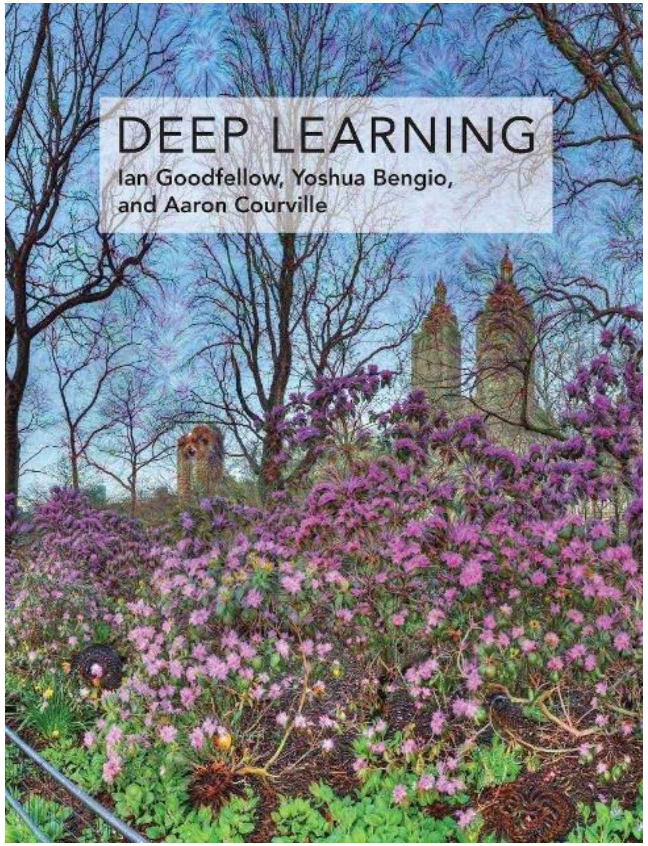


James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

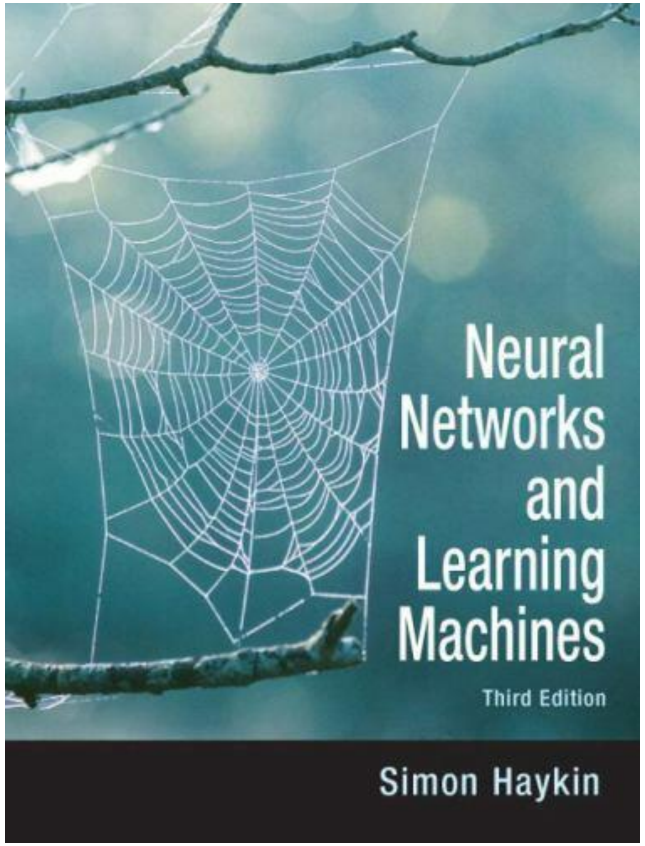


Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc."..

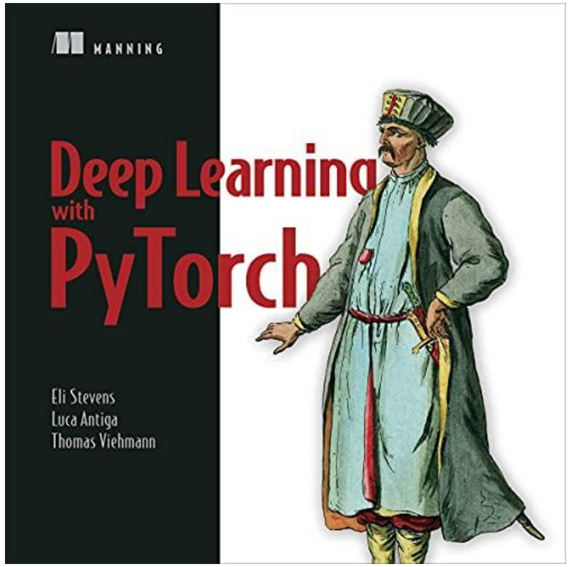
What to read after this lectures?



Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.



Haykin, S. (2009). *Neural networks and learning machines, 3/E*. Pearson Education India.



Stevens, E., Antiga, L., & Viehmann, T. (2020). *Deep learning with PyTorch*. Manning Publications.

Introducción a pandas

Pandas (**Pan**el **da**ta) is a Python library focused on Data manipulation an analysis. Pandas offers a data structure (DataFrame) and Time series.

Let's get hands on! Open TRS01_Pandas.ipynb from the repository.

Jericallas!

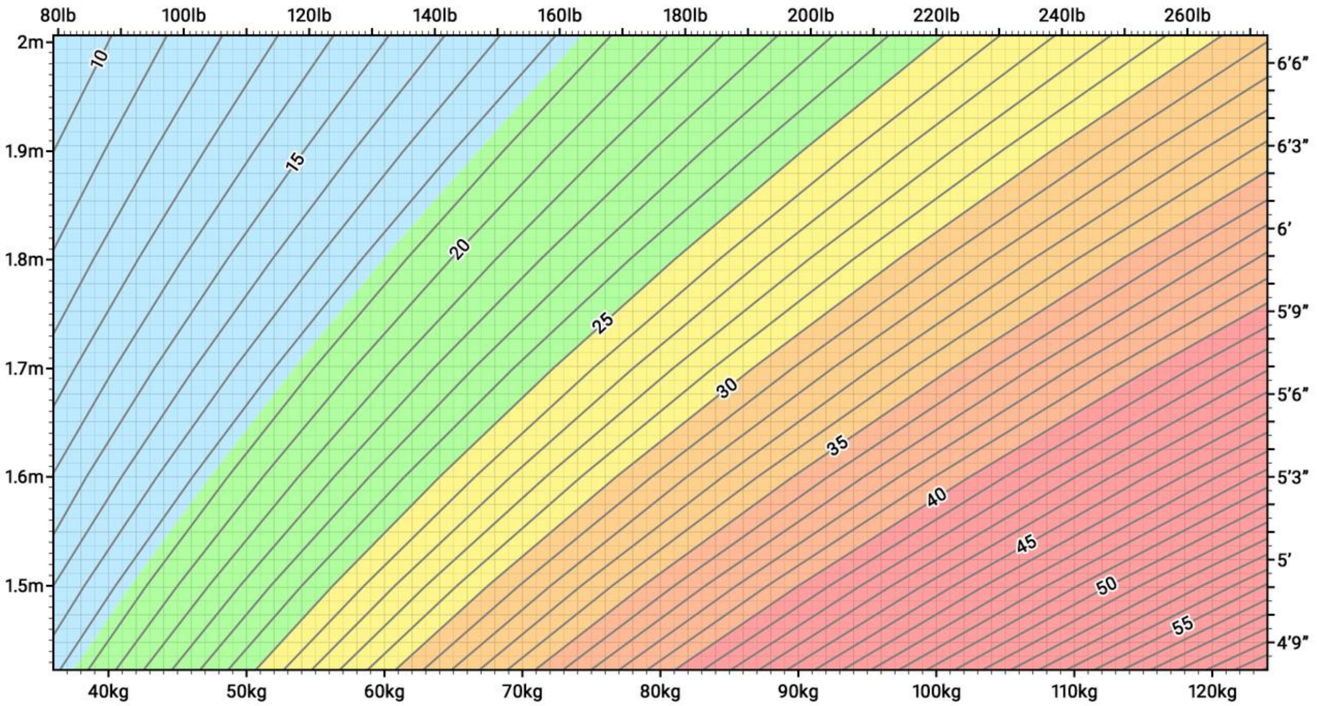
Jericalla is a Mexican dessert from the city of Guadalajara, it is made of egg, milk, vanilla, cinnamon and sugar.

Ask for one: “Quiero una jericalla”

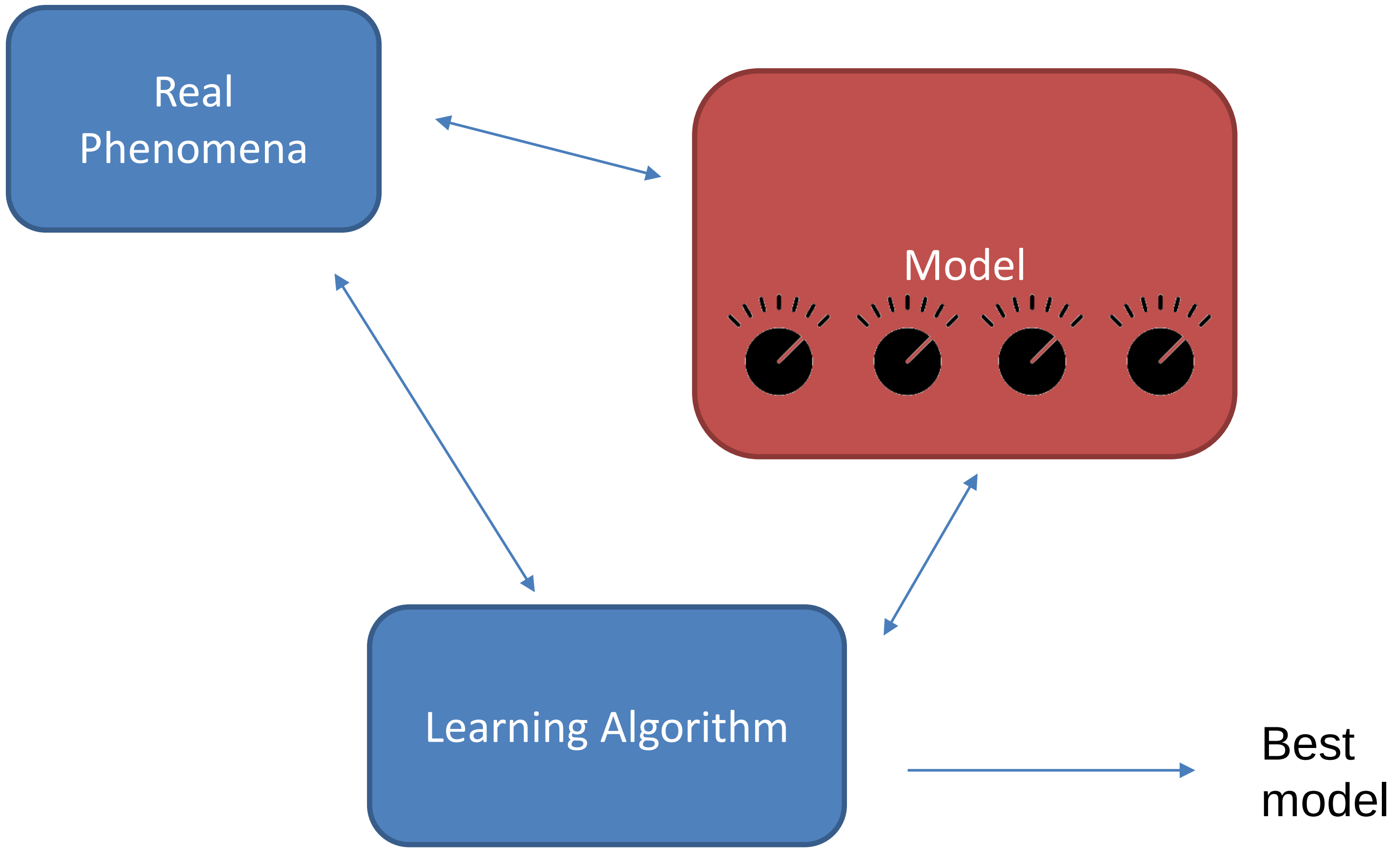
What is learning?

- Develop a program that predicts if a person is overweighted

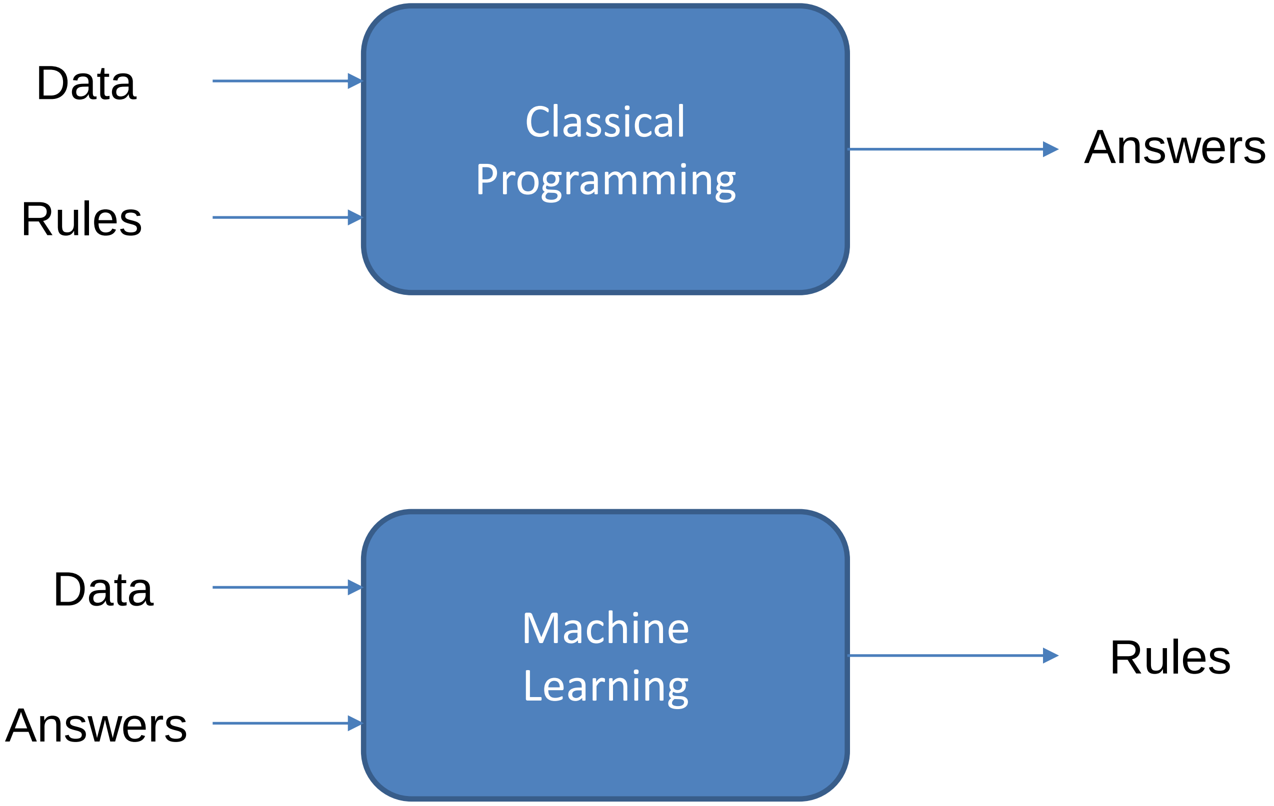
```python
def overweight(mass_kg, height_m):
    bmi = mass_kg / height_m**2
    if bmi < 25:
        return True
    else:
        return False
```



What is learning?



What is learning?



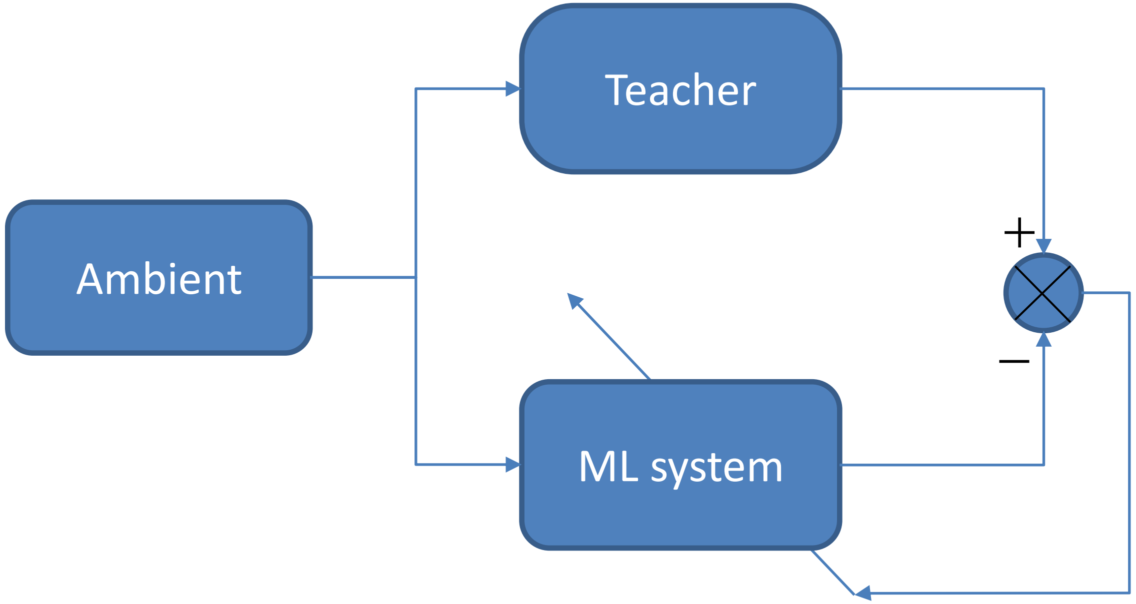
What is learning?

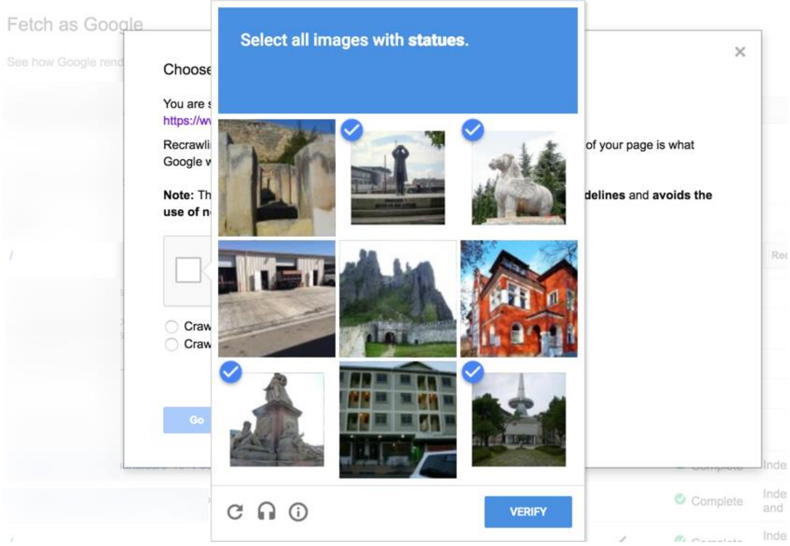
- Tom Mitchell definition:

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Tom M. Mitchell
Professor at the Carnegie Mellon University

Supervised learning

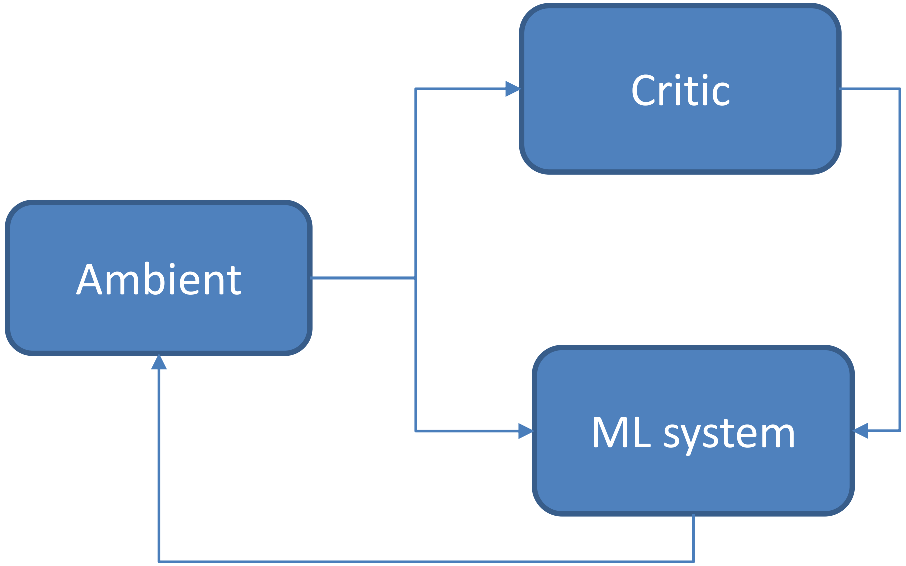




Non-supervised learning

Ambient → ML system →

Reinforcement learning



Prototypical problems

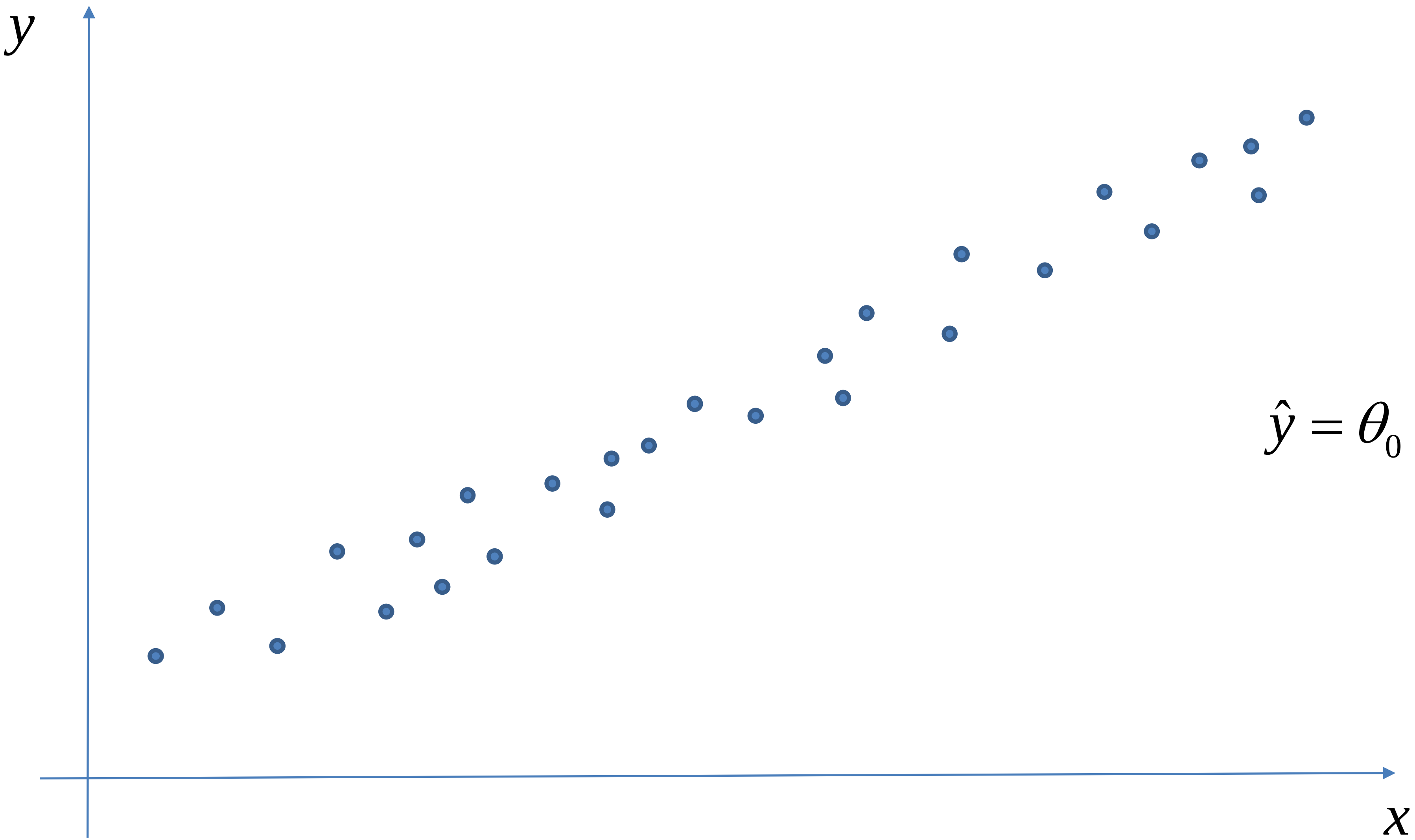
		Wished output	
		Categorical	Continuous
Paradigm	Supervised	Classification	Regression
	Non-supervised	Clustering	Dimensionality reduction

Tejuino

Tejuino is a cold fermented beverage made from corn and popularly consumed in the Mexican states of Jalisco, Colima, Nayarit and Oaxaca. Tejuino is usually made from corn dough, the same kind used for tortillas.

Let's take a rest!

Linear Regression



$$\hat{y} = \theta_0 + \theta_1 x$$

Let's get hands on! Open TRS02_Linear_regression.ipynb

Regression performance metrics

Consider a supervised dataset

$$x^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$$

$$\mathrm{D}=\left\{\left(x^{(1)}, y^{(1)}\right),\left(x^{(2)}, y^{(2)}\right),\cdots,\left(x^{(p)}, y^{(p)}\right)\right\}$$

- Mean Absolute Error (MAE)

$$\mathrm{MAE}\left(y, \hat{y}\right)=\frac{1}{p}\sum_{i=1}^{p}\left|y^{(i)}-\hat{y}^{(i)}\right|$$

Métricas de Desempeño en Regresión

- Median Absolute Error (MedAE)

$$\mathrm{MedAE}\left(y,\hat{y}\right)=\mathrm{median}\left(\left|y^{(1)}-\hat{y}^{(1)}\right|,\cdots,\left|y^{(p)}-\hat{y}^{(p)}\right|\right)$$

- Mean Squared Error (MSE)

$$\mathrm{MAE}\left(y,\hat{y}\right)=\frac{1}{p}\sum_{i=1}^{p}\left(y^{(i)}-\hat{y}^{(i)}\right)^2$$

Métricas de Desempeño en Regresión

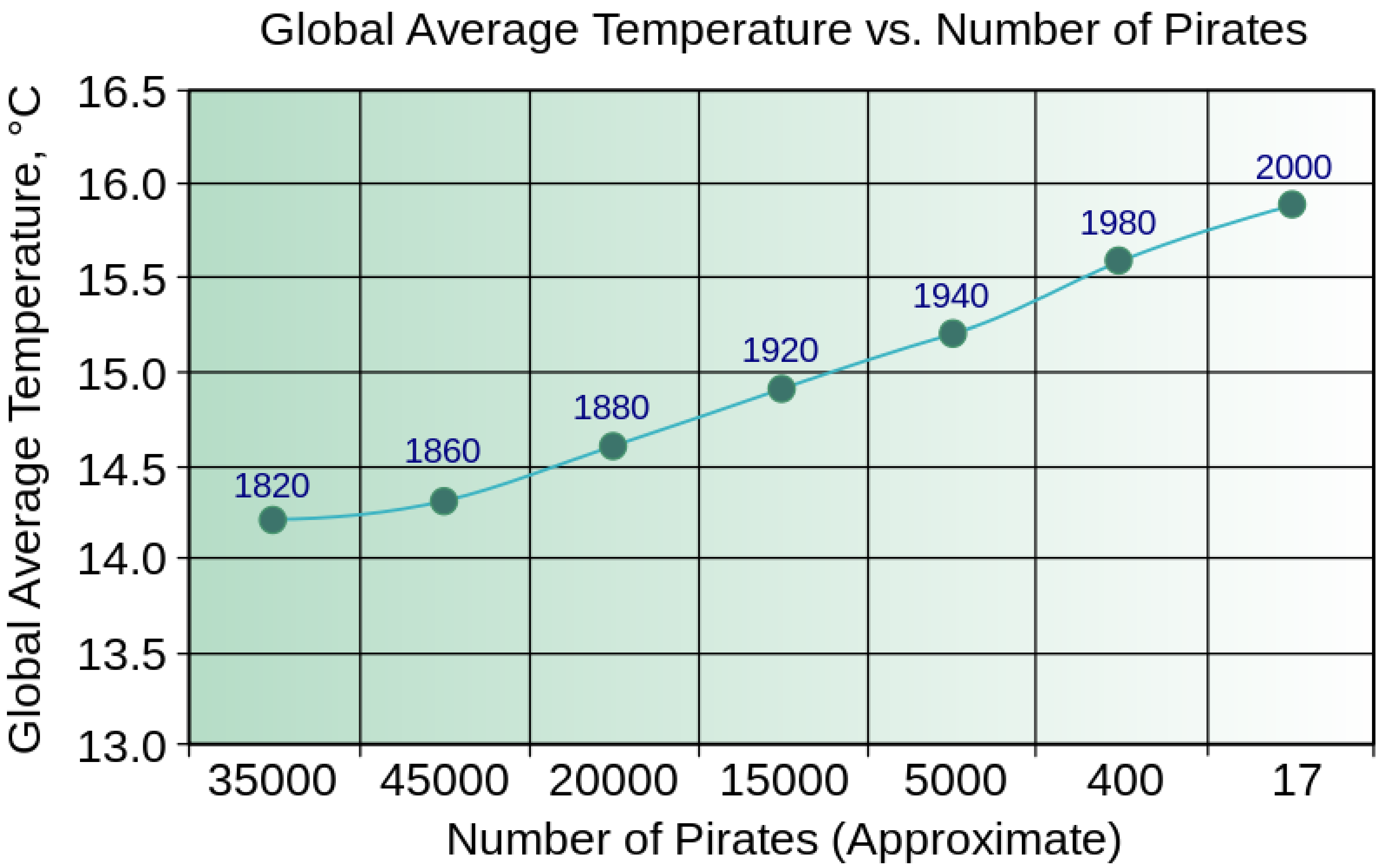
- Explained Variance Score (EVS)

$$\mathrm{EVS}(y,\hat{y}) = 1 - \frac{\mathrm{var}(y-\hat{y})}{\mathrm{var}(y)}$$

- R2-score

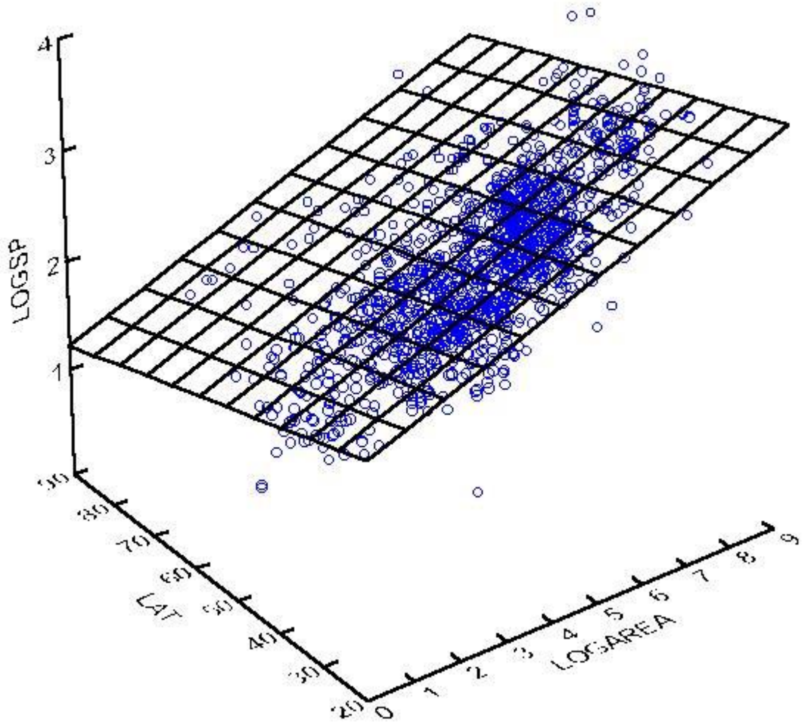
$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{p}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{\sum_{i=1}^{p}\left(y^{(i)} - \overline{y}\right)^2}$$

Correlation vs causality



Linear regression

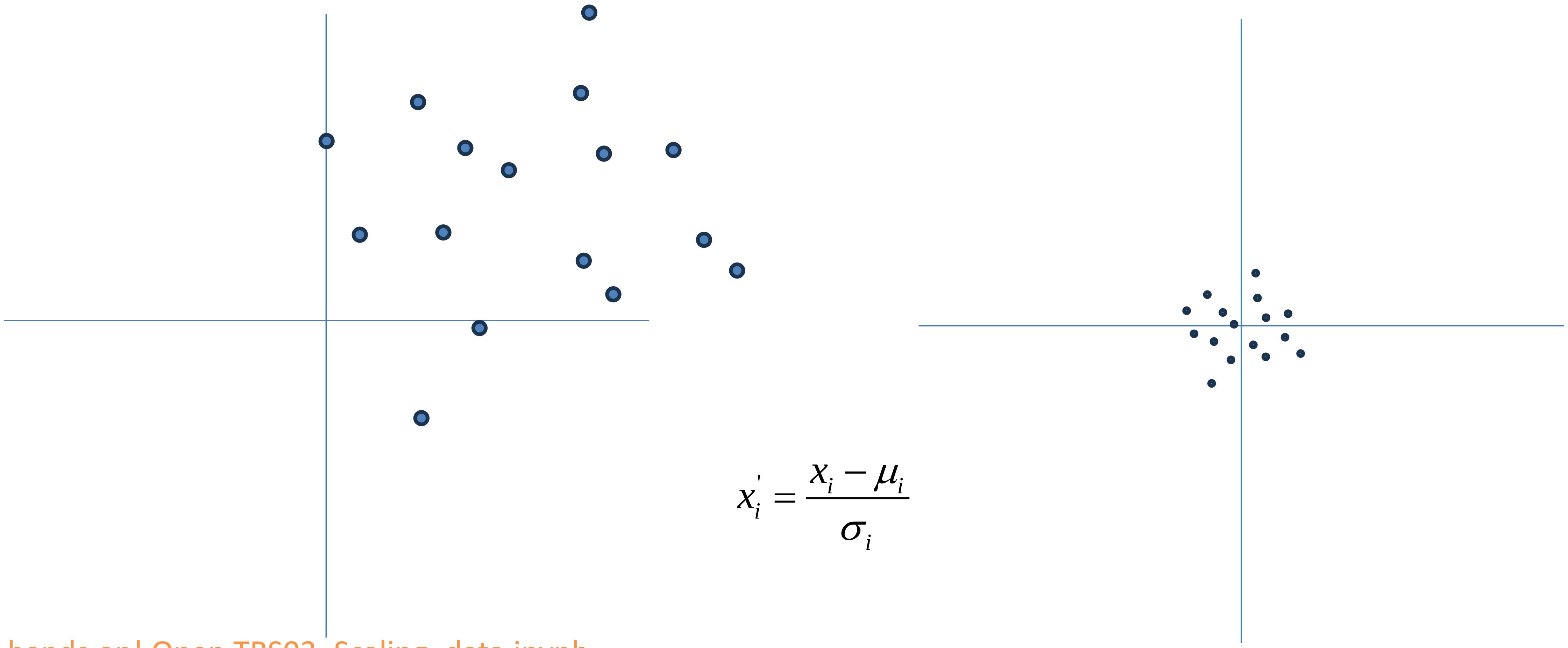
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$



$$\text{Loss}\left(y,\hat{y}\right) = \frac{1}{2p}\sum_{i=1}^{p}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Scaling data

Standard scaler

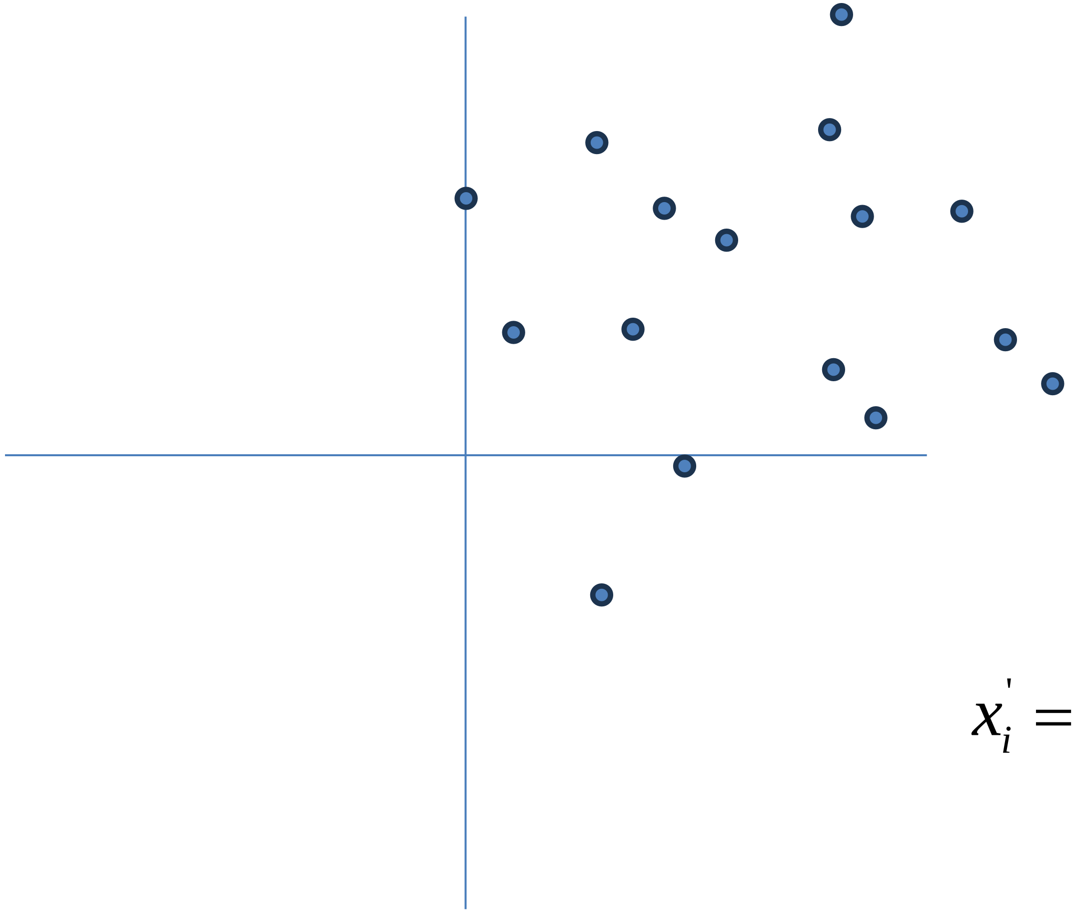


$$x_i' = \frac{x_i - \mu_i}{\sigma_i}$$

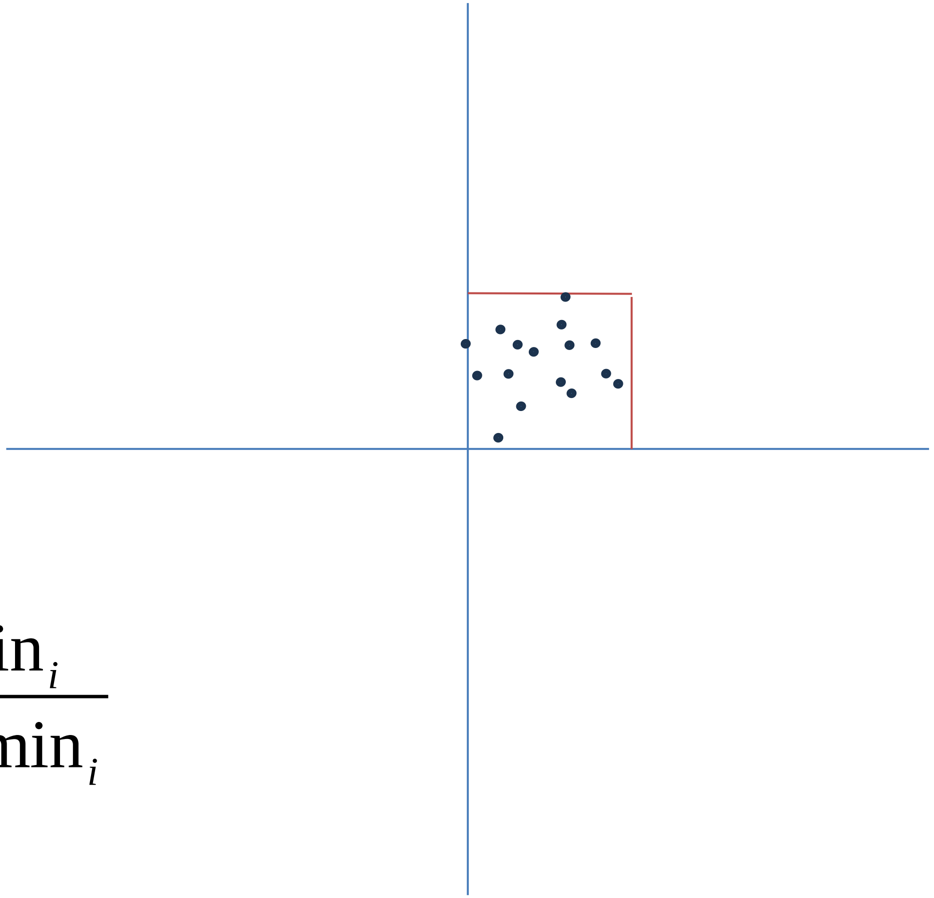
Let's get hands on! Open TRS03_Scaling_data.ipynb

Scaling data

MinMax scaler

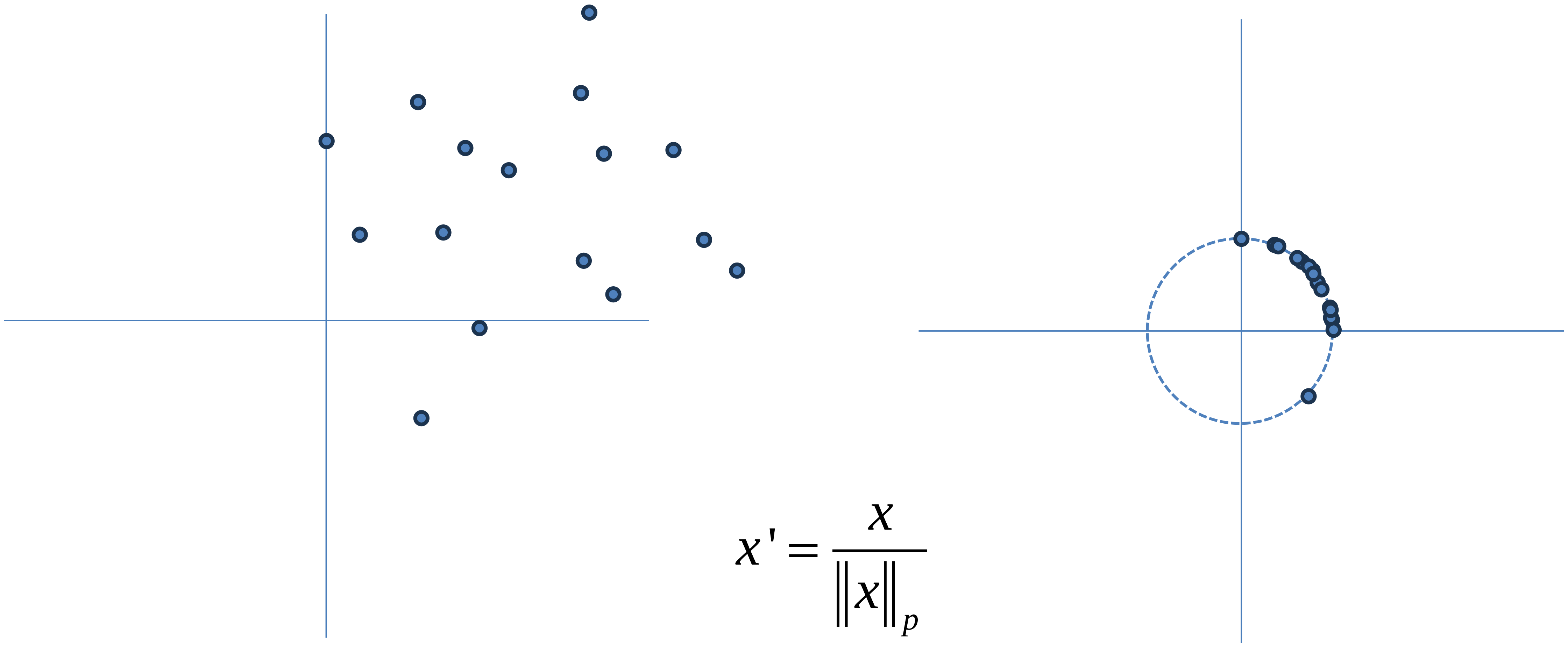


$$x_i' = \frac{x_i - \min_i}{\max_i - \min_i}$$



Scaling data

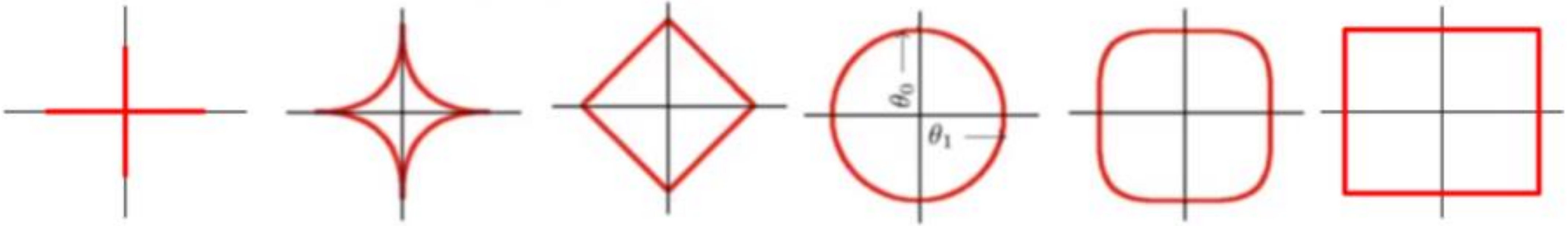
Normalizing scaler



$$x' = \frac{x}{\|x\|_p}$$

Scaling data

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

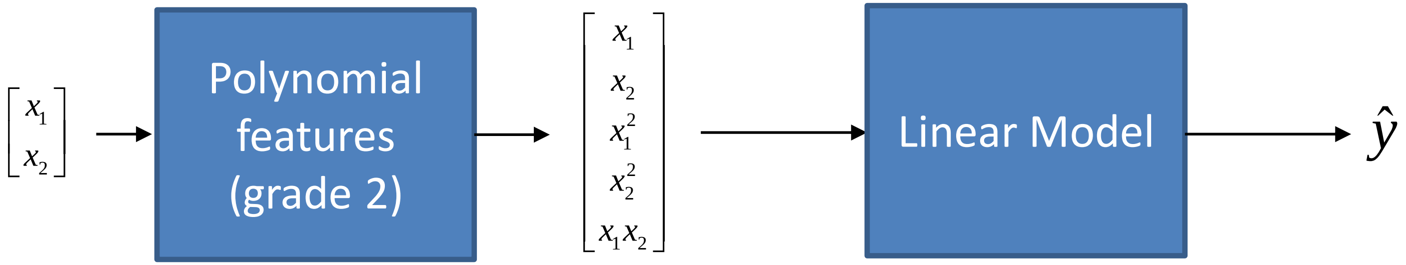


p → 0	p = 0.5	p = 1	p = 2	p = 4	p → ∞
$\|x\|_0 = (\sum_i \|x_i\|^0)^\infty$	$\|x\|_{0.5} = (\sum_i \|x_i\|^{0.5})^2$	$\|x\|_1 = (\sum_i \|x_i\|^1)^1$	$\|x\|_2 = (\sum_i \|x_i\|^2)^{1/2}$	$\|x\|_4 = (\sum_i \|x_i\|^4)^{1/4}$	$\|x\|_\infty = (\sum_i \|x_i\|^\infty)^0$
The number of non-zero parameters					The size of the largest parameter

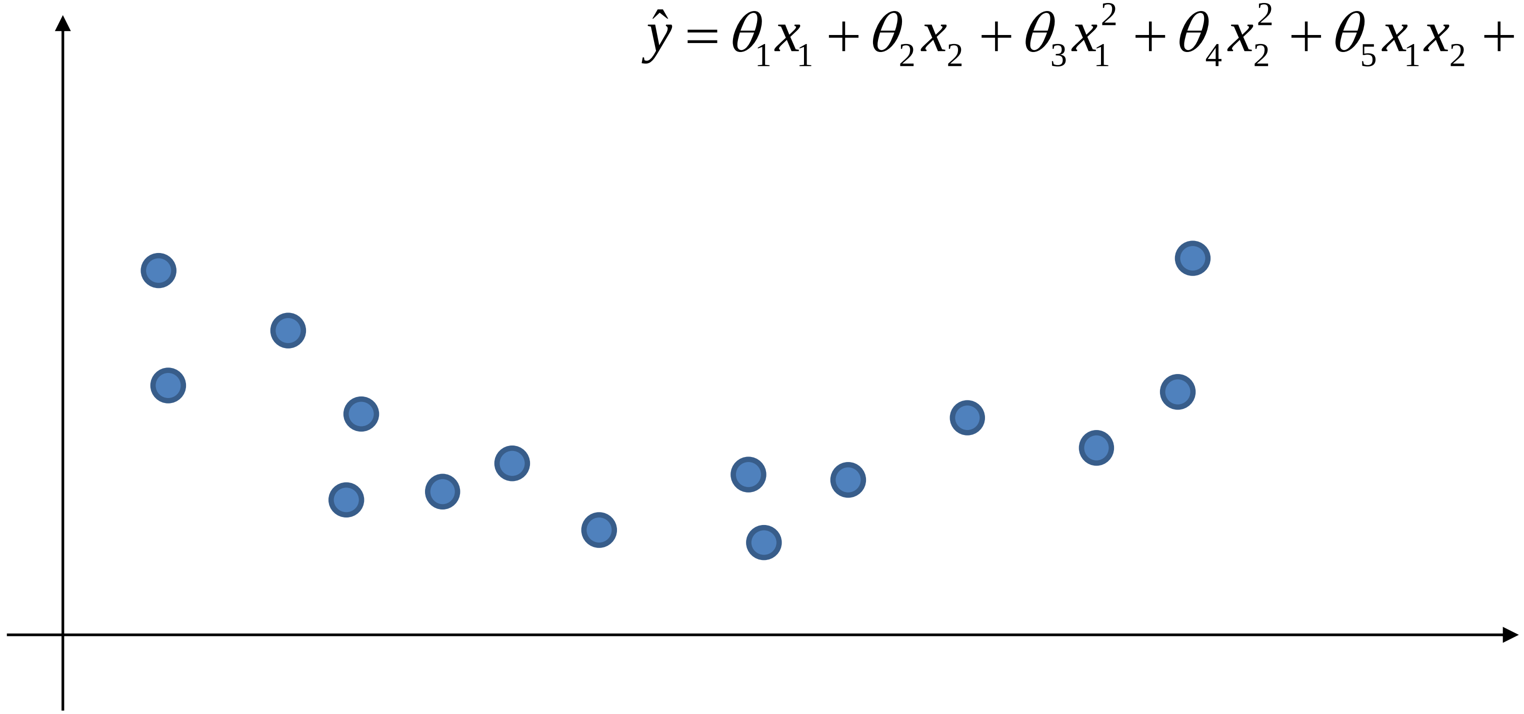
Polynomial Regression

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \xrightarrow{\text{Polynomial features (grade 2)}} \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$$

Polynomial Regression



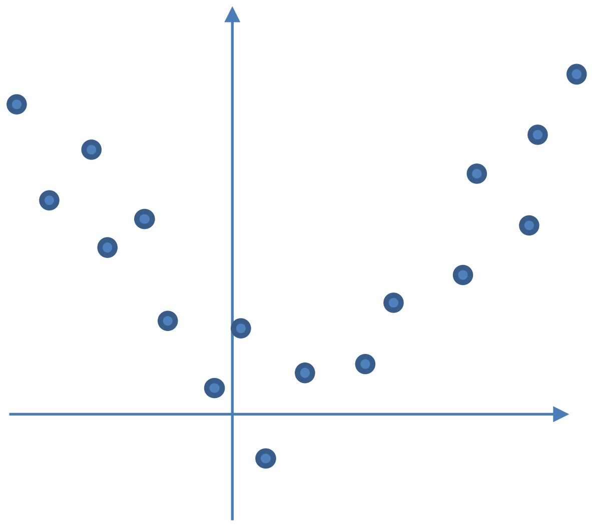
$$\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_0$$



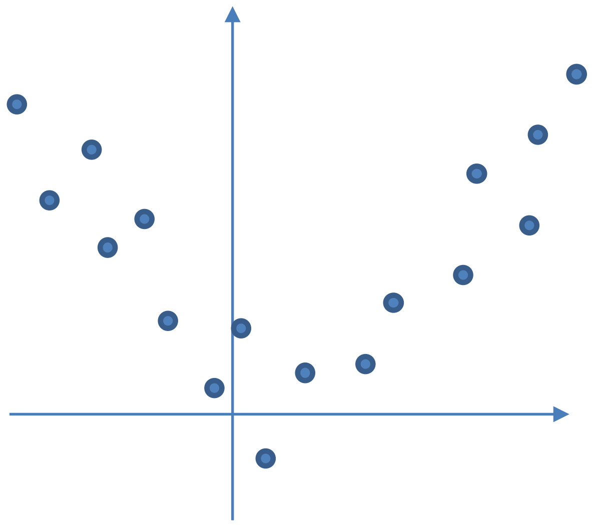
Let's get hands on! Open TRS04_Poly_reg.ipynb

Underfit and overfit

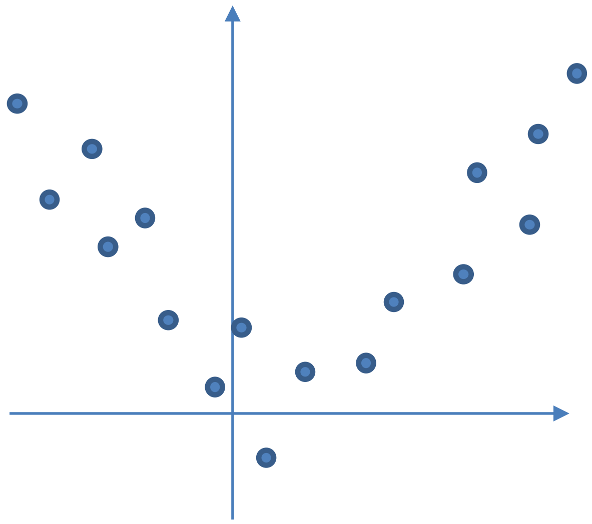
Underfitted



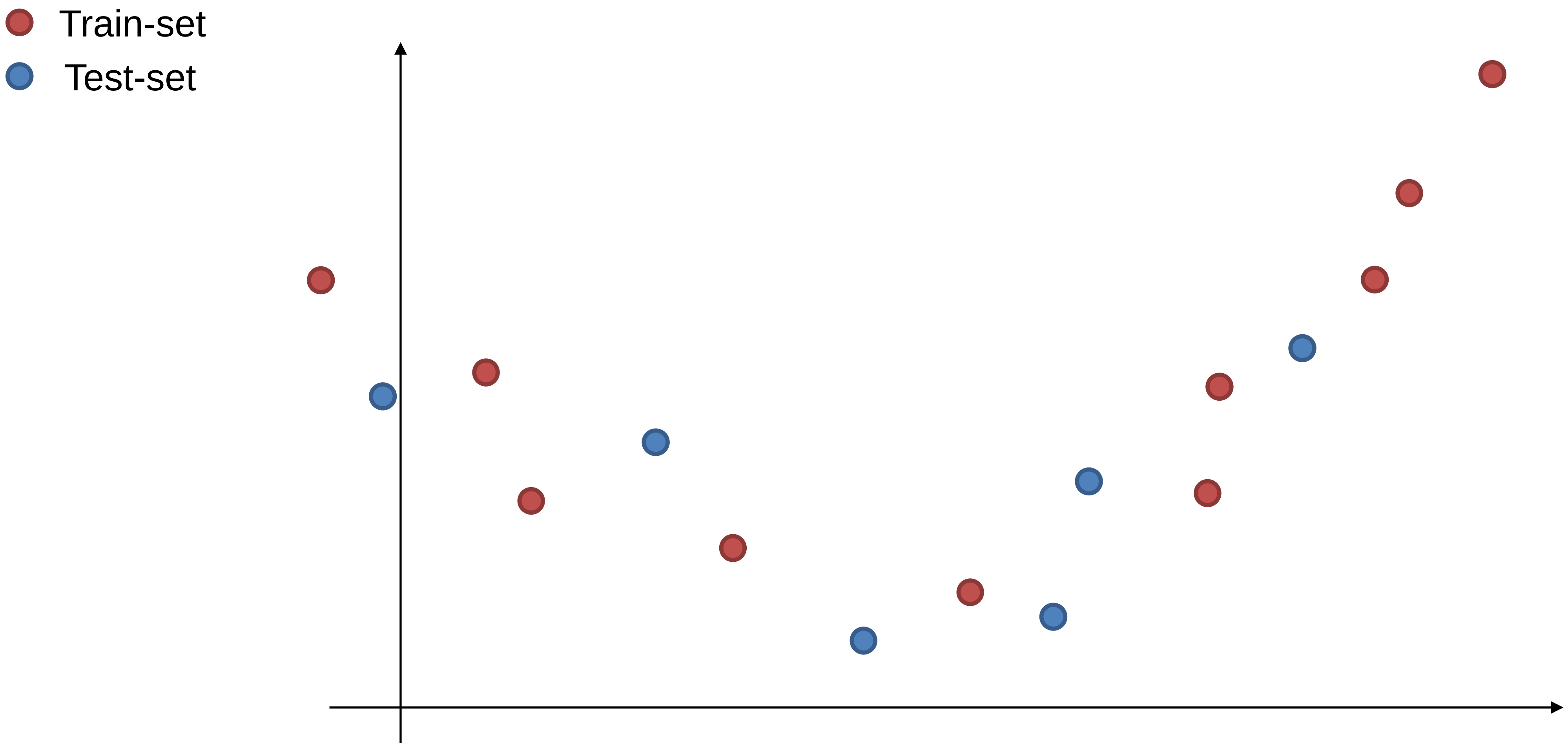
Good generalization



Overfitted



Proving generalization



Train/Test split

Samples	Train-Set	Test-Set
500	75%	25%
2000	80%	20%
10000	90%	10%
100000	98%	2%

Underfit and overfit

[0-1]	Case 1	Case 2	Case 3	Case 4
Train-set	High	Low	High	Low
Test-set	Low	Low	High	High
	Overfit	Underfit	Good Generalization	Miss-match
	Miss-match			

El día de los muertos

Cempasúchil flowers

Let's take a rest!

Non-linear regression

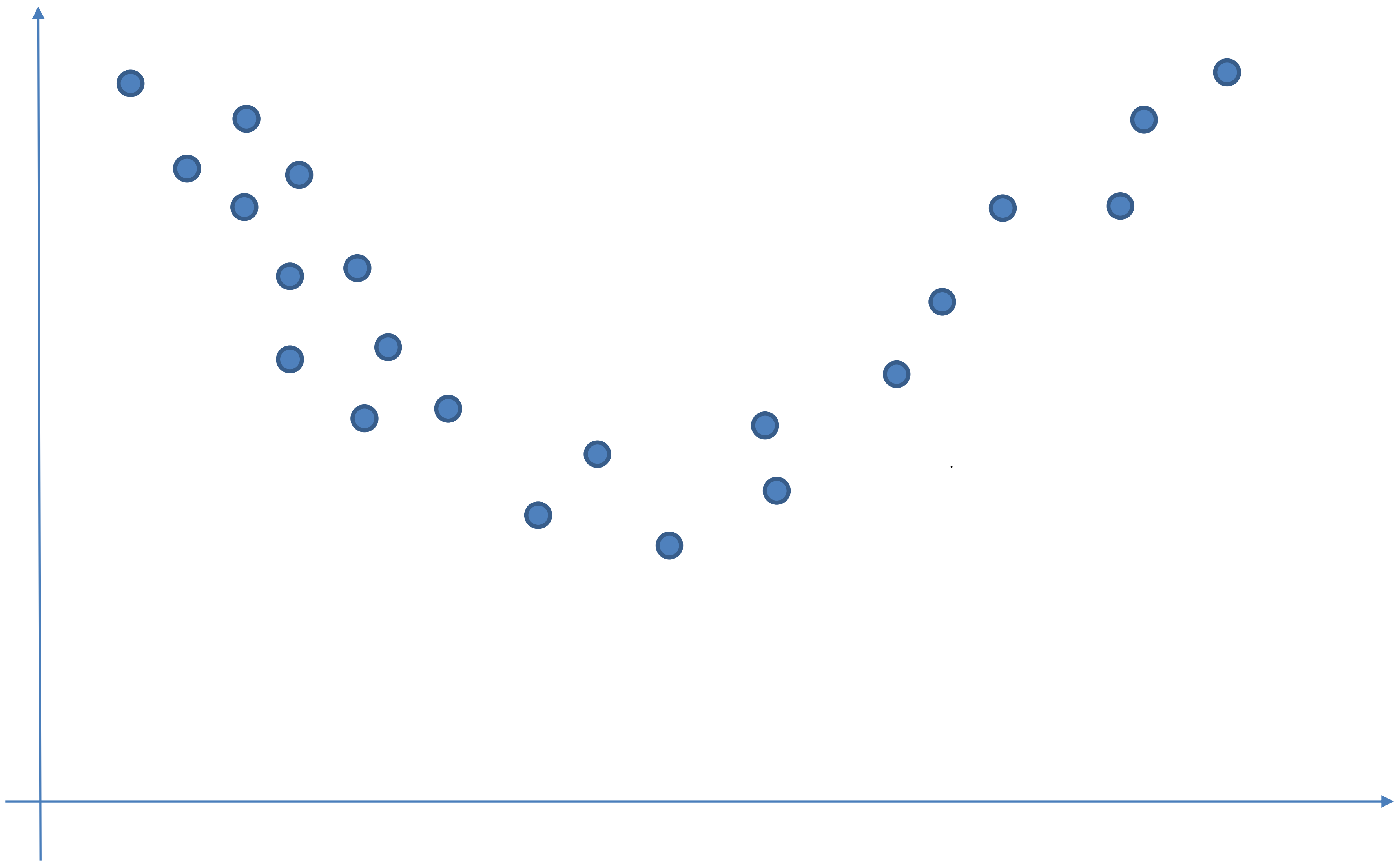
We will review the following ML models

- K Nearest Neighbors (KNN)
- Decision Tree (DT)
- Support Vector Machine (SVM)
- Multi-Layer Perceptron (MLP)

To be a user of these techniques we will experiment with their principal hyper-parameters.

Let's get hands on! Open TRS05_Non_linear_Reg.ipynb

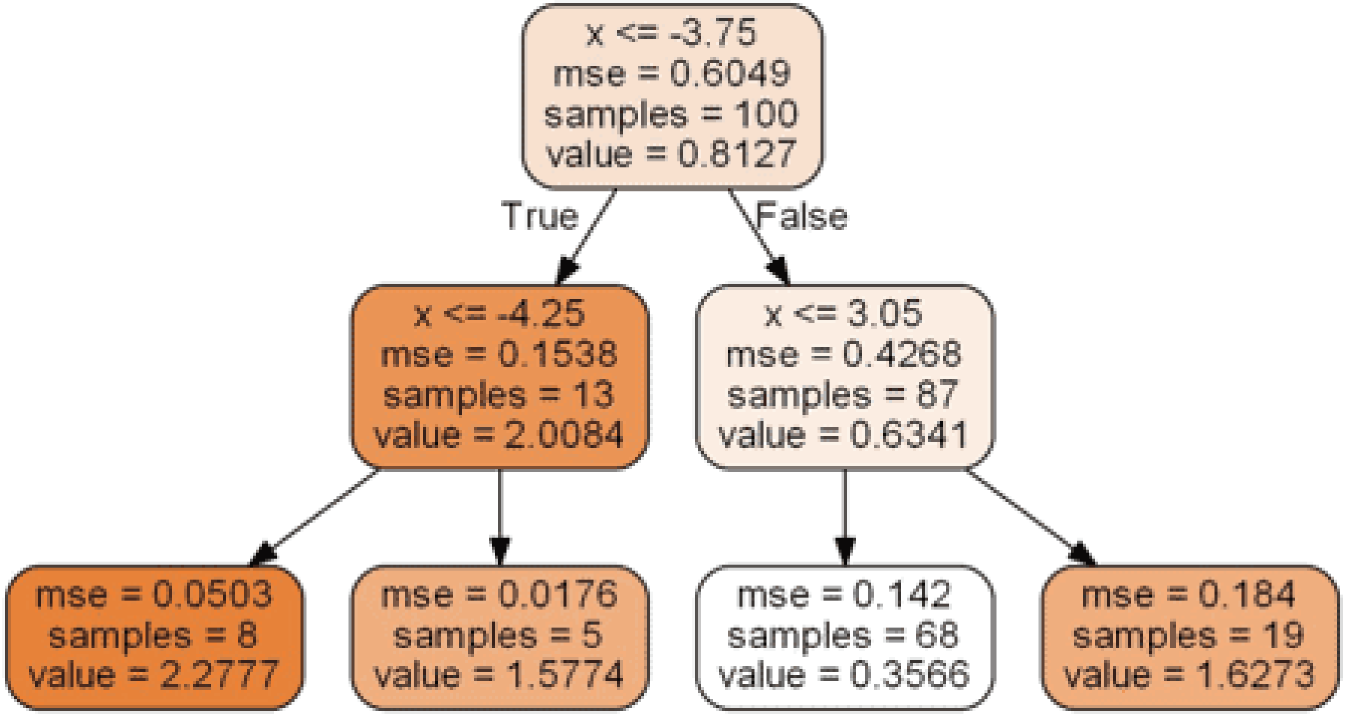
K Nearest Neighbors (KNN)



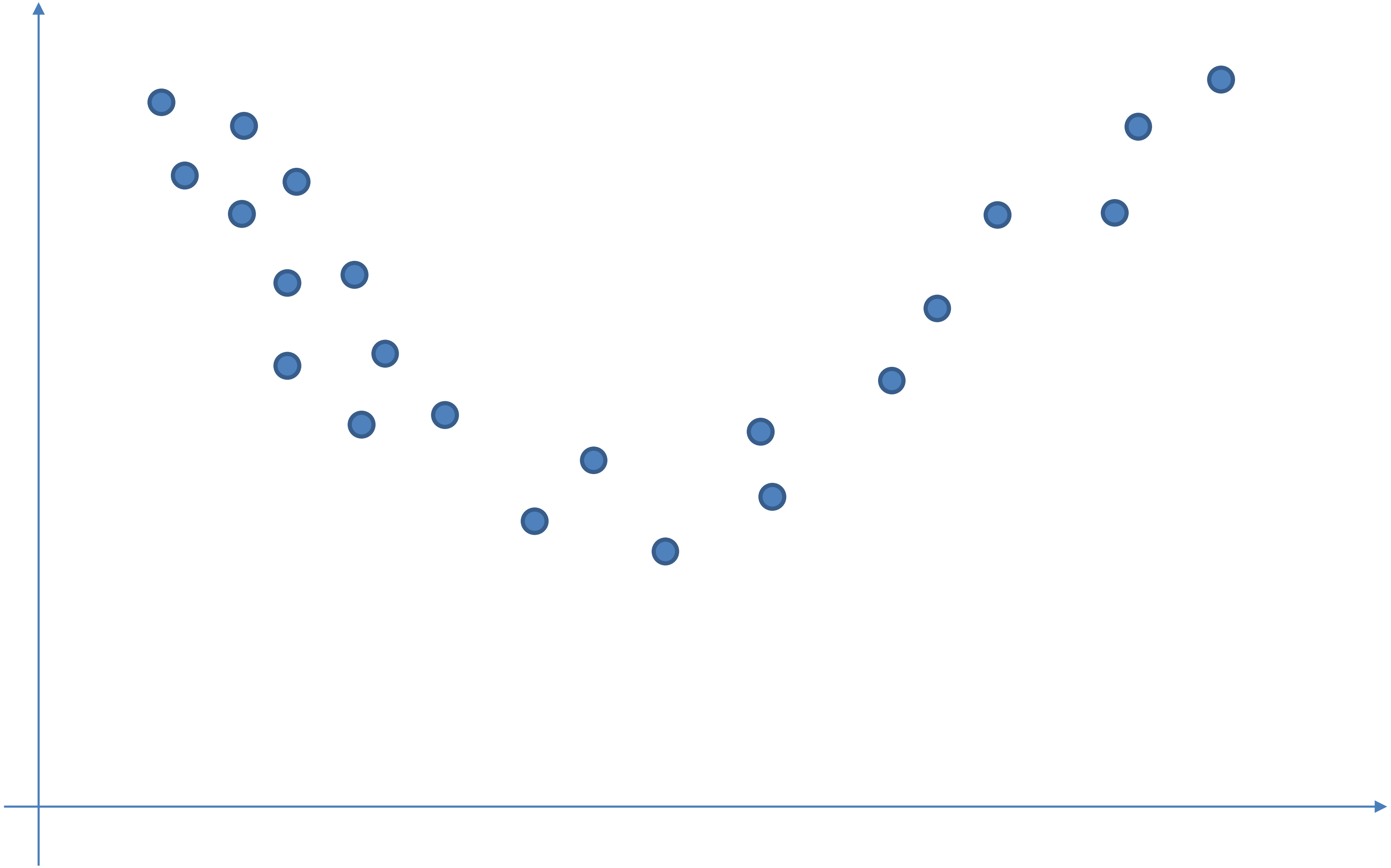
K Nearest Neighbors (KNN)

- Hyper-parameters:
  - **n_neighbors** = número de vecinos,
  - **weights**: {‘uniform’, ‘distance’},
  - **algorithm**:{‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’},
  - **metric**: {‘euclidean’, ‘minkowski’}

Arboles de decisión



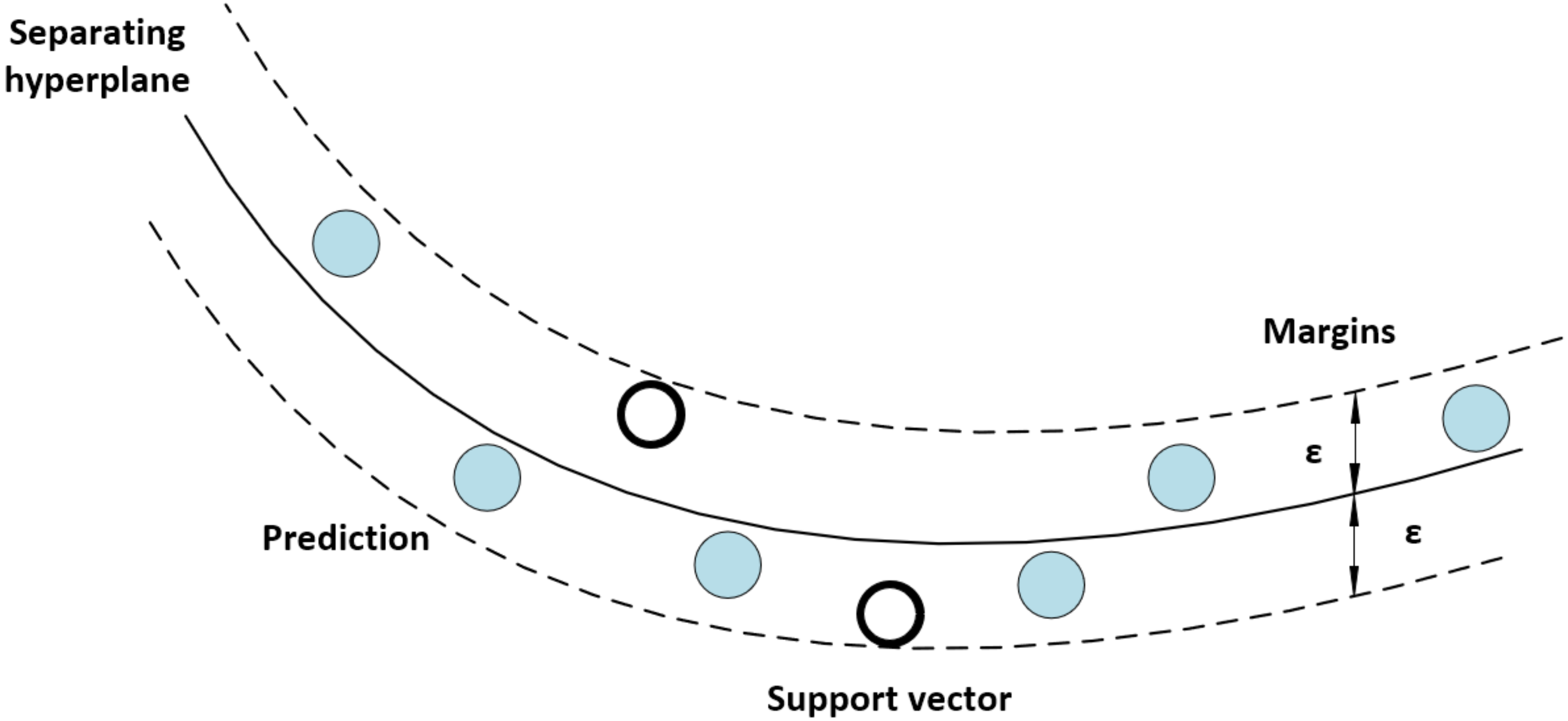
Decision Tree



Decision Tree

- Hyper-parameters:
  - **max_depth**
  - **min_samples_split**
  - **min_samples_leaf**

Support Vector Regressor (SVR)



Support Vector Regressor (SVR)

- Hyper-parameters:
  - **kernel**: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}
  - **gamma:** {‘scale’, ‘auto’}
  - **C** : Regularization

Multi-Layer Perceptron(MLP)

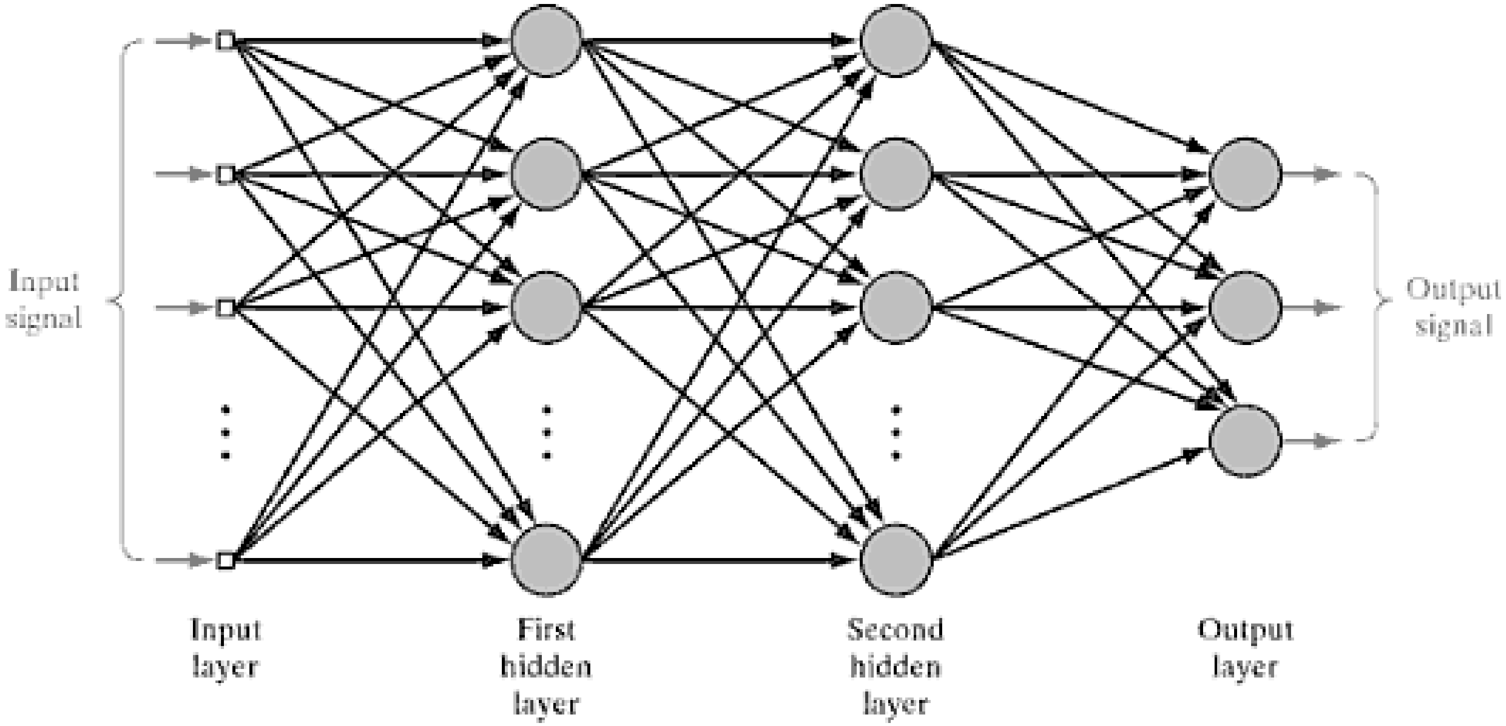


FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

Multi-Layer Perceptron (MLP)

- Hyper-parameters:
  - **hidden_layer_sizes** = (100,)
  - **activation:** {‘identity’, ‘logistic’, ‘tanh’, ‘relu’}
  - **solver**: {‘lbfgs’, ‘sgd’, ‘adam’}
  - **learning_rate:** {‘constant’, ‘invscaling’, ‘adaptive’}
  - **max_iter:** None

Real example of a regressor

Let's get hands on! Open TRS06_Reg_real_example.ipynb