Introduction to Data Science

with Python

Alexis Bogroff

May 25, 2022

Presenter

Alexis Bogroff
Lecturer and Mentor in Data Science
at Paris 1 Panthéon-Sorbonne, ESILV,
Openclassrooms, EM-Lyon

Presenter

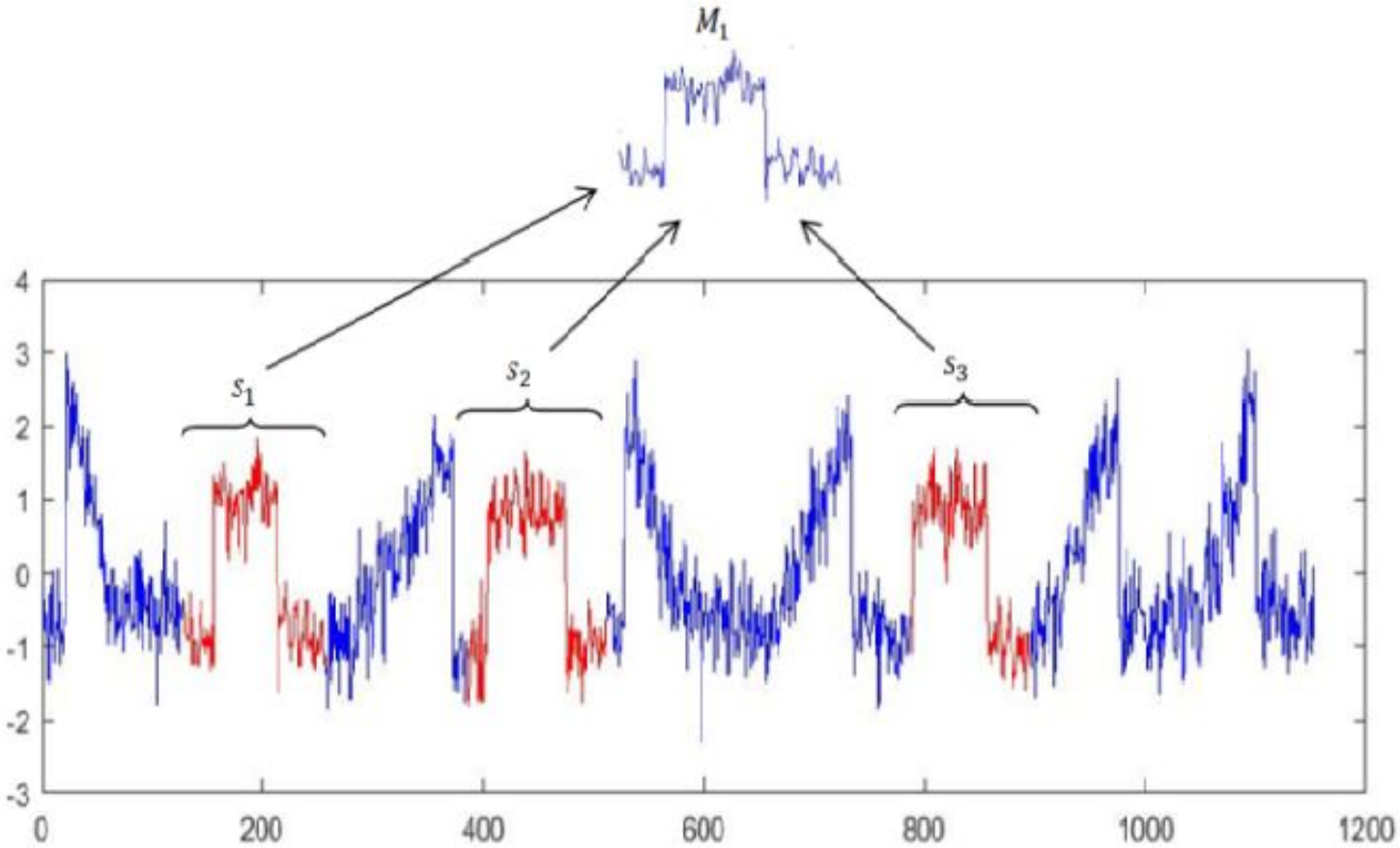
Alexis Bogroff
Lecturer and Mentor in Data Science
at Paris 1 Panthéon-Sorbonne, ESILV,
Openclassrooms, EM-Lyon

- 4 years Teaching Assistant and lecturer in VBA, Python for finance, SQL, Data Analysis and Data Science
- 9 months Researcher Assistant at Paris 1 Panthéon-Sorbonne within H2020 European Project
- 1 year Data Scientist at Pléiade Asset Management

Predictions: What does that mean?

What is modeled?

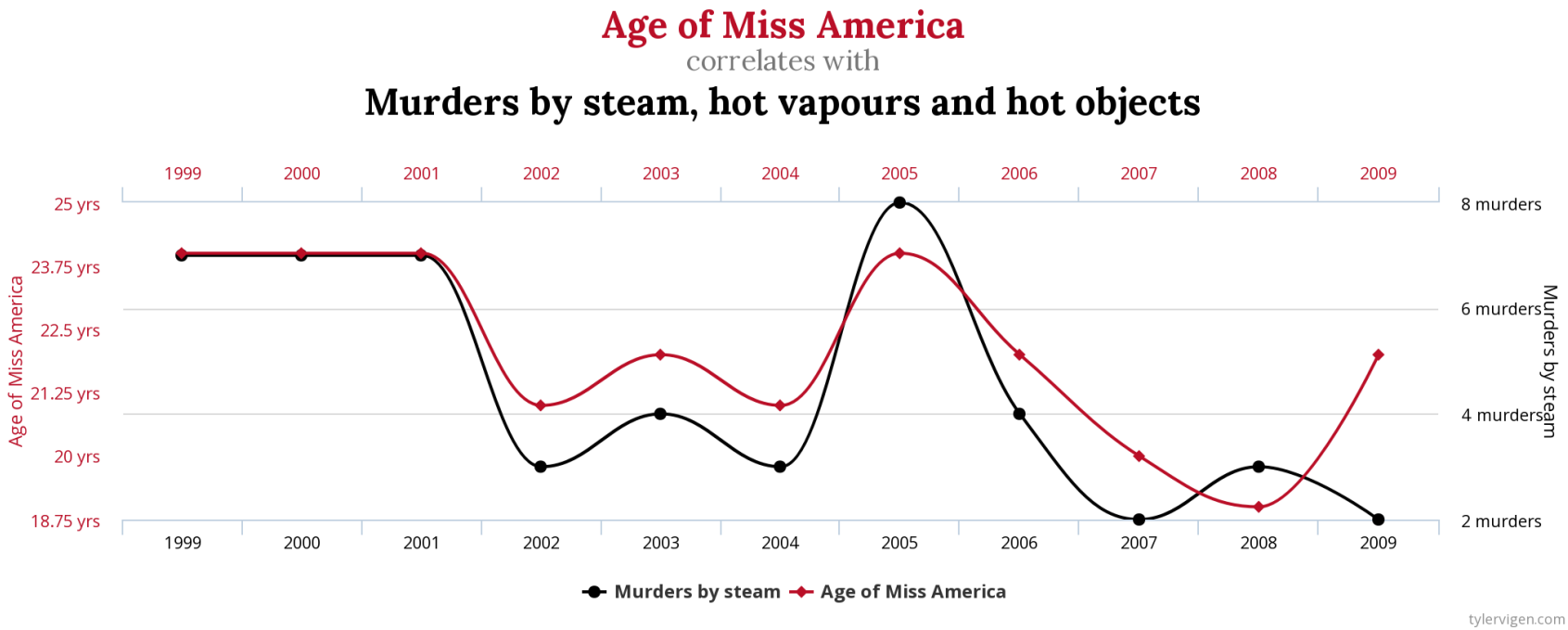
- Continuity (stationarity)
- Correlation (pattern)



Predictions: What does that mean?

What is modeled?

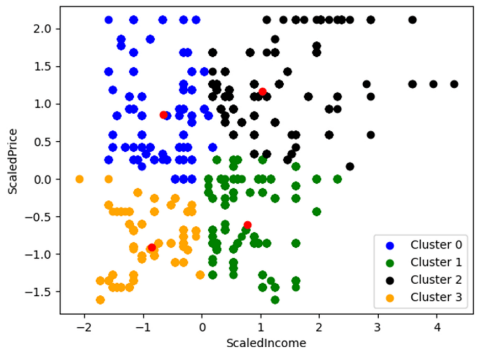
- Correlation vs Causality[1]



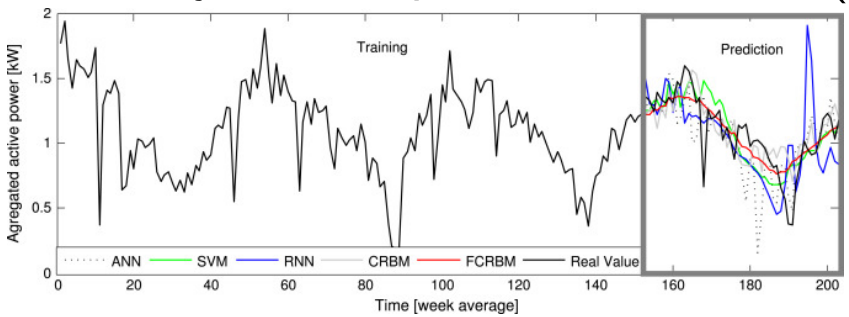
[1] https://www.tylervigen.com/spurious-correlations

Predictions: Examples

- Present:
  - Electricity consumption based on other cities (e.g. Seattle)
  - Missing values (interpolation, extrapolation)
  - Client category



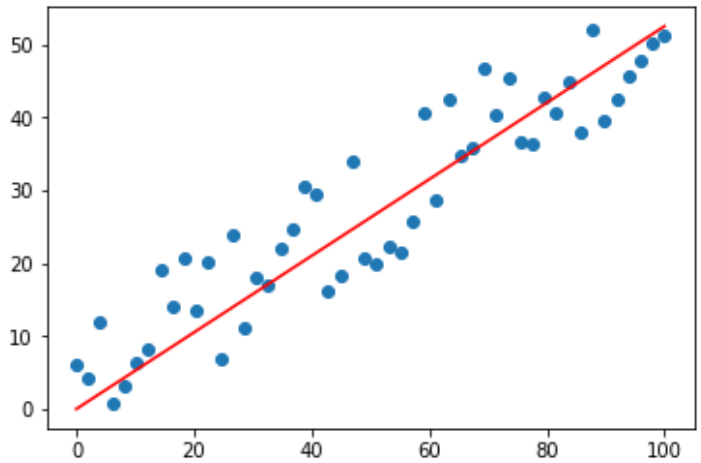
- Future:
  - Electricity consumption next month (time series)



  - Client clicking add (recommander sys.)
  - Pedestrian and cars trajectories (RL)

Regression

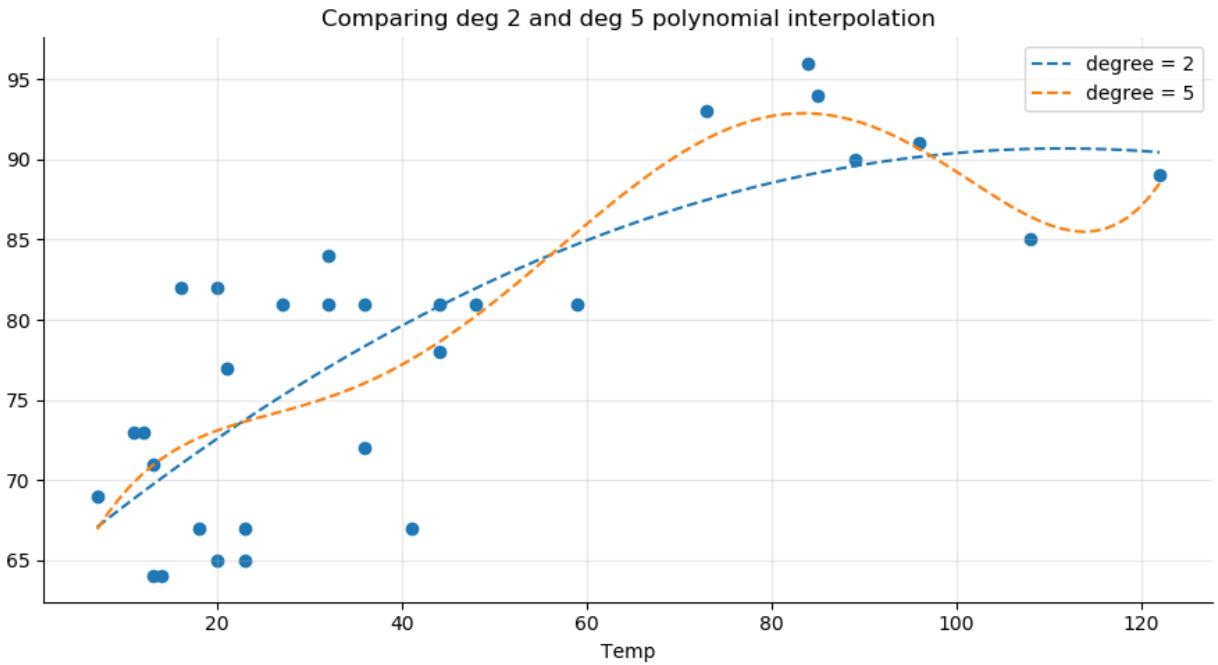
- Linear regression
- Simple: $Y = aX + b$
- Multiple: $Y = a_1X_1 + a_2X_2 + \cdots + a_nX_n + b$



Regression

- Polynomial regression
- Simple: $Y = a_1X + a_2X^2 + \cdots + a_nX^n + b$
- Multiple:

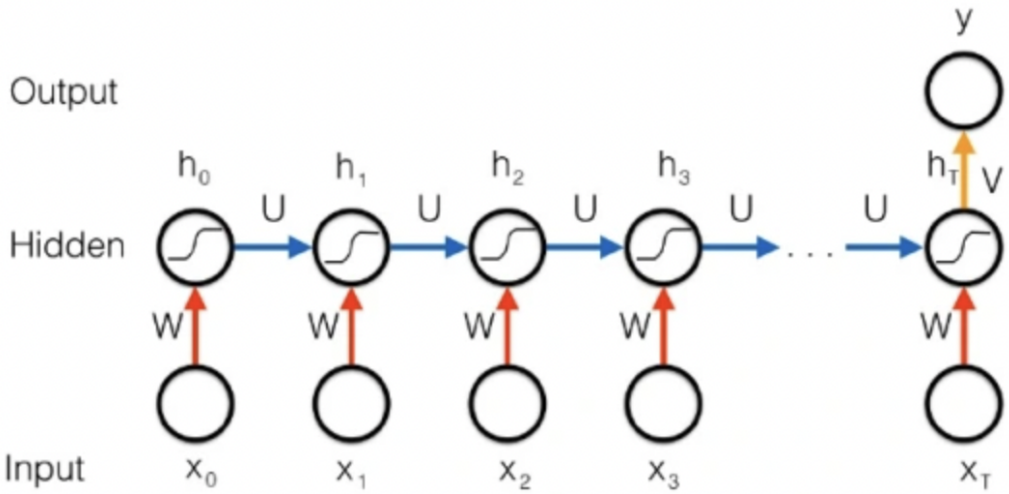
$$Y = a_{11}X_1 + a_{21}X_2 + a_{n1}X_n + a_{12}X_1^2 + a_{22}X_2^2 + a_{n2}X_n^2 +$$
$$\cdots + a_{1m}X_1^m + a_{2m}X_2^m + a_{nm}X_n^m + b$$

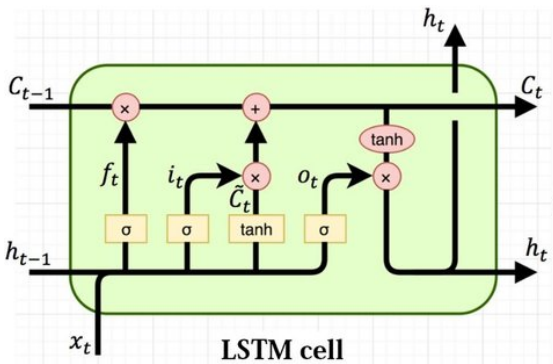


Regression

- SARIMA: seasonal, auto-regressive, integrated, moving average
- Parametric model that capture patterns like:
  - Trend
  - Cycle
  - Season

- Deep Learning (RNN, LSTM)

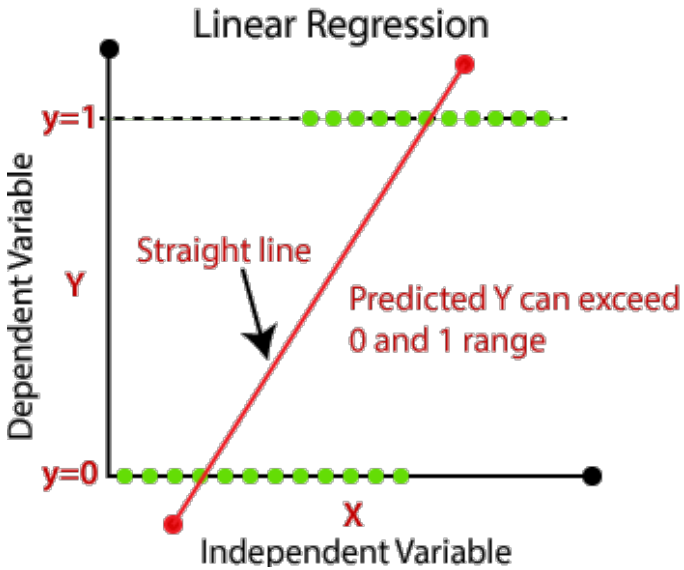


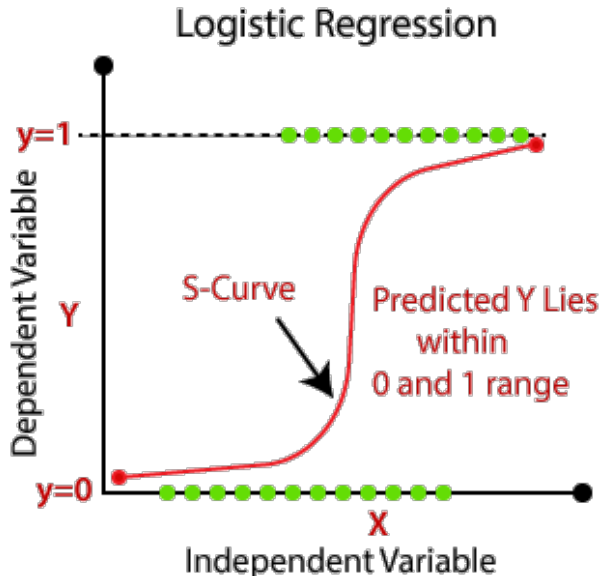


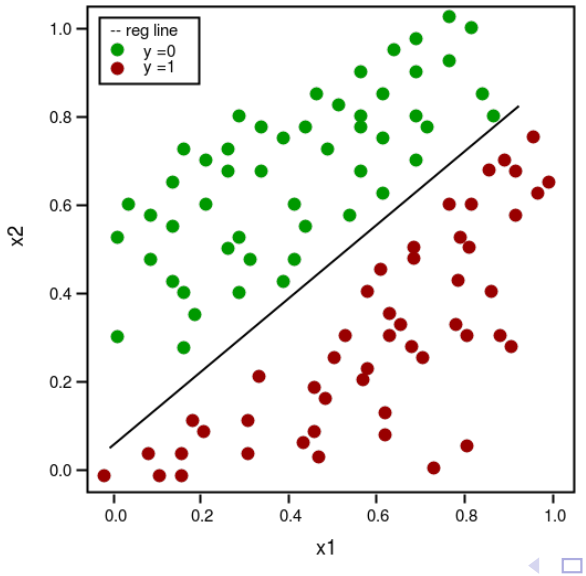
$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Classification

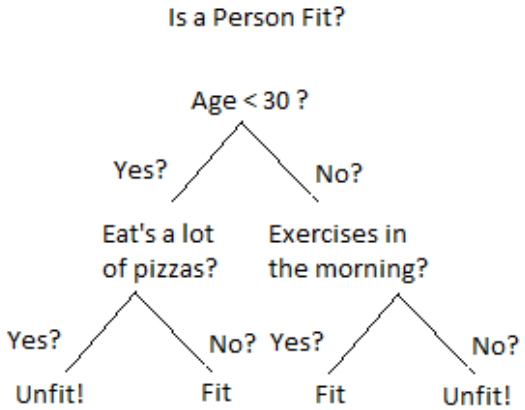
- Logistic regression $\frac{1}{1+e^{-z}}$ with z linear (polynomial) regression

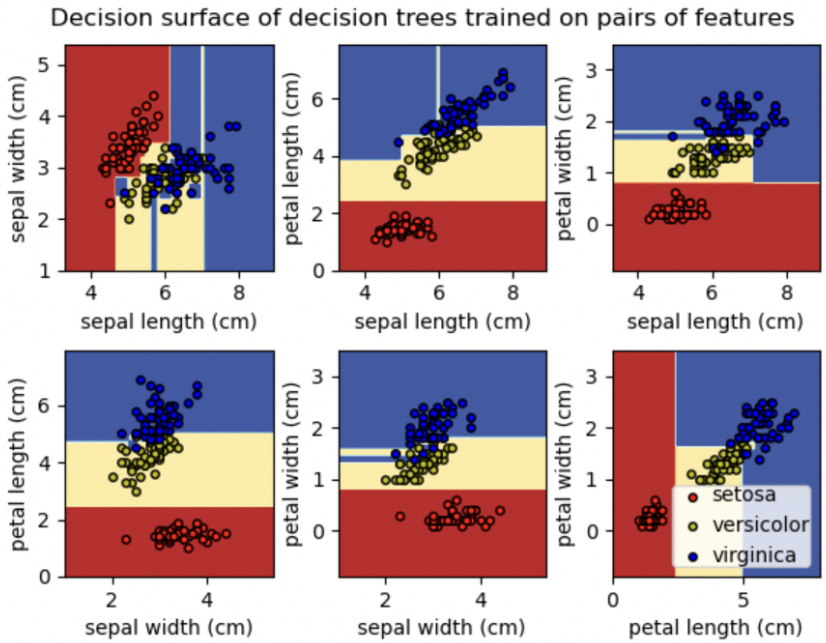






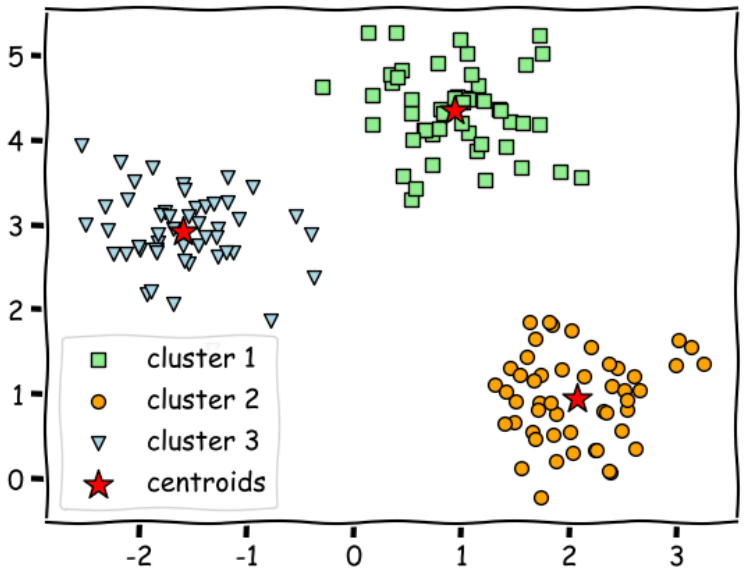
- Tree (ensemble models, RF, XGBoost)





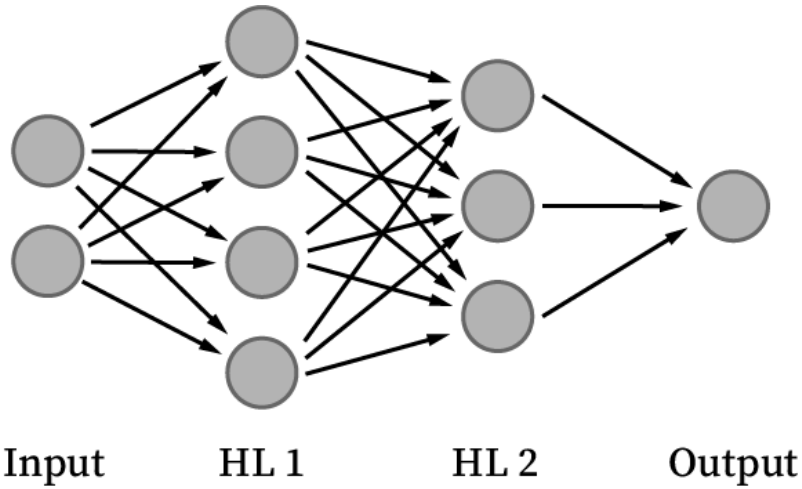
- K-means
  - Positionnement

1. Capture (iter 1)
2. Recentrage (iter 2)

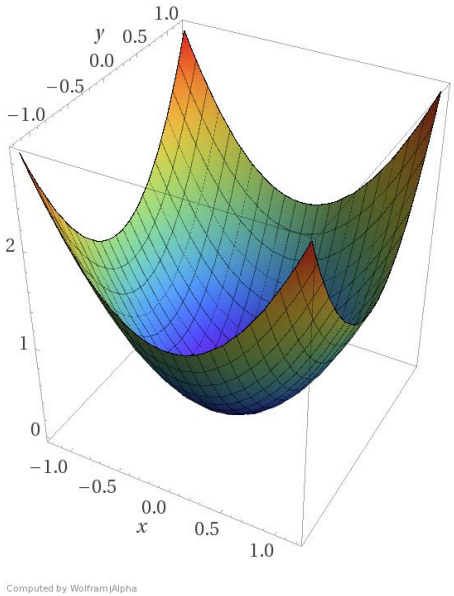


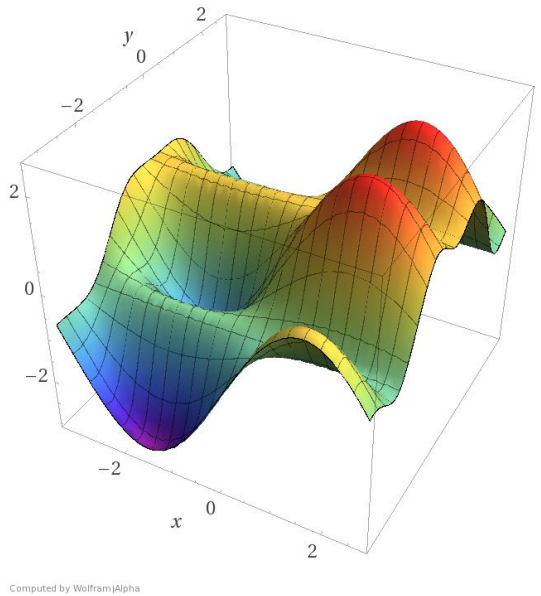
Classification

- Neural Network (Deep Learning: CNN)[2]



- Gradient descent





Unsupervised ML

- Clustering (grouping)
- Dimensionality reduction
- Reducing multicolinearity
- Models:
  - K-Means: entropy minimisation principle (min var intra, max var inter)
  - Hierarchical Clustering
  - PCA

General

- Parameters
- Hyperparameters
- Train, cross-validate, test
- Feature importance
- Data Leakage

Fit

- Generalization
- Complexity
- Over/Under-sampling
- Unbalanced Datasets (weights on cost function, SMOTE, Auto-encoder)

Metrics

- Regression
  - RMSE
  - R^2
- Classification
  - Accuracy
  - AUC, ROC Curve
  - Other metrics based on confusion matrix

Transfer Learning, Why?

- Training can be complicated, long and expensive
- Specific but complex (and similar) task (NLP)
- Few samples

What has been learnt?

- Weights value (or centroids)
- Hyperparameters

Deep Learning

- Optimize target objective on long term, intermediate steps on short term:
  - Increase task difficulty gradually
  - Better generalisation: Multi-task learning (RL, learn recognize unrelated objects)
  - Improve Neural Network architecture (genetic algorithms)

Some code examples

- Sklearn simple 4 lines of code
- More advanced Sklearn
- Deep Learning with Pytorch