# Spark ML Lib

Marco Milanesio

UCA - MSc DATA SCIENCE 2019

# The ML problem

- from **Data** to **Real (exploitable) knowledge**
  - is this email spam or not?
  - are estate prices related to pollution?
  - is there a face in the picture?
- Problem: knowledge is not concrete
- Solution: **mine** the data

# Knowledge discovery

- Preprocessing
- Data mining
- Result validation

# Preprocessing

- Data cleaning
- Data integration
- Data reduction (i.e., sampling)
- Data transformation (i.e., normalization)

# Data mining

- Classification and regression (supervised learning)
- Clustering (unsupervised learning)
- Frequent patterns
- Outliers detection

# Result validation

- Evaluate the performance of a model
- Depends on application requirements
- Multiple “scores”
  - RMSE
  - $R^2$
  - accuracy
  - ROC
  - . . .

# Supervised learning

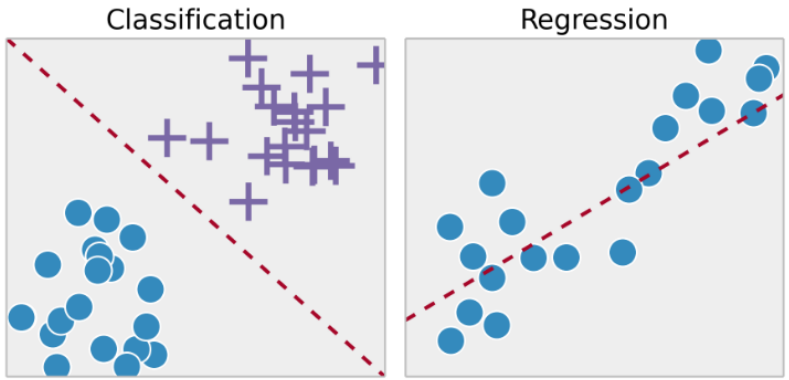
- Right answers are given
- Training data is labeled
- A model is prepared through a training process...
- ... up until a certain level of accuracy

# Supervised learning

- $N$ **training examples**: $(x_1, y_1), \ldots, (x_n, y_n)$
- $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ is the **feature vector** of the $i^{th}$ sample
- $y_i$ is the $i^{th}$ feature vector **label**
- A learning algorithm seeks $y_i = f(X_i)$

# Classification vs. Regression

- Classification $\rightarrow$ labels
- Regression $\rightarrow$ continuous values



# In Spark

- Linear models
- Decision trees
- (Naive Bayes models)

# Linear Models

- Training dataset: $(x_1, y_1), \ldots, (x_n, y_n)$
- $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$
- Model the target as a function of a **linear predictor** applied to the input variables: $y_i = g(\mathbf{w}^T x_i)$
  - e.g., $y_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in}$
- **Loss function**:

$$f(\mathbf{w}) = \sum_{i=1}^{n} L(g(\mathbf{w}^T x_i), y_i)$$

- Optimization problem

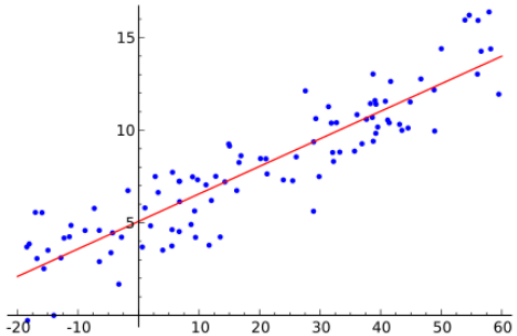
$$\min_{\mathbf{w} \in R^m} f(\mathbf{w})$$

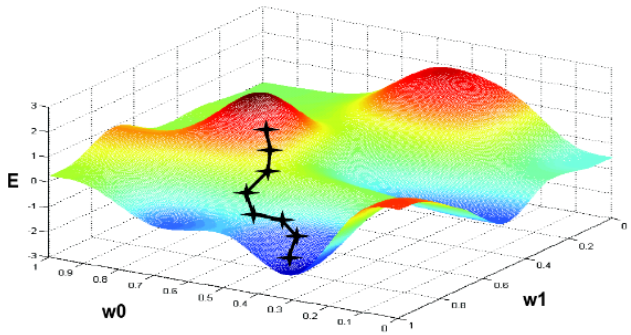
# Linear Regression

- $g(\mathbf{w}^T x_i) = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in}$
- **RMSE**:

$$\frac{1}{2}(\mathbf{w}^T x_i - y_i)^2$$

- Gradient descent



# Classification (Logistic Regression

- Binary classification: outputs $[0, 1]$
- $$g(\mathbf{w}^T x_i) = \frac{1}{1 + e^{-\mathbf{w}^T x}}$$
- sigmoid function
- if $g(\mathbf{w}^T x_i) > 0.5$ then $y_i = 1$ else $y_i = 0$

# Decision Tree

- Recursive binary partitioning of the feature space
- Greedy algorithm:
  - Find the best split condition (impurity measure)
    - how well two classes are separated
    - in Spark: *variance* (regression); *gini* and *entropy* (classification)
  - Stops when no improvement is possible
    - *minInfoGain, minInstancesPerNode, maxDepth, ...*

# Random Forest

- Train a set of decision trees separately
- Training done in parallel
- Inject randomness into training process to diversify all trees

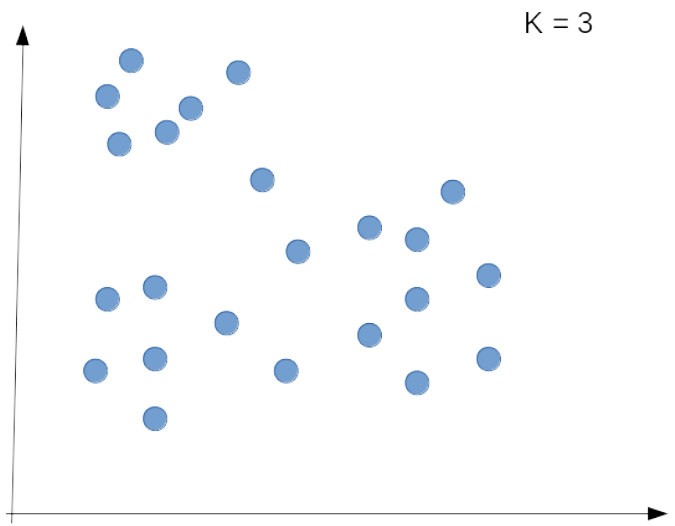
# Clustering

- Clustering is a technique for finding similarity groups (clusters)
- It groups *similar* data in one cluster and *different* data into different clusters
- No class values denoting an *a-priori* grouping is given

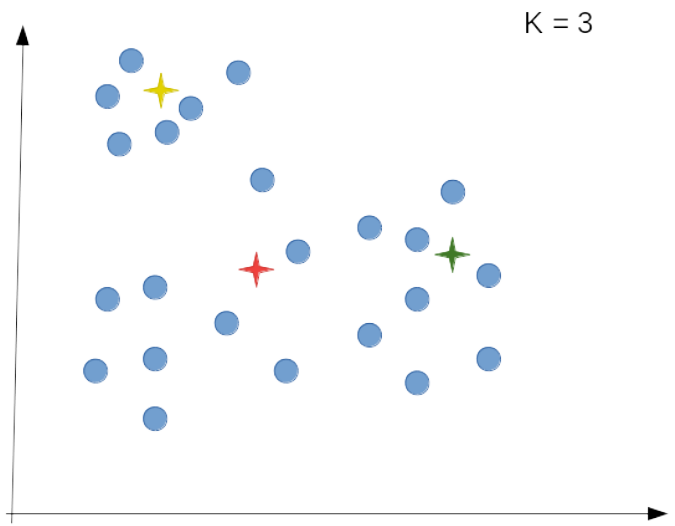
# K-means

- $K$ number of clusters (given)
- One **mean** per cluster
- Initialization: pick $K$ centers at random
- Iteration: assign each point to the nearest mean and move mean to center of its cluster
- Stop: difference to the center.

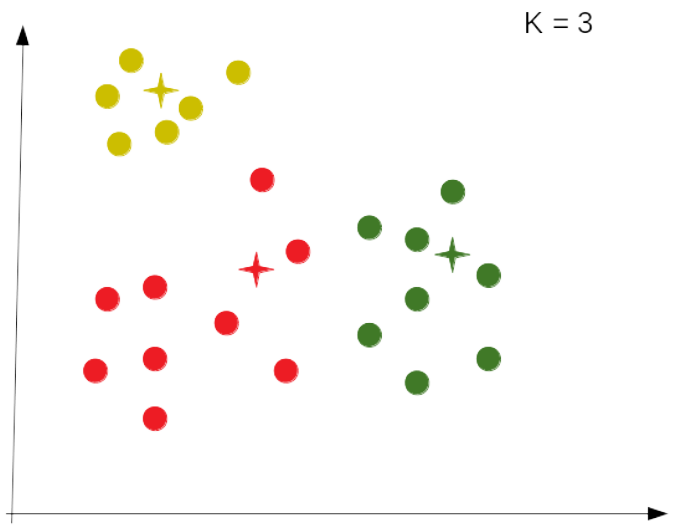
# K-means



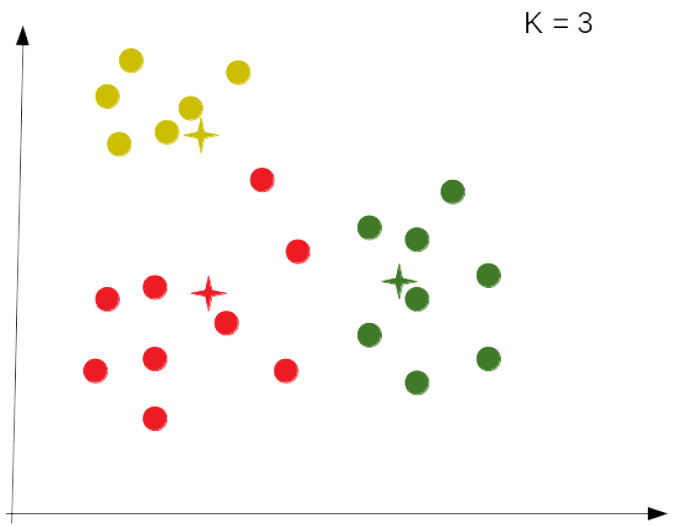
# K-means



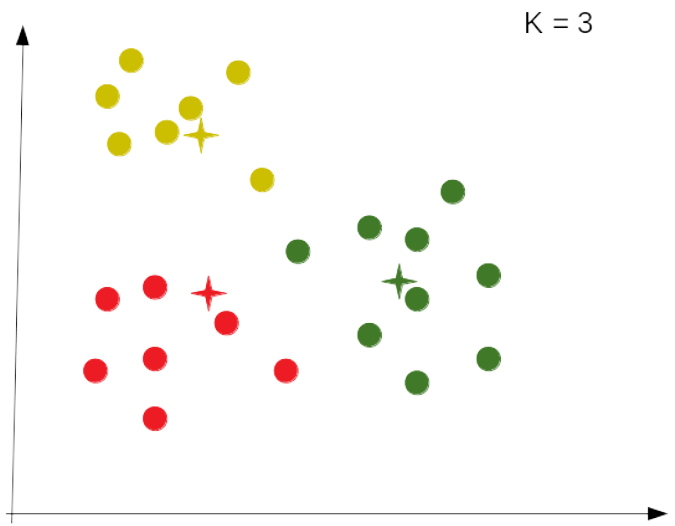
# K-means



# K-means



# K-means



# Model evaluation: Classification

- For each data point:
  - True output
  - model-generated prediction
- Four categories:
  - True positive (TP): *label* = 1 ; *prediction* = 1
  - True negative (TN): *label* = 0 ; *prediction* = 0
  - False positive (FP): *label* = 0 ; *prediction* = 1
  - False negative (FN): *label* = 1 ; *prediction* = 0

# Model evaluation: Classification

- Precision (positive predictive value): the fraction of retrieved instances that are relevant
- Recall (sensitivity): the fraction of relevant instances that are retrieved
- F-measure:

$$(1 + \beta^2) \frac{precision * recall}{\beta^2 * precision + recall}$$

# Model evaluation: Regression

- MSE
- RMSE
- MAE
- ...