Machine Learning

Università Roma Tre
Dipartimento di Ingegneria
Anno Accademico 2021 - 2022

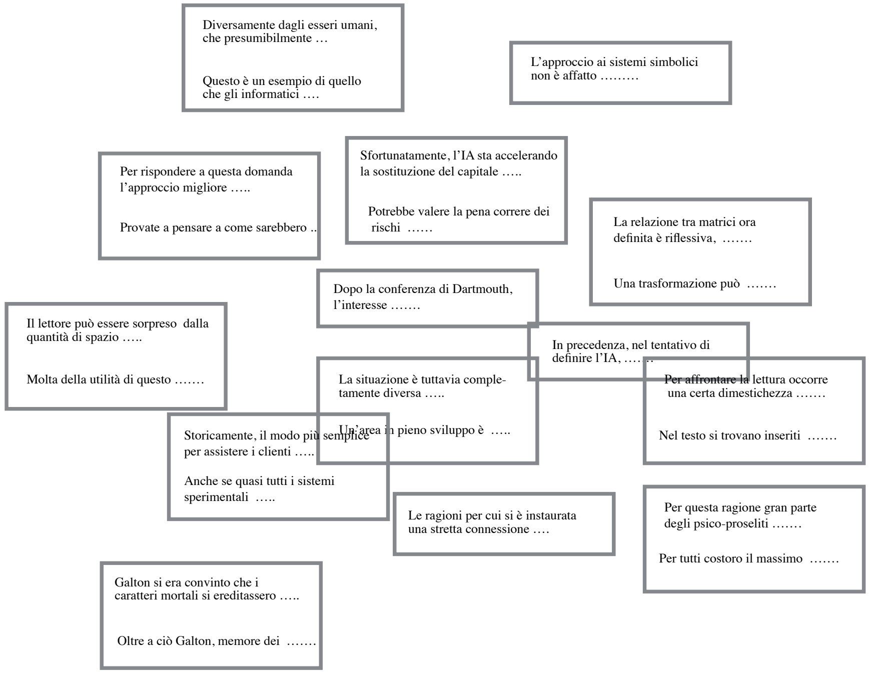
Algoritmo K-NN

Sommario

- Ripasso su Information Retrieval
- Algoritmo k-NN
- kd-trees per k-NN

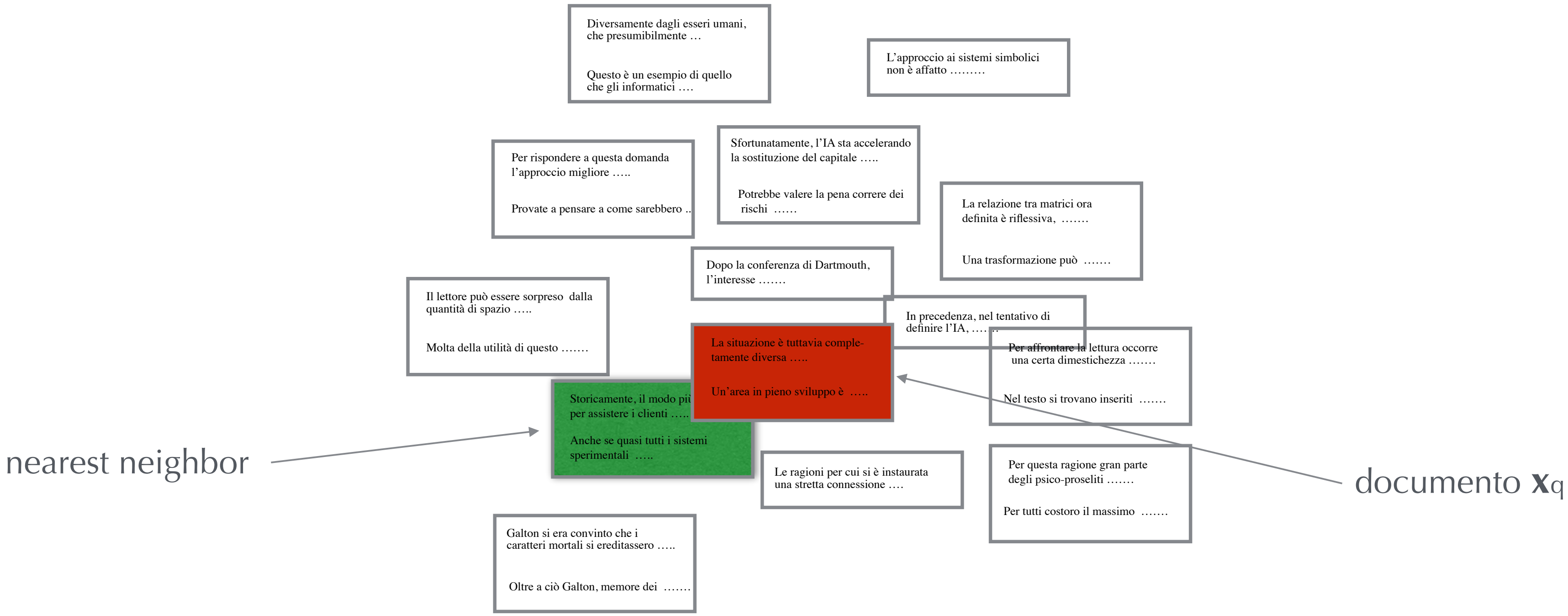
Document Retrieval

- Supponiamo di avere disponibile un corpus di documenti: Come possiamo misurare la similarità tra di loro? Come possiamo effettuare ricerche?



Nearest Neighbor

- Obiettivo: dato un documento $\mathbf{x}_q$, trovare l'articolo più simile nel corpus di documenti disponibili:



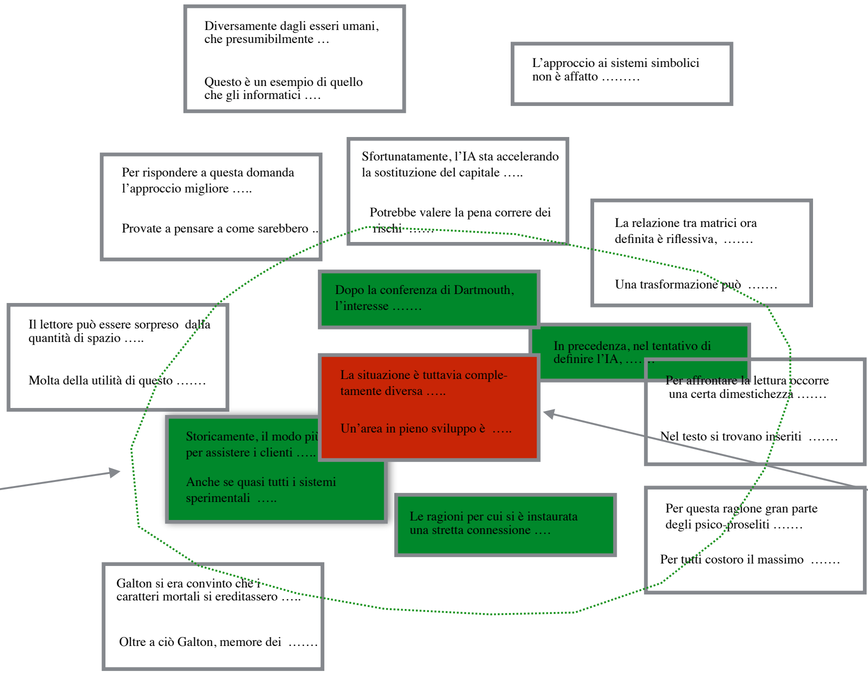
Algoritmo 1-NN

- Input: documento $\mathbf{x}_q$ per la query e documenti $\mathbf{x}_1$, $\mathbf{x}_2$, … , $\mathbf{x}_N$
- Output: documento $\mathbf{x}_i$ più vicino (nearest_doc) a $\mathbf{x}_q$

```
dist_min = ∞
nearest_doc = ∅
for  i = 1, ..., N
    δ = distanza(x_q, x_i)  ; distanza tra documento query e documento i-esimo
    if  δ < dist_min
        nearest_doc = x_i  ; documento più vicino corrente
        dist_min = δ       ; distanza minima corrente
return  nearest_doc
```

k Nearest Neighbors

- Obiettivo: dato un documento $\mathbf{x}_q$, trovare i k articoli più simili nel corpus di documenti disponibili:



k nearest neighbors

documento $\mathbf{x}_q$

Algoritmo k-NN

- Input: documento $\mathbf{x}_q$ per la query e documenti $\mathbf{x}_1$, $\mathbf{x}_2$, … , $\mathbf{x}_N$
- Output: lista dei k documenti più vicini a $\mathbf{x}_q$

```
lista_k_dist_min = sort(δ_1, δ_2, ..., δ_k)
lista_k_nearest_doc = sort(x_1, x_2, ..., x_k)
for  i = k + 1, ..., N
    δ = distanza(x_q, x_i)  ; distanza tra documento query e documento i-esimo
    if  δ < lista_k_dist_min[k]
        inserisci δ in lista_k_dist_min        ; inserimento in lista ordinata
        inserisci x_i in lista_k_nearest_doc   ; inserimento in lista ordinata
return  lista_k_nearest_doc
```

Criticità nella NN search

- Per effettuare una ricerca dei nearest neighbors occorre risolvere i seguenti problemi:
    - Come rappresentare gli item coinvolti (nel nostro esempio i documenti).
    - Come valutare la distanza tra gli item, ossia definire una metrica che consenta di calcolare la similarità tra i vari item.

Richiami su Rappresentazione dei Documenti

- Vediamo ora due possibili metodi per la rappresentazione dei documenti non strutturati:
  - bag of words
  - *tf-idf* (term frequency - inverse document frequency)

Modello Bag-of-Words

- In questo modello è ignorato l'esatto ordine dei termini nel documento.
- Viene preso in considerazione solo il numero di occorrenze (*term frequency*: *tf*) di ogni termine nel documento.
- In tal modo è possibile rappresentare ogni documento mediante un vettore di occorrenze:

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Modello Bag-of-Words

- Un problema che emerge in questa semplice rappresentazione è relativa ai termini poco frequenti ("rare words").
- In effetti, in tale rappresentazione tutti i termini sono considerati ugualmente importanti.
- In realtà certi termini hanno poca capacità discriminante ai fini della determinazione della rilevanza di un documento (e.g., quando ne calcoliamo la distanza rispetto ad un altro).
- Ad esempio, nel caso di una collezione di documenti relativi all'industria automobilistica, è piuttosto probabile avere il termine "automobile" in quasi ogni documento.
- Tali termini dominerebbero dunque quelli più rari.

Modello TF-IDF

- Una rappresentazione alternativa che possiamo considerare è quella chiamata *tf-idf*.
- Come vedremo, questa rappresentazione enfatizza i termini "importanti", individuati dalle seguenti caratteristiche:
  - appaiono frequentemente in un documento ("common locally")
  - appaiono raramente nel corpus ("rare globally")

Modello TF-IDF

- Definiamo *Document Frequency* (*df*) per il termine *t* come il numero di documenti nel corpus che contengono *t*.
- Definiamo inoltre l'*Inverse Document Frequency* come segue:

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$$

dove N è la cardinalità del corpus.

Modello TF-IDF

ESEMPIO:

Nella seguente tabella sono riportati alcuni esempi di valori *df* e *idf* relativi alla collezione Reuters, costituita da 806.791 documenti:

termine	df_t	idf_t
car	18.165	1,65
auto	6.723	2,08
insurance	19.241	1,62
best	25.235	1,5

Modello TF-IDF

- Il tf-idf è definito come segue:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \cdot \text{idf}_t$$

- In sostanza il *tf-idf* per un termine *t* in un documento *d* assegna al termine un peso nel documento che è:
  - molto elevato quando *t* è molto frequente in un piccolo numero di documenti;
  - più basso quando il termine è poco frequente nel documento, oppure quando è presente in molti documenti;
  - il più basso quando il termine compare in tutti i documenti.

Metriche

- Vediamo ora come possiamo calcolare la distanza tra due *item*.
- Nel semplice caso di una dimensione possiamo definire la funzione distanza come segue (Distanza Euclidea):

$$\text{distanza}(x_i, x_q) = |x_i - x_q|$$

- Nel caso di *d* dimensioni, la funzione *distanza* può assumere la seguente forma (Distanza Euclidea):

$$\text{distanza}(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{(\mathbf{x}_i[1] - \mathbf{x}_q[1])^2 + \cdots + (\mathbf{x}_i[d] - \mathbf{x}_q[d])^2}$$

che possiamo riscrivere come segue, in forma matriciale:

$$\text{distanza}(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{(\mathbf{x}_i - \mathbf{x}_q)^T \cdot (\mathbf{x}_i - \mathbf{x}_q)}$$

Metriche

- Nel caso in cui vogliamo pesare in modo diverso le varie dimensioni, possiamo usare una *Scaled Euclidean distance*:

$$\text{distanza}(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{a_1(\mathbf{x}_i[1] - \mathbf{x}_q[1])^2 + \cdots + a_d(\mathbf{x}_i[d] - \mathbf{x}_q[d])^2}$$

che possiamo riscrivere come segue, in forma matriciale:

$$\text{distanza}(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{(\mathbf{x}_i - \mathbf{x}_q)^T \cdot \mathbf{A} \cdot (\mathbf{x}_i - \mathbf{x}_q)}$$

dove:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_d \end{bmatrix}$$

Cosine Similarity

- Una metrica largamente utilizzata per quantificare la similarità tra due documenti $\mathbf{x}_i$ e $\mathbf{x}_q$ è la *cosine similarity*, che si avvale della rappresentazione vettoriale dei documenti:

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_q) = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_q}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_q\|}$$

  dove il numeratore rappresenta il prodotto scalare tra i due vettori e il denominatore il prodotto tra i moduli dei due vettori.

- L'effetto del denominatore è dunque quello di normalizzare i vettori $\mathbf{x}_i$ e $\mathbf{x}_q$ ottenendone i corrispondenti versori. Possiamo dunque riscrivere la precedente espressione come segue:

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_q) = \hat{\mathbf{x}}_i^T \cdot \hat{\mathbf{x}}_q$$

Cosine Similarity

- Consideriamo ad esempio i documenti in figura a), rappresentati mediante i vari *tf*. La quantità:

$$\|\mathbf{x}\| = \sqrt{\sum_{j=1}^{d} x_j^2}$$

ha i valori 30,56, 46,84 e 41,30 per Doc1, Doc2 e Doc3. Applicando la normalizzazione otteniamo la figura b):

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

a)

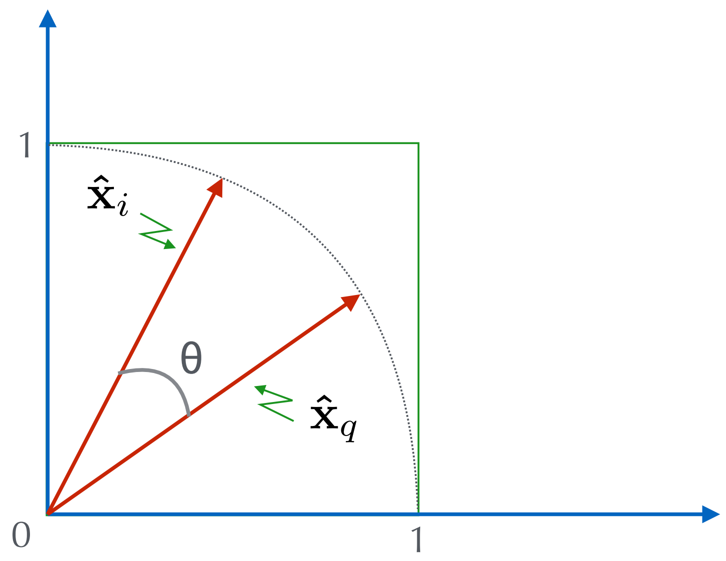
	Doc1	Doc2	Doc3
car	0,88	0,09	0,58
auto	0,10	0,71	0
insurance	0	0,71	0,70
best	0,46	0	0,41

b)

Cosine Similarity

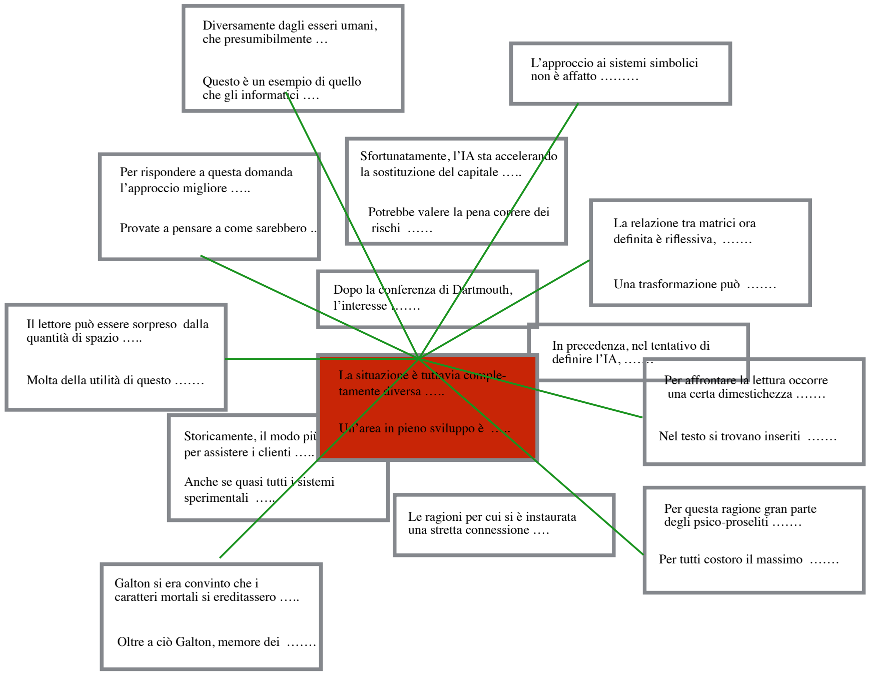
- La similarità definita in precedenza corrisponde al coseno dell'angolo tra i due vettori:

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_q) = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_q}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_q\|} = \cos(\theta)$$



K-NN: Complessità della ricerca

- Il calcolo delle distanze tra documenti può essere molto pesante computazionalmente quando N è molto elevato:



K-NN: Complessità della ricerca

- Dato un *query point*, il costo della scansione su tutti i punti è:
  - O(N) per una query per 1-NN
  - O(N log k) per una query per k-NN
- Per rendere più efficiente la ricerca è possibile utilizzare una particolare struttura dati, i *KD-Trees*.

KD-Trees

- Permette un’organizzazione strutturata degli item:
  - partiziona ricorsivamente i data point in “axis aligned boxes”.
- Comporta un più efficiente pruning dello spazio di ricerca.
- Ottiene buoni risultati in dimensioni “low-medium”.

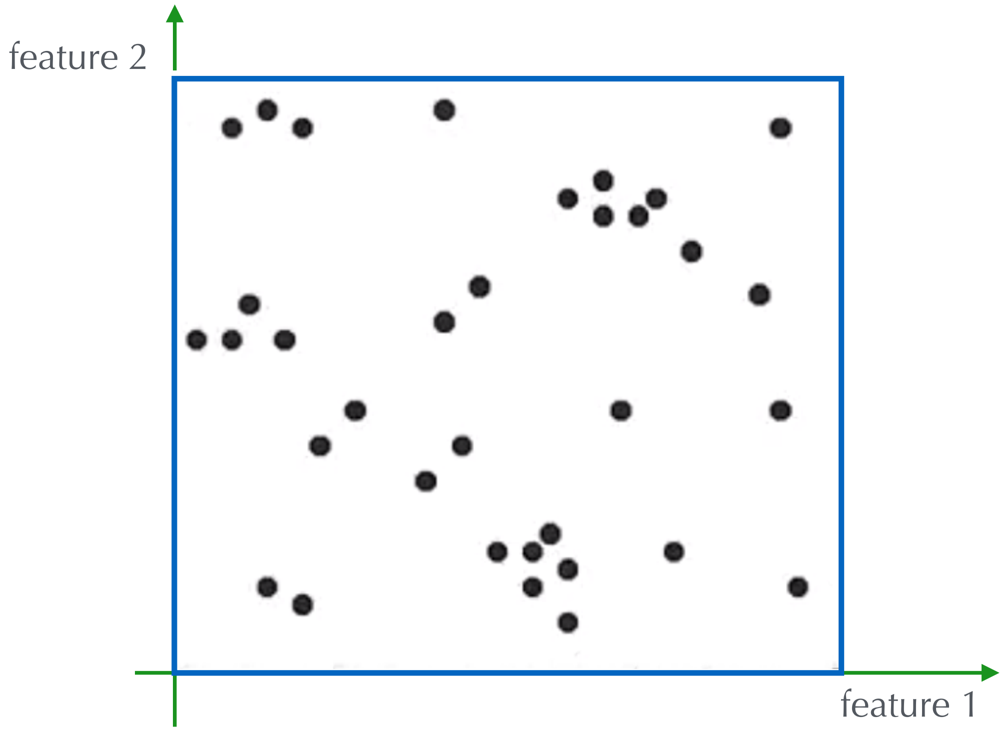
- Riferimenti:

Bentley, J.L. “Multidimensional Binary Search Trees Used for Associative Searching”, in: *Communications of the ACM*, **18**(9), 1975, pp. 509-517.

Friedman, J.H., Bentley, J.L., Finkel, R.A. “An Algorithm for Finding Best Matches in Logarithmic Expected Time”, in: *ACM Transactions on Mathematical Software*, **3**(3), 1977, pp. 209-226.

KD-Trees

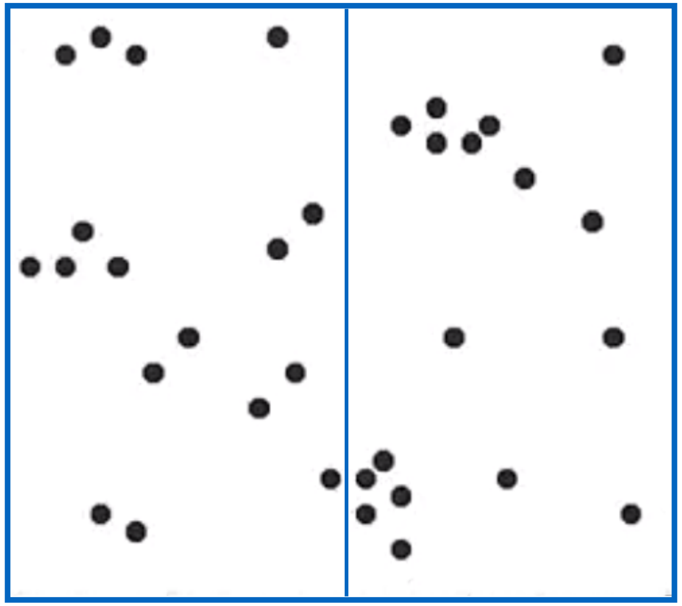
- Costruzione dell'albero:



Data Point	x[1]	x[2]
1	0,00	0,00
2	1,00	4,31
3	0,13	2,85
…	…	…

KD-Trees

- Split relativo alla prima feature:



0,5

$x[1] \leq 0,5$

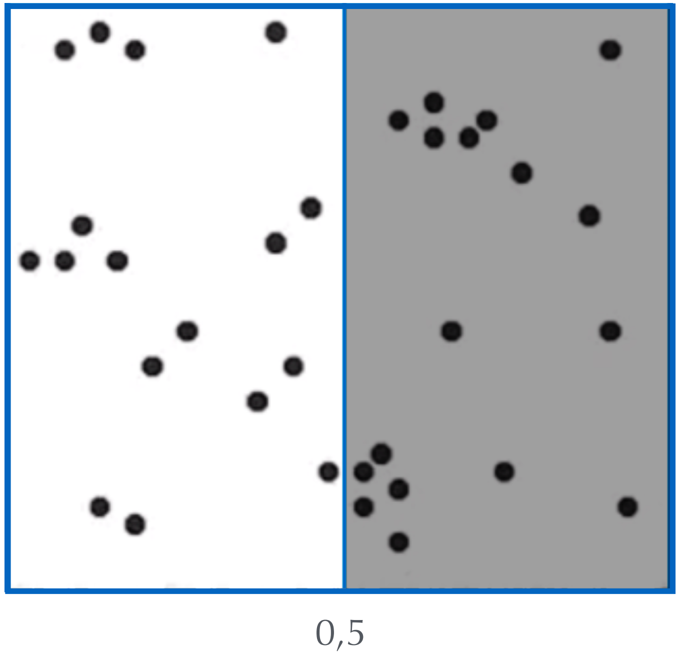
Data Point	x[1]	x[2]
1	0,00	0,00
3	0,13	2,85
…	…	…

$x[1] > 0,5$

Data Point	x[1]	x[2]
2	1,00	4,31
…	…	…

KD-Trees

- Consideriamo ora la parte sinistra:



$x[1] \leq 0,5$

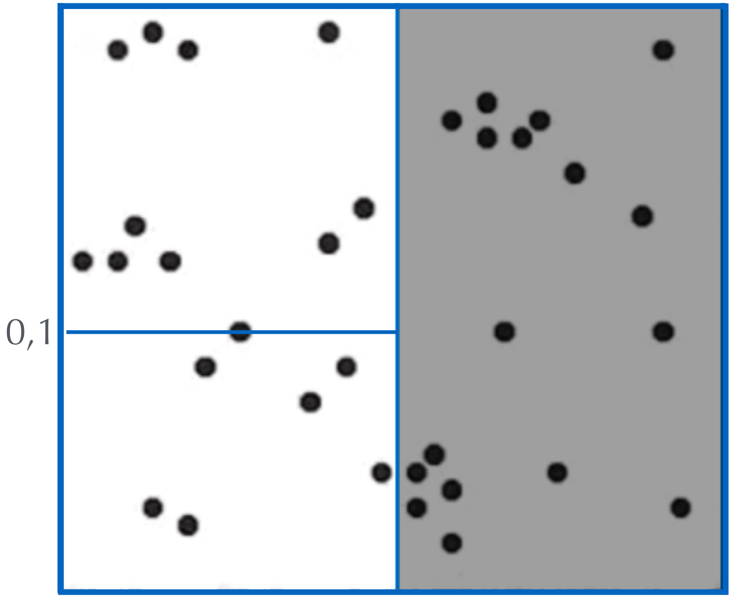
Data Point	x[1]	x[2]
1	0,00	0,00
3	0,13	2,85
…	…	…

$x[1] > 0,5$

Data Point	x[1]	x[2]
2	1,00	4,31
…	…	…

KD-Trees

- Split relativo alla seconda feature:



$x[2] \leq 0{,}1$

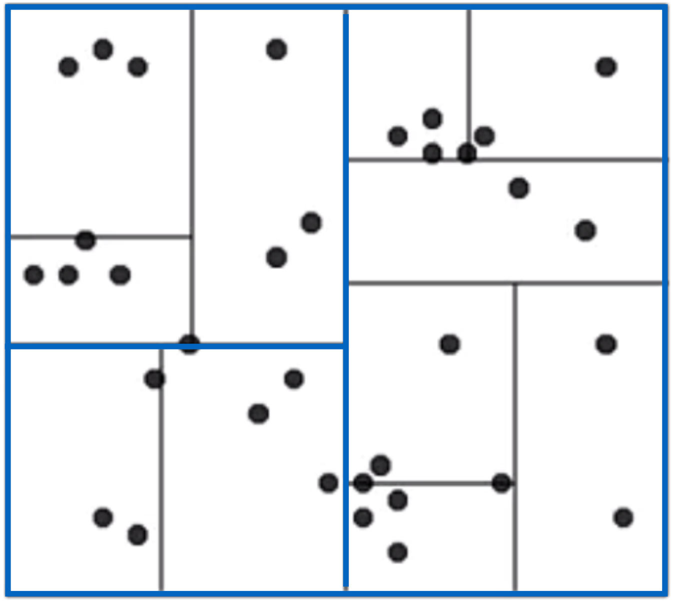
Data Point	x[1]	x[2]
3	0,13	2,85
…	…	…

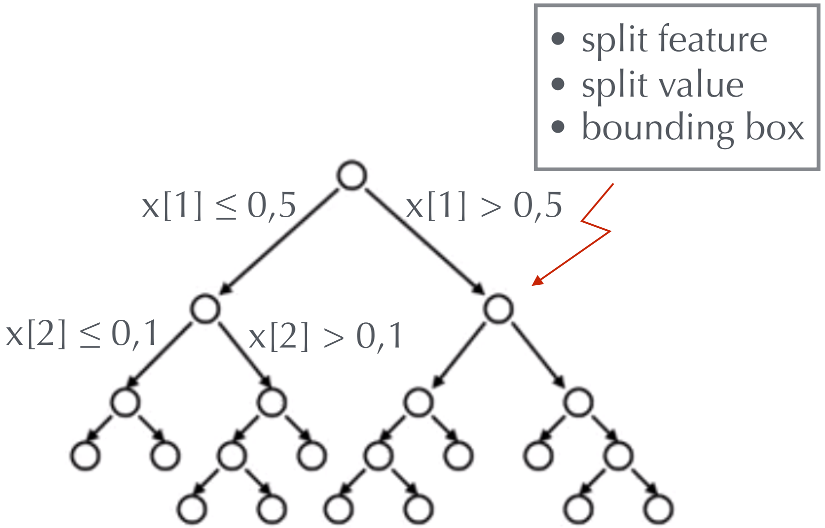
$x[2] > 0{,}1$

Data Point	x[1]	x[2]
1	0,00	0,00
…	…	…

KD-Trees

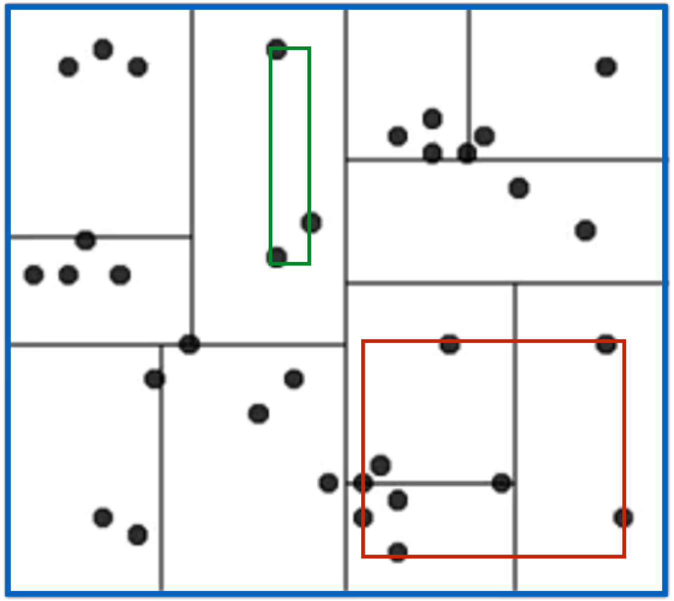
- Si procede in tal modo fino a completare l'albero:

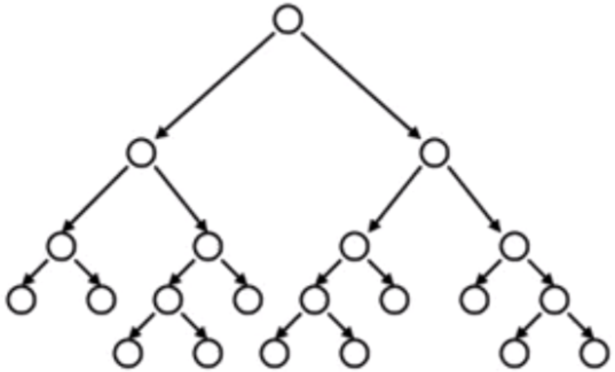




KD-Trees

- Esempi di bounding box:



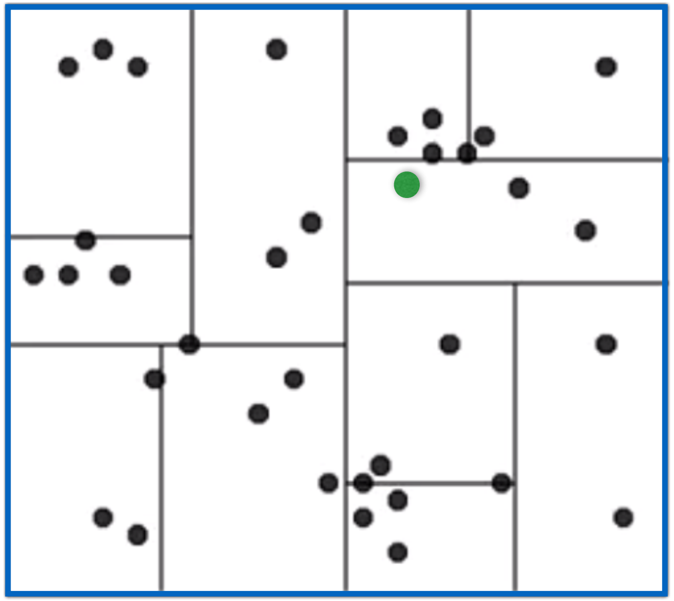


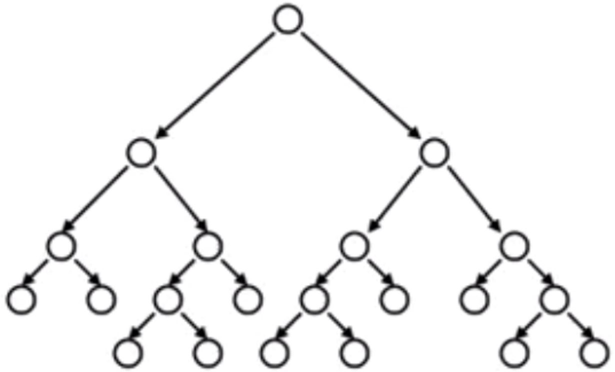
KD-Trees

- Euristiche per effettuare le decisioni sugli splitting:
    - Scelta della dimensione (la più ampia, dim. alternate)
    - Valore della feature a cui effettuare lo split (mediana, centro del box)
    - Condizione di terminazione (numero di punti sotto una determinata soglia, larghezza del box sotto una determinata soglia)

KD-Trees

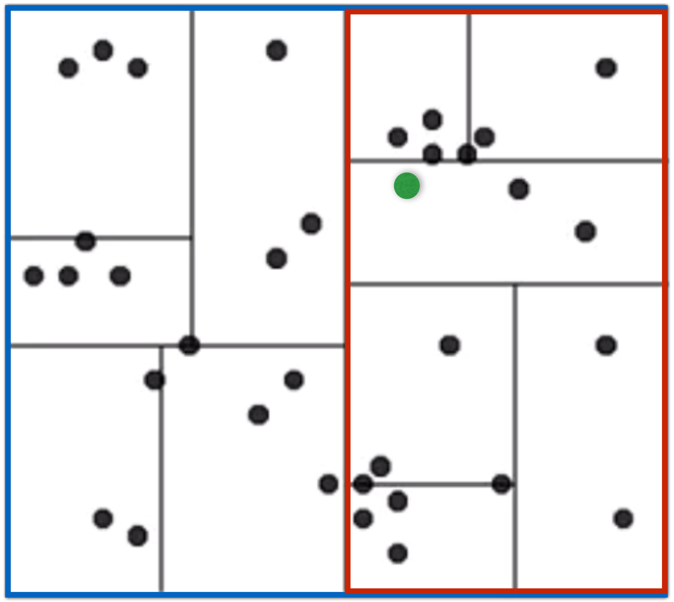
- Dato un query point (in verde), attraversiamo l'albero alla ricerca del nearest neighbor.

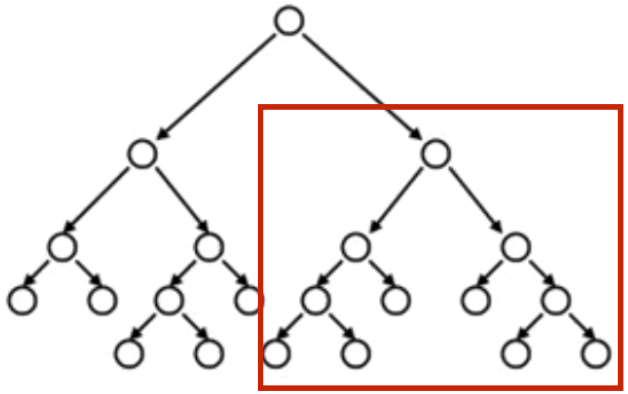




KD-Trees

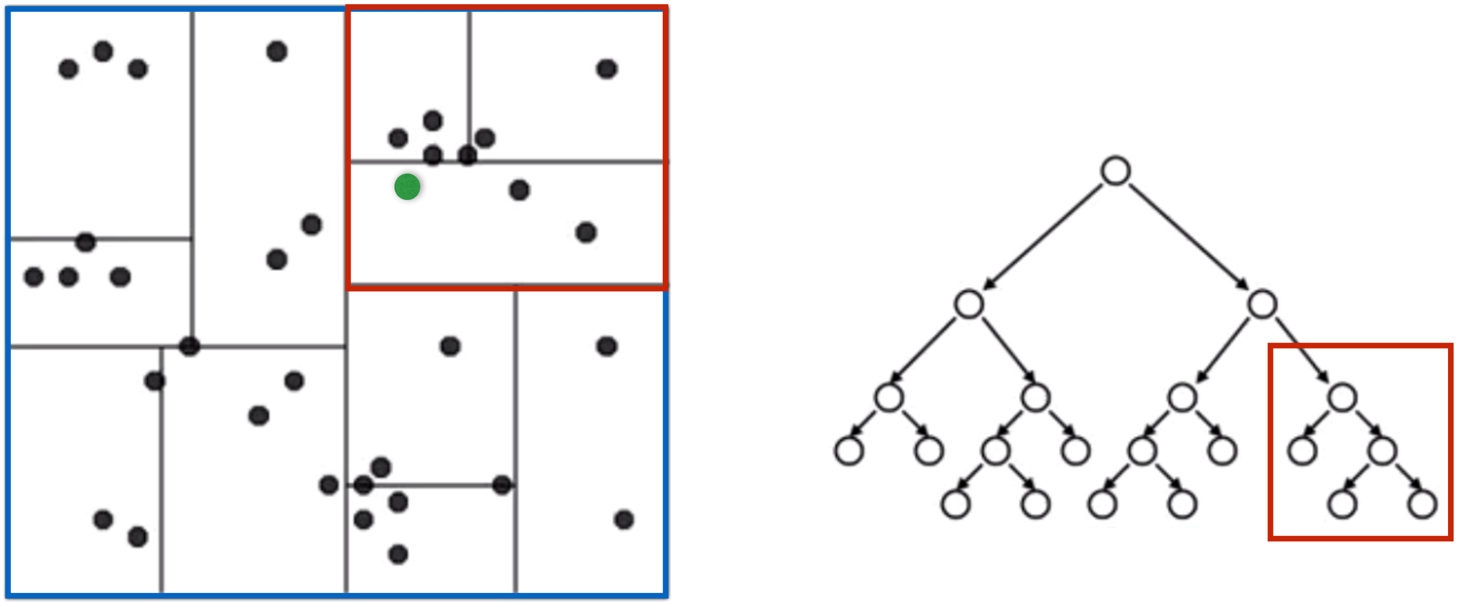
- Prima metà dell’area:





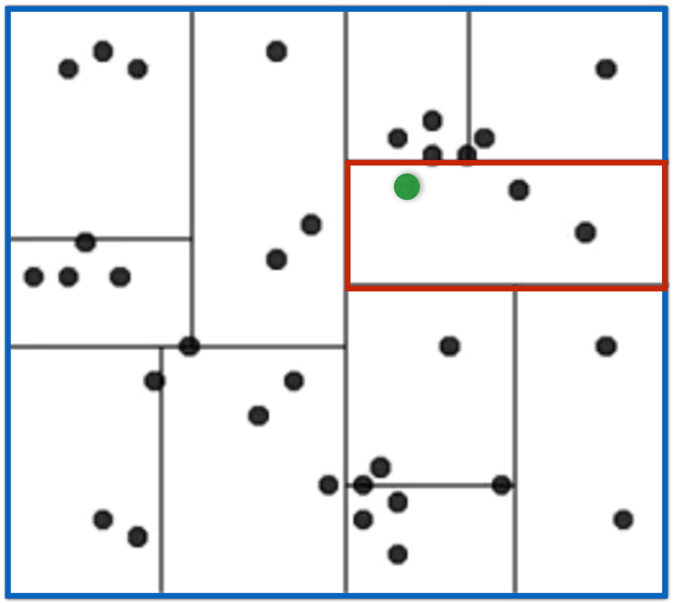
KD-Trees

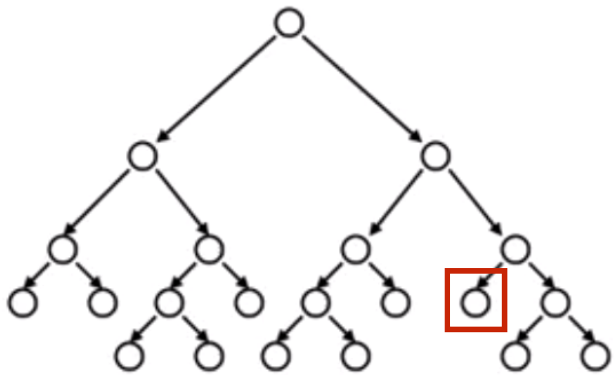
- .. e così via …



KD-Trees

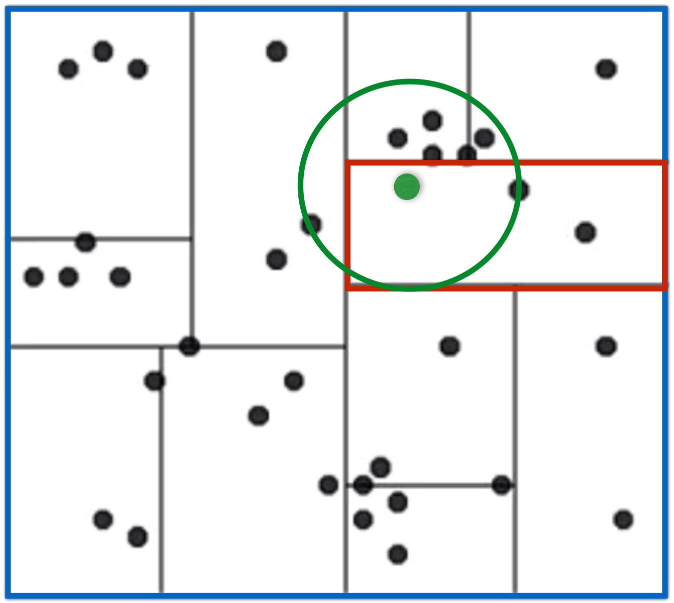
- Abbiamo raggiunto la foglia che contiene il query point:

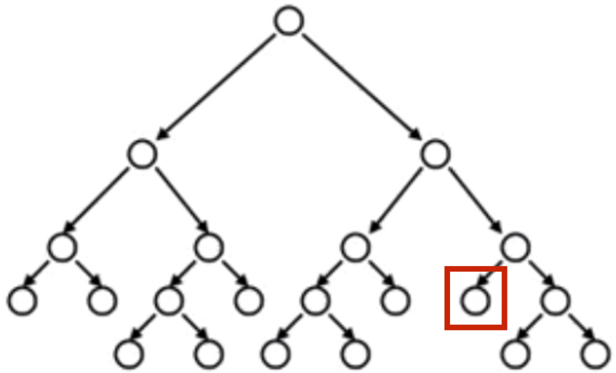




KD-Trees

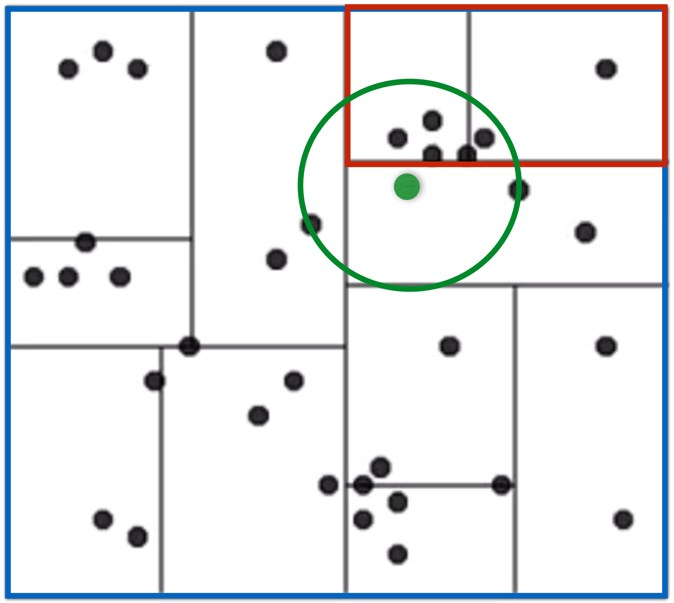
- Calcolo della distanza del NN tra i punti contenuti nella foglia:

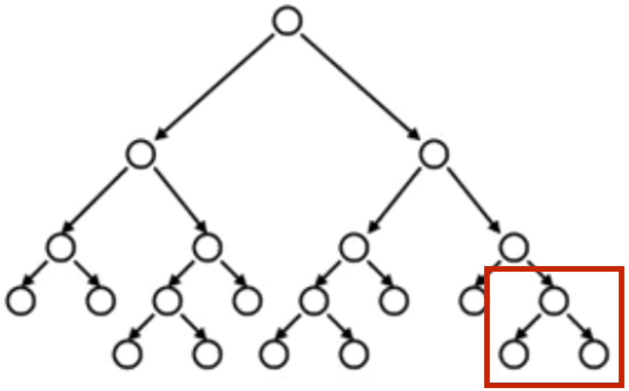




KD-Trees

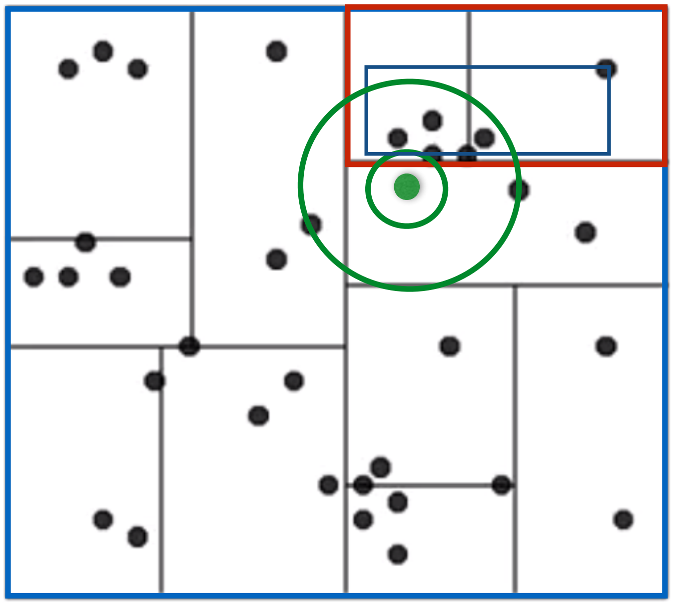
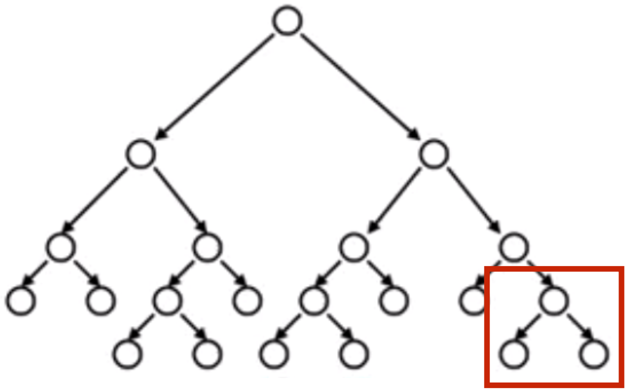
- Backtrack e proviamo altri rami per ogni nodo visitato:





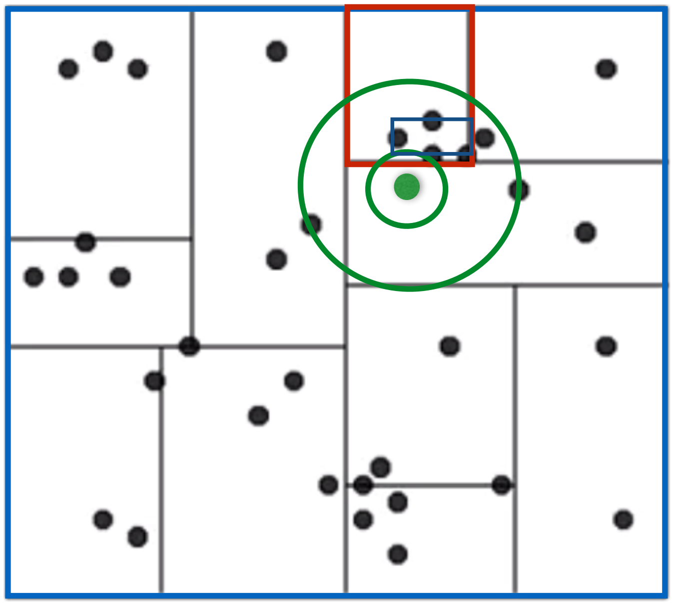
KD-Trees

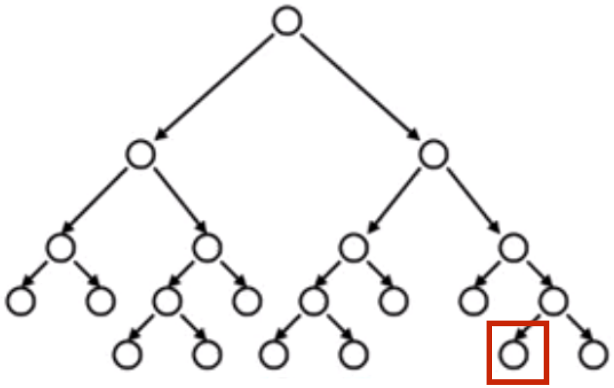
- Valutiamo la distanza dal bounding box:

KD-Trees

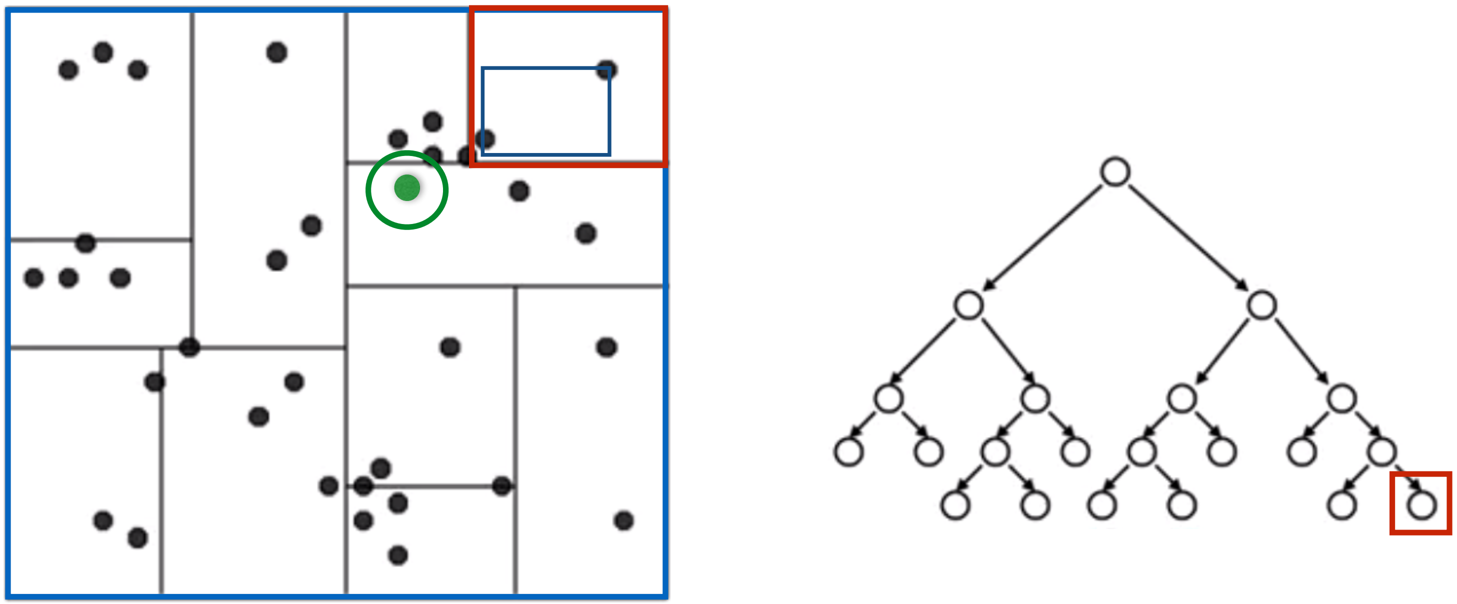
- La distanza è minore di quella corrente, perciò visitiamo i sottoalberi (in questo caso le foglie). La prima ha distanza minore:





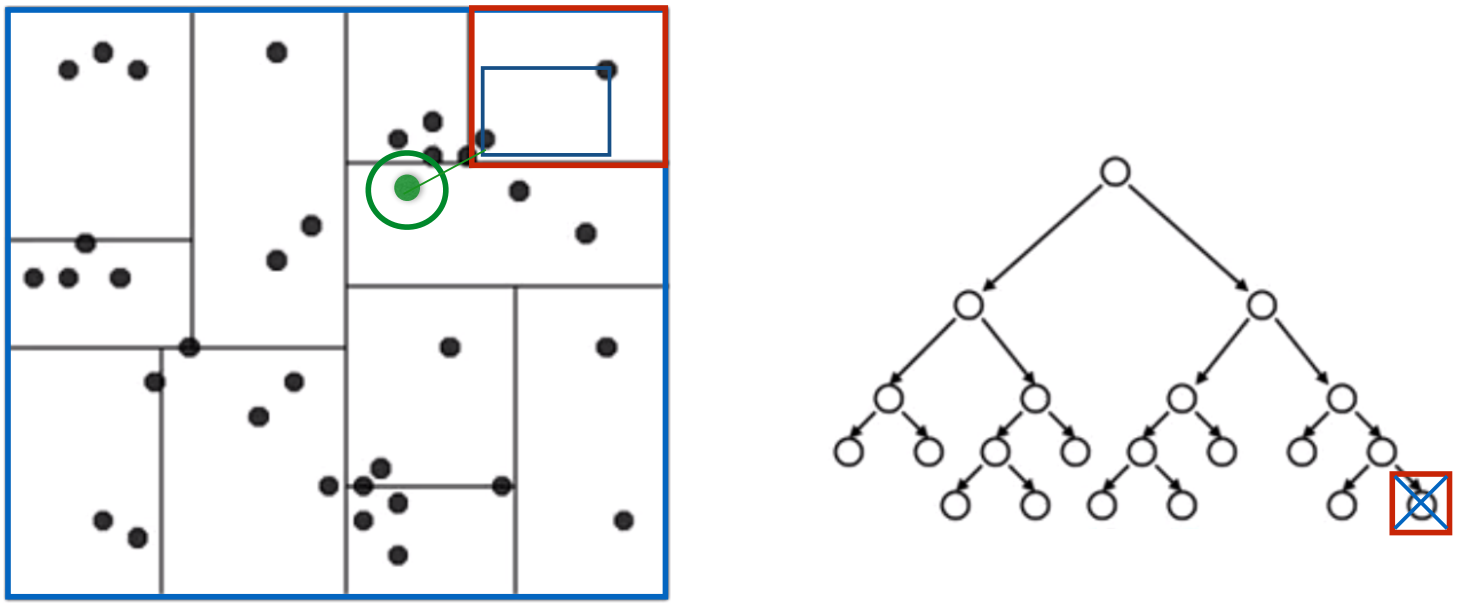
KD-Trees

- Backtrack e visitiamo l'altra foglia:



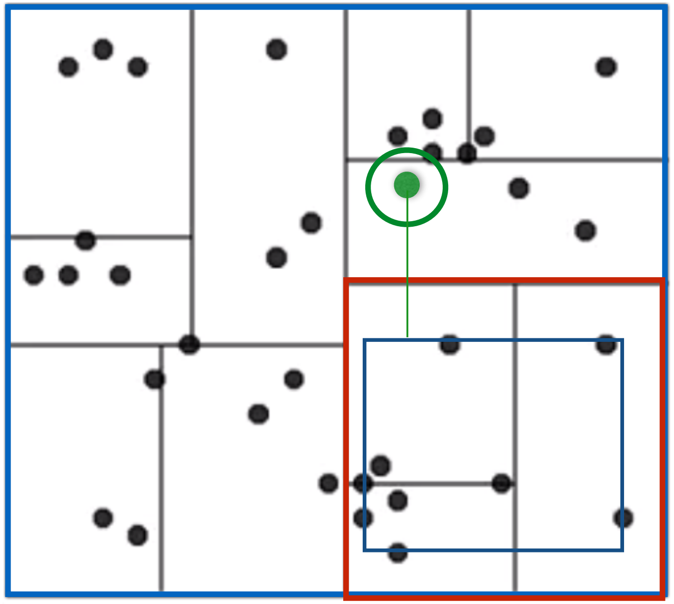
KD-Trees

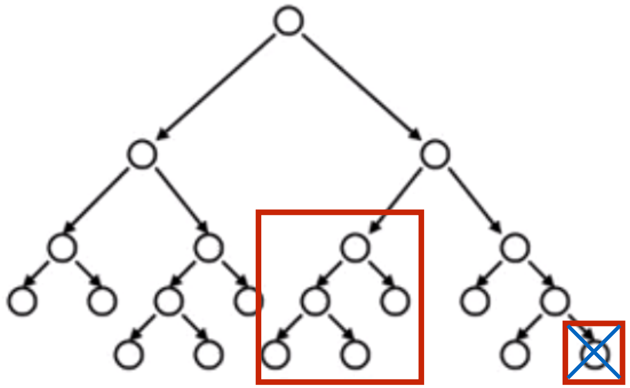
- La distanza dal bounding box è superiore alla minima, perciò possiamo potare il ramo:



KD-Trees

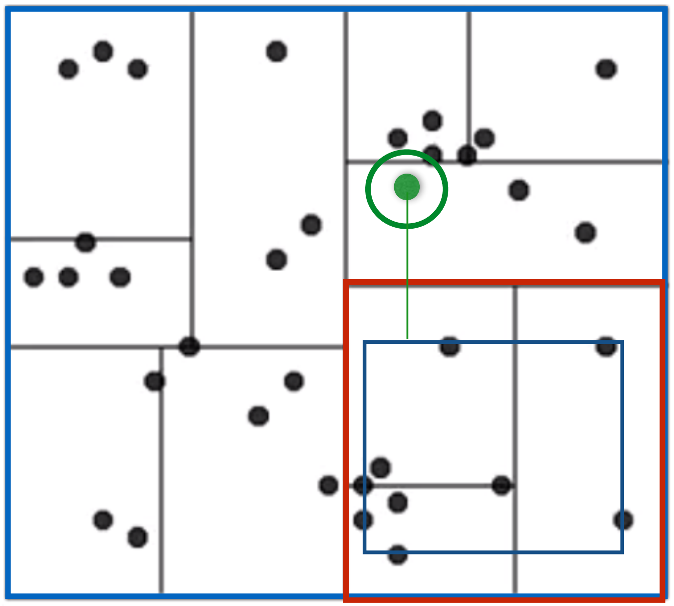
- Backtrack e proviamo altri rami per ogni nodo visitato:

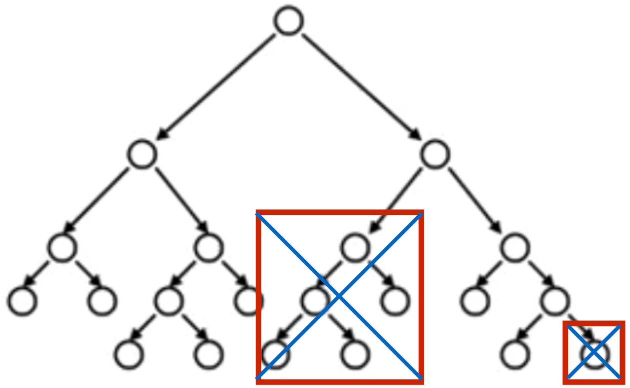




KD-Trees

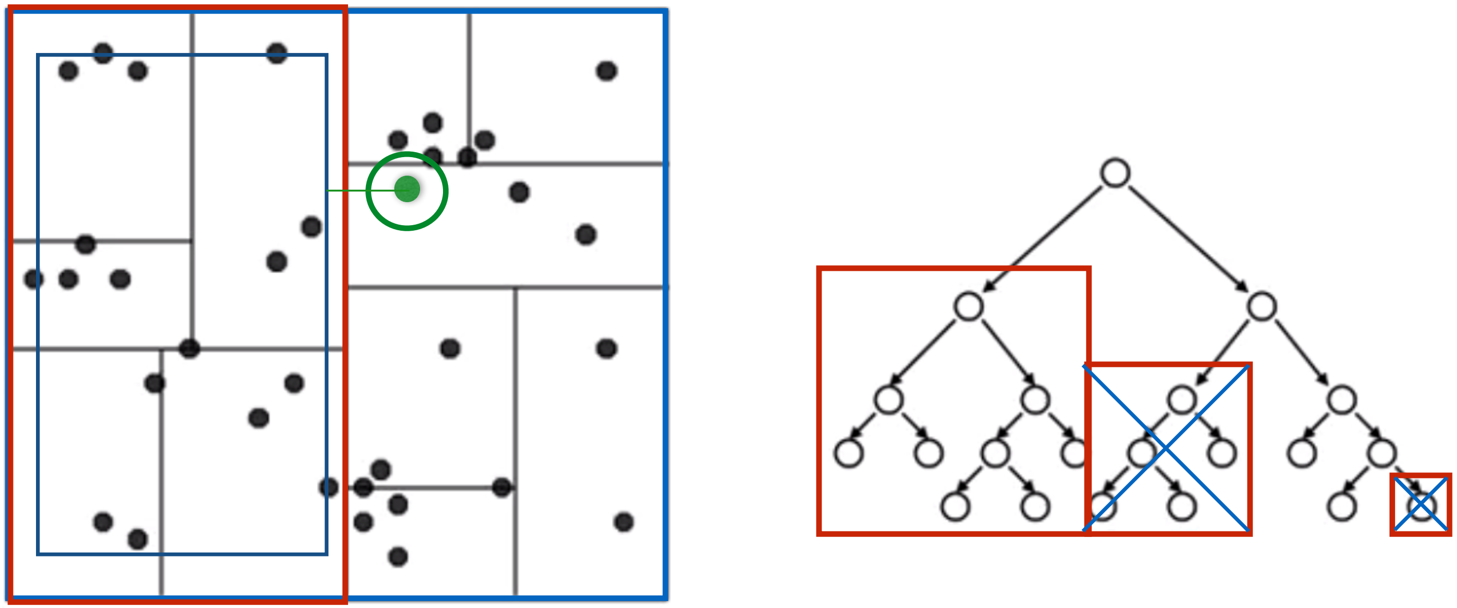
- La distanza dal bounding box è superiore alla minima corrente, perciò possiamo potare il ramo:





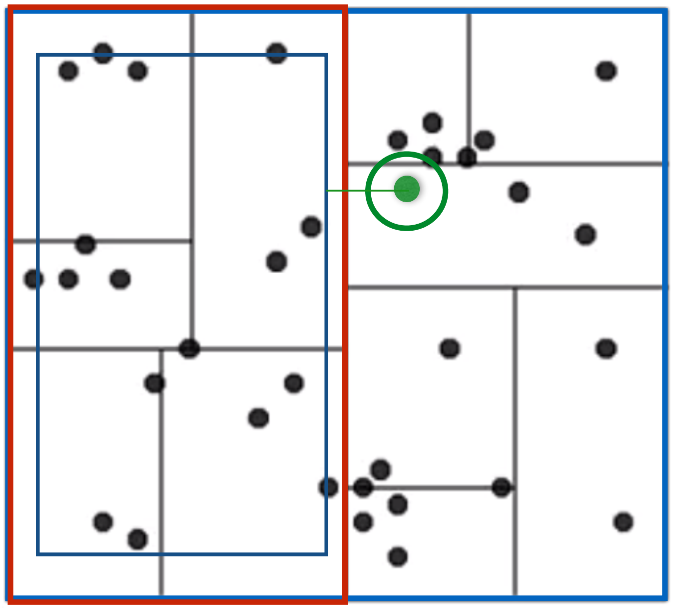
KD-Trees

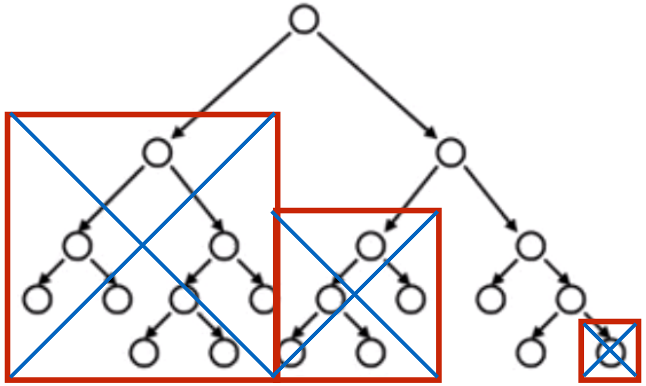
- Backtrack e proviamo altri rami per ogni nodo visitato:



KD-Trees

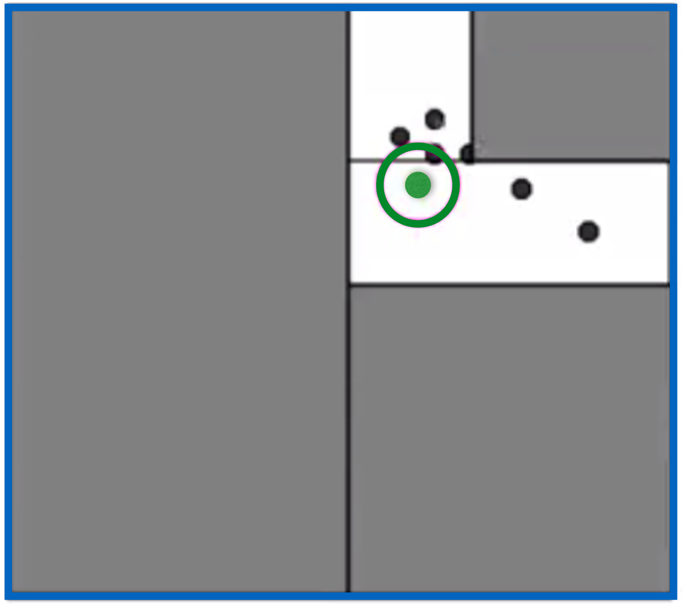
- La distanza dal bounding box è superiore alla minima corrente, perciò possiamo potare il ramo:

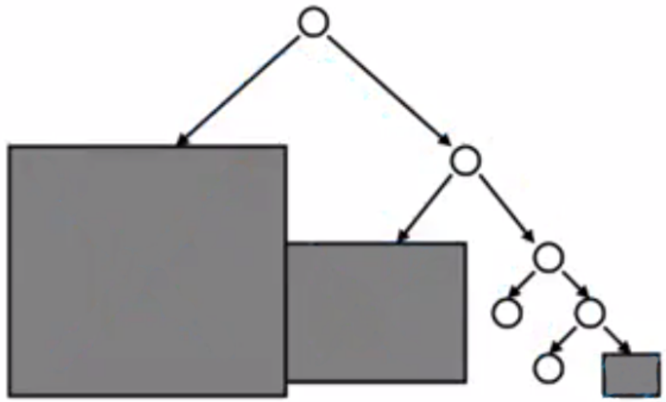




KD-Trees

- Pruning complessivo:





Riferimenti

- Watt, J., Borhani, R., Katsaggelos, A.K. *Machine Learning Refined*, 2nd edition, Cambridge University Press, 2020.
- James, G., Witten, D., Hastie, T., Tibishirani, R. *An Introduction to Statistical Learning*, Springer, 2013.
- Ross, S.M. *Probabilità e Statistica per l'Ingegneria e le Scienze*, Apogeo, 3a edizione, 2015.
- *Machine Learning: Clustering & retrieval*, University of Washington - Coursera, 2017.
- Flach, P. *Machine Learning - The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, 2012.
- Murphy, K.P. *Machine Learning - A Probabilistic Approach*, The MIT Press, 2012.