**Apprendimento basato sulle istanze**

**Apprendimento**: Memorizzazione di tutte gli esempi di training.

**Classificazione**: Reperimento degli esempi simili e classificazione sulla base di quegli esempi.

È un **apprendimento lazy**, ritarda l’elaborazione fino a quando non è necessario classificare un’istanza. Ma il **costo di classificazione** può essere **elevato**, perché devono essere considerati tutti gli esempi (archiviazione).

**K-NN (K nearest neighbour)**

Classificatore supervisionato non parametrico. Usa la prossimità per fare classificazioni.

Le istanze sono punti di RN e i vicini sono definiti con la distanza Euclidea.

Gli svantaggi:

* Tutte le feature sono considerate per reperire esempi simili. Se il concetto dipende da poche, gli esempi simili sono a grande distanza.

Altre distanze che possono essere usate: Minkowski (p>0), Euclidea (p=2), Manhattan (p=1), Chebyshev (p=∞)

Gli attributi possono avere scale diverse. È necessario portare tutti gli attributi in un’unica scala (0-1). a’ = (a-amin)/(amax-amin).

Se gli attributi sono invece nominali, è necessario definire una distanza specifica.

**K-NN pesato sulla distanza**

Il contributo di ciascun vicino è pesato in base alla distanza dal punto da classificare. Maggio peso ai vicini più vicini.

**Pesando sulla distanza**, si potrebbero considerare **tutti gli esempi**, invece dei k più vicini. Lo svantaggio sarebbe computazionale, agirà più lentamente, e il **metodo diventa globale**, non più locale.

**Bias induttivo**: assunzione che la classificazione di una istanza xQ sia simile a quella di altre istanze che sono vicine secondo la distanza Euclidea.

**Curse of Dimensionality**

K-NN considera **tutte le feature**. Es: 20 attributi di cui solo 2 rilevanti per la classificazione delle istanze. Istanze che hanno valori identici per questi 2 attributi potrebbero essere distanti nello spazio delle istanze (20-NN). La distanza tra vicini sarebbe dominata dal gran numero di attributi irrilevanti.

**Soluzione**: Pesare ogni attributo diversamente. Variare la lunghezza degli assi nello spazio euclideo. Assi più corti per feature meno rilevanti, Assi più lunghi per feature più rilevanti. **Oppure** porre **alcuni pesi = 0**. Si eliminano quindi alcune feature.

**Valori bassi** di K: **Overfitting**.

**Valori alti**, si considerano aree più grandi, si appiana la previsione: **Underfitting**.

Nello scegliere K, meglio valori dispari, per non avere lo stesso #positivi = #negativi.