Appunti Data Mining

Prof. Dino Pedreschi, Mirco Nanni, Riccardo Guidotti, Salvatore Citraro

Anno accademico 2020/2021

Contents

Chapter 1

Introduzione alla disciplina

Data mining is the use of efficient techniques for the analysis of very large collections of data and the extraction of useful and possibly unexpected patterns in data (hidden knowledge)

1.1 Introduzione

Il *data mining* è una tecnologia che fonde metodi tradizionali di analisi dei dati con algoritmi capaci di processare e gestire di grandi quantità di dati, con l'obiettivo di estrarre informazioni. Un ruolo determinante in questa direzione lo hanno avuto gli avanzamenti che la tecnologia ha subito a partire dalla seconda metà del Novecento, dove si è registrata una modifica radicale della società, che ha portato alla produzione di enormi quantità di dati in tutti i campi: basti pensare che ora è normale avere dei dispositivi che ci permettono di compiere diverse azioni tradizionali, ma in modo digitale (es. navigatore, contapassi/smartwatch, etc.), che lasciano alle loro spalle delle tracce, ovvero dei *dati*.
La produzione di dati può essere paragonata ad un vero e proprio tesoro di informazioni che possono essere conservate e utilizzate in futuro per comprendere meglio, e in un'ottica più generale, fenomeni e gli aspetti positivi o negativi (per esempio, si possono prendere i dati delle lezioni online su Teams e altre piattaforme per confrontarli). Fin dagli anni Cinquanta, i dati sono diventati un problema principale all'interno degli ambienti di ricerca, e negli anni Novanta si era già consapevoli che i dati sarebbero stati importanti in futuro, ancor prima della diffusione di Internet e degli smartphone.
Negli anni successivi c'è stata una vera e propria "cascata": la quantità di dati è aumentata sempre di più, le persone stesse sono diventate più con-

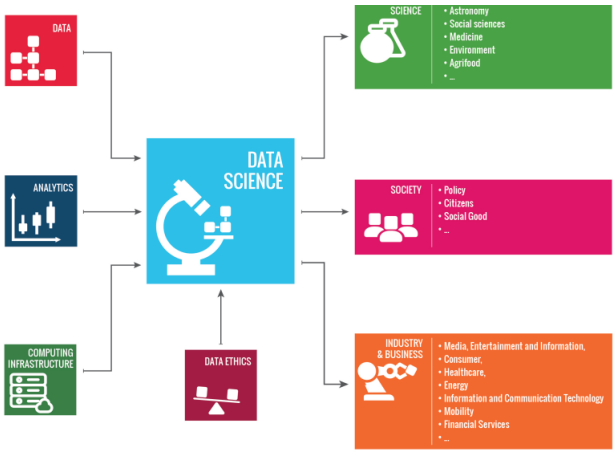


Figure 1.1: Ambiti di pertinenza della data science

sapevoli dei dati generati e della loro importanza in una società digitale per le sfide del domani, come il progresso verso un'economia sostenibile. Risulta quindi necessaria la *data science*, il cui punto focale è la trasformazione di *raw data* in conoscenza utile in diversi domini.
In questi ultimi decenni si è quindi assistito a:

- Aumento della quantità di dati prodotta che possono rappresentare fenomeni nuovi in una maniera più profonda;
- Sviluppo di modelli e algoritmi per analizzare i dati in maniera nuova, rispetto alla statistica ordinaria: sono nati *modelli predittivi* per la classificazione, clustering di gruppi, associazione;
- Sviluppo dell'infrastruttura dei computer, sia hardware che software, che ora risulta molto più performante. Questo ha permesso di far girare algoritmi molto complessi o con un tempo di esecuzione o, ancora, dati confusionari.

Il punto d'incontro con l'Intelligenza Artificiale è proprio relativo alla grande quantità di dati, che in quel caso vengono analizzati da algoritmi di Machine Learning.
Esempi di combinazione di ML e Big Data:

- linguaggio: ci sono molti più esempi di dati testuali in digitale, gli algoritmi hanno imparato correttamente a partire dall'esperienza, grazie ai testi inseriti dagli utenti;

- riconoscimento del contenuto di un'immagine: esistono modelli capaci di categorizzare in automatico le immagini partendo da immagini categorizzate.

L'utilizzo di big data ha provocato però anche problemi etici, legati alla privacy delle persone che generano i dati stessi (spostamenti, uso dello smartphone, social media e quindi amici, interessi, informazioni mediche). Si individuano due aspetti:

- In generale, quando ci sono dei grossi progetti di data science, le persone accettano delle policy in cui viene esplicitato cosa si può e cosa non si può fare con quei dati; il data scientist deve sapere i limiti dei GDPR (in Europa) ed evitare delle violazioni.
  Principio del **privacy by design**: qualunque progetto di data science deve tener conto fin dall'inizio delle disposizioni sulla privacy, non solo alla fine.

- Deve essere legale ed etico anche l'uso finale della conoscenza estratta. Un esempio può essere la raccolta di dati medici per prevenire il diabete e aiutare pazienti e SSN.

Chapter 2

Lezione 2: Data understanding

Il processo di Knowledge Discovery Database (KDD), ovvero quel processo di ricerca di conoscenza celata nei dati, si basa su una pipeline, una sequenza di operazioni che prevedono la raccolta dei dati, la loro pulizia, preparazione, per poi passare finalmente all'applicazione di algoritmi di machine learning, e alla successiva lettura critica dei risultati ottenuti (vedi figura 2.1). A volte in questa pipeline può accadere di dover ricompiere passi già effettuati in precedenza: si parla per questo di un processo **interattivo**.
I dati, sempre più in quantità, oltre che di alta qualità sono ormai reperibili da diverse fonti negli ultimi decenni: si possono per esempio trovare nella sanità, nel mondo dell'educazione, nella cyber security. Essi possono essere generati da sistemi di simulazione computazionale, oppure, per esempio, da sensori, quindi coprono una sfera molto larga di campi. Possono essere utili anche in **campi differenti** rispetto a quello di origine: i dati provenienti dai GPS possono essere utili, in retrospettiva, per essere aggregati ad altri dati

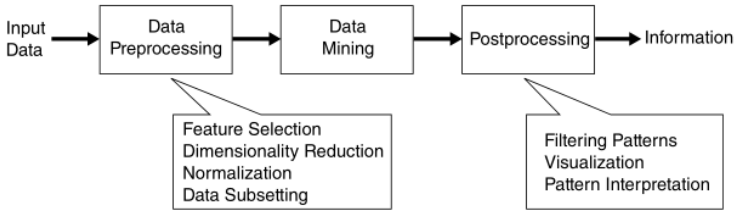


Figure 2.1: KDDprocess

sugli spostamenti dei cittadini per individuare le zone di traffico più intenso, creare sistemi per evitarlo etc. Questo ha sollevato problematiche relative alla privacy, in quanto a volte i dati invadono la nostra sfera privata.
Da un punto di vista commerciale, le grandi compagnie raccolgono moli enormi di dati e questi sono importantissimi per la relazione tra azienda e clienti, per esempio per offrire un servizio "custom" ad ogni singolo cliente.
Da un punto di vista scientifico, invece, i dati possono essere raccolti da satelliti, telescopi, simulazioni scientifiche e le tecniche di data mining possono aiutare ad analizzare in maniera automatica dataset molto grandi e formulare delle ipotesi a partire dai risultati ottenuti.

2.1 Data mining tasks

Ci sono principalmente due tipi di metodi per il data mining:

- **Metodi predittivi**: fanno uso di variabili per comprendere valori sconosciuti di altre variabili o di valori futuri;
- **Metodi descrittivi**: permette di comprendere dei fenomeni rappresentati per mezzo dei dati.

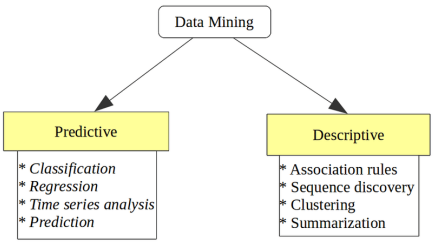


Figure 2.2: Principali tecniche di data mining

2.1.1 Modelli predittivi: classificazione

Si parte da un insieme di dati in input, dove abbiamo una serie di features accompagnate da un valore. Questi dati costituiscono il nostro train set sul quale il modello verrà addestrato. Successivamente, tale modello dovrà predire la classe di un insieme di dati mai visti prima, il test set (vedi figura 2.3). Si può fare l'esempio di una banca, che può avere classificato i clienti relativamente al loro comportamento verso la banca stessa. Il modello addestrato utilizzerà i dati in input per capire quali siano le caratteristiche

associate ai "buoni clienti" e quali siano quelle dei "cattivi clienti"; in tale modo, si potrà predire quanto saranno affidabili i clienti futuri per mutui/prestiti.
I classificatori possono essere usati anche in campo medico, per esempio per predire se le cellule tumorali siano maligne o benigne, classificare la struttura secondaria delle proteine o ancora per la classificazione delle transazioni bancarie o dei testi.

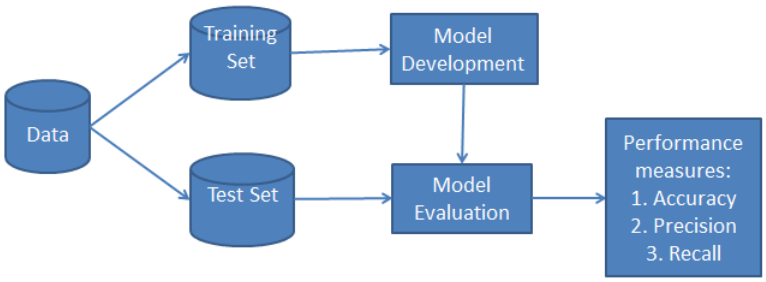


Figure 2.3: Funzionamento di un algoritmo di classificazione

2.1.2 Modelli predittivi: clustering

Nel nostro dataset di partenza non abbiamo delle classi, ma vogliamo individuare degli oggetti che sono simili tra loro e distinguerli da altri oggetti che non sono correlati e appartengono ad altri gruppi (fig. 2.4). Il clustering può essere utilizzato sia per la **comprensione**, quindi per la profilazione dei clienti, per usare strategie di marketing differenti per diverse tipologie di clienti o ancora per "raggruppare" documenti simili osservando la frequenza delle parole, ma anche per individuare geni e proteine che hanno funzioni simili, che per la *summarization* di grandi quantità di dati per ridurre le dimensioni dei dataset.

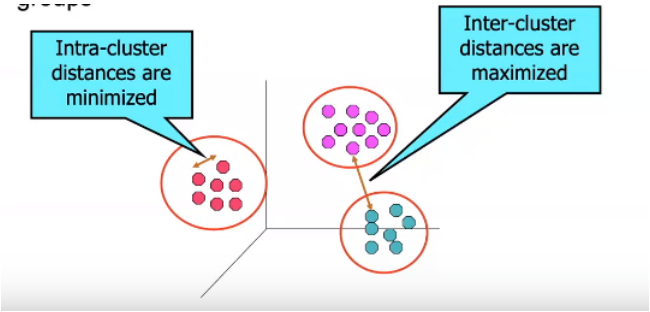


Figure 2.4: Clustering

2.1.3 Modelli predittivi: Association Rule Discovery

Si cerca di individuare delle associazioni interessanti in un dataset tramite la ricerca di pattern che ricorrono diverse volte all'interno dei dati. Questi pattern di associazione permettono di predire l'occorrenza di un item sulla base della presenza di altri item. Anche in questo caso vi sono delle applicazioni legate al marketing, per la promozione di prodotti, la gestione degli scaffali e dell'inventario. Altri interessanti ambiti di applicazione sono quelli delle telecomunicazioni e della medicina, dove vengono usati per analizzare la compresenza di sintomi legati a certe malattie.

2.2 I dati

Un dataset può essere visto spesso come collezione di *data objects*, dove ogni oggetto viene descritto per mezzo di features, che catturano e descrivono alcune sue caratteristiche.

	attribute_1	...	attribute_m
record_1			
⋮			
record_n			

Figure 2.5: Esempio di *dataset*

Osservando la figura 2.5), si possono osservare gli oggetti (le righe), accompagnati in ogni colonna dai loro attributi. Oltre a questi tipi di dataset tabellari, ne esistono anche altri più complessi, come quelli a grafo, utilizzati

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Figure 2.6: Document data

per la rappresentazione di molecole o per la struttura delle pagine web, o sequenze ordinate, usate per la rappresentazione del genoma.
Un altro esempio sono quei dataset che rappresentano dei documenti sotto forma di vettori (vedi fig. 2.6) dove ogni oggetto viene descritto dal numero di occorrenza delle parole (che costituiscono le colonne della tabella
Un particolare tipo di dataset è quello delle **matrici**, dove gli oggetti hanno un numero fisso di attributi, quindi i dati vengono interpretati come punti in uno spazio multidimensionale, dove ogni dimensione rappresenta un attributo distinto.

La qualità dei dati è fondamentale, in quanto questo affligge direttamente i task di data mining.Tra i principali problemi si ricordano:

- Rumore o presenza di outliers
- Valori mancanti
- Valori duplicati
- Dati errati

Chapter 3

Lezione 3: Data Understanding

Spesso il dataset viene fornito come semplice tabella, dove le righe prendono il nome di **istanze**, **record** o **data object**, mentre le colonne vengono chiamate ***features***, **variabili** o **attributi**. Gli attributi, di tipo numerico o categorico, possono essere a loro volta di diversa tipologia

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Figure 3.1: Differenti tipi di attributi

- **Categorici nominali**: piccolo insieme di valori qualitativi, senza un ordinamento intrinseco per le sue categorie. Di solito ci si chiede se due valori siano gli stessi o se sono differenti (es.maschio, femmina o

blu, verde, castano)

- **Categorici ordinali**: attributi che possono essere ordinati in una scala.
- **Numerici discreti**: numeri interi, reali.
- **Numerici continui**: numeri continui all'interno dei numeri reali

Occorre sempre controllare la qualità dei dati, in quanto una bassa qualità potrebbe compromettere i risultati dell'analisi. Per quanto riguarda l'accuratezza dei dati, quindi la vicinanza del valore nei dati rispetto al valore reale, potrebbe essere bassa per motivazioni differenti a seconda del tipo di attributi.
Se l'attributo è numerale, allora una bassa accuratezza potrebbe essere ricondotta al rumore, ad una precisione limitata, a misurazioni errate o ad una erronea digitazione delle cifre, quando i dati vengono inseriti manualmente. Per quanto riguarda gli attributi categorici, gli errori sono solitamente legati a typos. Alcuni problemi:

- **Accuratezza sintattica**: es.fmale in *Gender*;
- **Accuratezza semantica**: l'entry è nel dominio ma non è corretta. Es. *John Smith* con attributo *Gender* female;
- **Completezza**: mancano record completi
- **Dati non bilanciati**: errore nel modo in cui i dati sono messi assieme. es. modello per distinguere lupi e cani. Per i lupi metti sempre il background con la neve, quindi il modello riconoscerà lo sfondo e non le caratteristiche dell'animale.
- **Timeliness**: i dati sono aggiornati?

3.1 Data Visualization

3.1.1 Bar chart e istogrammi

Per le variabili individuali, dobbiamo osservarne la distribuzione: quando l'attributo è **categorico**, la maniera più semplice per visualizzarne la frequenza è un **bar chart**. Questo tipo di grafico anche essere usato per visualizzare i valori pari a 0.
L'istogramma si differenzia perché viene applicato ad attributi numerici.

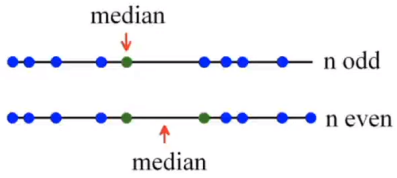


Figure 3.2: Rappresentazione mediana

Ciascuna barra non conta il numero di valori dei dati con esattamente lo stesso valore come nel grafico a barre, ma il numero di data object che ricadono in un certo intervallo. Il range degli attributi numerici viene discretizzato in un numero preciso di intervalli, o **bins**, solitamente di eguale misura. Per ogni intervallo, viene calcolata la frequenza dei valori che ricadono al suo interno, che viene poi indicata dall'altezza delle barre. Un'altra differenza è che l'ordinamento nell'asse delle X è **naturale** (dal valore più piccolo al più grande); nel bar chart, l'ordinamento è **arbitrario**.
È molto importante il numero di bin quando si fa il grafico e uno dei metodi per regolare il numero di bin è la legge di Sturges:

$$k = log_2(n) + 1 \tag{3.1}$$

dove n è il numero di item nel dataset. Questo è tipicamente una buona scelta se le distribuzioni sono normali e il dataset è di dimensioni medie.

3.1.2 Mediani, Quantili, Quartili, Interquartili

- **Mediana**: valore al centro, quando si parla di valori incrementali;
- **q%-quantile (0 ¡ q ¡ 100)**: il valore per il quale il q% dei valori sono più piccoli e il 100-q% sono più larghi.
- **Quartile**: primo quartile = 25%-quantile, mediana = secondo quantile, 75%-quantile = terzo quartile.
- **IQR** o **Interquartile range**: terzo quantile - primo quantile.

I boxplot, ideati da J. W. Tukey, sono direttamente connessi a quantili e quartili e sono un modo rapido per rappresentarli.
La linea "solida" all'interno di un boxplot corrisponde alla **mediana**, la scatola è il limite tra il secondo e il terzo quantile.

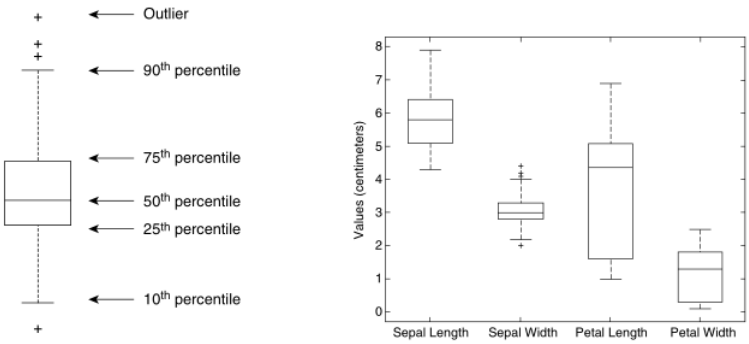
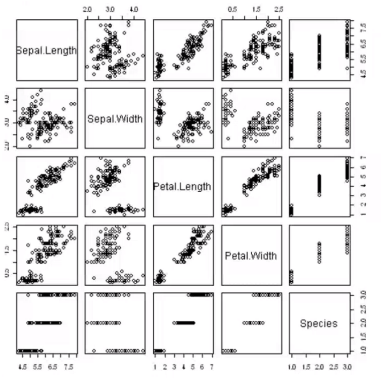


Figure 3.3: Boxplot, dal libro *Introduction to Data Mining*

3.1.3 Scatter plot

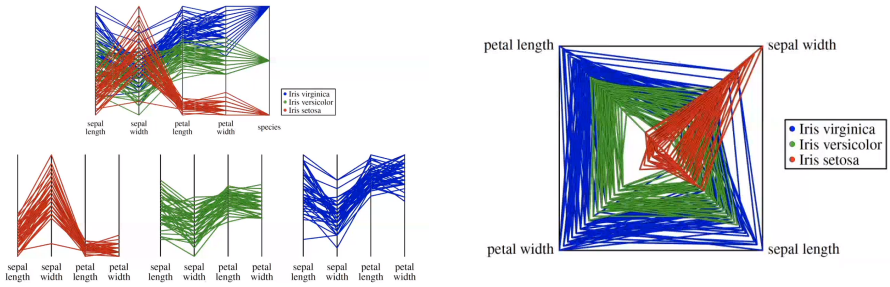
Gli scatter plot, o grafici a dispersione, permettono di visualizzare due variabili in un grafico bidimensionale. Nell'*iris dataset*, dove abbiamo 4 attributi numerici e uno categorico, possiamo plottare cinque grafici per ogni attributo.



Gli scatter plot possono essere arricchiti con ulteriori informazioni, attraverso l'uso di colori e differenti simboli.
Una cosa utile da fare, è l'utilizzo dello **jitter**, per diversificare i punti differenti per visualizzare tutti i dati, anche quelli che si potrebbero sovrapporre, facendoci perdere quindi delle informazioni dai cluster.
Gli scatter plot sono anche molto utili per visualizzare gli outlier.
Si può anche usare una versione 3D degli scatter plot se il dataset non è molto grande, ma è difficile suddividere bene i diversi cluster.

3.1.4 Parallel coordinates e Radar plot

Sono due modi simili di visualizzare i dati. Nel caso dell'Iris Datates, abbiamo tutte le nostre variabili in delle colonne, dove ognuna rappresenta l'intera barra che corrisponde all'intervallo dei dati per quella variabile. Ogni



(a) (b)

Figure 3.4: Parallel coordinates e Radar plot

fiore è una linea che può essere o meno colorata. Se si hanno connessioni molto inteconnesse tra loro, dove ogni variabile dipende dall'altra, si possono creare dei grafici "cubici", molto uniformi, che non ci aiutano più di tanto.
I radar plot sono basati sulla stessa ide delle coordinate parallele, ma al posto di avere colonne parallele, queste partono dalla stessa origine e vi ritornano, quindi ogni fiore è rappresentato per mezzo di una linea che termina nella stessa origine da cui è partita.
Una variante dei radar plot è lo star plot, dove però ogni data object viene disegnato separatamente.

3.2 Correlation Analysis

Oltre a farla con dei grafici, si possono usare anche diverse formule statistiche per definire la correlazione tra attributi.
Uno di questi è il coefficiente di correlazione di Pearson, che viene definito come:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)s_x s_y} \tag{3.2}$$

dove:

- $\overline{x}$ e $\overline{y}$, sono i valori medi degli attributi x e y
- $\sum_{i=1}^{n}(x_i - \overline{x})$ è la covarianza
- s_xs_y indicano la deviazione standard di x e di y

Tale indice di correlazione lineare è compreso sempre tra -1 e 1 e più è alto il valore assoluto del coefficiente di correlazione, più è forte la relazione lineare tra gli attributi. Quando il coefficiente = 1, allora c'è una correlazione

perfetta tra gli elementi; quando il valore è negativo, invece, c'è una perfetta correlazione lineare negativa. Se è 0, allora le variabili non sono legate tra loro.
Un'alternativa a Pearson, è il coefficiente di Spearman, che permette di definire una correlazione di qualunque tipo (es. quadratica), senza guardare esclusivamente alla correlazione lineare. Si osserva il **rank**: se la posizione nel rank è simile, c'è una forte relazione tra le posizioni, senza guardare gli attributi[1] .

$$\rho = 1 - 6\frac{\sum_{i=1}^{n}(r(x_i) - r(y_i))^2}{n(n^2-1)} \tag{3.3}$$

- $r(x_i)$ e $r(y_i)$ sono il rank i-esimo degli attributi+ x e y
- n è il numero complessivo di osservazioni

3.3 Checklist per la data understanding

- Determinare la qualità dei dati;
- Trovare gli outlier (con tecniche di visualizzazione);
- Trovare e esaminare i valori mancanti;
- Trovare nuove dipendenze o correlazioni tra gli attributi;
- Verificare ipotesi dipendenti dalla specifica applicazione;
- Confrontare il comportamento statistico con quello che vediamo effettivamente dai dati.

[1] permette di stabilire quanto bene una relazione tra due variabili può essere descritta usando una funzione monotona. - Da Wikipedia

Chapter 4

Lezione 4: Data preparation

Se la *data understanding*, l'analisi esplorativa dei dati, ci permette di ricavare informazioni interessanti sui dati, quali l'assenza di valori, la presenza di outlier o ancora, informazioni statistiche come la correlazione tra attributi, la fase di preparazione dei dati invece, parte dalle problematiche che affliggono i dati per risolverle e aumentare la loro qualità generale o renderli più funzionali ai successivi passi dell'analisi, senza, ovviamente, intaccare la conoscenza in essi contenuta.
La *data preparation* si occupa di:

- Selezionare gli attributi;
- Ridurre la dimensione del dataset;
- Selezionare solo una parte di record per esempio per ragioni legate alla qualità di dati;
- Trattare valori mancanti e outliers;
- Integrare, unire e trasformare i dati per facilitarne l'analisi;
- Migliorare la qualità dei dati;

Esempio 1 - Estrazione di feature: costruire delle nuove feature da quelle presenti

Si vogliono trovare i migliori lavoratori in una compagnia, partendo da alcuni attributi, ovvero il *numero di lavori conclusi ogni mese*, *il numero di ore di lavoro*, *il numero di ore necessarie a concludere ogni task*.

Questi attributi contengono delle informazioni riguardanti l'efficienza dei lavoratori, ma ancora non esiste una feature che espliciti tale parametro. È necessario quindi calcolare l'efficienza come:

$$\text{Efficienza} = \frac{\text{ore } impiegate\ per\ concludere\ i\ task}{\text{ore } normalmente\ impiegate\ per\ concludere\ i\ task} \tag{4.1}$$

È molto comune che le feature che abbiamo originariamente nei dati di partenza vengano combinate attraverso operazioni matematiche (feature engineering) per ottenere così delle feature più raffinate. Questo è un task piuttosto complesso, ma esistono sistemi, come apposite reti neurali, capaci analizzare in maniera automatica diverse combinazioni di dati.

4.1 Data reduction

Ci si deve interrogare su come ridurre la quantità di dati, sia per concentrarsi meglio su un sottoinsieme di questi su cui siamo più sicuri riguardo la quantità e la completezza, sia per oltrepassare problemi di complessità computazionale. Gli algoritmi di Data Mining possono scalare linearmente gli attributi rispetto alla dimensione dei dati, senza abbassare le proprie performance.
Per ridurre il numero di dati si può diminuire il numero di righe, per esempio campionando i dati o usando tecniche di clustering, mentre per ridurre il numero delle feature nelle colonne è possibile effettuare la *feature selection*, rimuovendo le informazioni ridondanti; alcuni metodi possono anche trasformare i dati per ottenere un nuovo insieme di attributi, più piccolo, che è più funzionale per risolvere il problema che dobbiamo affrontare.

4.1.1 Sample

Il campionamento ci permette di velocizzare il tempo di esecuzione degli algoritmi di data mining, usando un algoritmo che mantiene solo parte dei dati originari, considerati rappresentativi dell'insieme. È un'operazione delicata, perché all'inizio non sappiamo quali dati siano effettivamente rappresentativi; è possibile usare delle procedure che campionano randomicamente per selezionare un insieme arbitrario di righe, ma rimane comunque un'operazione rischiosa, che può portare alla perdita di dati importanti riguardanti, per esempio, quei dati che hanno una bassa frequenza e che non vengono campionati. Proprio per questi motivi si cerca di effettuare un campionamento stratificato (*stratified sampling*), che cerca di mantenere

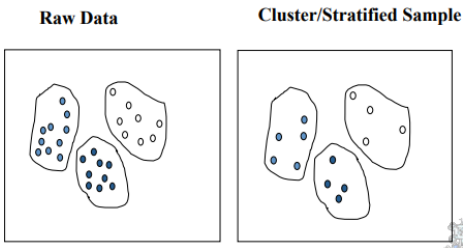


Figure 4.1: Stratified sampling

la percentuale di elementi rappresentativi di ciascuna classe. Il problema è che all'inizio, non si sa nulla sui differenti "cluster" all'interno del dataset, quindi non possiamo effettuare un campionamento in base alla classe; per ottenere questo però, ci sono numerosi algoritmi di clustering efficienti da un punto di vista computazionale, che possono essere applicati ad un dataset di larghe dimensioni. Non va troppo d'accordo con il Machine Learning e il Data Mining, questi funzionano meglio con grandi quantità di dati.

4.1.2 Riduzione della dimensionalità

Viene selezionato un sottoinsieme di attributi quanto più piccolo possibile ma comunque abbastanza grande per effettuare le nostre analisi: se si hanno feature molto correlate tra di loro, se ne può rimuovere una perché portano un'informazione molto simile sullo stesso fenomeno.
Perché si dovrebbero ridurre le variabili?
In generale, sebbene sia meglio avere un maggiore numero di informazioni, si ha un aumento nella difficoltà di risoluzione di un problema all'aumentare del numero delle colonne di un dataset, anche senza aumentare il numero delle righe. I dati diventano sempre più sparsi, sempre più difficili da analizzare per gli algoritmi, per cui è sempre meglio eliminare feature ridondanti o irrilevanti, andando magari a crearne delle nuove.
Per comprendere quali tra queste siano più irrilevanti, si fa uso di misure di qualità che indichino quanto bene un sottoinsieme di feature performino rispetto al task di data mining che abbiamo in mente. Questo non è solitamente un problema del tutto non supervisionato, ma è legato al problema specifico che dobbiamo affrontare: questo si ricollega al fatto che sia importante creare delle nuove feature che racchiudano in sé diversi valori ridondanti presenti nel dataset originale. Per rimuovere la ridondanza si può in alternativa usare una misura di correlazione, così da trovare variabili fortemente correlate ed eliminarne una.

Di solito non si effettua su dataset di piccole dimensioni, ma con quelli più grandi; per fare ciò si fa uso di algoritmi automatici che selezionano le feature "migliori" dal punto di vista statistico o qualitativo (*feature selection*):

- Selezionare le feature con il ranking più alto: si assegna un peso a ciascuna feature quando vengono valutate. È un approccio brutale, perché quello che vogliamo fare è valutare quali siano le feature migliori tra il totale delle caratteristiche, non le migliori "in sé".
- Selezionare il sottoinsieme di feature con il ranking più alto: questo prevede che vengano effettuate diverse ricerche per ottenerlo, quindi anche solo per 20 feature ci sono già più di un milione di possibili subset.
- Forward selection: si parte da un insieme vuoto, cui si aggiungono le feature una alla volta. Ad ogni passo si seleziona la feature che dà il miglioramento più elevato (greedy - euristic - search).
- Backward elimination: funziona in maniera opposta rispetto alla forward selection. Si parte infatti con l'intero insieme di feature e si eliminano una alla volta quelle feature che diminuiscono di meno le prestazioni.

Gli ultimi due metodi euristici ci permettono di risolvere un problema molto complesso dal punto di vista computazionale, quello della ricerca completa; quindi si usano delle soluzioni *greedy*, dove con il termine "greedy" si intende che non vanno a controllare tutte le possibili combinazioni ma solo un piccolo numero di esse che sono selezionate andando "avidamente" verso la migliore performance, come se a scacchi si facesse solo ed esclusivamente la mossa che ti dà un maggiore vantaggio nell'immediato.

4.2 Data Cleaning

Il *data cleaning* è quell'operazione che riguarda la gestione di valori anomali, outliers e la trasformazione dei dati.

4.2.1 Valori anomali

Sono quei valori mancanti, come un "NULL" o un "?" (es. una persona che non ha risposto ad una domanda del questionari), oppure dati che non sono validi, come quelli che lascia dietro di sé un sensore rotto quando registra la

temperatura. Altri ancora possono non avere un significato.
Per gestire questa tipologia di dati ci sono tre modi:

- Se è una singola colonna, dove ci sono molti record mancanti o non validi, allora una buona soluzione è eliminare direttamente tale colonna;
- Se siamo nella situazione opposta, in cui in certe colonne ci sono tutti i valori ben specificati, tranne alcuni nelle stesse righe, si potrebbero eliminare le righe pensando che quelli non siano record affidabili;
- Si potrebbe avere il caso intermedio in cui ci sono solo alcuni valori mancanti e la parte restante contiene dati validi, non si può cancellare la riga perché si perderebbero dati importanti; qui si decide per la sostituzione e anche in questo caso si possono attuare diverse strategie.
  - Riempire i record con la media, mediana o moda;
  - Stimare i valori mancanti usando la distribuzione di probabilità dei valori presenti;
  - Predire i valori usando modelli di classificazione o regressione a partire dai dati contenuti nella colonna

4.3 Data transformation

Perché trasformare i dati?
Si intende l'applicazione di una funzione di trasformazione Y che mappi le colonne in differenti colonne con delle migliori proprietà da un punto di vista statistico, senza distorcere i valori originali contenuti nel dataset. Certe trasformazioni dei dati possono risolvere problemi legati alla loro distribuzione sparsa o quando sono caratterizzati da numerosi picchi.

$$Y = T(X) \tag{4.2}$$

Y deve preservare le informazioni di X e allo stesso tempo deve eliminare almeno uno dei problemi di X e Y deve essere più utile di X.
Gli obiettivi principali sono quelli di stabilizzare la varianza, normalizzare le distribuzioni, realizzare una relazione lineare tra variabili. Secondariamente, deve semplificare l'elaborazioni di dati e rappresentarli in una scala considerata più adatta, in quanto molti metodi statistici lavorano meglio quando i dati hanno una distribuzione lineare e una correlazione normalizzata.

Normalizzazione

Si usa affinché i valori assoluti nelle singole variabili non influenzino la nostra analisi. Si vuole evitare che certe variabili, i cui valori sono in intervalli più grandi rispetto agli altri, rischino di essere considerate come più importanti proprio perché con un valore assoluto più elevato.

- Min-max normalization: $v' = \frac{v - min_A}{max_A - min_A}$
  Le variabili sono scalate per rientrare in un range compreso tra il valore massimo e il valore minimo.

- Z-score normalization: $v' = \frac{v - mean_A}{stand\_dev_A}$
  In questo caso si sottrae il valore reale dalla media della popolazione e si divide il tutto per la deviazione standard. Si ottiene un numero compreso tra +1 e -1.

- Normalization by decimal scale: $v' = \frac{v}{10^j}$
  i dati vengono scalati nell'appropriata scala decimale.

Qualunque normalizzazione si scelga, occorre stare attenti riguardo alla distribuzione dei dati, poiché se vi è "scarsità" di dati (*data sparsity*), quindi vi sono pochi dati con valori molto elevati che possono apparire alla stregua di outliers, ma non lo sono realmente, in quanto rappresentativi di un fenomeno. Si può fare l'esempio della distribuzione della ricchezza nel mondo: ci sono pochissime persone che possiedono enormi ricchezze, mentre il resto della popolazione possiede il minimo indispensabile. Se dovessimo analizzare un dataset con tali informazioni, si deve fare estrema attenzione a scegliere una trasformazione dei dati che preservi tali informazioni, come l'utilizzo della *scala logaritmica*(log_{10}), con cui si passa dal valore assoluto all'ordine di grandezza, quindi l'esponente di 10 che mi serve per ottenere quel numero. I dati trasformati sono ancora rappresentativi, ma molto più facili da sottoporre a future analisi.

4.3.1 Discretizzazione

Le variabili, rimanendo nel generico, possono essere di due tipologie: numeriche o categoriche. Il problema è che, in certi casi, potremo voler passare da variabili categoriche a variabili discrete che rappresentano essenzialmente la stessa informazione, per diversi motivi, di cui uno è che certi metodi lavorano meglio con le variabili categoriche. La discretizzazione si può quindi definire come il trovare una "mappatura" per trasformare i valori continui in valori discreti.

Figure 4.2: Discretizzazione di diversi attributi (altezza, peso, guadagni)

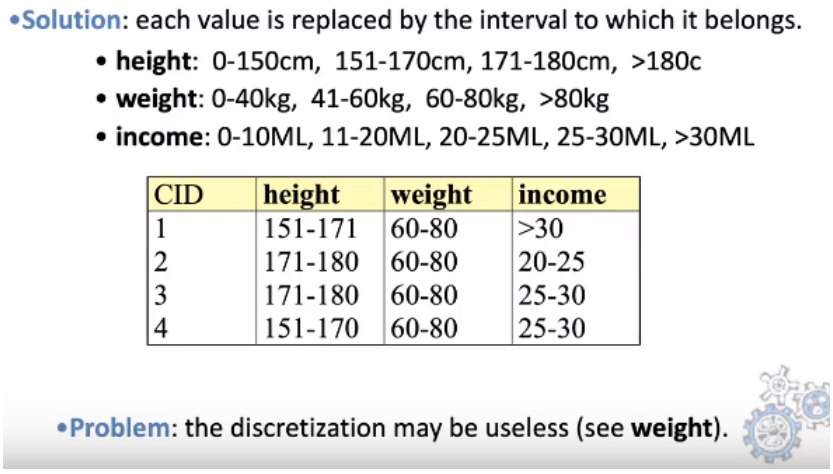
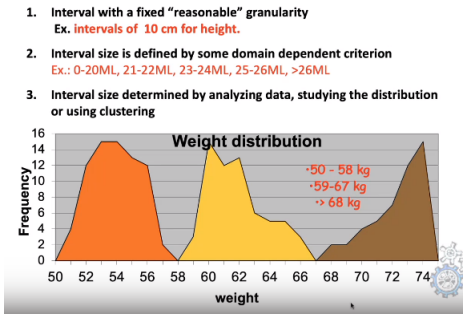
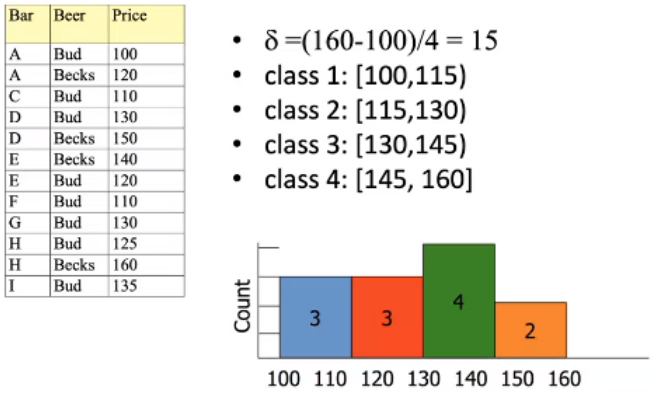


Figure 4.3: Scelta degli intervalli



Un altro motivo è che i dati originali sono sparsi, quindi una discretizzazione può renderli più semplici da interpretare e sottoporre ad indagini più profonde.
In generale, si partiziona in intervalli il dominio delle variabili dal minimo al massimo all'interno del dominio delle variabili del dataset, e si assegna ogni data point al nome dell'intervallo nel quale andranno i dati. Gli intervalli sono da considerare come dei cesti (*bin*) in cui andranno inserite le nostre variabili.

Figure 4.4: Esempio di sbilanciamento delle classi dopo l'applicazione del *natural binning*



Bar	Beer	Price
A	Bud	100
A	Becks	120
C	Bud	110
D	Bud	130
D	Becks	150
E	Becks	140
E	Bud	120
F	Bud	110
G	Bud	130
H	Bud	125
H	Becks	160
I	Bud	135

Tecniche di data binning

Come si può vedere in figura 4.3, analizzando la distribuzione dei dati o utilizzando algoritmi di clustering, si potrebbe avere una situazione analoga, in cui emergono naturalmente degli intervalli. Negli altri casi, invece, si può effettuare il binning:

- Natural binning: intervalli con la stessa ampiezza
- Equal frequency binning: non hanno la stessa ampiezza, è uguale la proporzione di dati che ricadono nello stesso *bin*
- Statistical binning: usa informazioni statistiche, come la media, i quartili etc.

Natural binning: ha il vantaggio di essere semplice e permette di dividere i valori in k parti di uguale misura. Lo svantaggio, invece, è quello di creare delle classi sbilanciate (vedi fig. 4.4), perché ogni bin ha la stessa ampiezza ma non la stessa popolazione.

$$\delta = \frac{x_{max} - x_{min}}{k} \tag{4.3}$$

L'elemento x_j appartiene alla classi$i$ se $x_j \in [x_{min} + i\delta, x_{min} + (i+1)\delta)$
Uno dei problemi del natural binning è però che genera delle distribuzioni dei dati sbilanciate (fig. 4.5). **Equal Frequency Binning**: ordina e conta

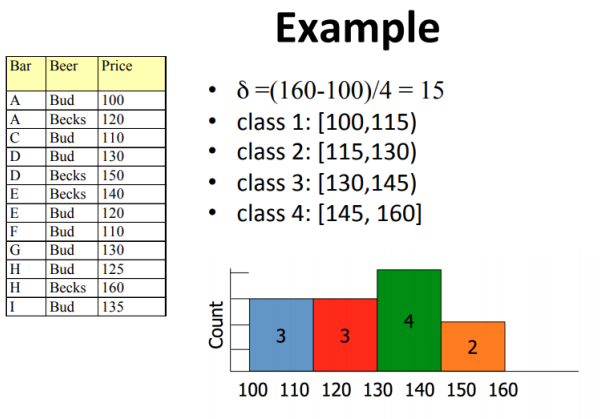


Figure 4.5: Esempio di un caso problematico di natural binning

gli elementi e definisce k invervalli di f, dove

$$f = \frac{N}{k} \tag{4.4}$$

dove N è il numero di elementi nel campione.
L'elemento $_i$ appartiene alla classe j se $j * f \leq i < (j+1) * f$. Anche in questo caso, c'è un lato negativo, ovvero che non sempre è adatto per evidenziare delle correlazioni interessanti (vedi fig. 4.6).

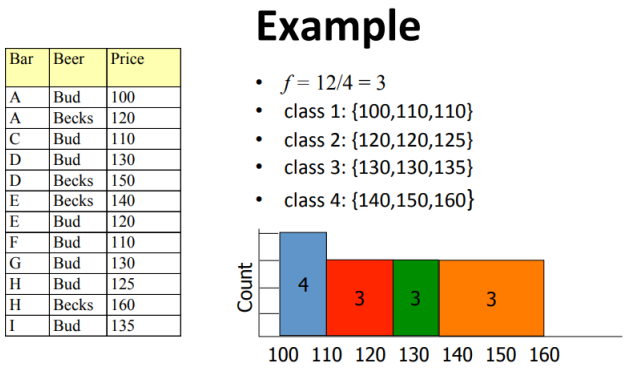


Figure 4.6: Caption

Anche la scelta del numero corretto di classi è fondamentale, in quanto se ne vengono lasciate troppe poche, allora si verifica una perdita di informazione, mentre se sono troppe si ha una dispersione dei valori che non permette di mettere in evidenza la loro distribuzione. Il numero ottimale di classi può essere deciso per mezzo di formule matematiche.
Un caso è quello di (*Sturges, 1929*), per cui il numero ottimale di classi è funzione degli N elementi:

$$C = 1 + \frac{10}{3} log_1 0(N) \tag{4.5}$$

e la ampiezza delle classi, che dipende dalla varianza e dal numero dei dati (*Scott, 1979*):

$$h = \frac{3,5 * s}{\sqrt{N}} \tag{4.6}$$

Discretizzazione supervisionata
Essa ha le caratteristiche di avere un "obiettivo" quantificabile e un numero di classi noto. Tra le tecniche utilizzate per la realizzazione della discretizzazione supervisionata si ricordano:

- *ChiMerge*;
- discretizzazione basata sull'*entropia*
- discretizzazione basata su percentili

ChiMerge: è un processo *bottom-up*, in cui inizialmente ogni valore corrisponde ad un intervallo. Successivamente gli intervalli adiacenti vengono fusi tra loro se sono simili e questa somiglianza viene misurata sulla base del *target attrubute*, che misura quanto due intervalli sono differenti tra loro.

Entropy based approach: si cerca di minimizzare l'entropia (4.7

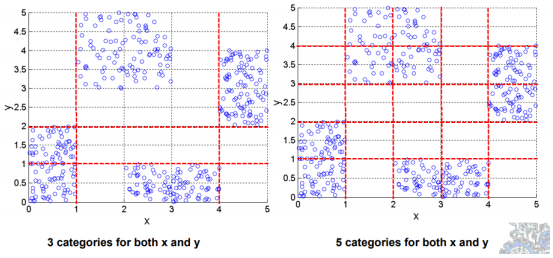


Figure 4.7: Entropy based approach per la discretizzazione

4.3.2 Similarity

Con il termine *similarità*, in data mining si intende la misura numerica di quanto simili due *data object*: più è alta, più gli oggetti sono simili. Solitamente, tale valore è incluso nell'intervallo [0,1].
Il termine *dissimilarità* si intende invece quella misura numerica che indica quanto due *data object* differiscano tra di loro, che assume valori bassi quando gli oggetti sono simili. La dissimilarità minima è spesso pari a 0, mentre il limite massimo tende invece a variare.
Il termine *prossimità* fa riferimento alla similarità o dissimilarità.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d-min\_d}{max\_d-min\_d}$

Figure 4.8: Similarità e dissimilarità per un attributo

Quando gli attributi sono più di uno, allora si usano diverse misure.
Distanza Euclidea

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \tag{4.7}$$

dove n è il numero di dimensioni o attributi e p_k e q_k sono il valore del *k-esimo* attributo o *data object* p e q.
Quando si usa la distanza euclidea è necessaria la standardizzazione se differiscono le scale.

Minkowski Distance
È la generalizzazione della distanza euclidea:

$$dist = \sum_{k=1}^{n} |p_k - q_k|^r)^{\frac{1}{r}} \tag{4.8}$$

dove r è un parametro, n è il numero di dimensioni o attributi, p_k e q_k sono, rispettivamente, il *k-esimo* attributo o data object p e q.
Esempi di distanza di Minkowski:

- $r = 1 \rightarrow$ ***City block*** distance (Manhattan, taxicab, L1 norm). Tra queste è importante la *distanza di Hamming*, che indica il numero di bit per i quali sono differenti due oggetti che hanno soli attributi binari;
- $r = 2 \rightarrow$ **Distanza Euclidea**;
- $r \rightarrow$ Supremum distance $\rightarrow \infty$(L_{max}norm, L_∞norm). Queste sono anche le differenze massime tra i componenti dei vettori.

N.b. Le distanze sono definite per **tutte le dimensioni**.

Curse of dimensionality
All'aumentare delle dimensionalità, i dati diventano sempre più sparsi nello spazio, quindi questo comporta delle problematiche nel definire la densità dei punti, che causa problemi soprattutto nel clustering e nell'outlier detection.

Proprietà delle distanze

- *d(p, q)* ≥ 0 per tutte le p e q e *d(p, q)* $= 0$ solo se p = q (positive definiteness);
- *d(p, q)* = *d(p, q* per tutte le p e q (simmetria);
- *d(p, r)* $\leq$ *d(p, q)* + *d(q, r)* per tutti i punti p, q e r (triangle inequality).

d(p, q) è la distanza o dissimilarità tra i punti p e q.
Se una distanza rispetta queste proprietà prende il nome di **metrica**.

Proprietà delle similarità Anche le similarità possiedono delle proprietà ben note:

- *s(p,q) = 1* (o max. similarità) solo se $p = q$;
- *s(p, q) = s(q, p)* per tutte le p e q.

dove *s(p,q)* è la similarità tra punti p e q

Similarità tra vettori binari Una situazione comune è che gli oggetti p e q abbiano solo attributi binari. In questi casi si calcolano le similarità come "quantità"

- M01 = il numero di attributi dove p era 0 e q era 1;

Categorical	insufficient	sufficient	good	very good	excellent
p1	0	0	1	0	0
p2	0	0	1	0	0
p3	1	0	0	0	0
p4	0	1	0	0	0

item	bread	butter	milk	apple	tooth-past
p1	1	1	0	1	0
p2	0	0	1	1	1
p3	1	1	1	0	0
p4	1	0	1	1	0

Figure 4.9: Esempio di dati binari

- M10 = il numero di attributi dove p era 1 e q era 0;
- M00 = il numero di attributi dove p era 0 e q era 0;
- M11 = il numero di attributi dove p era 1 e q era 1.

A questo punto si applica il *Simple Matching* o il *Jaccard Coefficients*: *SMC* = il numero di match / il numero di attributi = $(M_{11} + M£_{00})$ / $(M_{01} + M_{10} + M_{11} + M_{00})$
J = numero di 11 match / numero di attributi con valori diversi da 00 = (M_{11})/ $(M_{01} + M_{10} + M_{11}$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

M_{01} = 2 (the number of attributes where p was 0 and q was 1)
M_{10} = 1 (the number of attributes where p was 1 and q was 0)
M_{00} = 7 (the number of attributes where p was 0 and q was 0)
M_{11} = 0 (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00})$ = (0+7) / (2+1+0+7) = 0.7

J = (M_{11}) / $(M_{01} + M_{10} + M_{11})$ = 0 / (2 + 1 + 0) = 0

Figure 4.10: SMC vs Jaccard

Document data

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Figure 4.11: Esempio di documento rappresentato come vettore

Un caso particolare è quello della rappresentazione dei documenti come vettori (fig. 4.11) In questo caso, come misura di similarità si usa quella della *similarità del coseno*, per cui: se d_1 e d_2 sono due vettori di documenti, allora

$$cos(d1, d2) = (d1 \cdot d2)/||d1||||d2|| \tag{4.9}$$

dove $\cdot$ indica il prodotto scalare dei vettori e ||d|| è la lunghezza del vettore.

Correlazione
La correlazione misura la relazione lineare tra oggetti binari o continui. Per fare ciò è necessario standardizzare gli oggetti ed effettuare il prodotto scalare (covarianza/dev.standard)

$$p'_k = p_k - mean(p)) \tag{4.10}$$

$$q'_k = k_k - mean(q)) \tag{4.11}$$

$$correlation(p, q) = (p' \cdot q')/(n-1)std(p)std(q) \tag{4.12}$$