

# Pianificazione Automatica e sistemi di Supporto delle Decisioni

Matteo Aprile

Professore: Giampaolo Ghiani, Enamuele Manni

## INDICE

<b>I</b>	<b>Definizioni - 22/23.09.22</b>	<b>0</b>
I-A	Business Analytics . . . . .	0
I-B	Decisioni . . . . .	0
I-C	Business Intelligence (BI) . . . . .	0
I-D	Data visualization . . . . .	0
I-E	Decision Support Systems (DSS) . . . . .	0
I-F	Operations Research (OR) . . . . .	0
I-G	Agents . . . . .	0
I-H	Artificial Intelligence (AI) . . . . .	0
I-I	Machine Learning (ML) . . . . .	0
I-J	Deep Learning . . . . .	0
I-K	Data Mining (DM) . . . . .	0

## II Software solutions and languages for AP and DSS - 29.09.22

### Riferimenti bibliografici

#### I. DEFINIZIONI - 22/23.09.22

Ci occuperemo di 2 tipi di scenari:

- usare **algoritmi a supporto delle decisioni**
- usare algoritmi che **sostituiscono completamente l'uomo**

#### A. Business Analytics

Disciplina che utilizza dati, statistiche, modelli matematici per **aiutare a prendere delle decisioni in base a dei dati**.

Possiamo racchiudere i suoi **passaggi** in:

- 1) **descriptive analytics**: capire **cosa sia successo nel passato** tramite i dati disponibili
- 2) **predictive analytics**: cercare di **fare delle previsioni** in base ai dati già disponibili
- 3) **prescriptive analytics**: **creare un piano di azione** per poter massimizzare il KPI (Key Performance Indicator)



Figura 1. Fasi della business analytics

#### B. Decisioni

Rappresenta la **scelta di un elemento tra soluzioni** dopo aver ponderato le opzioni.

Possiamo avere più casi d'uso:

- **simplest case**: abbiamo **poche alternative** quindi una semplice scelta
- **multiple criteria**: abbiamo **più metri di paragone** delle performance, quindi si dovranno tenere in conto:
  - **soluzioni migliori** di altre (dette di Pareto)
  - **vincoli** dovuti dai clienti o da casi logistici da gestire (es: spedizioni)
  - ottimizzazioni matematiche
  - **conflitti tra i vincoli**
- **incertezze e rischi**:
  - **decisioni operative**: di **breve periodo reversibili e limitate** a "n" persone del team
  - **decisioni tattiche**: **coinvolge una parte dell'organizzazione** per un medio periodo
  - **decisioni strategiche**: di **lungo periodo non reversibili e coinvolgono denaro**
  - **decisioni strutturate**: hanno una **procedura di risoluzione specifica**
  - **decisioni non strutturate**: richiedono **creatività ed esperienza** in un dato settore

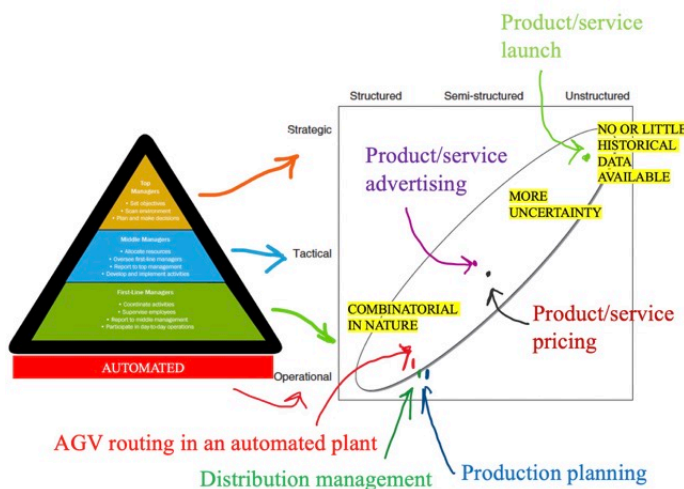


Figura 2. Diagonale decisionale

### C. Business Intelligence (BI)

Usato per indicare un sistema dedicato alla raccolta di dati e alla loro elaborazione al fine di un reporting, infatti per "Intelligence" si intende investigazione. Venivano usati su dati atomici per avere delle conoscenze approfondite in un determinato business.

### D. Data visualization

Consiste nel prendere dati e plottare un grafico, ma in realtà ora si ha una trattazione più metodologica, cioè se visualizzare in modo statico o meno i dati.

### E. Decision Support Systems (DSS)

Si indicava un sistema computerizzato dotato di un sistema di "data management" per creare un modello di ottimizzazione, fornendo un feedback tramite un'interfaccia. Ora indica una varietà di sistemi per visualizzare i dati in larga misura o meno.

### F. Operations Research (OR)

Attività organizzative per portare avanti un sistema logistico. Per "research" si indica la ricerca delle operation per conseguire dei risultati, avremo come sottocategorie:

- **ottimizzazione matematica**
- **queueing theory**: studio matematico delle linee in attesa il limite è che funzionano solo con sistemi semplici e con richieste di servizio in ordine stocastico
- **simulazione**: per usarle è necessario generare dei numeri randomici quindi inconvenienti (bisogna fare un'analisi statistica dei risultati dalle quali si farà una stima)
- **game theory**: decisioni con più players

### G. Agents

È un sistema che si muove in un environment (ambiente), ha dei sensori tramite i quali percepisce alcuni aspetti del mondo che lo circonda quindi si crea una rappresentazione del mondo circostante che può vedere. È capace di influenzare l'ambiente tramite degli attuatori come ruote o braccia (intendiamo anche agenti software).

Possiamo classificarli come:

- **agenti autonomi**: se è concepito in modo tale che tramite un'istruzione sintetica raggiunge un goal sviluppando le azioni per raggiungerlo. In realtà può anche non essere una sequenza di azioni dato che potrebbero esserci degli imprevisti
- **agenti intelligenti**: se
- **impara dall'esperienza**
- crea una rappresentazione dell'ambiente che lo circonda e ci ragiona sopra per un possibile risultato delle proprie azioni
- **si adatta ad un ambiente mutevole**

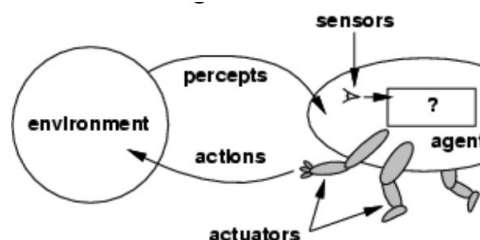


Figura 3. Schematizzazione di un agente e sue caratteristiche

### H. Artificial Intelligence (AI)

Comprende tante sottodiscipline:

- **automated reasoning**: legato alla rappresentazione del mondo e come ragionare su di essa ma anche calcolandone le probabilità
- **automated planning**: usato in ambienti industriali
- **automated learning**
- **natural language processing**: sviluppare agenti software per fare sintesi di testi, scrivere automaticamente articoli, chat bot, ecc
- **perception**: visione artificiale
- **manipulation**: avere un agente che può modificare l'agente circostante

### I. Machine Learning (ML)

Consiste nell'apprendimento automatico e quindi lo sviluppo degli agenti che apprendo tramite la loro esperienza pregressa. Ci sarà allora una fase di training. Una delle possibili architetture che permette di farlo sono le Neural Networks prima avevano solo 2/3 neuroni, ora ne hanno vari strati il che fornisce delle prestazioni impressionanti

### J. Deep Learning

Si basa sull'apprendimento automatico con reti neurali tramite un gran numero di strati di neuroni.

## K. Data Mining (DM)

Usare metodi di Machine Learning per **estrarre manualmente** i pattern dai dati, cioè una **regola o un trend**. È quindi la parte nobile del knowledge discovery in db, dato che i dati sono in genere disponibili su db o da altre piattaforme.

La **sequenza** nella quale interviene è:

- 1) **prendere** i dati
- 2) trovare i vari **target**
- 3) **preprocessare** i dati
- 4) trasformare i dati tramite il **data mining**
- 5) trovare dei **patterns** (dopo il data mining)

## II. SOFTWARE SOLUTIONS AND LANGUAGES FOR AP AND DSS - 29.09.22

tipo operativo strutturato: una classe importante è eulad e delle decisioni di tipo operativo o strutturato cioè che si possono prendere tramite una procedura standard che può seguire un manuale o delle normative, automatizzata o no. queste decisioni di breve periodo si collocano in basso a destra in figura 2. Quindi non sono decisioni dove possiamo solo supportare allora si possono andare a codificare in un linguaggio di programmazione procedurale come c++ java ecc. questo approccio si oppone ad un approccio dove devo far generare delle azioni in seguito di un obiettivo. detto approccio procedurale

approccio dichiarativo: in 2 fasi nella prima modello il problema e nella seconda fase lo descrivo tramite un linguaggio di modellazione (modelling language) che è un linguaggio di programmazione matematico come AMPL, oppure in linguaggi come python con Amply e Pulp. abbiamo altri paradigmi basati sull'inferenza logica e quindi cambia sia il paradigma che il linguaggio. la terza fase abbiamo un solver of the shelf, che può essere un planner che ci darà delle istruzioni per il nostro contesto.

notare che se avviciniamo una decisione strutturata implemento una procedura tramite la quale creo un piano con certi vincoli oppure uso un approccio dichiarativo quindi analisi dei requisiti -> linguaggio -> solver

le limitazioni sono che in genere gli approcci dichiarativi sono più flessibili, dato che se cambia un vincolo operativo non devo mettere le mani al codice ma dovrò solo cambiare il modello senza cambiare il solver quindi è quello più economico e flessibile, in genere però è meno performante se in ambienti realtime devo prendere soluzioni in tempi molto stretti. quindi in questi casi servono approcci procedurali

in caso di alcuni sistemi al soluzione deve prendere soluzioni nel breve allora si usa C C++, C

tipo decisioni non strutturate o destrutturate: non possiamo automatizzare. si parte allora dai dati si tirano fuori i dati aggregati per creare statistiche ecc e con l'aiuto di modelli di ottimizzazione. in questi casi si parte da un'analisi dei dati sempre diversi. si usa uno spreadsheet per tirare fuori delle statistiche l'unico problema è che non riescono a gestire big data e tendono a generare errori

abbiamo anche soluzioni altre di tipo proprietario come SaS, Minitab, IBM

il linguaggio più usato è python ma non è la soluzione più efficiente per tutte quelle applicazioni dove il tempo di calcolo è importante dato che è un linguaggio precompilato????

strumenti usati: colab <https://colab.research.google.com/>

nell'ambito delle decisioni semistrutturate dove vogliamo solo valutare le prestazioni di un certo sistema. tipo quando parliamo di sistemi che presentano un comportamento random per un tot di motivi: 1. i server hanno un tempo di risposta che possiamo modellare 2. le richieste del sistema arrivano in maniera stocastica.

si usano in questi casi metodi simulativi tramite dei visual interactive modelling system per simulare la rete per la quale passano le informazioni e i server ognuno con diverse proprietà di ciascun linker

## RIFERIMENTI BIBLIOGRAFICI

- [1] <https://en.wikibooks.org/wiki/LaTeX/Hyperlinks>