UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale IN INGEGNERIA MECCANICA

STUDIO DELLE PERFORMANCE DI LINEE PRODUTTIVE TRAMITE TECNICHE DI ASSOCIATION RULES

STUDY OF THE PERFORMANCE OF PRODUCTION LINES THROUGH ASSOCIATION RULES TECHNIQUES

Relatore: Chiar.mo
Prof. CIARAPICA FILIPPO EMANUELE

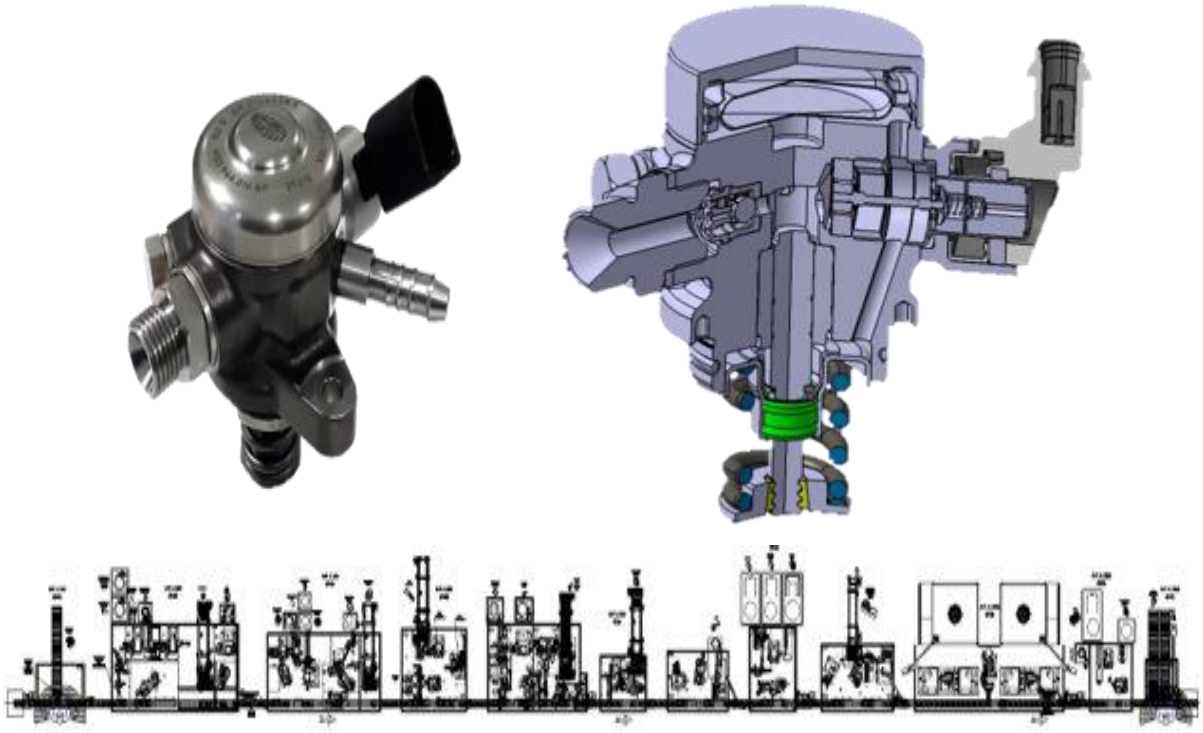
Tesi di Laurea di:
BORDI MATTIA

A.A. 2019/2020

INDICE

1. Studio linee ed indicatore OEE

1.1. Introduzione



Questa tesi tratta l'analisi di dati di produzione di una linea di assemblaggio di pompe della Magneti Marelli, in specifico lo studio delle performance di linee produttive tramite tecniche di Association Rules.

1.2. Linea Magneti Marelli

Magneti Marelli è una multinazionale italiana specializzata nella fornitura di prodotti e sistemi ad alta tecnologia per l'industria automobilistica con sede a Corbetta (MI, Italia). Fondata nel 1919, Magneti Marelli diventa sempre più conosciuta all'interno del settore automotive grazie al suo spirito pionieristico, e al contributo apportato alla mobilità intelligente e sostenibile. Durante la sua storia centenaria ha servito clienti dalla sede italiana per poi espandere le operazioni in Europa, Nord e Sud America, India e Cina, diventando un'azienda leader nel campo dell'illuminazione, dell'elettronica, della propulsione e del motorsport.

Nel 2018, Calsonic Kansei e Magneti Marelli hanno annunciato l'intenzione di effettuare una fusione per dare vita al settimo fornitore automotive indipendente a livello globale per fatturato.

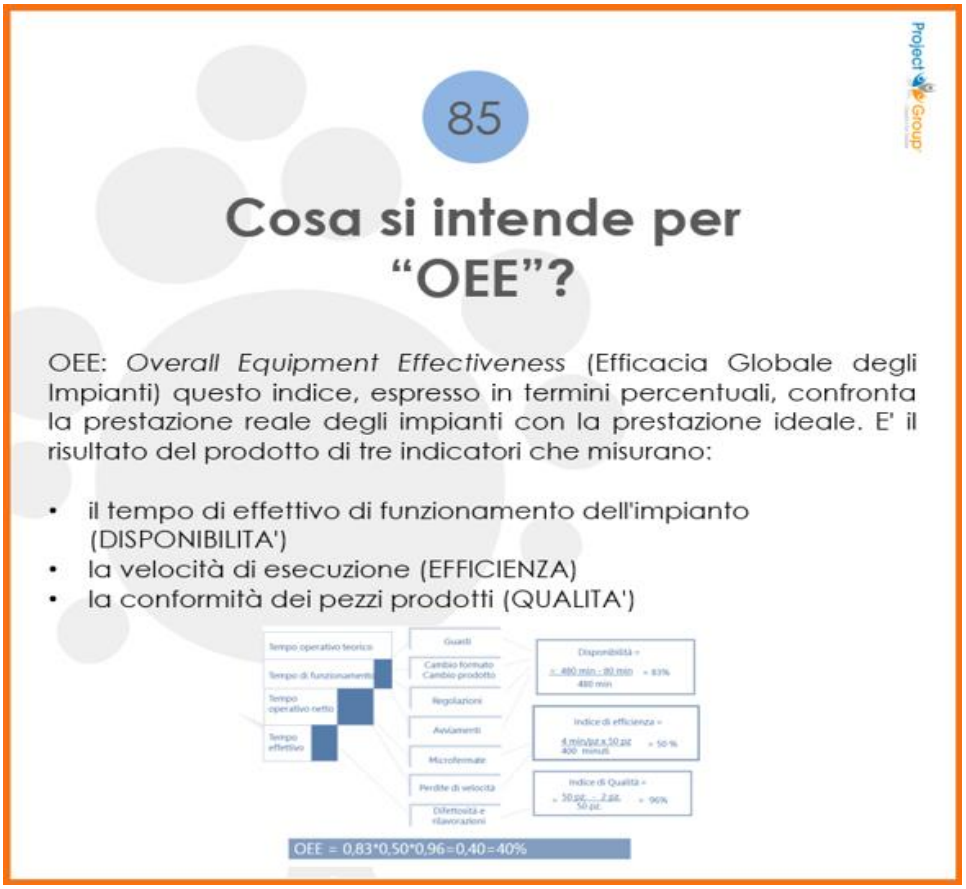
Nel corso dei suoi 80 anni di storia, Calsonic Kansei ha consolidato una reputazione di spicco per la qualità e l'eccellenza produttiva (Monozukuri). Dalla sua sede storica in Giappone, Calsonic Kansei ha ampliato la propria area operativa in Asia e in Europa, diventando un'azienda leader nelle soluzioni integrate per l'abitacolo (cockpit/interni per abitacolo), nei sistemi di climatizzazione, negli scambiatori di calore e nei compressori.

Nel 2019 è stata fondata ufficialmente MARELLI. L'unione di questi due giganti del settore ha consentito la fusione di una straordinaria esperienza industriale e di un patrimonio storico unico. Le due aziende infatti sono tra loro altamente complementari, sia in termini di linee di prodotto che di presenza geografica. La nascita di Marelli è fondata quindi su un'unione di qualità e innovazione.

Marelli, grazie alla sua ampia offerta rivolta ai principali player del settore, copre le principali aree di prodotto: illuminazione, comfort dell'abitacolo, propulsori elettrici, sistemi elettronici, tecnologia green, soluzioni integrate per l'abitacolo, propulsori, sospensioni, soluzioni termiche, motorsport.

Con circa 62.000 dipendenti nel mondo, il perimetro di MARELLI conta 170 fra stabilimenti e centri di Ricerca e Sviluppo in Asia, America, Europa e Africa e un fatturato di 14,6 miliardi di € (1.825 miliardi di yen) nel 2018.

1.3. L'OEE



L'**Overall Equipment Effectiveness** (OEE) è la misura di efficacia totale di un impianto. È un indice espresso in punti percentuali che riassume in sé tre concetti molto importanti dal punto di vista della produzione manifatturiera: la disponibilità, l'efficienza ed il tasso di qualità di un impianto.

La **DISPONIBILITÀ**, chiamata anche con il termine inglese *availability*, è la frazione del tempo allocato in cui l'impianto è effettivamente disponibile. Viene anche indicata con il termine Available Time o Scheduled Time.

È il rapporto tra il **Tempo Operativo** e il **Tempo Disponibile per la Produzione** e si esprime in percentuale.

Per **Tempo Disponibile per la Produzione** si intende il tempo totale disponibile della macchina (quindi tutto l'anno) meno le chiusure pianificate (quindi le manutenzioni programmate, le manutenzioni preventive, mancanza di ordini, chiusure aziendali, ecc…)

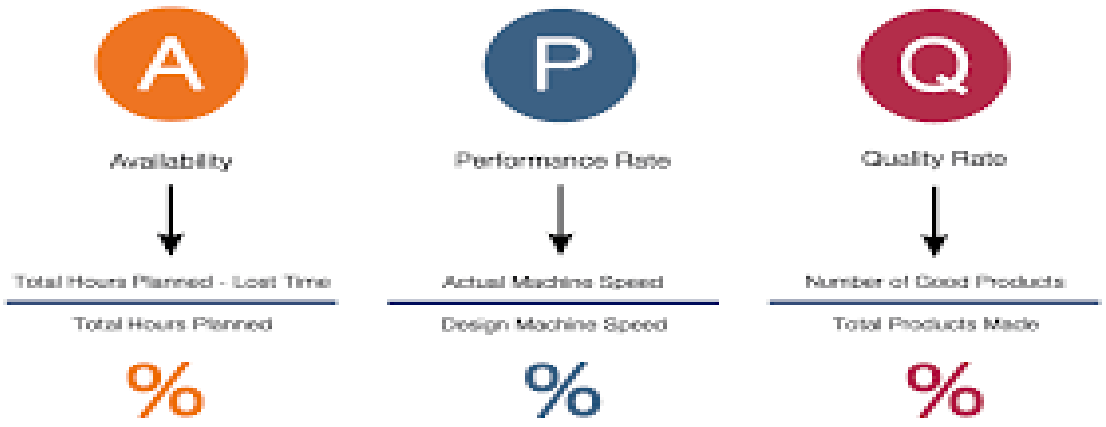
Per **Tempo Operativo** si intende il Tempo Disponibile per la Produzione meno il tempo in cui la macchina è occupata per attività non pianificate (rotture, setup, cambi versione, aggiustamenti).

L'EFFICIENZA, in inglese *throughput*, rappresenta la velocità con cui l'impianto sta lavorando come frazione rispetto a quella di progetto.
È il rapporto tra **Tempo Operativo Netto** e il **Tempo Operativo** e si esprime in percentuale.
Il **Tempo Operativo** è quello già calcolato per la Disponibilità.
Il **Tempo Operativo Netto** è il Tempo Operativo meno la quantità di tempo persa a causa di inefficienze produttive (microfermate, tempo ciclo più alto, ecc…).

Il **TASSO DI QUALITÀ**, in inglese *quality*, indica la percentuale di unità in specifica rispetto a tutte quelle prodotte.
È il rapporto tra **Tempo Operativo a Valore** e **Tempo Operativo Netto** e si esprime in percentuale.
Il **Tempo Operativo Netto** è quello già calcolato per l'Efficienza.
Il **Tempo Produttivo a Valore** è il Tempo Operativo Netto meno il tempo perso per difetti di qualità (scarti, tempi di startup, rilavorazioni).

Questi tre elementi includono quelle che sono definite tradizionalmente le "**Sei Maggiori Perdite** ", in inglese "Six Big Losses", correlate agli elementi principali sopraelencati.

Disponibilità	**Qualità**	**Efficienza**
• Guasti • Tempi di Set up	• Scarti e rilavorazioni • Tempo di start up	• Arresti dovuti a piccoli inconvenienti • Ridotta velocità di lavorazione



L'OEE è quindi un numero adimensionale (per intenderci, %) che tiene quindi conto delle tre principali categorie di perdite produttive.

Le perdite causate da questi tre elementi riducono l'ammontare dei pezzi conformi che una macchina può produrre.

Le perdite per **disponibilità** sono dovute principalmente a inattività, causata da guasti e tempi necessari al set-up delle macchine. Altre ulteriori perdite possono essere causate da strumenti di taglio e tempi di Start up.

Le perdite di prestazioni (**efficienza**) sono dovute prevalentemente alla ridotta velocità delle macchine. Interruzioni interne (eventi che interrompono il flusso produttivo senza fare fermare le macchine) e riduzione della velocità di lavoro (le macchine spesso lavorano a velocità minori di quelle per cui sono state progettate), sono la causa principale di diminuzione della produzione reale.

Il fattore **qualità** è influenzato negativamente dalla produzione di pezzi difettosi, i quali possono essere prodotti nella fase di avviamento (Start up), in molti casi necessaria per portare le macchine alle condizioni operative ottimali. Scarti e rilavorazioni sono altre perdite causate da errori in produzione.

La macchina ideale e completamente efficace dovrebbe lavorare tutto il tempo (o finchè necessario) alla velocità massima o standard, senza generare alcun tipo di problema per la qualità dei prodotti, ma la maggior parte delle macchine non raggiunge queste condizioni ideali. Le macchine non possono lavorare in maniera continuata o a velocità massima, in quanto subiscono vari arresti e producono (spesso) pezzi difettosi. Questi

problemi sono la causa della riduzione dell'efficienza delle macchine, come misurato dall'OEE.

Raccogliendo i dati dell'OEE su base fissa, è possibile individuare i procedimenti e le interferenze che causano problemi all'attrezzatura produttiva. Inoltre, i dati raccolti, permettono di valutare se gli interventi messi in atto per migliorare le prestazioni delle macchine hanno dato risultati positivi. Affinché il processo di misurazione e applicazione dei dati OEE risulti efficace dovrebbe essere coinvolto il personale operativo, il quale inoltre dovrebbe ricevere feedback (informazioni di ritorno) sui risultati dell'OEE.
La raccolta dati deve coinvolgere tutte le fasi di lavorazione ed avere la precisione necessaria per distinguere i pezzi scartati da quelli buoni. Richiede l'integrazione con i sistemi informativi aziendali, quali MES, ERP o software gestionali per poter raccogliere informazioni riguardo tempi di lavorazione e di funzionamento oltre che dati su commesse e ordini di lavoro. Inoltre è necessario poter rilevare quando un certo macchinario è fermo e le cause che hanno interrotto la produzione, solitamente direttamente dai PLC di controllo.
Le fonti di questi dati sono quasi sempre eterogenee e spesso, soprattutto se si vuole misurare l'OEE di un impianto di una certa dimensione, l'integrazione tra sistemi si rende necessaria. L'interconnessione tra macchinari e reparti della fabbrica diventa quindi un prerequisito fondamentale, come sottolineato anche dal piano Impresa 4.0. Un unico punto di raccolta ed integrazione dei dati semplifica il calcolo dell'OEE, perché tutte le informazioni raccolte sono già disponibili.

Prima di iniziare ad applicare l'OEE, è necessario decidere quali dati relativi a macchine e prodotti saranno misurati e utilizzati nel calcolo. I valori principali da misurare sono: le perdite che riducono la disponibilità, le prestazioni e la qualità. Tali perdite variano a seconda del macchinario, ma lo schema delle "Sei Maggiori Perdite" (Six Major Losses) fornisce un buon punto di partenza.
Le **perdite per inattività** sono misurate in unità di tempo. Esse includono:

- guasti e tempi di riparazione

- tempi di set-up e regolazione
- altre perdite causa di inattività.

Le **perdite di velocità** sono misurate in unità di produzione, ricavate dalla differenza tra produzione reale e produzione potenziale. Per produzione potenziale si intende: la produzione che si otterrebbe se le macchine lavorassero costantemente alla velocità standard ottimale, diversa per ogni prodotto.

Le **perdite dei difetti**, sono anch'esse misurate in unità di produzione. In questo caso il valore sarà ottenuto dalla differenza tra produzione reale totale e produzione che soddisfa le richieste dai clienti (prodotti conformi).

Lo scopo della documentazione sull'OEE non è quello di preparare documentazione cartacea in più. Una forma ben progettata permetterà facilmente di registrare i dati OEE come gli altri dati che è necessario registrare durante la produzione giornaliera. È necessario processare i dati per trasformarli in informazioni utili. Questo procedimento coinvolge il calcolo e immagazzina i dati in un modo che ci permetterà di ricavare differenti tipi di informazioni. È importante avere un sistema sul posto per immagazzinare i dati OEE. Esistono alcuni software che possono essere d'aiuto per il calcolo dei valori e l'organizzazione dei dati da usare nelle relazioni. Riportare i dati su tabelle nella postazione di lavoro è una via per migliorare i risultati futuri. Gli operatori devono essere informati sui risultati dell'OEE. Condividere le informazioni è un aspetto cruciale per ridurre le perdite.

Segnando questi dati nel corso del tempo, si potrà vedere l'andamento dell'OEE per le macchine e rispondere ad altre domande tipo:

- Quali sono i principali problemi di inattività?
- Quand'è capitato quell'incidente?
- Com'era la qualità il mese scorso?
- Come stiamo utilizzando la macchina?

In letteratura e sul web, l'**OEE** viene affrontato secondo un modello "**classico**" che ben si adatta all'industria di processo (o comunque, laddove esistono linee di produzione

automatizzate) ma che è difficile applicare in realtà organizzate per reparti, specie se queste producono per parti discrete (pezzi), e ancor più se le lavorazioni sono manuali. Nel corso degli anni, invece, è stata introdotta una metodologia di calcolo **innovativa** che, pur riconducendosi allo stesso modello di base, è decisamente più adatta ai tipici contesti produttivi per reparti, oppure misti reparti/linee, con produzione di parti discrete (pezzi).

La definizione di OEE "**classica**":

(i) OEE = Disponibilità x Prestazione x Qualità

Il risultato dell'OEE quindi calcola tutti i problemi del processo produttivo come i guasti, la riduzione della velocità di produzione, gli scarti, i prodotti qualitativamente non all'altezza e le rilavorazioni. Nei contesti attuali questa formula è però di difficile attuazione in quanto solitamente, in un'azienda, vi sono più reparti produttivi, ognuno dei quali dovrebbe veder applicato l'OEE per una parte del prodotto.

Questo modello, seppure teoricamente valido, diventa di difficile applicazione nella maggioranza dei contesti produttivi.

Il modello **innovativo:**

OEE = "Tempo redditizio" / Tempo disponibile

L'**OEE**, al pari di tutti gli indicatori di efficienza, per definizione può essere espresso attraverso un rapporto **OUTPUT/INPUT.**

Esso dà infatti una indicazione globale sulla capacità di un insieme di risorse di produrre valore per il cliente (output) con le risorse produttive a disposizione (input).

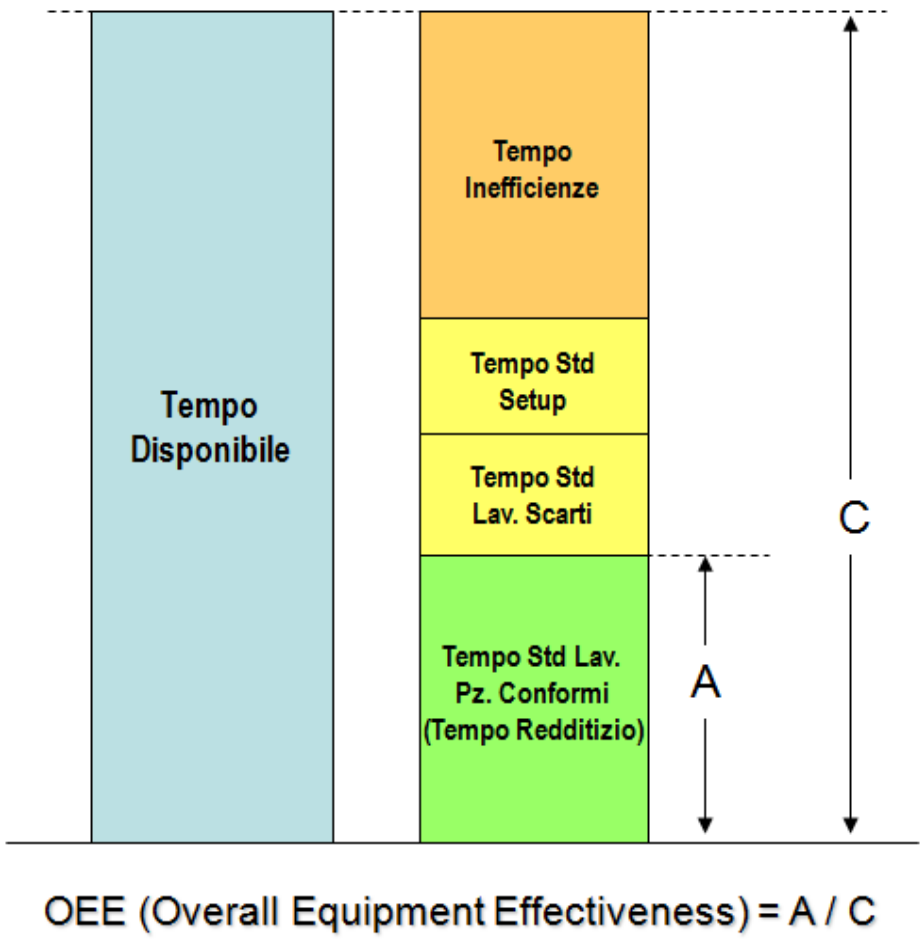
Come misurare output e input? Ovviamente, il numero di pezzi non va bene, a causa delle differenze nei cicli e nei tempi di produzione.

Di certo, l'**input** deve essere proporzionale all'impegno che l'azienda investe nel sistema produttivo, impegno che è ben rappresentato dalla **disponibilità oraria delle risorse**,

siano esse manodopera o macchine/impianti. Non a caso, entrambe le categorie di risorse tipicamente hanno costi orari ad essi associati.

L'**output** deve essere espresso in una unità di misura confrontabile, e cioè temporale, in modo che l'OEE risulti un rapporto di elementi tra loro omogenei, dunque una percentuale.

In questo contesto acquista valore il concetto di **tempo standard di lavoro**: è il tempo necessario per l'esecuzione di una data operazione a fronte di strumenti, metodi e procedure operative stabiliti. Il tempo standard è definito dall'azienda per lo specifico ciclo di lavoro, manuale o automatico che sia: è il "tempo giusto" che serve per eseguire una lavorazione, né più, né meno. In questa accezione, il tempo standard si avvicina (anche se non è esattamente la stessa cosa) al concetto di **tempo a valore aggiunto**. Per non creare ambiguità, possiamo chiamarlo **tempo redditizio**.



OEE (Overall Equipment Effectiveness) = A / C

Il tempo standard per le attività basate su macchine o impianti a controllo numerico è tipicamente "predeterminato", in quanto dipende dal ciclo tecnologico. Per le lavorazioni manuali, invece, questo è più complesso, in quanto per la sua determinazione è necessario avvalersi di analisi sperimentali secondo le tecniche Tempi e Metodi.

Calcolare l'OEE apporta numerosi vantaggi per l'azienda: solitamente un sistema produttivo che non viene gestito calcolando l'OEE si attesta sul 50-60% dell'efficienza. Chi invece intuisce l'importanza dell'OEE, come le aziende più efficienti, riesce ad arrivare all'85% di efficacia generale dell'attrezzatura. Adoperarsi per raccogliere i dati e calcolare l'OEE è dunque fondamentale se si vuole migliorare la produzione aziendale.

I migliori produttori, invece, raggiungono e mantengono nel tempo un OEE pari all'**85%**, considerato un obiettivo **"world class"**.

Raggiungere la condizione ideale del 100% è virtualmente impossibile, in quanto rappresenterebbe un sistema che non si ferma mai e che non effettua mai attrezzaggi/setup. Se l'OEE risultasse maggiore del 100%, anzi, sarebbe sintomo di inaccuratezza del modello impostato (ad esempio, tempi standard sovradimensionati e quindi inesatti). Anche valori alti (maggiori del 70%), se rilevati in contesti che non hanno mai affrontato un processo strutturato di miglioramento dell'efficienza, devono essere validati approfonditamente.

Il calcolo dell'OEE non migliora automaticamente la produttività. Esso deve essere abbinato ad una analisi dettagliata ed accurata dei motivi alla base della ridotta produttività.

Per raggiungere il "world class" dell'85% servono non solo una buona gestione tecnica delle risorse, ma anche e soprattutto una ottima gestione organizzativa. In questo senso, l'esperienza maturata negli anni presso numerose realtà manifatturiere può rendere molto efficace un intervento in azienda da parte di personale esterno, che può impostare correttamente il metodo, progettare le attività di miglioramento e monitorarne i risultati.

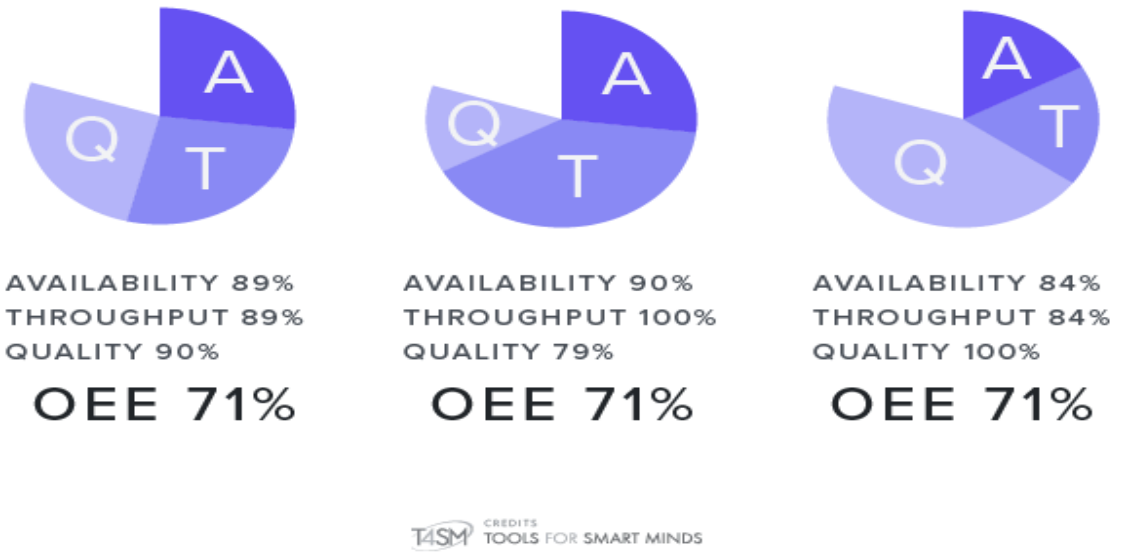
Come ogni valore statistico, tuttavia, gli indici sono un'arma a doppio taglio: il loro più grande pregio infatti può essere anche il loro maggiore difetto. Siccome gli indici statistici sono per loro natura sintetici, riassumono molto bene una grande quantità di informazioni ma, al contempo, mascherano i dettagli che possono essere importanti per capire le cause di inefficienze e problemi. Calcolare la media, ad esempio, nasconde la presenza di pochi valori fuori dai limiti normali. Per questo motivo gli indici non vanno mai utilizzati da soli e devono essere contestualizzati: per avere un quadro completo bisogna utilizzare più indici in concerto tra loro.

Un singolo indice, appunto, utilizzato da solo può non fornire indicazioni sufficienti a chi deve prendere decisioni vitali per la gestione di un impianto produttivo. In particolare, **l'OEE** da solo non è in grado di rispondere a due domande molto importanti, che rappresentano due facce della stessa medaglia:

- Quali sono le cause di un OEE troppo basso per gli obiettivi aziendali?
- Come si può intervenire per migliorare un OEE insufficiente?

Un'indicazione iniziale per rispondere alla prima domanda si può ottenere analizzando le tre componenti di cui l'OEE è costituito.

Il valore più basso tra i tre è quello che maggiormente incide nel causare prestazioni insufficienti.



Questo dato ci servirà per capire se bisogna verificare la scarsa qualità dei pezzi prodotti, concentrare l'analisi sulla bassa velocità dell'impianto o se esaminare le cause che

troppo spesso fermano i macchinari di produzione. Chiaramente per rispondere con precisione alla prima delle due domande è necessario approfondire oltre la semplice analisi dei tre valori, ma questi indicano la strada su cui proseguire.

Rispondere alla seconda domanda può invece essere più complesso. Sebbene possa sembrare che la risposta provenga ancora una volta dalle tre componenti citate sopra, per aumentare l'efficienza di un impianto è importante fare più attenzione.

Prendendo il caso in cui la produttività sia il fattore più basso della moltiplicazione, si potrebbe essere tentati di alzare il numero di pezzi prodotti aumentando la velocità di produzione. Questo potrebbe portare ad avere molti più pezzi prodotti fuori dalla tolleranza prevista ed essere quindi scartati. Come effetto complessivo quindi, a fronte di un elevato aumento di velocità, si potrebbe avere un modesto incremento dell'OEE, con un aggravio dei costi di produzione a causa del maggior numero di scarti. Per questo motivo, oltre ad una attenta considerazione riguardo le modifiche dell'impianto di produzione sotto esame, è importante poter tenere traccia dei cambiamenti che subisce l'OEE a fronte degli interventi effettuati. Questo indice viene quindi utilizzato anche per misurare l'impatto che differenti strategie lavorative possono avere, appunto, sull'efficienza complessiva dell'impianto.

L'OEE è utilizzato come strumento di misurazione nel TPM (Total Productive Maintenance) e nei programmi di Lean Manufacturing, dove riesce a fornire un'importante chiave di lettura dell'efficacia delle misure adottate fornendo al tempo stesso un supporto per la misurazione dell'efficienza.

2. Il Machine Learning e le Association Rules

2.1. Il Machine Learning

Anche se oggi parlare di apprendimento automatico, di intelligenza artificiale, di computer e macchine intelligenti sembra quasi la normalità, per arrivare ai risultati

odierni la strada è stata molto complessa, perché divisa tra sperimentazioni e scetticismo.

Le prime sperimentazioni per la realizzazione di macchine intelligenti risalgono agli inizi degli anni Cinquanta del Novecento, quando alcuni matematici e statistici iniziarono a pensare di utilizzare i metodi probabilistici per realizzare macchine che potessero prendere decisioni proprio tenendo conto delle probabilità di accadimento di un evento.

Il primo grande nome legato al machine learning è sicuramente quello di Alan Turing, che ipotizzò la necessità di realizzare algoritmi specifici per realizzare macchine in grado di apprendere.

In quegli stessi anni, anche gli studi sull'intelligenza artificiale, sui sistemi esperti e sulle reti neurali vedevano momenti di grossa crescita alternati da periodi di abbandono, causati soprattutto dalle molte difficoltà riscontrate nelle possibilità di realizzazione dei diversi sistemi intelligenti, nella mancanza di sussidi economici e dallo scetticismo che circondava spesso chi provava a lavorarci.

A partire dagli anni Ottanta, una serie di interessanti risultati ha portato alla rinascita di questo settore della ricerca: una rinascita che è stata resa possibile da nuovi investimenti nel settore.

Alla fine degli anni Novanta l'apprendimento automatico trova nuova linfa vitale in una serie di innovative tecniche legate ad elementi statistici e probabilistici: si trattava di un importante passo che permise quello sviluppo che ha portato oggi l'apprendimento automatico ad essere un ramo della ricerca riconosciuto e altamente richiesto.

Quando si parla di machine learning, si parla di una particolare branca dell'informatica che può essere considerata una parente stretta dell'intelligenza artificiale.

Ad oggi la definizione più accreditata dalla comunità scientifica è quella fornita da un americano, Tom Michael Mitchell, direttore del dipartimento Machine Learning della Carnegie Mellon University:

«si dice che un programma apprende dall'esperienza E con riferimento a alcune classi di compiti T e con misurazione della performance P, se le sue performance nel compito T, come misurato da P, migliorano con l'esperienza E».

Definire in maniera semplice le caratteristiche e le applicazioni del machine learning non è sempre possibile, visto che questo ramo è molto vasto e prevede differenti modalità, tecniche e strumenti per essere realizzato.

Inoltre, le differenti tecniche di apprendimento e sviluppo degli algoritmi danno vita ad altrettante possibilità di utilizzo che allargano il campo di applicazione dell'apprendimento automatico, rendendone difficile una definizione specifica.

Si può tuttavia dire che quando si parla di machine learning si parla di differenti meccanismi che permettono a una macchina intelligente di migliorare le proprie capacità e prestazioni nel tempo. La macchina, quindi, sarà in grado di imparare a svolgere determinati compiti migliorando, tramite l'esperienza, le proprie capacità, le proprie risposte e funzioni.

Alla base dell'apprendimento automatico ci sono una serie di differenti algoritmi che, partendo da nozioni primitive, sapranno prendere una specifica decisione piuttosto che un'altra o effettuare azioni apprese nel tempo.

L'aspetto più importante del machine learning è la ripetitività, perché più i modelli sono esposti ai dati, più sono in grado di adattarsi in modo autonomo.

I computer imparano da elaborazioni precedenti per produrre risultati e prendere decisioni che siano affidabili e replicabili.

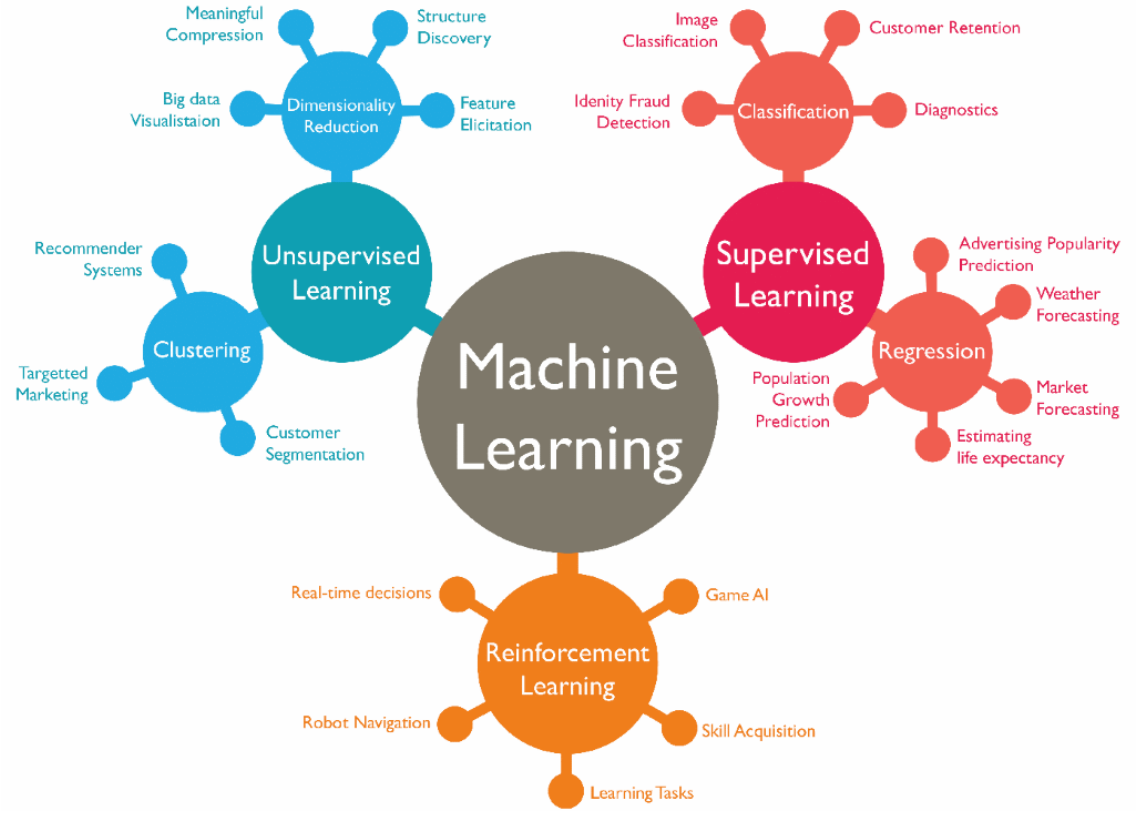
Grazie alle nuove tecnologie di elaborazione, il machine learning di oggi non è il machine learning del passato.

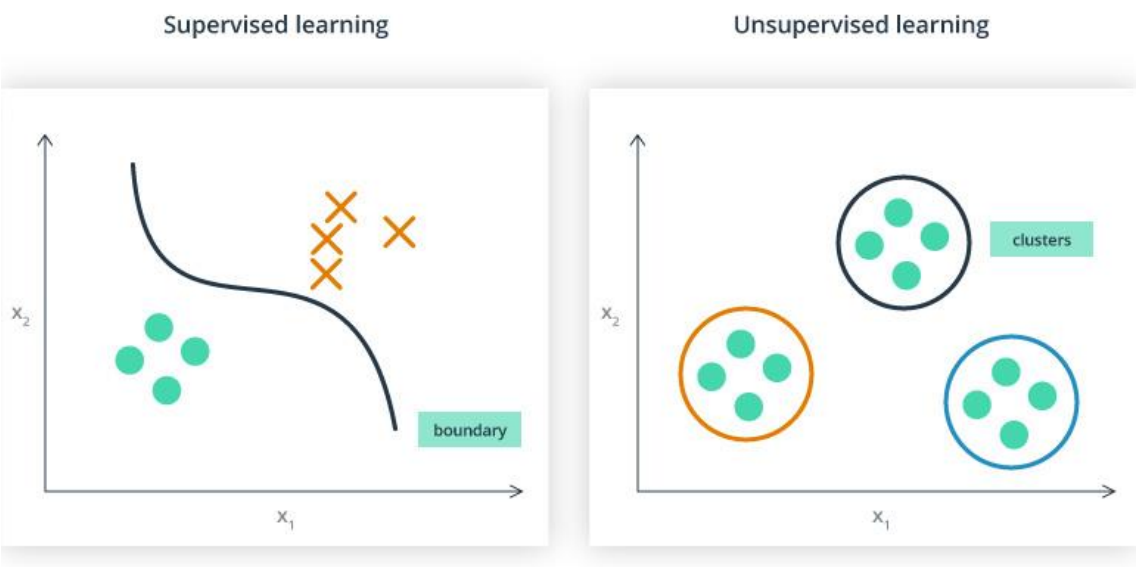
Questa scienza non è nuova ma sta acquisendo un nuovo slancio e sebbene molti algoritmi di machine learning siano in circolazione da molto tempo, la capacità di applicare calcoli matematici complessi ai big data è uno sviluppo più recente.

Il primo a coniare il termine Machine Learning fu Arthur Lee Samuel, scienziato americano pioniere nel campo dell'Intelligenza Artificiale, nel 1959, il quale identificò due approcci distinti, sui quali si basa il funzionamento del Machine Learning stesso. Essi permettono di distinguere l'apprendimento automatico in due sottocategorie del Machine Learning a seconda del fatto che si diano al computer esempi completi da utilizzare come indicazione per eseguire il compito richiesto (apprendimento

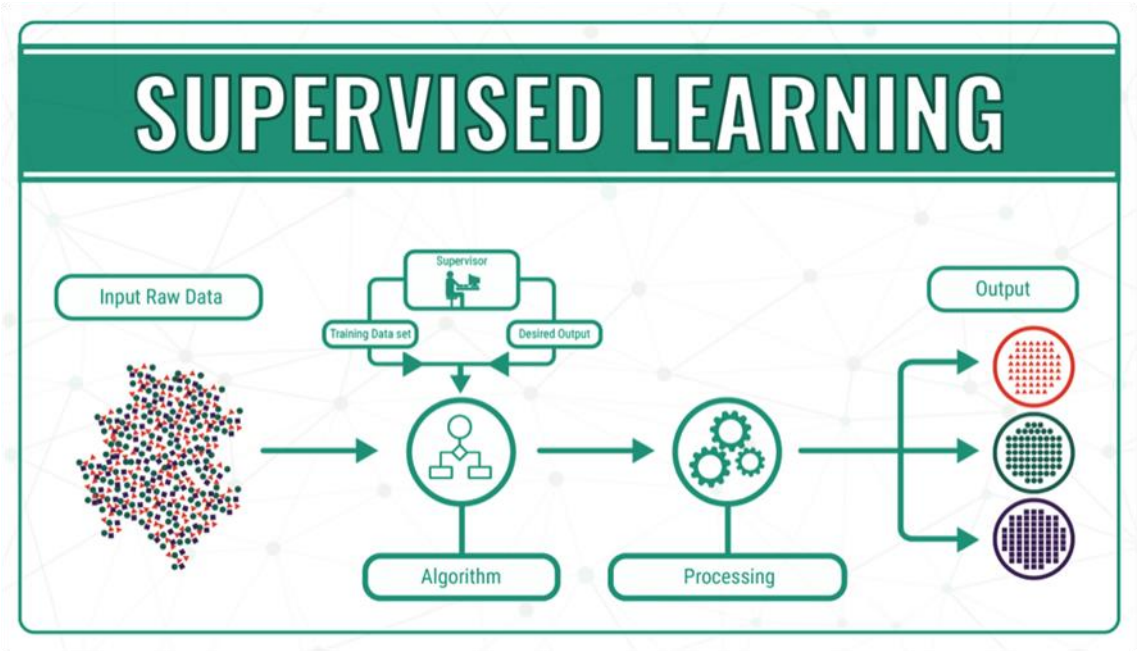
supervisionato) oppure che si lasci lavorare il software senza alcun "aiuto" (apprendimento non supervisionato).

In realtà, ci sono poi dei sottoinsiemi che consentono di fare un'ulteriore classificazione ancora più dettagliata del Machine Learning proprio in base al suo funzionamento.





Apprendimento supervisionato (supervised learning)



L'apprendimento supervisionato consiste nel fornire al sistema informatico della macchina una serie di nozioni specifiche e codificate, ossia di modelli ed esempi che permettono di costruire un vero e proprio database di informazioni e di esperienze.

In questo modo, quando la macchina si trova di fronte ad un problema, non dovrà fare altro che attingere alle esperienze inserite nel proprio sistema, analizzarle, e decidere quale risposta dare sulla base di esperienze già codificate.

Questo tipo di apprendimento è, in qualche modo, fornito già confezionato e la macchina deve essere solo in grado di scegliere quale sia la migliore risposta allo stimolo che le viene dato.

Gli algoritmi che fanno uso di apprendimento supervisionato vengono utilizzati in molti settori, da quello medico a quello di identificazione vocale: essi, infatti, hanno la capacità di effettuare ipotesi induttive, ossia ipotesi che possono essere ottenute scansionando una serie di problemi specifici per ottenere una soluzione idonea ad un problema di tipo generale.

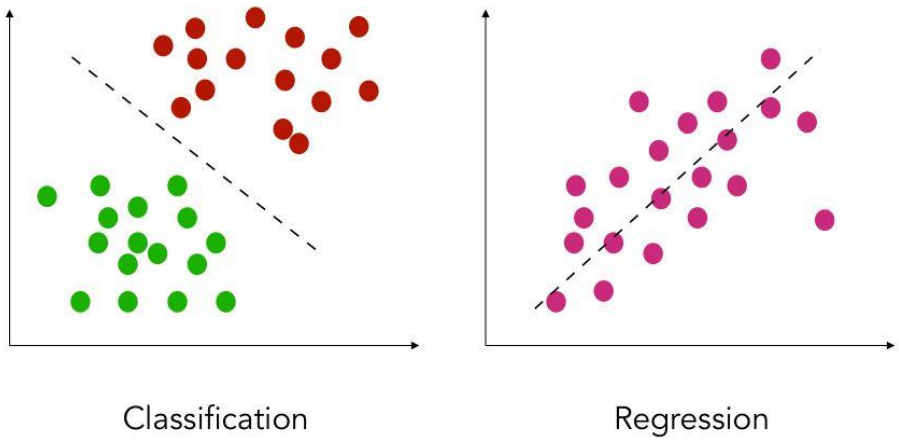
Dal punto di vista logico, una classica implementazione di apprendimento supervisionato è costituita da:

- Un insieme esperienza “E” che contiene esempi del comportamento che si desidera nel sistema. È rappresentato come un insieme di coppie di input-output.
- Degli input “I” che rappresentano gli input al sistema e che tipicamente sono forniti sotto forma di vettori.
- Degli output “O” che rappresentano le risposte del sistema e che possono assumere forma di valori continui o di etichetta numerica.
- Una funzione “ha”, chiamata ipotesi induttiva, che ad ogni dato in ingresso I associa l'ipotetica risposta corretta del sistema “O”. “ha” rappresenta la parte di sistema che deve cambiare per ottimizzare l'efficienza del suo comportamento.
- Un'ipotetica funzione “hb”, chiamata funzione obiettivo, che ad ogni dato in ingresso I associa la risposta corretta desiderata dal progettista-utilizzatore.
  È una formalizzazione teorica dei voleri del progettista-utilizzatore.
- Un parametro di efficienza “F” che rappresenta l'efficienza del sistema. Generalmente, a parità di input, consiste nella differenza di output tra “ha” e “hb”.

Tutti gli algoritmi di apprendimento supervisionato partono dal presupposto che, se forniamo al sistema un numero adeguato di esempi, questo accumulerà un'esperienza

“E” sufficiente da permettergli di creare una funzione “ha” adeguata ad approssimare la funzione “hb” (e quindi il comportamento desiderato da chi ha fornito gli esempi). Data la similitudine tra le funzioni “ha” e “hb”, quando proporremo al sistema dei dati in ingresso non presenti nella sua esperienza “E”, la funzione “ha” dovrebbe essere in grado di approssimare in maniera sufficientemente precisa la funzione “hb” e fornire delle risposte “O” sufficientemente soddisfacenti.
Per raggiungere questo obiettivo il sistema sfrutta spesso due principi che sono quello della distribuzione (matematica) e quello della funzione di verosimiglianza.
Una volta identificata la distribuzione matematica che lega il variare dei valori degli input ai valori degli output desiderati il sistema sceglie i parametri che massimizzano la probabilità dei dati ed esprime la funzione di verosimiglianza appropriata.
Si può facilmente intuire che il funzionamento corretto ed efficiente di questi algoritmi dipende in modo significativo dall'esperienza; se si fornisce poca esperienza, l'algoritmo potrebbe non creare una funzione interna efficiente, mentre con un'esperienza eccessiva la funzione interna potrebbe divenire molto complessa tanto da rendere lenta l'esecuzione dell'algoritmo.
Le principali categorie dell'apprendimento supervisionato sono:

- La classificazione: quando la variabile output desiderata è categorica (qualitativa nominali o ordinale).
- Regressione: quando la variabile output desiderata è quantitativa.



Nella **classificazione** la macchina viene addestrata alla classificazione dal supervisore tramite l'aggiunta di etichette sui dati in cui giudica il risultato.
Ogni etichetta è una classe discreta che identifica il risultato atteso oppure un giudizio di valore.

Alla macchina spetta il compito di trovare una relazione tra i valori di input (valori descrittivi) e di output.

Gli algoritmi più comuni sono il Naive Bayes Classifier, l'Albero decisionale, la Regressione logistica, K-Nearest Neighbours (K-NN) e Support Vector Machine (SVM). Si possono anche usare metodi abbinati (combinazioni di modelli), come Random Forest e altri metodi metodi di potenziamento come AdaBoost e XGBoost.

L'**analisi della regressione,** invece, è una tecnica usata per analizzare una serie di dati che consistono in una variabile dipendente e una o più variabili indipendenti.

Lo scopo è stimare un'eventuale relazione funzionale esistente tra la variabile dipendente e le variabili indipendenti.

La variabile dipendente nell'equazione di regressione è una funzione delle variabili indipendenti più un termine d'errore.

Quest'ultimo è una variabile casuale e rappresenta una variazione non controllabile e imprevedibile nella variabile dipendente.

I parametri sono stimati in modo da descrivere al meglio i dati. Il metodo più comunemente utilizzato per ottenere le migliori stime è il metodo dei "minimi quadrati" (OLS), ma sono utilizzati anche altri metodi.

Il data modeling può essere usato senza alcuna conoscenza dei processi sottostanti che hanno generato i dati; in questo caso il modello è un modello empirico.

Inoltre, nella modellizzazione, non è richiesta la conoscenza della distribuzione di probabilità degli errori.

L'analisi della regressione richiede ipotesi riguardanti la distribuzione di probabilità degli errori.

Test statistici vengono effettuati sulla base di tali ipotesi.

Nell'analisi della regressione il termine "modello" comprende sia la funzione usata per modellare i dati che le assunzioni concernenti la distribuzione di probabilità.

L'analisi della regressione può essere usata per effettuare previsioni (ad esempio per prevedere dati futuri di una serie temporale), inferenza statistica, per testare ipotesi o per modellare delle relazioni di dipendenza.

Questi usi della regressione dipendono fortemente dal fatto che le assunzioni di partenza siano verificate.

L'uso dell'analisi della regressione è stato criticato in diversi casi in cui le ipotesi di partenza non possono essere verificate.

Un fattore che contribuisce all'uso improprio della regressione è che richiede più competenze per criticare un modello che per adattarlo.

L'algoritmo più semplice e veloce è la regressione lineare (metodo dei minimi quadrati), ma bisognerebbe fermarsi qui, perché spesso offre un risultato mediocre. Altri algoritmi di regressione di machine learning comuni (escluse le reti neurali) includono Naive Bayes, Albero decisionale, K-Nearest Neighbors, LVQ (Learning Vector Quantization), Least Angle Regression (LARS), Elastic Net, Random Forest, AdaBoost e XGBoost.

Per poter comprendere meglio la differenza tra classificazione e regressione può risultare utile vedere degli esempi concreti.

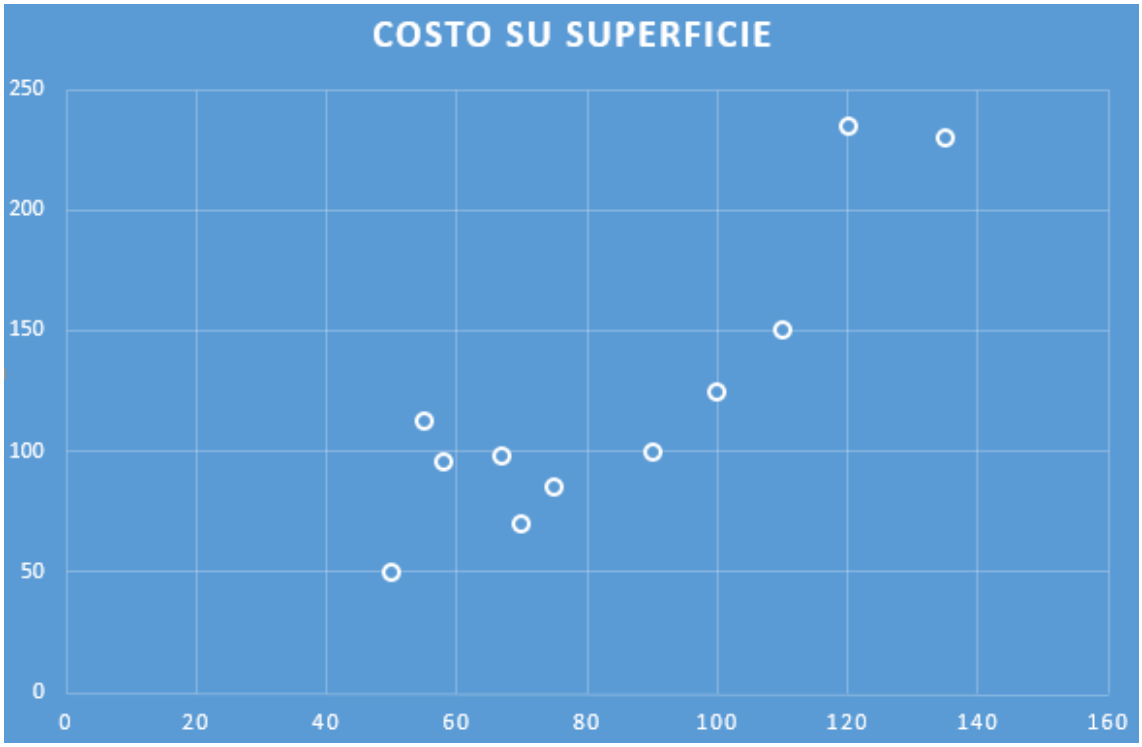
Supponiamo di voler vendere una casa e di conseguenza stabilire un prezzo.

Se volessimo fare in modo che, ad esempio, il prezzo si adegui, giorno per giorno, ai movimenti di mercato, avremmo bisogno di un modello più complesso, di un algoritmo che sia in grado di determinare il prezzo in funzione delle vendite che sono state fatte nei giorni precedenti.

Abbiamo bisogno, quindi, di un algoritmo che apprenda, che impari dalle vendite che sono state fatte e ci fornisca il prezzo più "consono".

Per semplificare il problema possiamo ipotizzare che il prezzo sia funzione di una sola variabile: la superficie della casa.

Reperendo una tabella con superfici e costi annessi, se inseriamo in un piano cartesiano questi dati, posizionando la superficie (variabile indipendente) sull'asse delle x e il prezzo (variabile dipendente, nel nostro esempio, dalla sola superficie) sull'asse delle y, otteniamo questo:



Quello che vogliamo trovare, quindi, è la retta che approssima meglio questa distribuzione.

Tale problema viene denominato problema di regressione lineare.

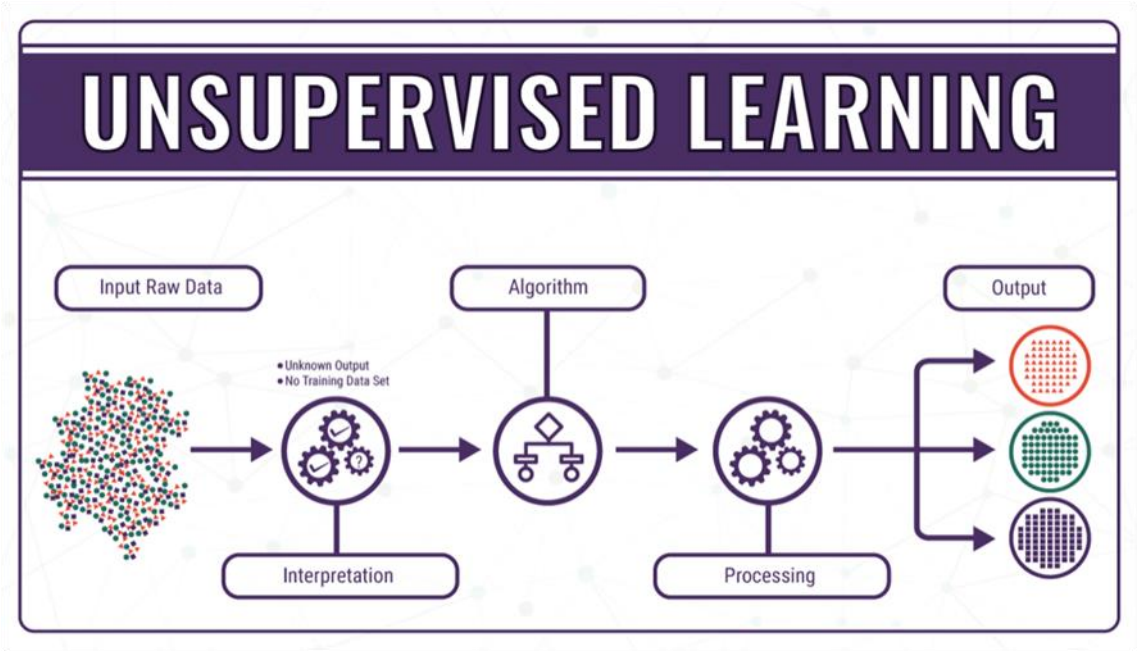
Una volta individuata la retta ottima, ovvero quella che ottimizza i dati mostrati in tabella sarà possibile stimare un qualunque prezzo, data una qualunque superficie.

Ora, invece, supponiamo di avere a disposizione una tabella che metta in relazione l'età del paziente e il valore di un determinato parametro del sangue (es. quantità globuli bianchi per millilitro) con l'effettivo riscontro o meno di una malattia del sangue (es. leucemia).

In questo caso, non vogliamo stimare un valore continuo, ma la presenza o meno ("sì" o "no") di una determinata malattia.

Tale problema è, appunto, un problema di classificazione.

Apprendimento non supervisionato (unsupervised learning)



L'apprendimento non supervisionato prevede invece che le informazioni inserite all'interno della macchina non siano codificate, ossia la macchina ha la possibilità di attingere a determinate informazioni senza avere alcun esempio del loro utilizzo e, quindi, senza avere conoscenza dei risultati attesi a seconda della scelta effettuata.

Dovrà essere la macchina stessa, quindi, a catalogare tutte le informazioni in proprio possesso, organizzarle ed imparare il loro significato, il loro utilizzo e, soprattutto, il risultato a cui esse portano.

L'apprendimento senza supervisione offre maggiore libertà di scelta alla macchina che dovrà organizzare le informazioni in maniera intelligente e imparare quali sono i risultati migliori per le differenti situazioni che si presentano.

Le principali tecniche di machine learning senza supervisione sono le seguenti:

- Clustering
- Riduzione della dimensione dei dati

Il **Clustering** è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e raggruppamento di elementi omogenei in un insieme di dati.

Le tecniche di clustering si basano su misure relative alla somiglianza tra gli elementi. In molti approcci questa similarità, o meglio, dissimilarità, è concepita in termini di distanza in uno spazio multidimensionale.

La bontà delle analisi ottenute dagli algoritmi di clustering dipende molto dalla scelta della metrica, e quindi da come è calcolata la distanza.

Gli algoritmi di clustering raggruppano gli elementi sulla base della loro distanza reciproca, e quindi l'appartenenza o meno a un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme stesso. Le tecniche di clustering si possono basare principalmente su due "filosofie":

-Dal basso verso l'alto (metodi aggregativi o bottom-up): questa filosofia prevede che inizialmente tutti gli elementi siano considerati cluster a sé, e poi l'algoritmo provvede ad unire i cluster più vicini.

L'algoritmo continua ad unire elementi al cluster fino ad ottenere un numero prefissato di cluster, oppure fino a che la distanza minima tra i cluster non supera un certo valore, o ancora in relazione ad un determinato criterio statistico prefissato.

-Dall'alto verso il basso (metodi divisivi o top-down): all'inizio tutti gli elementi sono un unico cluster, e poi l'algoritmo inizia a dividere il cluster in tanti cluster di dimensioni inferiori.

Il criterio che guida la divisione è naturalmente quello di ottenere gruppi sempre più omogenei.

L'algoritmo procede fino a che non viene soddisfatta una regola di arresto generalmente legata al raggiungimento di un numero prefissato di cluster.

Esistono varie classificazioni delle tecniche di clustering comunemente utilizzate.

Una prima categorizzazione dipende dalla possibilità che un elemento possa o meno essere assegnato a più cluster:

-Clustering esclusivo: ogni elemento può essere assegnato ad uno e ad un solo gruppo. Quindi i cluster risultanti non possono avere elementi in comune.

Questo approccio è detto anche hard clustering.

-Clustering non-esclusivo: un elemento può appartenere a più cluster con gradi di appartenenza diversi.

Questo approccio è noto anche con il nome di soft clustering o fuzzy clustering, dal termine usato per indicare la logica fuzzy.

La logica fuzzy è una logica in cui si può attribuire a ciascuna proposizione un grado di verità diverso da 0 e 1 e compreso tra di loro.

Con grado di verità o valore di appartenenza si intende quanto è vera una proprietà, che può essere, oltre che vera (= a valore 1) o falsa (= a valore 0) come nella logica classica, anche parzialmente vera e parzialmente falsa.

Formalmente, questo grado di appartenenza è determinato da un'opportuna funzione di appartenenza $\mu F(x) = \mu$.

La x rappresenta dei predicati da valutare e appartenenti a un insieme di predicati X.

La μ rappresenta il grado di appartenenza del predicato all'insieme fuzzy considerato e consiste in un numero reale compreso tra 0 e 1.

Un'altra suddivisione delle tecniche di clustering tiene conto del tipo di algoritmo utilizzato per dividere lo spazio:

-Clustering partizionale (detto anche non gerarchico, o k-clustering), in cui per definire l'appartenenza ad un gruppo viene utilizzata una distanza da un punto rappresentativo del cluster (centroide, medioide ecc...), avendo prefissato il numero di gruppi della partizione risultato. Si tratta di derivazioni del più noto algoritmo di clustering, quello detto delle k-means, introdotto da MacQueen nel 1967.

Gli algoritmi di clustering partizionali sono più adatti a data set molto grandi, per i quali la costruzione di una struttura gerarchica dei cluster porterebbe a uno sforzo computazionale molto elevato.

-Clustering gerarchico, in cui viene costruita una gerarchia di partizioni caratterizzate da un numero (de)crescente di gruppi, visualizzabile mediante una rappresentazione ad albero (dendrogramma), in cui sono rappresentati i passi di accorpamento/divisione dei gruppi.

Queste due suddivisioni sono del tutto trasversali, e molti algoritmi nati come "esclusivi" sono stati in seguito adattati nel caso "non-esclusivo" e viceversa.

La **riduzione della dimensione dei dati,** invece, è una tecnica in cui l'algoritmo di apprendimento elimina i dati non significativi (rumore) e combina le informazioni ridondanti (correlate) per concentrare l'analisi su quelli in cui emerge uno schema.
Risulta utile nel machine learning per eliminare dal dataset le informazioni ridondanti (correlate), meno o poco rilevanti per il problema da risolvere.
E' senza dubbio più semplice e meno dispendioso addestrare un algoritmo con uno spazio dati di dimensione inferiore.
La riduzione dei dati è usata anche per rappresentare i dati in una dimensione inferiore e più interpretabile, ad esempio, per visualizzare un diagramma 3D in 2D.
La riduzione della dimensione ha vantaggi e svantaggi.
Il principale vantaggio è quello di poter comprimere il volume dei dati, riducendo la complessità computazionale dell'algoritmo di apprendimento.
Il principale svantaggio risiede nel fatto che riducendo la dimensione si potrebbero degradare le informazioni e le prestazioni predittive dell'algoritmo di apprendimento.

L'**apprendimento per rinforzo** è una filosofia di programmazione che punta a realizzare algoritmi in grado di apprendere e adattarsi alle mutazioni dell'ambiente.
Questa tecnica di programmazione si basa sul presupposto di potere ricevere degli stimoli dall'esterno a seconda delle scelte dell'algoritmo.
Quindi una scelta corretta comporterà un premio mentre una scelta scorretta porterà ad una penalizzazione del sistema.
L'obiettivo del sistema è il raggiungimento del maggior premio possibile e di conseguenza del migliore risultato possibile.
In questo caso, quindi, alla macchina vengono forniti una serie di elementi di supporto, quali sensori, telecamere, GPS eccetera, che permettono di rilevare quanto avviene nell'ambiente circostante ed effettuare scelte per un migliore adattamento all'ambiente intorno a loro.
Questo tipo di apprendimento è tipico delle auto senza pilota, che grazie a un complesso sistema di sensori di supporto è in grado di percorrere strade cittadine e non, riconoscendo eventuali ostacoli, seguendo le indicazioni stradali e molto altro.

Questo tipo di apprendimento è solitamente modellizzato tramite i processi decisionali di Markov, che forniscono un framework matematico per la modellizzazione del processo decisionale in situazioni in cui i risultati sono in parte casuale e in parte sotto il controllo decisionale.

Gli MDP sono utili per lo studio di una vasta gamma di problemi di ottimizzazione e sono noti fin dal 1950.

Essi sono utilizzati in una vasta area di discipline in cui il processo di presa di decisione avviene in un intorno dinamico, tra cui la robotica, l'automazione, l'economia, e la produzione industriale.

L'apprendimento per rinforzo può essere effettuato con diverse tipologie di algoritmi, classificabili in base all'utilizzo di un modello che descriva l'ambiente, alle modalità di raccolta dell'esperienza (in prima persona o da parte di terzi), al tipo di rappresentazione degli stati del sistema e delle azioni da compiere (discreti o continui).

Questa tecnica si basa sul presupposto che all'interno di un sistema si possano predisporre:

- un meccanismo logico A in grado di scegliere degli output sulla base degli input ricevuti.
- un meccanismo logico B in grado di valutare l'efficacia degli output rispetto ad un preciso parametro di riferimento.
- un meccanismo logico C capace di cambiare il meccanismo A per massimizzare la valutazione di efficacia effettuata da B.

Il modo in cui questi meccanismi dovrebbero collaborare è descritto dai seguenti punti:

- se il meccanismo A effettua una scelta efficace allora il meccanismo B manda in output un premio proporzionale all'efficacia della scelta di A.
- se il meccanismo A effettua una scelta inefficace allora il meccanismo B manda in output una penalità proporzionale all'inefficacia della scelta di A.
- il meccanismo C, osservando l'agire di A e B, cerca di modificare la funzione matematica che regola il comportamento di A in modo da massimizzare la quantità e la qualità dei "premi".

I meccanismi B e C sono quelli che vanno a costituire il metodo di rinforzo proprio di questa metodica di apprendimento.

Per attuare i meccanismi ed i comportamenti descritti nelle righe precedenti, dal punto di vista logico, si necessita delle seguenti componenti:

-Insieme di Input: rappresenta i possibili input che il sistema può ricevere (servono per determinare lo stato del sistema).

Gli input al sistema possono provenire dai più svariati sensori.

Ad esempio, nel caso di un robot che deve imparare a muoversi all'interno di un percorso, gli input potrebbero essere forniti da dei sensori di prossimità che dovrebbero essere poi rimappati in opportuni stati che nel caso dell'esempio potrebbero essere "ostacolo di fronte", "strada libera", "muro sul lato" ecc…

Per mappare i valori dei sensori a particolari stati si sono rivelate particolarmente efficaci le tecniche basate su controllori fuzzy.

-Funzione valore di stato: questa funzione associa un parametro di valutazione ad ogni stato del sistema.

La funzione valore di stato, in particolare, è quella che ad ogni stato identificato dal sistema e determinato sulla base degli input, associa un valore relativo al grado di bontà della situazione.

-Funzione valore di azione: questa funzione associa un parametro di valutazione ad ogni possibile coppia stato-azione.

-Tecnica di rinforzo: consiste in una funzione di rinforzo che, a seconda delle prestazioni attuali e dell'esperienza passata, fornisce delle direttive con cui cambiare la funzione di valore di stato e la funzione di valore d'azione.

Dal punto di vista modellistico tutte le funzioni di rinforzo possono essere ricondotte alla seguente formula base:

$$v_{t+1} = (1-\alpha)v_t(s) + \alpha\Delta_{t+1}$$

dove $0 < \alpha \leq 1$

Δ_{t+1} è il "premio" o la "penalità" che è stata associata alla corrente azione da parte della funzione di azione.

Questa funzione, come si può intuire dalla formula, altera la funzione di valore di stato a partire dal prossimo istante in cui verrà richiamata e in base alla valutazione dell'azione corrente effettuata dalla politica di premio (o di penalità).
Le più diffuse politiche di premio (o di penalità) sono: rinforzo con premio ad orizzonte infinito, dove il rinforzo ha sempre la stessa intensità ed è valutato per tutti gli istanti temporali; rinforzo con premio ad orizzonte finito, in cui il rinforzo ha sempre la stessa intensità ed è valutato per un periodo di tempo limitato; rinforzo con premio medio, nel quale il rinforzo ha intensità via via decrescente ma viene valutato per tutti gli istanti temporali (in pratica man mano che il tempo passa, i valori di rinforzo vengono attenuati dando più importanza alle valutazioni effettuate negli istanti iniziali); rinforzo con premio scontato, dove il rinforzo è distribuito per tutti gli istanti temporali ma aumenta a seconda di un parametro legato agli istanti temporali in cui viene applicato.
-Insieme di Output: rappresenta le possibili decisioni che il sistema può intraprendere. La scelta è effettuata in modo da massimizzare il valore della funzione di valore di azione ed è strettamente dipendente dal rinforzo distribuito durante gli istanti passati.

L'apprendimento per rinforzo è in grado di prelevare vantaggi provenienti sia dall'apprendimento supervisionato che dall'apprendimento non supervisionato.
Come nell'apprendimento non supervisionato, nel RF la macchina non è legata a una tabella di esempi con input e output scritti da un progettista.
Quindi, è meno legata al contenuto del training set e può prendere le decisioni con meno vincoli e un maggiore grado di libertà.
Tuttavia, a differenza dell'apprendimento non supervisionato, l'agente non inizia il processo di apprendimento senza conoscenza pregressa.
Nel reinforcement learning la macchina può distinguere fin da subito le azioni positive e negative tramite una funzione di rinforzo.
Come nell'apprendimento supervisionato, nel RF, l'agente è aiutato nel processo di apprendimento.
Tuttavia, i feedback non sono etichette aggiunte da un supervisore negli esempi dell'insieme di training ma una funzione matematica di rafforzamento.

Pertanto, a differenza dell'apprendimento supervisionato, nel RF, la macchina è in grado di valutare anche situazioni non previste inizialmente dal progettista.

Uno degli algoritmi di apprendimento con rinforzo più diffusi ed utilizzati è il Q-learning. Fa parte della famiglia di algoritmi adottati nelle tecniche delle differenze temporali, adottate nel caso di modelli a informazione incompleta.

Il suo obiettivo è quello di permettere ad un sistema di apprendimento automatico di adattarsi all'ambiente che lo circonda migliorando la scelta delle azioni da eseguire

Il modello del problema può essere descritto da un agente, un insieme di stati S e un insieme di azione per stato A.

Effettuando un'azione $a \in A$ l'agente si muove da uno stato ad un altro stato.

Ogni stato fornisce all'agente una ricompensa (un numero reale o naturale).

L'obiettivo dell'agente è quello di massimizzare la ricompensa totale.

L'agente fa questo apprendendo quali sono le azioni ottimali associate ad ogni stato.

Quindi l'algoritmo è provvisto di una funzione per calcolare la Qualità di una certa coppia stato-azione: $Q: S\ X\ A \rightarrow R$.

Prima che l'apprendimento inizi, Q restituisce un valore fisso, scelto dal progettista. Poi, ogni volta che l'agente riceve una ricompensa (lo stato è cambiato) vengono calcolati nuovi valori per ogni combinazione stato-azione.

Il cuore dell'algoritmo fa uso di un processo iterativo di aggiornamento e correzione basato sulla nuova informazione.

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{vecchio valore}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{tasso di apprendimento}} \times \left[\overbrace{\underbrace{r_t}_{\text{ricompensa}} + \underbrace{\gamma}_{\text{fattore di sconto}} \underbrace{\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})}_{\text{valore futuro massimo}}}^{\text{valore appreso}} - \underbrace{Q(s_t, a_t)}_{\text{vecchio valore}} \right]$$

R_{t+1} è una ricompensa osservata dopo aver eseguito a_t in s_t e il tasso di apprendimento (o learning rate) è identificato da $\alpha_t(s, a)$ $(0 < \alpha \leq 1)$.

Il fattore di sconto γ è tale che $0 < \gamma \leq 1$.

Un episodio dell'algoritmo termina quando lo stato s_{t+1} è uno stato finale.

Il tasso di apprendimento determina con quale estensione le nuove informazioni acquisite sovrascriveranno le vecchie informazioni.

Un fattore 0 impedirebbe all'agente di apprendere, al contrario un fattore pari ad 1 farebbe sì che l'agente si interessi solo delle informazioni recenti.

Il fattore di sconto determina l'importanza delle ricompense future.
Un fattore pari a 0 renderà l'agente "opportunista" facendo sì che consideri solo le ricompense attuali, mentre un fattore tendente ad 1 renderà l'agente attento anche alle ricompense che riceverà in un futuro a lungo termine.

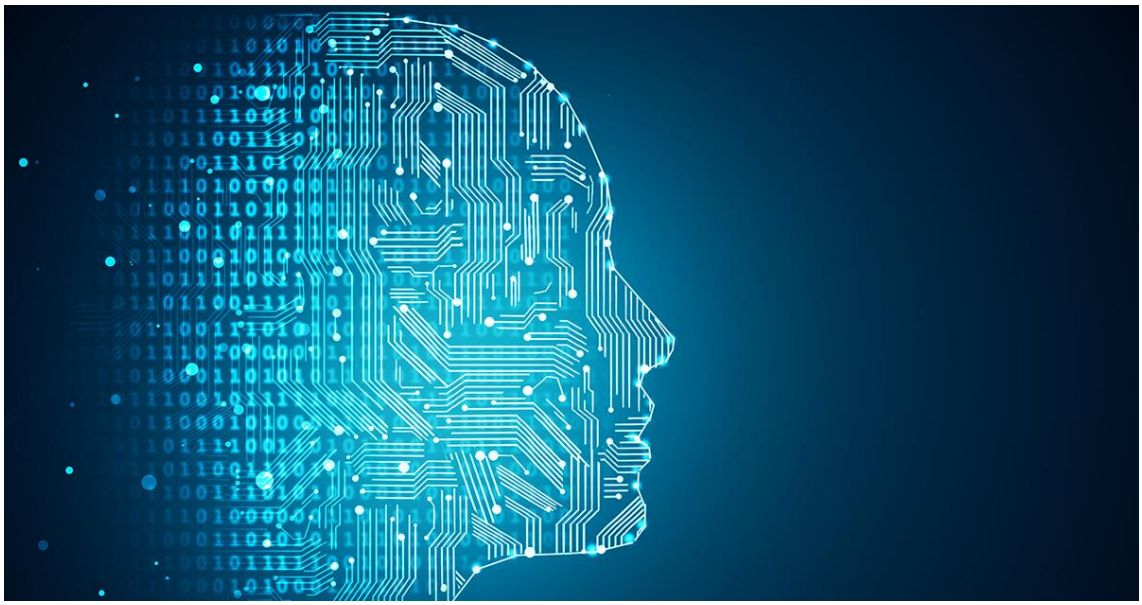
Il **modello semi-supervisionato** è un sistema di apprendimento con punti dati sia “non labaled” (non etichettati) sia etichettati.
L’apprendimento semi supervisionato si snoda tra l’apprendimento supervisionato e quello non supervisionato.
I modelli semi-supervisionati mirano a utilizzare una piccola quantità di dati di allenamento etichettati insieme a una grande quantità di dati di allenamento senza etichetta.
Ciò si verifica spesso in situazioni reali in cui l’etichettatura dei dati è molto costosa e/o si dispone di un flusso costante di dati.
Ad esempio, se stessimo cercando di rilevare messaggi inappropriati in un social network, non c’è modo di ottenere informazioni etichettate a mano su ogni messaggio, poiché ce ne sono semplicemente troppe e sarebbe troppo costoso.
Possiamo, invece, etichettare a mano un sottoinsieme di essi e sfruttare le tecniche semi-supervisionate per utilizzare questo piccolo set di dati etichettati per aiutarci a comprendere il resto del contenuto dei messaggi appena arrivano.
Alcuni metodi semi-supervisionati comuni sono le macchine vettoriali di supporto trasversali e i metodi basati su grafici, ad esempio la propagazione delle etichette.
I metodi semi-supervisionati devono fare alcune ipotesi (presupposti) sui dati al fine di giustificare l’utilizzo di una piccola serie di dati etichettati per trarre conclusioni sui punti dati non etichettati.
Questi possono essere raggruppati in tre categorie.
La prima categoria riguarda il presupposto di continuità: si presume che i punti dati “vicini” tra loro abbiano maggiori probabilità di avere un’etichetta comune.

Il secondo presupposto è l'ipotesi del cluster: si presume che i dati formino naturalmente cluster discreti e che i punti nello stesso cluster abbiano maggiori probabilità di condividere un'etichetta.

La terza categoria riguarda il presupposto molteplice: si presume che i dati si trovino approssimativamente in uno spazio di dimensioni inferiori (o collettore) rispetto allo spazio di input.

Questo scenario è rilevante quando un sistema non osservabile o difficile da osservare con un numero ridotto di parametri produce output osservabile ad alta dimensione.

Applicazioni reali



Dopo aver analizzato nel dettaglio il machine learning, ci si potrebbe chiedere da chi può essere utilizzato il machine learning e dove lo ritroviamo nella società odierna.

Molti settori che lavorano con grandi volumi di dati hanno riconosciuto il valore di tale tecnologia.

Raccogliendo informazioni dai dati, anche in tempo reale, le organizzazioni sono in grado di lavorare con più efficienza e acquisire un vantaggio competitivo.

Banche e altre aziende nell'industria finanziaria utilizzano le tecnologie di machine learning con due principali scopi: identificare le informazioni importanti nei dati e prevenire le frodi.

Le informazioni possono identificare opportunità d'investimento e aiutare gli investitori a sapere quando agire.

Il machine learning sta crescendo velocemente anche nell'industria dell'assistenza sanitaria, grazie all'avvento dei dispositivi indossabili e ai sensori che utilizzano i dati per verificare in tempo reale lo stato di salute di un paziente.

Questa tecnologia, inoltre, aiuta i consulenti medici nell'analisi, identificando tendenze o i segnali d'allarme che potrebbero condurre a diagnosi e a migliori trattamenti farmacologici.

L'analisi dei dati al fine di identificare schemi e tendenze è fondamentale nell'industria dei trasporti che, per incrementare il profitto, fa affidamento sulla creazione di rotte più efficienti e sulla previsione dei potenziali problemi.

Gli strumenti presenti nel machine learning per l'analisi dei dati e la creazione di modelli sono utili alle società di consegne, ai trasporti pubblici e alle altre ditte di trasporto.

Altri tipi di applicazioni

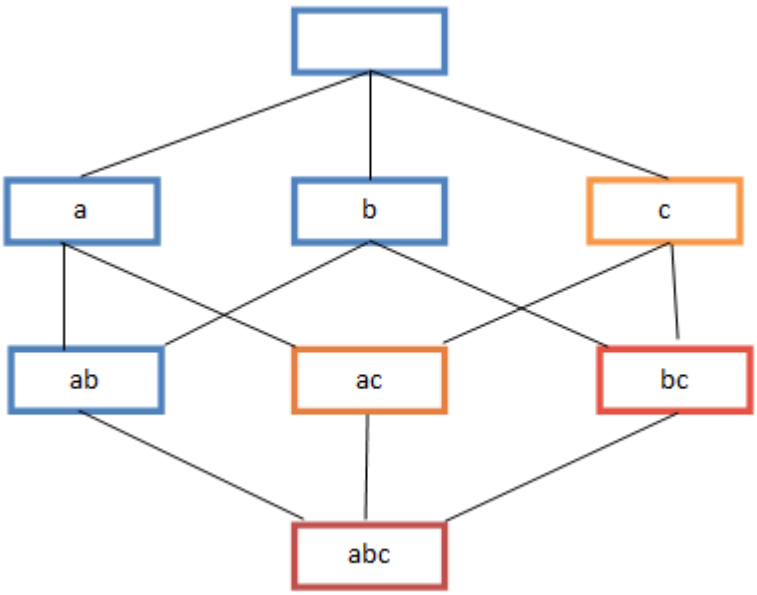
Un’altra applicazione classica di machine learning, ad esempio, è quella del riconoscimento vocale di cui sono dotati molti smartphone e che permettono di attivare comandi tramite la propria voce.

Ancora, molto comuni sono gli strumenti intelligenti che fanno uso di riconoscimento vocale per le diverse applicazioni di domotica, e che imparano nuovi vocaboli o modi di dire seguendo i comandi vocali che vengono impartiti.

Un altro utilizzo dell'apprendimento automatico legato al comune utilizzo dei computer e della rete, ad esempio, è quello che permette alle aziende di realizzare pubblicità traccianti.

Questo significa che, a seconda dell'utente di internet, vengono effettuate proposte pubblicitarie strettamente collegate agli interessi dell'utente stesso, le cui necessità e gusti vengono riconosciuti tramite l'analisi delle ricerche maggiormente effettuate in rete.

2.2. Association Rules



Le Association Rules sono un metodo di apprendimento automatico basato su regole per scoprire relazioni interessanti tra variabili in grandi database.

Ha lo scopo di identificare regole forti scoperte nei database usando alcune misure d' interesse.

Rakesh Agrawal, Tomasz Imieliński and Arun Swami introdussero le regole di associazione per la scoperta di regolarità all'interno delle transazioni registrate nelle vendite dei supermercati.

Per esempio, la regola $\{cipolla, patate\} => \{hamburger\}$ individuata nell'analisi degli scontrini di un supermercato indica che se il cliente compra insieme cipolle e patate è probabile che acquisti anche della carne per hamburger.
Tale informazione può essere utilizzata come base per le decisioni riguardanti le attività di marketing, come ad esempio le offerte promozionali o il posizionamento dei prodotti negli scaffali.
Le regole di associazione sono anche usate in molte altre aree, quali il Web mining, la scoperta di anomalie e la bioinformatica.
Una cosa molto importante, da constatare e sottolineare, in questo ambito, è che:

"Le regole non estraggono le preferenze di un individuo, piuttosto trovano relazioni tra un insieme di elementi di ogni transazione distinta. Questo è ciò che li rende diversi dal filtraggio collaborativo".

Per filtraggio collaborativo si intende una classe di strumenti e meccanismi che consentono il recupero di informazioni predittive relativamente agli interessi di un insieme dato di utenti a partire da una massa ampia e tuttavia indifferenziata di conoscenza.
Per comprendere meglio questo concetto, si può affermare che le regole non legano nel tempo le diverse transazioni degli utenti per identificare le relazioni.
Un elenco di elementi, che possiedono lo stesso ID transazione, è studiato come un unico gruppo.
Dall'altro lato, il filtraggio collaborativo lega tutte le transazioni corrispondenti a un ID utente per identificare la somiglianza tra le preferenze degli utenti.
Il primo è utile per il piazzamento dei prodotti sugli scaffali, mentre il secondo per consigliare articoli su siti Web di e-commerce e raccomandare canzoni su Spotify.

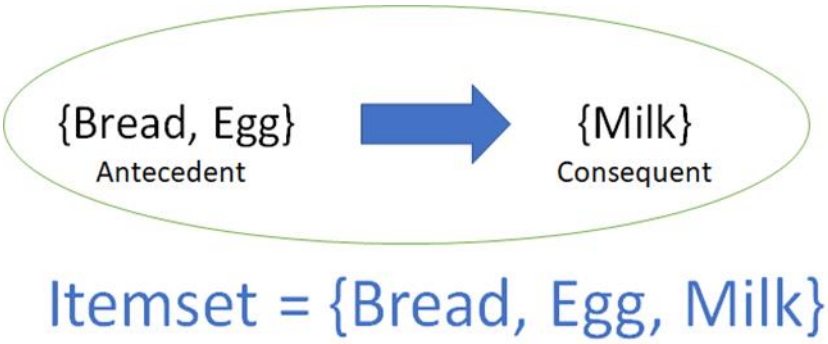
Association rules mining, a livello di base, prevede l'uso di modelli di apprendimento automatico per analizzare i dati per tipo o ricorrenza in un database.
Identifica le frequenti associazioni if-then, che sono chiamate regole di associazione.

Un' association rule è suddivisa in due parti: una parte antecedente (if) e una conseguente (then).

Si parla di antecedente quando un elemento è trovato all'interno dei dati, mentre si dice conseguente se è un elemento trovato in combinazione con l'antecedente.

Entrambe le parti consistono in una lista di elementi.

Per una determinata regola, l'itemset è l'elenco di tutti gli elementi nell'antecedente e nel conseguente.



Per riassumere, supponiamo di considerare l'insieme di n *attributi* binari (oggetti o item) $I = \{i_1, i_2 \dots, i_n\}$ e l'insieme di transazioni (database) $D = \{t_1, t_2, \dots, t_m\}$.

Ciascuna transazione appartenente a D possiede un codice identificativo (ID) e contiene un sottoinsieme degli oggetti contenuti in I.

Una regola è definita come un'implicazione nella forma $X => Y$, dove $X, Y \subseteq I$ e $X \cap Y = \emptyset$.

L'insieme di oggetti (o itemsets) X e Y vengono chiamati rispettivamente antecendente e conseguente della regola.

Per illustrare questo concetto, è possibile usare un esempio giocattolo riguardante un supermercato.

L'insieme di oggetti è $I = \{latte, pane, burro, birra\}$ e il database contenente gli oggetti è rappresentato nella tabella sottostante, dove 1 indica la presenza di un oggetto in una transazione e 0 l'assenza.

ID	latte	pane	burro	birra
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Un esempio di regola di associazione potrebbe essere $\{burro, pane\} => \{latte\}$.
Essa indica che se il cliente acquista pane e burro, comprerà anche il latte.

Sono disponibili vari parametri che aiutano a comprendere la forza di un'associazione:
-support
-confidence
-lift

Support: parametro che dà un'idea della frequenza con cui un insieme di elementi è presente in tutte le transazioni.
Siano X e Y gli itemset, $X => Y$ un'association rule, matematicamente il support è:

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

Se un set di elementi ha un support molto basso, non abbiamo abbastanza informazioni sulla relazione tra i suoi elementi e quindi non è possibile trarre conclusioni da tale regola.

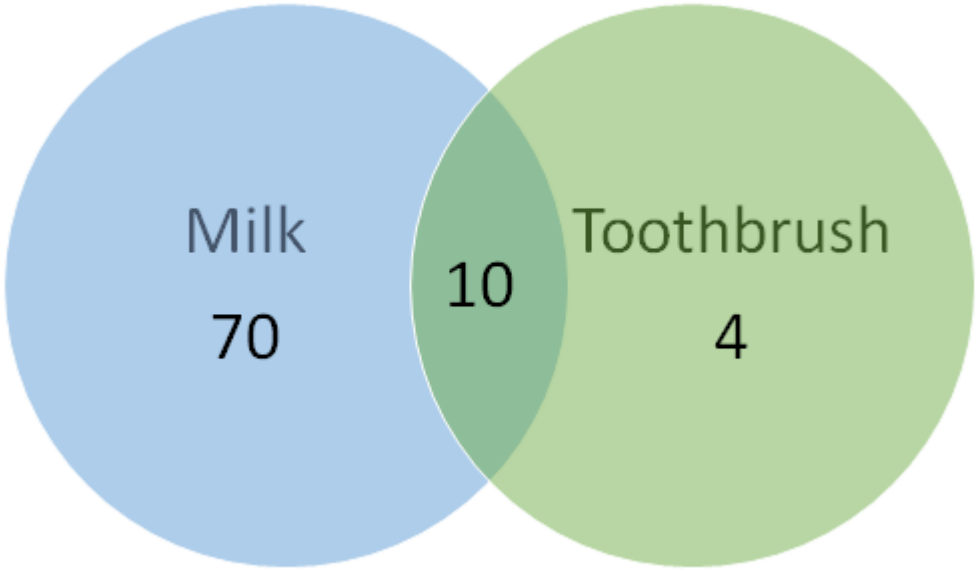
Confidence: è un'indicazione di quanto spesso la regola si è dimostrata vera.
In dettaglio:

$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

Lift: è definito come il rapporto tra il support osservato e quello atteso se X e Y fossero indipendenti.
In particolare:

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

Risulta essere molto interessante la relazione che lega la confidence con il lift e per mostrare ciò si illustra di seguito un semplice esempio.



La figura ci rivela come la confidence per {Toothbrush} → {Milk} sia pari a 0.7, il quale sembrerebbe essere un valore elevato.

Ma sappiamo intuitivamente che questi due prodotti hanno un'associazione debole e che c'è qualcosa di fuorviante in questo alto valore di confidence.

E' qui che il parametro lift viene in aiuto.

Consideriamo ora la probabilità di avere milk senza saper nulla riguardo a toothbrush: essa sarebbe pari all'80%.

Questi numeri mostrano che toothbrush sul carrello riduce effettivamente la probabilità di avere milk sul carrello da 0,7 a 0,8 e che quindi il valore del lift sarà pari a: 0,7 / 0,8 = 0,87.

Un valore di lift inferiore a 1 indica che avere toothbrush sul carrello non aumenta le possibilità di presenza di latte sul carrello nonostante la regola mostri un elevato valore di confidenza.

Un valore di lift superiore a 1 garantisce un'associazione elevata tra {Y} e {X}.

Lift è il parametro che aiuta i gestori dei negozi a decidere i posizionamenti dei prodotti sugli scaffali.

Le association rules sono in genere necessarie per soddisfare contemporaneamente un support minimo specificato dall'utente e una confidence minima specificata dall'utente.
La generazione delle association rules è generalmente suddivisa in due fasi separate:
-viene applicata una soglia di support minima per trovare tutti gli itemset frequenti in un database;
-viene applicato un limite minimo di confidence a questi itemset in modo da poter creare regole.
Mentre il secondo passo è semplice, il primo passo richiede più attenzione.

Vi sono vari algoritmi che permettono la generazione delle association rules: Eclat e FP-Growth, ma tra tutti il più importante è Apriori.

L'algoritmo Apriori è utilizzato per la generazione degli itemset frequenti, per approssimazioni successive, a partire dagli itemset con un solo elemento.
In sintesi, il presupposto teorico su cui si basa l'algoritmo parte dalla considerazione che se un insieme di oggetti (itemset) è frequente, allora anche tutti i suoi sottoinsiemi sono frequenti, ma se un itemset non è frequente, allora neanche gli insiemi che lo contengono sono frequenti (principio di anti-monotonicità).
Per ricavare le associazioni viene impiegato un approccio bottom up, dove i sottoinsiemi frequenti sono costruiti aggiungendo un item per volta (generazione dei candidati); i gruppi di candidati sono successivamente verificati sui dati e l'algoritmo termina quando non ci sono ulteriori estensioni possibili.
In questo processo, il numero delle iterazioni è $k_{max} + 1$ dove k_{max} indica la cardinalità massima di un itemset frequente.
Risulta poi possibile andare ad aumentare l'efficienza di tale algoritmo tramite degli accorgimenti:
-riducendo la dimensione della base di dati da considerare nei passaggi successivi;

-riducendo il numero di candidati da considerare, usando tecniche di indirizzamento e partizionamento;

-riducendo il numero di “scan” dell'intera base di dati;

-Transaction reduction: una transazione che non contiene nessun k-itemset frequente può essere trascurata nei passaggi successivi;

-Dynamic itemset counting: aggiungi un nuovo itemset candidato durante la scansione, sulla base dei dati analizzati fino a quel momento;

-campionamento.

I principali svantaggi, invece, che si riscontrano utilizzando tale algoritmo sono principalmente due:

-è molto esoso dal punto di vista della computazione.

Seppure riducendo il numero di candidati da considerare, il numero di questi è sempre molto grande quando il numero di elementi nei cestini della gente è alto o quando il valore limite di supporto è basso.

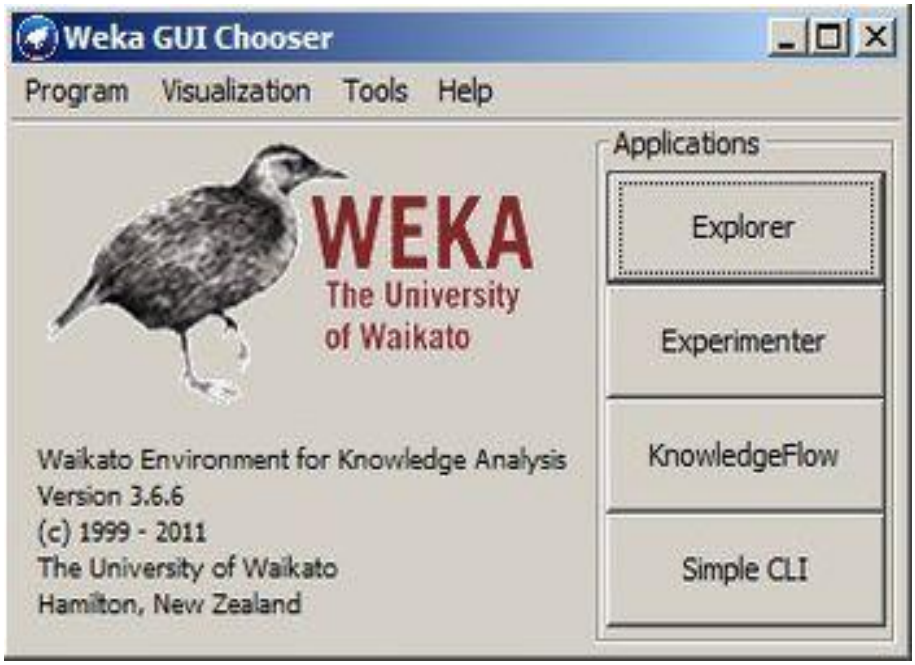
-Associazioni false.

Riducendo il valore limite di supporto per notare alcuni tipi di associazioni, può succedere che ci siano delle associazioni non giuste e quindi false.

Per ridurre questo problema occorre filtrare prima il Dataset o verificare il valore di supporto e confidenza in un Test Set separato.

3. Il software WEKA e le Association Rules in WEKA

3.1. Il software WEKA



WEKA, acronimo di "**W**aikato **E**nvironment for **K**nowledge **A**nalysis", è un software per l'apprendimento automatico (machine learning) sviluppato nell'università di Waikato in Nuova Zelanda. È open source e viene distribuito con licenza GNU General Public License. Curiosamente la sigla corrisponde al nome di un simpatico animale simile al Kiwi (vedi foto), presente solo nelle isole della Nuova Zelanda.

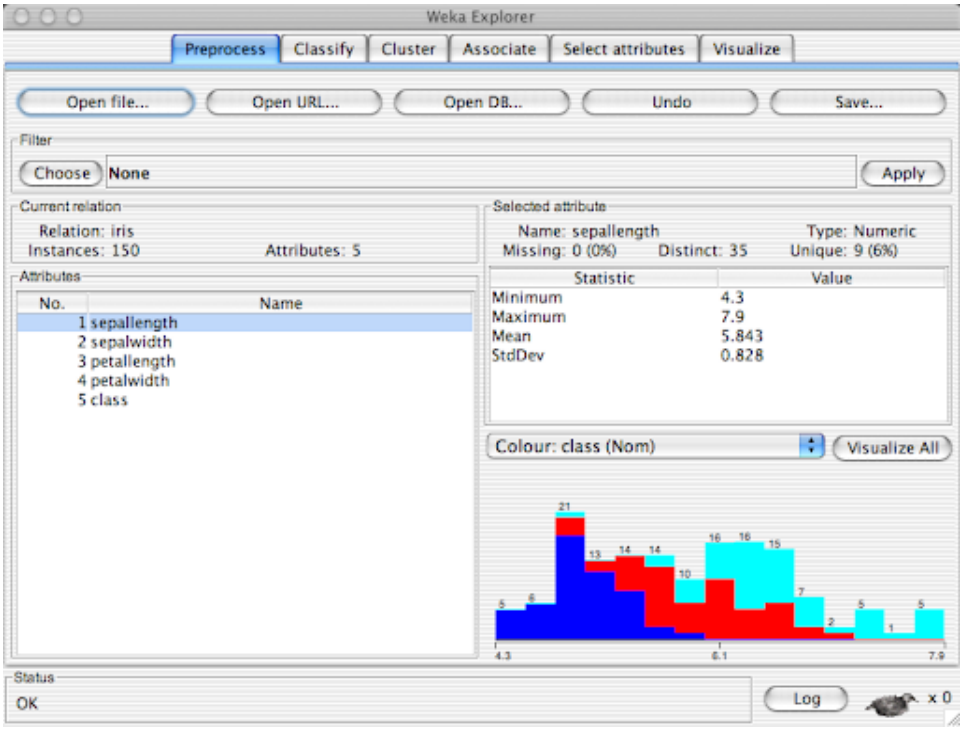
Weka è un ambiente software interamente scritto in Java. Un semplice metodo per utilizzare questo software consiste nell'applicare dei metodi di apprendimento automatici ad un set di dati (*dataset*), e analizzarne il risultato. Il **dataset** è l'insieme di valori e attributi presenti all'interno di una relazione. In una tabella di un database relazionale le istanze corrispondono alle righe e gli attributi alle colonne. Il formato utilizzato in Weka per la lettura dei dataset è l'ARFF(Attribute Relationship File Format), è simile al più famoso CSV (Comma-separated values) ed è equivalente alla tabella di un database relazionale.

È possibile attraverso questi metodi, avere quindi una previsione dei nuovi comportamenti dei dati.

L'interfaccia grafica di Weka è composta da quattro modalità di lavoro:

- **EXPLORER**: ambiente che consente di esplorare i dati attraverso i comandi Weka;
- **EXPERIMENTER**: compie test statistici fra i diversi algoritmi di **data mining**;
- **KNOWLEDGE FLOW**;
- **SIMPLE CLI**: l'interfaccia dalla riga di comando.

Esse possono essere utilizzate contemporaneamente e gestiscono in modo differente l'approccio con cui si affronta il lavoro sui dati di interesse.



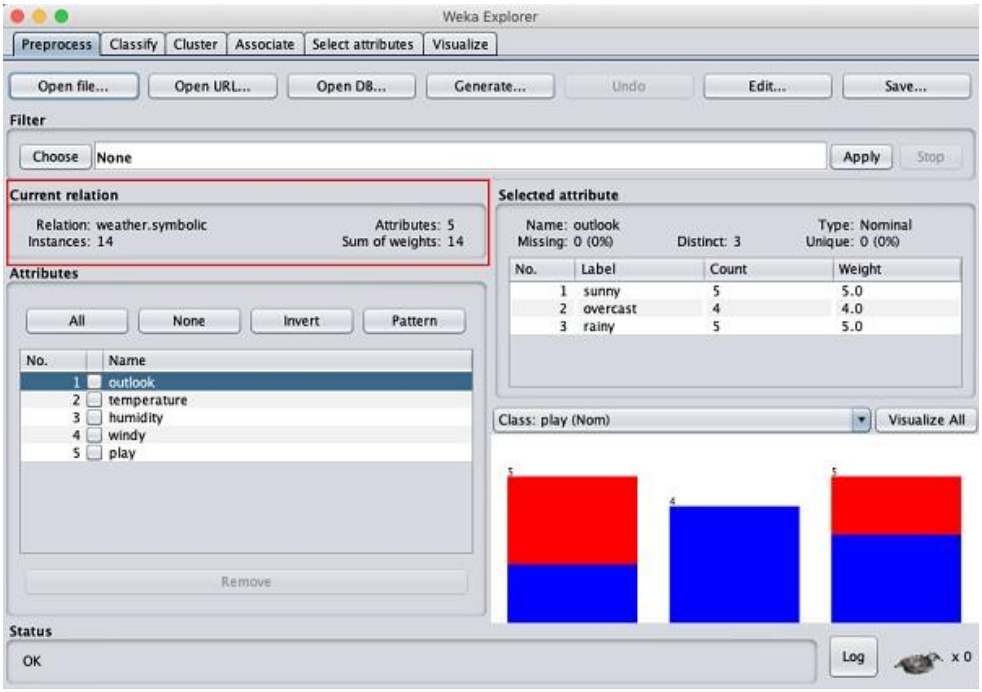
L'**EXPLORER** è diviso nei comandi:

- ***Preprocess*** permette di caricare i dati da una base dati o da un CSV e di applicare dei filtri ai dati;
- ***Classify*** applica algoritmi di classificazione e regressione;
- ***Cluster*** permette di usare tecniche di clustering;
- *Associate* cerca di estrarre delle Regole di associazione;
- ***Select attributes*** esegue degli algoritmi che permettono di valutare gli attributi in base alla loro utilità per la classificazione;
- ***Visualize*** visualizza un **Grafico di dispersione.**

Con esso si possono appunto caricare degli insiemi di dati, visualizzare in modo grafico la disposizione degli attributi, effettuare una serie di operazioni preliminari di preparazione, ed eseguire algoritmi di classificazione, clustering, selezione di attributi e determinazione di regole associative.

Per gli attributi nominali abbiamo l'elenco dei possibili valori e, per ognuno di essi, il numero di istanze con quel valore. Abbiamo anche il conteggio del numero di istanze in cui l'attributo manca e del numero di valori che appaiono una sola volta.

Per gli attributi numerici, abbiamo le informazioni sul valore massimo, minimo, sulla media e sulle deviazioni standard, oltre alle solite informazioni su numero di valori diversi, numero di valori unici e numero di istanze con valore mancante.



La scheda **Preprocess** serve per caricare il set di dati e applicare i filtri per trasformare i dati in una forma che esponga meglio la struttura del problema ai processi di modellazione. Fornisce anche alcune statistiche riassuntive sui dati caricati.

I primi quattro pulsanti nella parte superiore della sezione di Preprocess consentono di caricare i dati in WEKA:

- **Open file**: visualizza una finestra di dialogo che consente di cercare i file (preferibilmente in formato Csv o Arff) nel file system locale.
- **Open URL**: richiede un indirizzo URL dove i dati sono memorizzati.
- **Open DB**: legge i dati da un database.
- **Generate**: consente di generare dati artificiali da una varietà di DataGenerators.

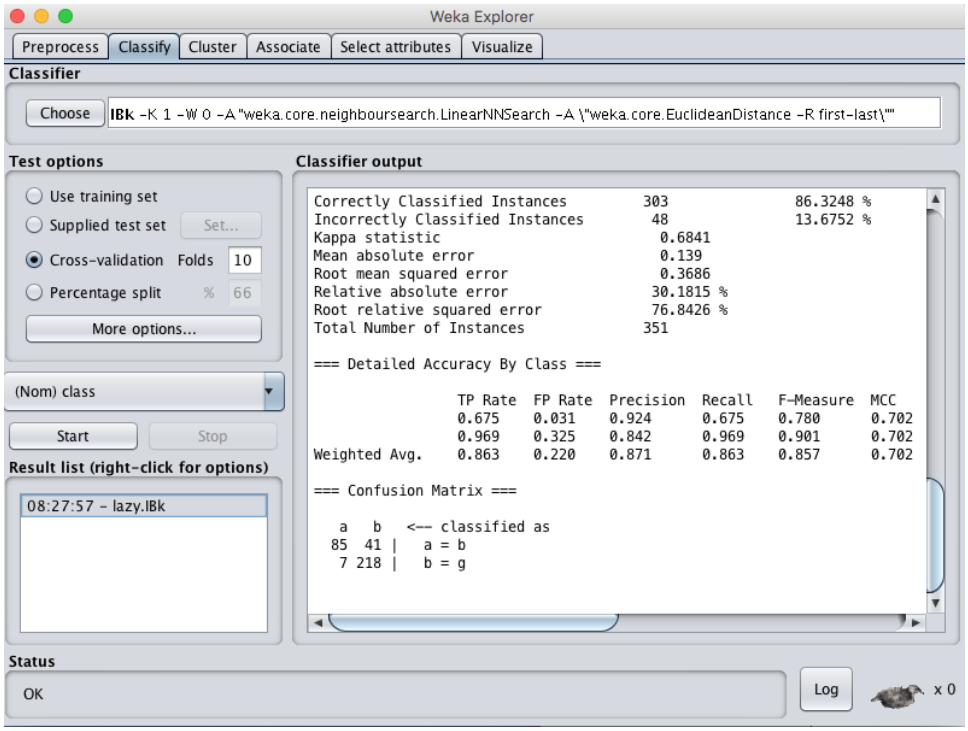
Tutti i valori possono essere modificati se ve ne è la necessità e possono essere nuovamente memorizzati in formato Arff, Csv o C45 in qualsiasi momento. Sotto a questi pulsanti, compare la voce **Filter**: essa permette di applicare filtri supervisionati e non

supervisionati ai dati caricati, per esempio è possibile riordinare attributi, aggiungere attributi con valori generati casualmente, ricampionare valori esistenti o rimuovere quelli che raggiungono una certa soglia. Un grafico a barre, riporta l'andamento dei valori per l'attributo scelto. Nella parte bassa di ogni finestra di Explorer (e relativamente alla sezione in cui ci si trova) è presente uno **status box**, un bottone di **log** e un disegno rimpicciolito di un animale weka: il primo mostra i messaggi su ciò che si sta elaborando, il secondo mostra, tramite doppio click con il mouse, informazioni sulle azioni che Weka ha eseguito nella sessione di lavoro corrente e il terzo si anima quando Weka è in attività e il numero di fianco al simbolo X indica quanti processi concorrenti sono in esecuzione nell'elaborazione corrente.

Una volta caricati i dati, il pannello Preprocess mostra una varietà di informazioni. La casella **Current relation** ha tre voci:

- **Relation**: Il nome della relazione, dato nel file da cui è stata caricata. I **filtri** modificano il nome di una relazione.
- **Instances**: Il numero di istanze nei dati.
- **Attributes**: Il numero di attributi nei dati.

Le rimanenti sezioni **Classify**, **Cluster**, **Associate** e **Select Attributes** sono tutte analoghe in termini di utilizzo: si seleziona tramite il pulsante in alto uno specifico algoritmo, poi l'insieme dei dati da elaborare nell'area "Cluster mode", successivamente premendo sul pulsante **Start** si dà inizio all'elaborazione e per ogni elaborazione si aggiorna la Result List che contiene tutte le elaborazioni fatte in ordine cronologico.



La scheda **Classify** serve per la formazione e la valutazione delle prestazioni di diversi algoritmi di apprendimento automatico sul problema di classificazione o regressione. Gli algoritmi sono suddivisi in gruppi, i risultati vengono mantenuti in un elenco di risultati e riepilogati nell'output principale del classificatore.

Nella parte superiore della sezione Classify si trova la casella **Classifier**. Questa casella ha un campo di testo che dà il nome del classificatore selezionato e le sue opzioni. Cliccando nella casella di testo con il tasto sinistro del mouse si apre una finestra di dialogo GenericObjectEditor che, esattamente come per i filtri, si può usare per configurare le opzioni del classificatore selezionato.

Il pulsante **Choose** consente di scegliere uno dei classificatori disponibili in WEKA.

Il risultato dell'applicazione del classificatore scelto verrà testato in base alle opzioni che vengono impostate cliccando nella casella **Test options**. Esistono quattro modalità di test:

- **Use training set**: il classificatore viene valutato in base a quanto bene prevede la classe delle istanze su cui è stato qualificato.

- **Supplied test set**: il classificatore viene valutato in base a quanto bene prevede la classe di un set di istanze caricate da un file. Cliccando sul pulsante **Set**, si apre una finestra di dialogo che consente di scegliere il file su cui testare.
- **Cross-validation: i**l classificatore viene valutato mediante cross validation, usando il numero di folds immesse nel campo di testo **Folds.**
- **Percentage split**: il classificatore viene valutato in base a quanto bene prevede una certa percentuale dei dati, che viene trattata per il test.

I classificatori in WEKA sono progettati per essere qualificati per prevedere una singola "class attribute", che è l'obiettivo della previsione. Alcuni classificatori possono apprendere solo classi nominali, altri possono apprendere solo classi numeriche (problemi di regressione) e altri ancora possono apprendere entrambe. Per definizione, la "class attribute" è considerata l'ultimo attributo nei dati.

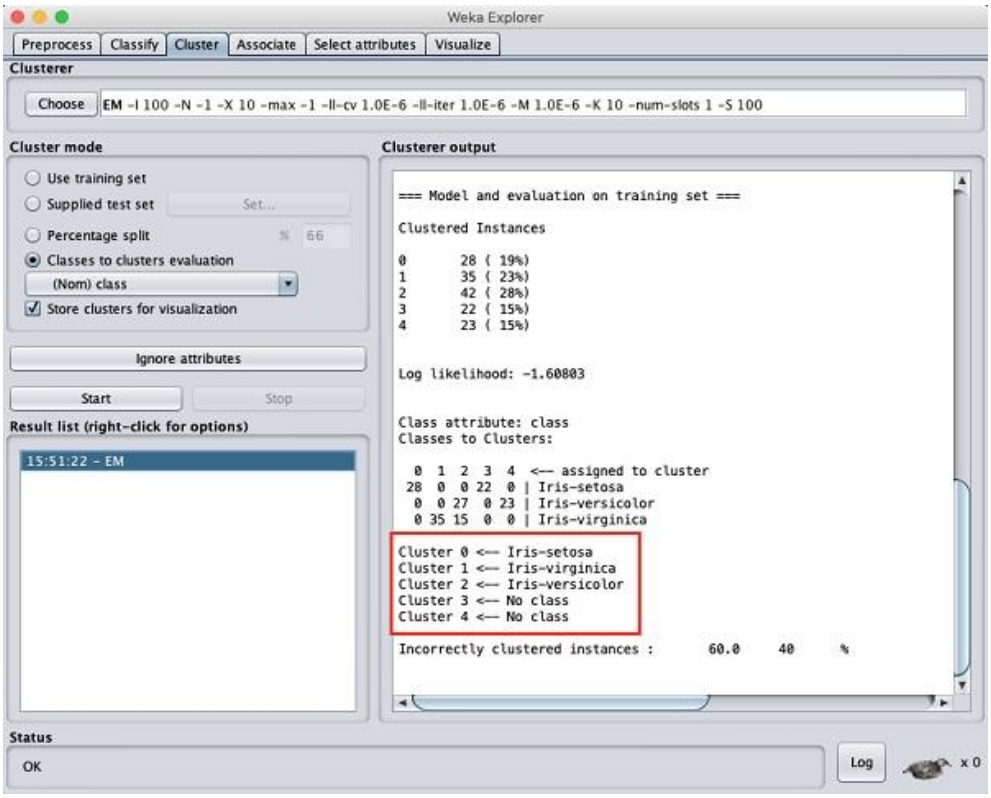
Una volta impostati classificatore, test options e classe, il processo di apprendimento viene avviato facendo clic sul pulsante **Start**. Si può interrompere il processo in qualsiasi momento, cliccando il pulsante **Stop**. Quando il processo è completato accadono diverse cose. L'area **Classifier output,** alla destra del display, è piena di un testo che descrive i risultati del processo. Viene visualizzata una nuova voce nella casella **Result list**.

Il testo nell'area Classifier output presenta barre di scorrimento che consentono di sfogliare i risultati. L'output è diviso in diverse sezioni:

- **Run information**: un elenco di informazioni che offrono opzioni dello schema di apprendimento, nome della relazione, istanze, attributi e modalità di test coinvolti nel processo.
- **Classifier model (full training set)**: una rappresentazione testuale del modello di classificazione che è stata prodotta sui dati di processo completi.
- **Summary**: un elenco di statistiche che riepiloga quanto il classificatore è stato capace di prevedere la vera classe delle istanze nella modalità di test scelta.
- **Detailed Accuracy By Class**: una suddivisione per classe più dettagliata dell'accuratezza della previsione del classificatore.

- **Confusion Matrix**: Mostra quante istanze sono state assegnate a ogni classe. Gli elementi mostrano il numero di esempi di test, la cui classe effettiva è la riga e la classe prevista è la colonna.

Dopo aver provato diversi classificatori, l'elenco dei risultati conterrà diverse voci.



La scheda **Cluster** serve per la formazione e la valutazione delle prestazioni di diversi algoritmi di clustering non supervisionati sul set di dati senza etichetta. Gli algoritmi sono divisi in gruppi, i risultati vengono mantenuti in un elenco di risultati e riepilogati nell'output principale.

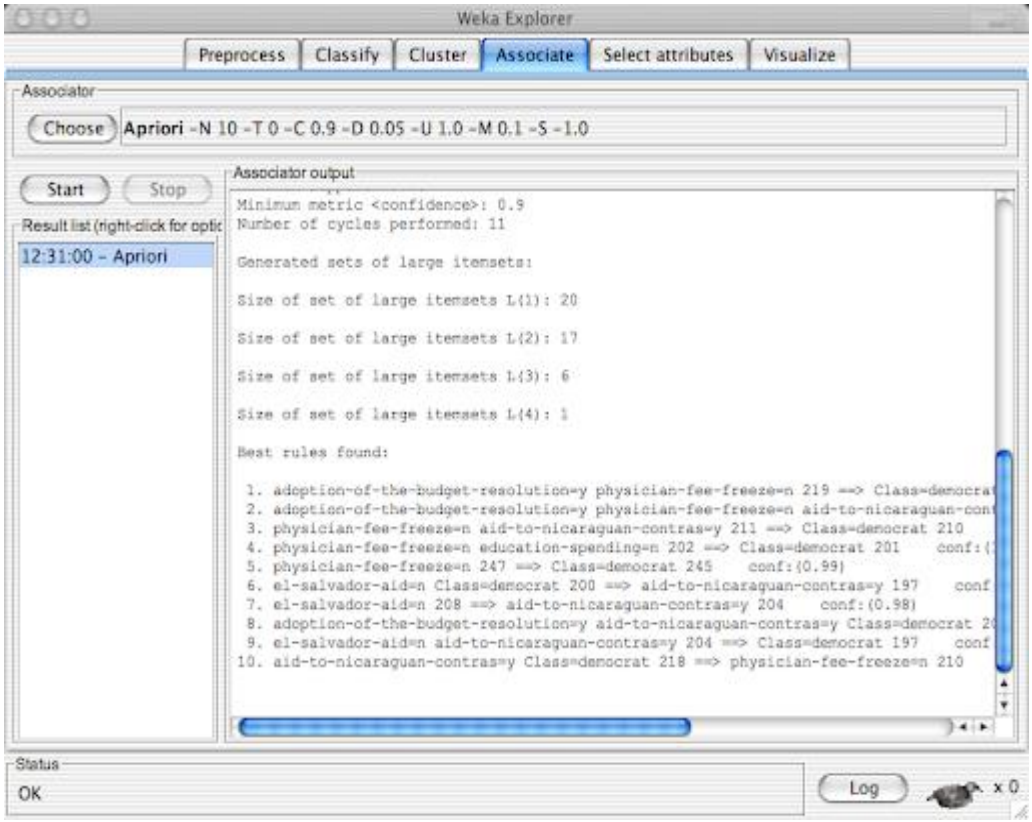
Cliccando sullo schema di clustering elencato nella casella **Clusterer** nella parte superiore della finestra, viene visualizzata una finestra di dialogo GenericObjectEditor con cui scegliere un nuovo schema di clustering.

La casella **Cluster mode** viene utilizzata per scegliere cosa raggruppare e come valutare i risultati. Le prime tre opzioni sono le stesse della classificazione: **Use training set**, **Supplied test set** and **Percentage split**, tranne per il fatto che ora i dati vengono assegnati ai cluster anziché cercare di prevedere una classe specifica. La quarta

modalità, **Classes to clusters evaluation**, confronta la corrispondenza tra i cluster scelti e una classe preassegnata nei dati. La casella a discesa sotto questa opzione seleziona la classe, proprio come nel pannello Classify. Un'opzione aggiuntiva nella casella **Cluster mode** è la casella di spunta **Store clusters for visualization**, che determina se sarà possibile visualizzare o meno i cluster una volta completato il processo. Quando si ha a che fare con set di dati così grandi che la memoria diventa un problema, può essere utile disabilitare questa opzione.

Spesso, alcuni attributi nei dati devono essere ignorati durante il clustering. Il pulsante **Ignore attributes** permette di visualizzare una piccola finestra che consente di selezionare gli attributi che si vogliono ignorare. Per annullare la selezione e tornare indietro bisogna cliccare il pulsante **Cancel**, per attivarla, bisogna cliccare il pulsante **Select**. Al successivo utilizzo del cluster, gli attributi selezionati vengono quindi ignorati.

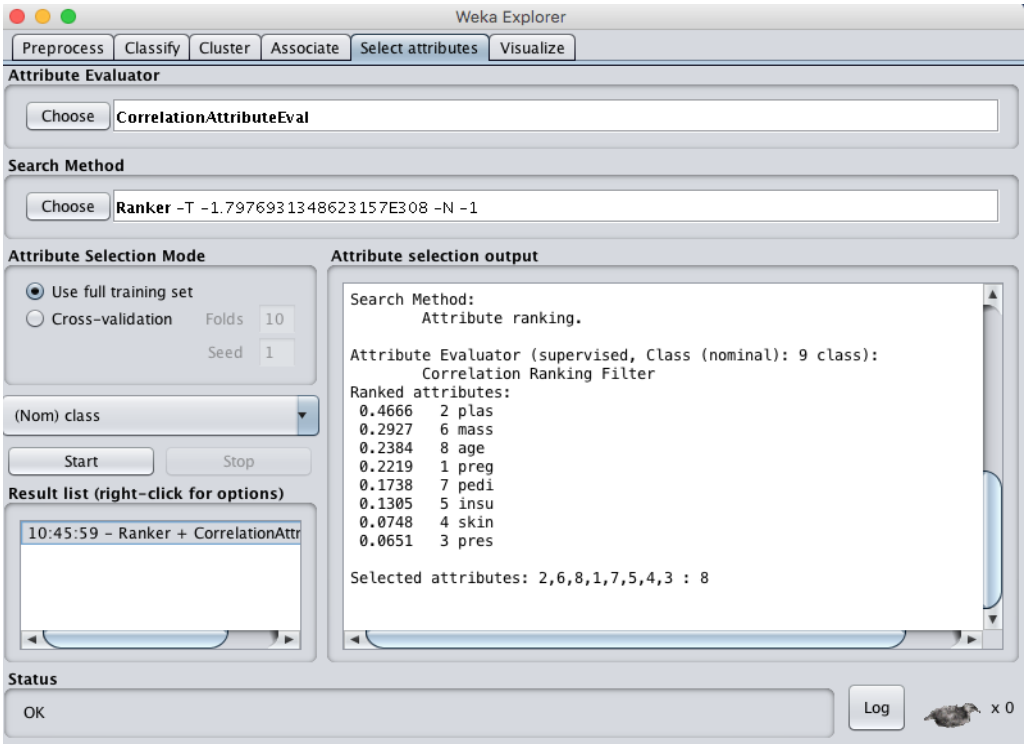
La sezione **Cluster**, come la sezione **Classify**, ha i pulsanti **Start/Stop**, una **result text area** e una **result list**. Tutti questi si comportano proprio come le loro controparti nella classificazione. Cliccando con il tasto destro del mouse su una voce nell'elenco dei risultati si apre un menu simile, eccetto che mostra solo due opzioni di visualizzazione: **Visualize cluster assignments** e **Visualize tree.** Quest'ultimo è disattivato quando non è applicabile.



La scheda **Associate** consente di trovare automaticamente le associazioni in un set di dati. Le tecniche sono spesso utilizzate per problemi di data mining di analisi del paniere di mercato e richiedono dati in cui tutti gli attributi sono categorie.

Questo pannello contiene schemi per l'apprendimento delle regole di associazione e i learners vengono scelti e configurati allo stesso modo dei clusterers, dei filtri e dei classificatori negli altri pannelli.

Una volta impostati i parametri appropriati per il learner della regola di associazione, bisogna cliccare il pulsante **Start**. Al termine, cliccando con il pulsante destro del mouse su una voce nell'elenco dei risultati, è possibile visualizzare o salvare i risultati.



La scheda **Select attributes** consente di eseguire la selezione delle funzioni sul set di dati caricato e di identificare quelle funzioni che sono più probabili essere rilevanti nello sviluppo di un modello predittivo.

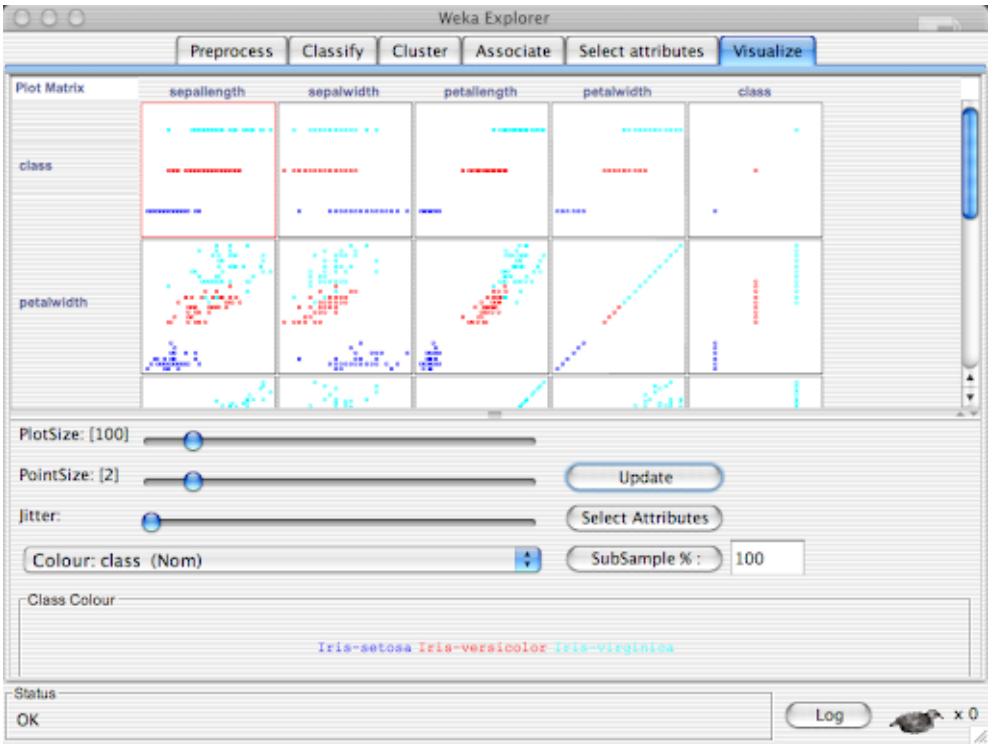
La selezione degli attributi implica la ricerca di tutte le possibili combinazioni di attributi nei dati per trovare quale sottoinsieme di attributi funziona meglio per la previsione. Per fare ciò, è necessario impostare due oggetti: un valutatore di attributi e un metodo di ricerca. Il valutatore determina quale metodo viene utilizzato per assegnare un valore a ciascun sottoinsieme di attributi. Il metodo di ricerca determina quale stile di ricerca viene eseguito.

La casella **Attribute Selection Mode** ha due opzioni:

- **Use full training set**: il valore del sottoinsieme di attributi viene determinato utilizzando l'intero set di dati di processo.
- **Cross-validation:** il valore del sottoinsieme di attributi è determinato da un processo di cross validation.

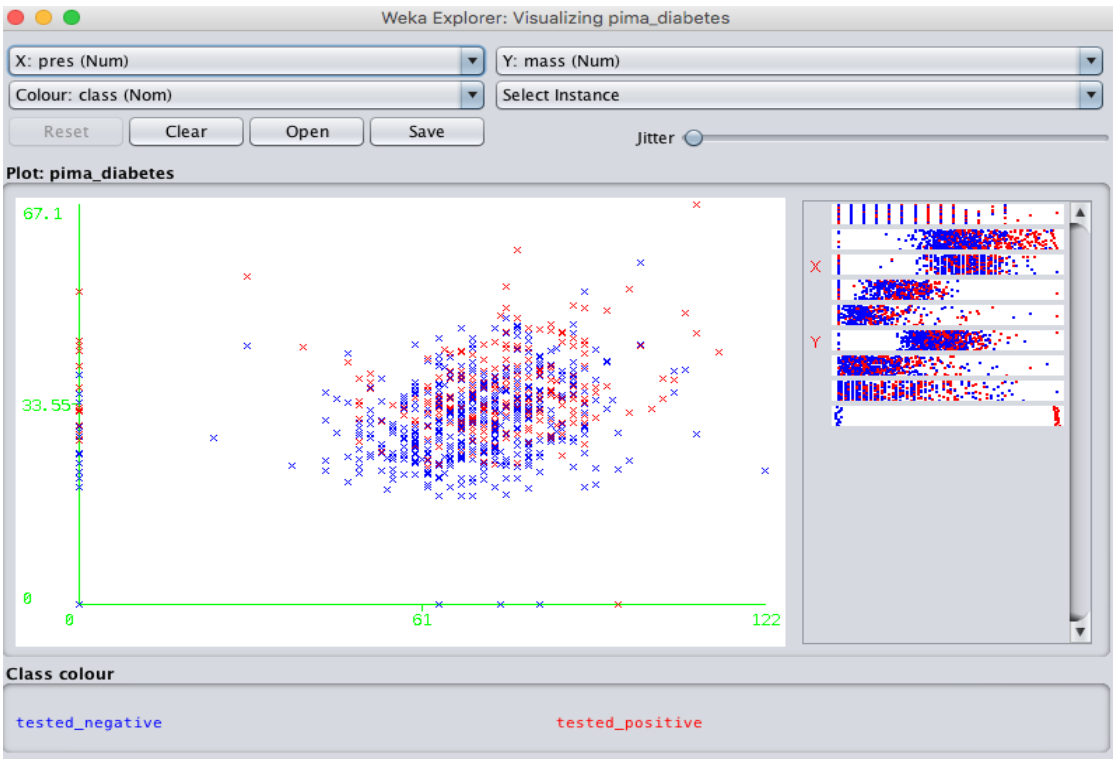
I campi **Fold** e **Seed** impostano il numero di folds da utilizzare e il seed casuale utilizzato quando si mescolano i dati.

Cliccando su **Start** si avvia il processo di selezione degli attributi. Al termine, i risultati vengono emessi nella **Result area** e una voce viene aggiunta alla **Result list**.



La scheda **Visualize** è per la revisione della matrice del diagramma a dispersione a coppie di ciascun attributo tracciato rispetto ad ogni altro attributo nel set di dati caricato.
Essa permette di avere una visione d'insieme di tutti grafici disponibili ottenuti con tutte le combinazioni degli attributi caricati da file. I grafici vengono visualizzati in piccolo come in un'anteprima, facendo doppio click con il mouse sul grafico scelto si apre una finestra separata che mostra il grafico nei dettagli: è possibile a questo punto intervenire cambiando gli attributi degli assi X e Y rispetto a quelli scelti inizialmente, scegliere di colorare i punti in base ai valori di un terzo attributo, selezionare zone del grafico da poter vedere nei dettagli facendo una sorta di "zoom".
Quando si seleziona il pannello Visualizza, viene visualizzata una matrice del diagramma a dispersione (scatter plot) per tutti gli attributi, codificata per colore in base alla classe attualmente selezionata.
Il **grafico di dispersione** (o grafico a dispersione o scatter plot o scatter graph) è un tipo di grafico in cui due variabili di un set di dati sono riportate su uno spazio cartesiano. I

dati sono visualizzati tramite una collezione di punti ciascuno con una posizione sull'asse orizzontale determinato da una variabile e sull'asse verticale determinato dall'altra.
Un grafico di dispersione è spesso usato quando una delle variabili è sotto controllo dello sperimentatore. Un parametro che è incrementato e/o decrementato sistematicamente è chiamato **parametro di controllo** o variabile indipendente, ed è arbitrariamente posto sull'asse orizzontale. La variabile **misurata** (o dipendente) è arbitrariamente posta sull'asse verticale. Se non esistono variabili dipendenti, ogni variabile può essere messa su un asse a piacere. Il grafico di dispersione può' essere utile per visualizzare il grado di correlazione (cioè di dipendenza lineare) tra le due variabili. Un grafico a dispersione può suggerire vari tipi di correlazione tra variabili con un certo intervallo di confidenza. Le correlazioni possono essere positive, negative o nulle.



Se il modello di punti sul grafico scende dall'alto a sinistra verso il basso a destra, suggerisce una correlazione negativa. Può essere disegnata una linea di andamento (o linea di trend) per studiare la correlazione tra le variabili in esame. Per una correlazione lineare, la migliore procedura (best-fit) è la regressione lineare (linear regression), e

garantisce di generare una soluzione corretta in un tempo finito. Sfortunatamente, non vi è una procedura universale che garantisca di generare una soluzione corretta per relazioni arbitrarie.

Un grafico di dispersione è molto utile anche quando vogliamo vedere quanto corrispondono due set di dati comparabili. Uno degli aspetti più interessanti dello scatter plot, tuttavia, è l'abilità di mostrare relazioni non lineari tra variabili. Inoltre, se i dati sono rappresentati da un modello misto di relazioni semplici, esse possono essere rese visibilmente evidenti come modelli sovrapposti. Il grafico di dispersione è uno degli strumenti basilari per il controllo della qualità.

Explorer è l'interfaccia che permette di accedere a tutti gli algoritmi presenti in Weka nell'intero processo di elaborazione dei dati, i **FILTRI** rappresentano la prima elaborazione per preparare i dati da elaborare e si impostano direttamente nella finestra **Preprocess**.

Un filtro effettua elaborazioni più o meno complesse sui dati di ingresso (così come si presentano al momento del caricamento in Weka) in base alle scelte fatte dall'utente. I filtri si distinguono in **supervisionati** e **non supervisionati**: nei supervisionati l'utente, attraverso i parametri impostati per l'algoritmo di filtraggio, imposta il grado di elaborazione e modifica da compiere sui dati in ingresso, i non supervisionati invece effettuano una elaborazione automatica senza fare distinzione alcuna dei dati che si trovano ad elaborare. All'interno di queste due categorie è possibile scegliere se applicare i filtri a livello di attributi o a livello di istanze: i primi operano su un singolo o più attributi selezionati, i secondi operano a livello di tuple prendendo in considerazione la totalità degli attributi. Una volta scelto il filtro, i parametri di configurazione si impostano in una finestra di dialogo che compare facendo doppio clic con il mouse sul nome del filtro stesso.

Filtri non supervisionati per gli attributi:

- **Add**: aggiunge un nuovo attributo a quelli esistenti con valori nulli.
- **AddCluster**: aggiunge un attributo con una etichetta che rappresenta il cluster assegnato a ognuna tupla in base a un algoritmo di cluster scelto dall'utente.

- **AddExpression**: crea un nuovo attributo con i valori risultanti da una funzione matematica basata sugli attributi già esistenti.
- **AddNoise**: cambia una certa percentuale di valori aggiungendo rumore.
- **ClusterMembership**: utilizza un algoritmo di clustering per generare valori appartenenti ai cluster trovati e che andranno a formare nuovi attributi.
- **Copy**: copia un intervallo di attributi nel dataset.
- **Discretize**: converte gli attributi numerici in etichette stringa.
- **FirstOrder**: applica l'operatore di differenza del primo ordine su un intervallo di valori.
- **MakeIndicator**: rimpiazza un attributo stringa con un attributo booleano.
- **MergeTwoValues**: fonde due valori per un attributo specificato.
- **NominalToBinary**: converte tutti gli attributi numerici dalla base dieci in base due.
- **NumerciTransform**: trasforma un attributo numerico utilizzando direttamente le funzioni Java.
- **Obfuscate**: "offusca" il dataset rinominando le relazioni, tutti gli attributi e il loro tipo.
- **PKIDiscretize**: discretizza attributi numerici.
- **RandomProjection**: elabora i dati tramite un sottospazio con dimensioni minori di quello di partenza e genera una matrice di valori casuali.
- **Remove**: rimuove gli attributi.
- **RemoveType**: rimuove gli attributi di un tipo specifico.
- **RemoveUseless** rimuove attributi costanti, insieme ad attributi stringa che variano troppo.
- **ReplaceMissingValues**: sostituisce i valori mancanti di attributi stringa o numerici con la moda e la media dei dati presenti.
- **Standardize**: standardizza tutti gli attributi numerici in modo che abbiano media zero e varianza unitaria.
- **StringToNominal**: converte un attributo stringa in etichetta.
- **StringToWordVector**: converte un attributo stringa in un vettore che rappresenta la frequenza di parole.

- **SwapValues**: scambia due valori di un attributo.
- **TimeSeriesDelta**: sostituisce i valori di attributi nella istanza (tupla) corrente con la differenza tra il valore corrente e il valore predetto analizzando altre tuple.
- **TimeSeriesTranslate**: sostituisce i valori di attributi nella tupla corrente con l'equivalente valore predetto analizzando altre tuple.

Filtri non supervisionati per le istanze:

- **NonSparseToSparse**: converte le istanze in formato "sparse" ovvero con valori zero per gli attributi mancanti.
- **Normalize**: considera gli attributi numerici come un vettore da normalizzare rispetto a una specifica lunghezza.
- **Randomize**: mescola in modo casuale l'ordine delle tuple in un dataset.
- **RemoveFolds**: riporta in output una specifica "fold" del dataset utilizzando la cross-validation.
- **RemoveMissclassified**: rimuove le istanze classificate come incorrette in base a uno specifico classificatore.
- **RemovePercentage**: rimuove una frazione del dataset espressa in percentuale.
- **RemoveRange**: rimuove un determinato intervallo di istanze da un dataset.
- **Resample**: produce un sottoinsieme di valori casuali del dataset originario.
- **SparseToNonSparse:** converte tutte le istanze in formato "sparse" in formato "nonsparse".

Filtri supervisionati per gli attributi:

- **AttributeSelection**: permette l'accesso alle funzioni di selezione di attributi così come nella sezione Select attributes di Explorer.
- **ClassOrder**: randomizza o altera in altro modo l'ordine dei valori di una classe selezionata.
- **Discretize**: converte gli attributi in formato binario, usando un metodo supervisionato se la classe è numerica.

Filtri supervisionati per le istanze:

- **Resample**: produce un sottoinsieme di valori casuali per un dataset, sostituendo i valori originari del dataset.

- **SpreadSubsample**: produce un sottoinsieme di valori casuali diffondendo i valori tra classi in base alla frequenza specificata.
- **StratifiedRemoveFolds**: crea una cross-validation per il dataset da aggiungere ai dati originari.

Algoritmi di apprendimento: si accede a questa tipologia di algoritmi nella sezione **Classify** di **Explorer** oppure in quella analoga di **Knowledge Flow** o in **Experimenter**. Esistono algoritmi di apprendimento supervisionato e non supervisionato dedicati alla classificazione dei dati nelle forme più varie: reti bayesiane, alberi di decisione, apprendimento di regole e funzioni matematiche per il calcolo di regressione, correlazione, ecc.

Algoritmi di classificazione attualmente presenti in WEKA:

Bayes:

- **AODE**: averaged, one-dependence estimators.
- **BayesNet**: apprendimento di reti bayesiane
- **ComplementNaiveBayes**: costruisce un classificatore bayesiano complementare.
- **NaiveBayes**: classificatore bayesiano probabilistico standard.
- **NaiveBayesMultinomial:** versione multinomiale del classificatore bayesiano.
- **NaiveBayesUpdateable:** classificatore bayesiano incrementale che apprende una istanza per volta.

Alberi di decisione:

- **ADTree**: costruisce un albero di decisione di tipo “alternating decision”.
- **DecisionStump**: costruisce un albero di decisione di primo livello.
- **ID3**: albero di decisione di tipo “divide-andconquer” di base.
- **J48**: albero di decisione basato sull’algoritmo C4.5.
- **LMT**: costruisce alberi di decisione logistici.
- **M5P**: albero di apprendimento basato sull’algoritmo M5.
- **NBTree**: costruisce un albero di decisione basandosi sul classificatore bayesiano.
- **RandomForest**: costruisce un albero seguendo la procedura di “Random forest”.
- **RandomTree**: costruisce un albero basandosi su un dato numero di caratteristiche scelte casualmente.

- **REPTree**: albero con algoritmo di apprendimento veloce che usa il pruning.
- **UserClassifier**: permette all'utente di costruire un albero in base alle proprie scelte.

Rules:

- **ConjunctiveRule**: semplice algoritmo di apprendimento di regole.
- **DecisionTable**: costruisce una tabella di decisione semplice.
- **JRip**: algoritmo RIPPER per regole a induzione.
- **M5Rules**: ottiene regole basandosi su alberi di decisione di tipo M5P.
- **Nnge**: genera regole in base al metodo "nearest-neighbor".
- **OneR**: classificatore 1R.
- **Part**: ottiene regole in base a porzioni di alberi creati con algoritmo J48.
- **Prism**: semplice algoritmo di copertura per generare regole.Ridor: Algoritmo di apprendimento di tipo "Ripple-down".
- **ZeroR**: predice i valori di maggior frequenza di una classe (se composta di etichette stringa) o il valore medio (se è composta da valori numerici).

Functions:

- **LastMedSq**: regressione utilizzando la mediana invece della media.
- **LinearRegression**: regressione lineare standard.
- **Logistic**: crea modelli logistici lineari.
- **MultilayerPerceptron**: rete neurale a retropropagazione.
- **PaceRegression**: crea modelli di regressione lineare usando il metodo Pace.
- **RBFNetwork**: implementa una funzione di rete neurale radiale.
- **SimpleLinearRegression**: apprendimento tramite modello a regressione lineare basato su un singolo attributo.
- **SimpleLogistic**: calcola la regressione lineare logistica in base alla selezione di attributi.
- **SMO**: algoritmo di ottimizzazione sequenziale minima per classificazione fatta tramite vettori.
- **SMOreg**: algoritmo di ottimizzazione sequenziale minima con supporto per vettori a regressione.
- **VotedPerceptron**: algoritmo del Percettrone.

- **Winnow**: algoritmo del Percettrone guidato dagli errori.

Lazy:

- **IB1**: algoritmo di apprendimento di base basato sul metodo “nearest-neighbor”.
- **IBk**: classificatore k-nearest-neighbor.
- **KStar**: algoritmo di tipo nearest-neighbor con funzione di distanza.
- **LBR**: classificatore bayesiano di tipo lazy.
- **LWL**: algoritmo per l’apprendimento di dati valutati localmente tramite “peso”.

Misc.:

- **Hyperpipes**: algoritmo di apprendimento veloce e molto semplice basato su ipervolumi nello spazio delle istanze.
- **VFI**: algoritmo del metodo dei voti.

Algoritmi di meta apprendimento: gli algoritmi di meta apprendimento costituiscono un sottoinsieme degli algoritmi di apprendimento classici, essi si applicano ai meta dati ovvero alle informazioni che descrivono i dati stessi: si basano su un insieme di regole che descrivono la struttura dei dati e si applicano a un determinato caso noto a priori. Sono:

- **AdaBoostM1**: algoritmo di boost che utiliza il metodo AdaBoostM1.
- **AdditiveRegression**: aumenta la performance di un metodo di regressione riempiendo iterativamente i dati mancanti.
- **AttributeSelectedClassifier**: riduce le dimensioni dei dati attraverso la selezione di attributi.
- **Bagging**: classificatore di tipo bagging, funziona anche per la regressione.
- **ClassificationViaRegression**: esegue la classificazione tramite un metodo di regressione.
- **CostSensitiveClassifier**: classificatore in base al costo.
- **CVParameterSelection**: seleziona parametri in base alla cross-validation.
- **Decorate**: utilizza un insieme di classificatori basandosi su esempi costruiti artificialmente.
- **FilteredClassifier**: esegue un classificatore sui dati filtrati.

- **Grading**: algoritmo che accetta in input dati di previsione di base che sono state precedentemente marcate come corrette o non corrette.
- **LogitBoost**: esegue la regressione logistica additiva.
- **MetaCost**: crea un classificatore sensibile al costo.
- **MultiBoostAB**: combina il metodo di boosting e bagging usando il metodo Multiboosting.
- **MultiClassClassifier**: usa un classificatore a due classi per i dataset a multiclasse.
- **MultiScheme**: utilizza la cross-validation per selezionare un classificatore da diversi candidati.
- **OrdinalClassClassifier**: applica gli algoritmi di classificazione standard a problemi con classi ordinate di valori.
- **RacedIncrementalLogitBoost:** algoritmo di apprendimento funzionante sul principio dell'elaborazione batch.
- **RandomCommittee**: crea un insieme di classificatore di base a caso.
- **Stacking**: combina diversi classificatori usando il metodo stacking.
- **StackingC**: versione più efficiente di Stacking.
- **ThresholdSelector**: ottimizza le f-misure di un classificatore probabilistico.
- **Vote**: combina diversi classificatori usando la media della probabilità stimata o le previsioni numeriche.

Algoritmi di clustering: algoritmi di classificazione mediante clustering. Sono:

- **EM**: clustering con algoritmo Expectation-Maximization.
- **Cobweb**: clustering con algoritmo Cobweb e Classit.
- **FarthestFirst**: clustering con algoritmo Farthest-first.
- **SimpleKMeans**: clustering con algoritmo delle k-medie standard.

Algoritmi per regole di associazione: questi algoritmi, in base a parametri decisi dall'utente, cercano associazioni non note a priori di elementi che ricorrono frequentemente nell'insieme di dati di input, infine riportano in output le associazioni trovate mostrando il valore di confidenza e supporto calcolato per ogni regola associativa corrispondente. Sono:

- **Apriori**: algoritmo Apriori per le regole.

- **PredictiveApriori**: algoritmo Apriori che trova regole di associazione ordinate per accuratezza nella predizione.
- **Tertius**: algoritmo a conferma guidata durante la scoperta di regole di associazione o classificazione.

Selezione e ricerca di attributi: questi algoritmi eseguono una serie di operazioni che riguardano la ricerca di attributi in base a parametri impostati dall'utente e anche la valutazione su singoli attributi o alle tuple stesse prese nella loro interezza. Tramite output e valutazioni numeriche viene stimato il grado di importanza e di utilità dei dati di input, permettendo così di prendere decisioni sull'effettiva validità dei dati disponibili. Il lavoro può consistere nell'eliminazione di attributi interi o di singoli valori di uno o più colonne di una tabella, oppure si possono operare correzioni nei valori già esistenti in base a parametri specificati dall'utente e possono essere creati nuovi attributi da aggiungere a quelli originari in base alle esigenze dell'utente.

Metodi per la selezione di attributi rilevanti:

Valutazione di più attributi contemporaneamente:

- **CfsSubsetEval**: considera il valore previsto di ogni attributo individualmente assieme al grado di ridondanza degli attributi stessi.
- **ClassifierSubsetEval**: usa un classificatore per valutare l'insieme di attributi.
- **ConsistencySubsetEval**: progetta il training set sul set di attributi e misura la consistenza in base ai valori assunti dalle classi.
- **WrapperSubsetEval**: usa un classificatore unito alla cross-validation.

Valutazione di un singolo attributo:

- **ChiSquaredAttributeEval**: calcola il valore di chiquadro per ogni attributo rispetto alla classe.
- **GainRatioAttributeEval**: valuta un attributo in base al rapporto di guadagno.
- **InfoGainAttributeEval**: valuta un attributo in base al guadagno di informazione.
- **OneRAttributeEval**: usa l'algoritmo OneR per valutare gli attributi.
- **PrincipalComponents**: analisi e trasformazione sui "principal components".
- **ReliefAttributeEval**: valutazione attributi a livello di istanze.

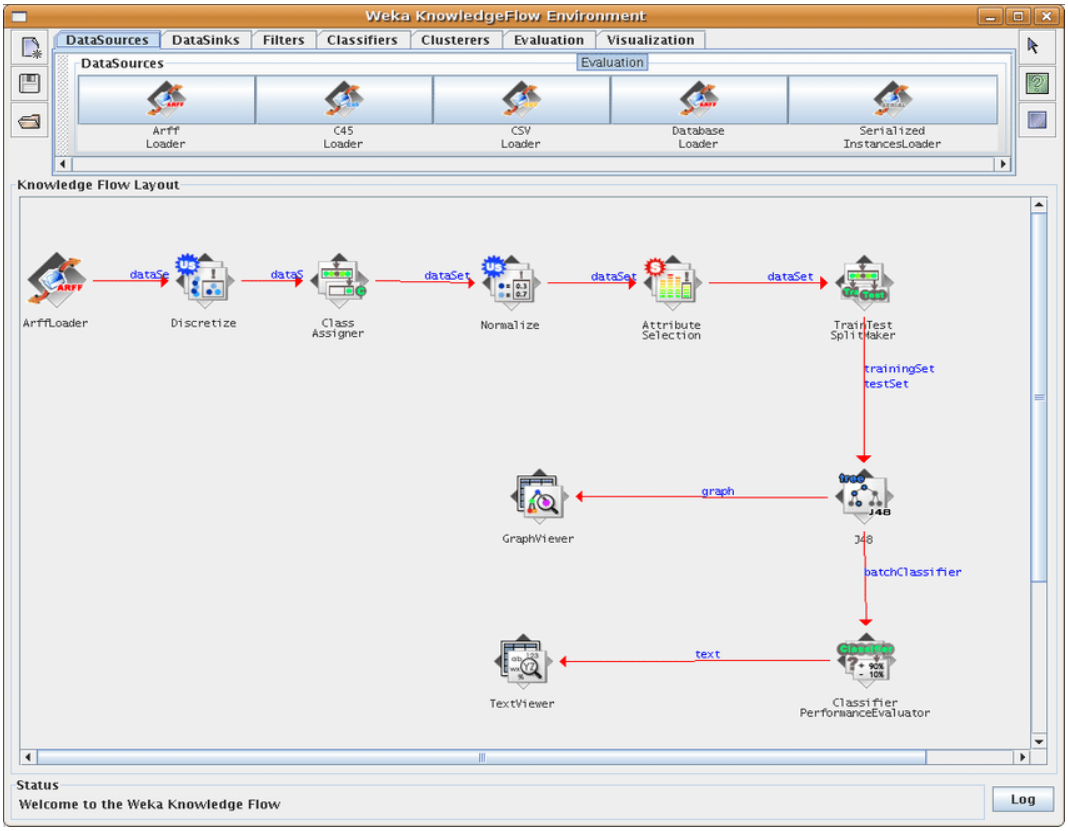
- **SVMAttributeEval**: usa il metodo “support vector machine” per determinare il valore degli attributi.
- **SymmetricalUncertAttributeEval**: valuta un attributo in base alla incertezza simmetrica.

Metodo di ricerca:

- **BestFirst**: metodo di ricerca di tipo “greedy” con tracciatura all’indietro.
- **ExhaustiveSearch**: ricerca esaustiva.
- **GeneticSearch**: ricerca utilizzando un algoritmo di base di ricerca genetica.
- **GreedyStepwise**: algoritmo di ricerca di tipo “greedy” senza tracciatura all’indietro.
- **RaceSearch**: utilizza la metodologia della ricerca “race”.
- **RandomSearch**: cerca in modo casuale.
- **RankSearch**: ordina gli attributi e li classifica usanto un algoritmo di valutazione per subset di attributi.

Metodo di classifica:

- **Ranker**: classifica attributi singoli (non sottoinsiemi) in base alla loro rilevanza.



L'interfaccia **KNOWLEDGE FLOW** è una variante dell'Explorer, in cui le operazioni da eseguire si esprimono in un ambiente grafico, disegnando un diagramma che esprime il "flusso della conoscenza".

Più in dettaglio l'utente seleziona un componente (rappresentato da un'icona) da una tool bar, lo posiziona nella finestra di lavoro e lo collega graficamente ad altri elementi già presenti nell'area di lavoro tramite frecce, ogni icona rappresenta una particolare elaborazione sui dati: apertura di file, salvataggio, applicazione di algoritmi di data mining e infine visualizzazione grafica. L'ordine con cui avvengono le operazioni viene stabilito al momento di collegare le frecce tra le icone, c'è sempre un'icona di partenza rappresentata dall'apertura di un file (arff o csv ad esempio) e da questa possono diramarsi una o più frecce che indicano una o più elaborazioni contemporanee di algoritmi.

Questa modalità di lavoro permette quindi di rappresentare e successivamente di eseguire in termini di flow chart le stesse procedure che Explorer permette di fare, ma aggiunge un livello di descrizione del lavoro più chiaro e conciso. Ogni elemento dell'area

di lavoro viene configurato individualmente tramite un menù che compare cliccando con il pulsante destro del mouse, tale menù ha tre voci: **Edit**, **Connections** e **Actions**. Con la voce **Edit** si cancellano i componenti o si apre la finestra di configurazione dell'elemento stesso. Gli algoritmi di classificazione o filtro si configurano come in Explorer, per il caricamento dati invece si sceglie un file da disco. La voce **Actions** comprende operazioni specifiche che riguardano il componente nell'area di lavoro che si vuole configurare al momento, infine tramite **Connections** si collegano i componenti tra di loro dall'icona sorgente e quella destinazione cliccando nei punti di connessione evidenziati da Weka. Diversamente da Explorer, i componenti (mostrati successivamente) per visualizzare e valutare i risultati sono solo presenti in Knowledge Flow: vengono utilizzati prelevandoli dalla barra delle icone e connettendo con le frecce i flussi di dati interessati.

Componenti di visualizzazione e valutazione di Knowledge Flow:

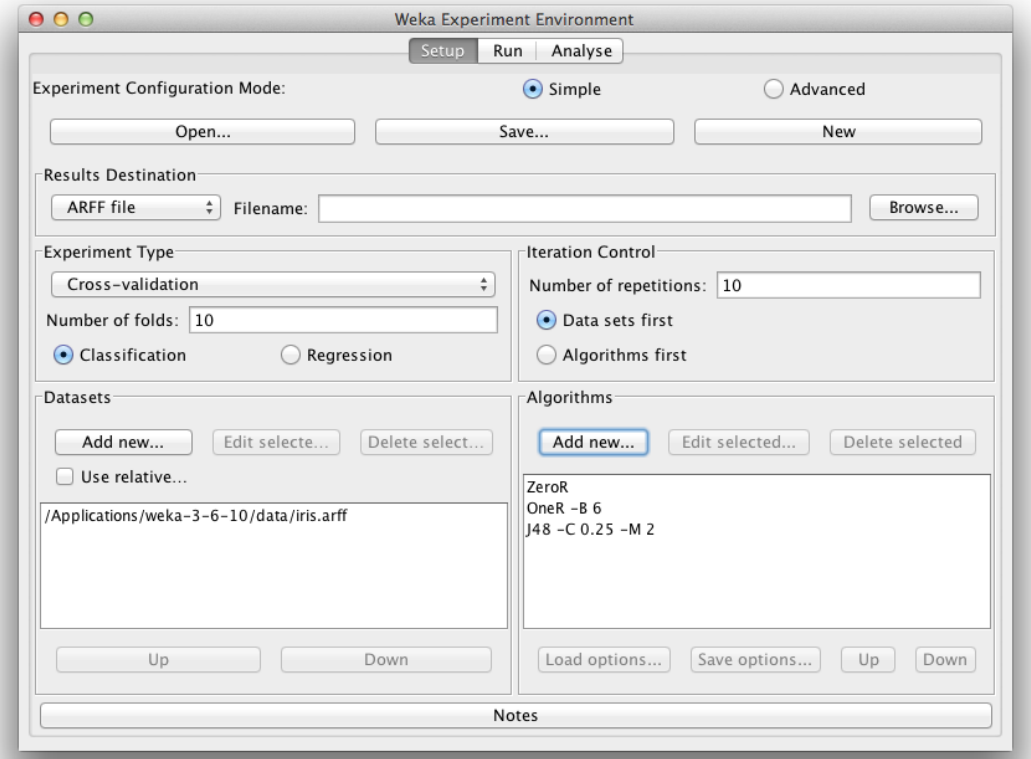
Visualization:

- **DataVisualizer**: visualizza i dati in grafici a due dimensioni.
- **ScatterPlotMatrix**: visualizza il riepilogo di tutti i grafici.
- **AttributeSummarizer**: mostra istogrammi per ogni attributo.
- **ModelPerformanceChart**: disegna curve ROC e altre curve di soglia.
- **TextViewer**: visualizza i dati in formato testo.
- **GraphViewer**: visualizza i grafi ad albero.
- **StripChart**: mostra un grafico a scorrimento dei dati.

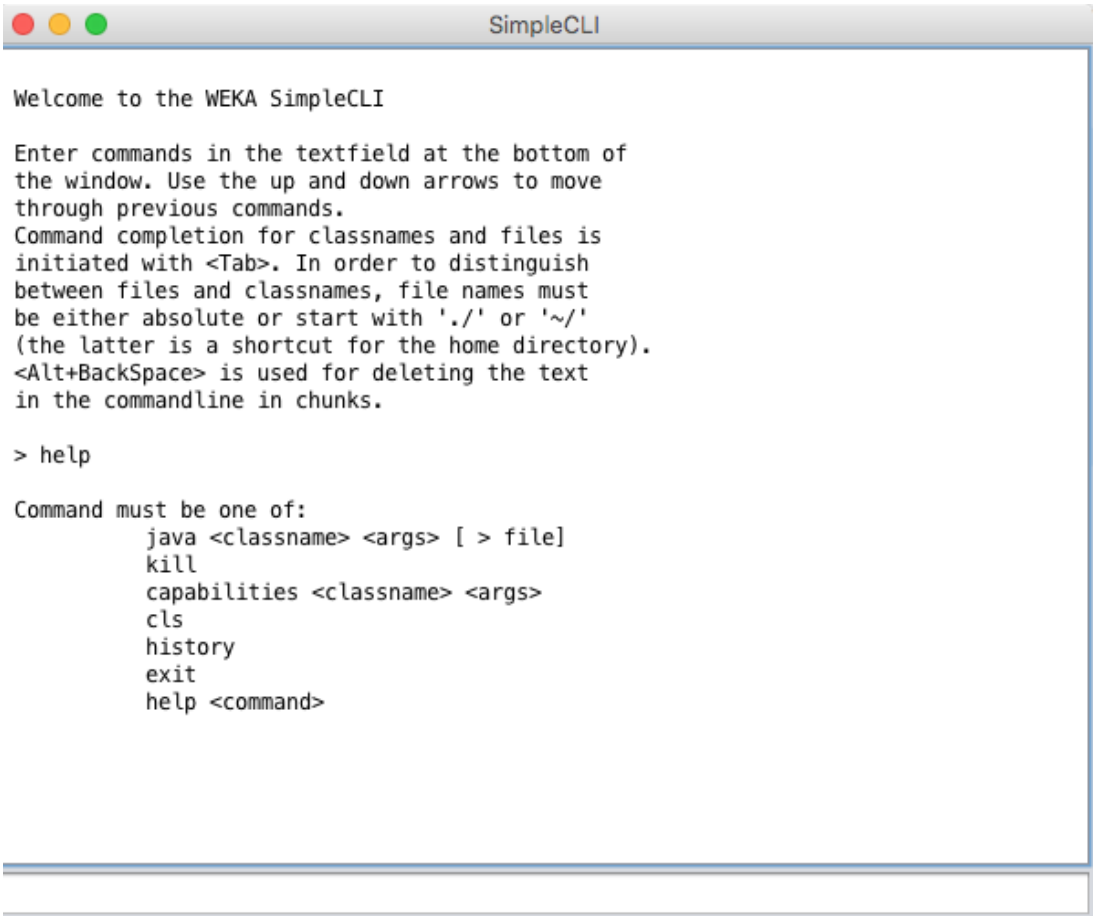
Evalutation:

- **TrainingSetMaker**: rende i dati di input come training set corrente.
- **TestSetMaker**: rende i dati di input come test set.
- **CrossValidationFoldMaker**: divide un dataset in fold.
- **TrainTestSplitMaker**: divide un dataset in training set e test set.
- **ClassAssigner**: imposta un attributo come classe di confronto.
- **ClassValuePicker**: sceglie un valore per la classe positiva.
- **ClassifierPerformanceEvaluator**: statistiche di valutazione sui risultati.
- **IncrementalClassifierEvaluator**: statistiche incrementali di valutazione sui risultati.
- **ClustererPerformanceEvaluator**: statistiche per il clustering.

- **PredictionAppender**: aggiunge al dataset i risultati delle predizioni di un algoritmo classificatore.



L'interfaccia **EXPERIMENTER** è dedicata alla effettiva "sperimentazione" di più algoritmi in serie che operano su una mole molto vasta di dati. Si caricano più file corrispondenti a diversi insiemi di dati, si impostano gli algoritmi e le iterazioni necessarie e infine il file di output da creare con i risultati. Una particolare e interessante caratteristica di Experimenter è quella di poter distribuire le elaborazioni su più processori ovvero su più postazioni che operano in parallelo.



L'interfaccia **SIMPLE CLI** permette, tramite una semplice shell, di utilizzare Weka con una interfaccia a linea di comando.

Tutto quello che si può fare dalla SimpleCLI è possibile farlo anche da un ambiente a linea di comando come il "prompt di DOS" di Windows o la shell di Unix.

3.2. Association Rules in WEKA

Per utilizzare le regole di associazione, dopo aver avviato WEKA ed essere entrati all'interno della sezione Explorer, si deve premere su "Associate", che mostra un'interfaccia per gli algoritmi di tali regole.

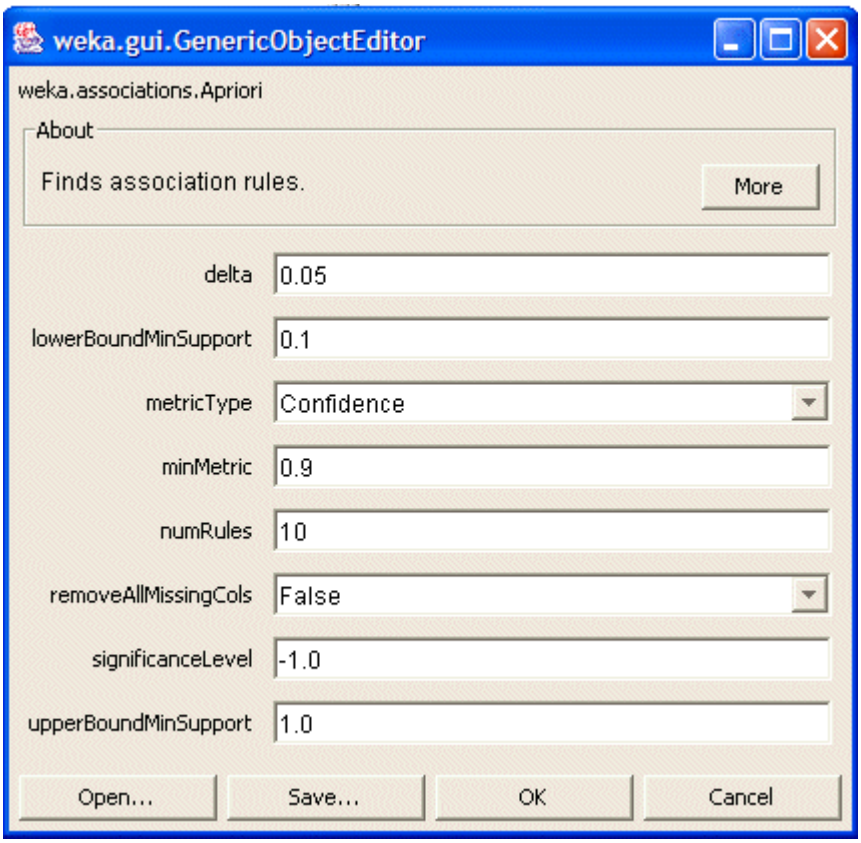
Essendo, come già affermato in precedenza, Apriori algorithm il più utilizzato, la trattazione delle association rules riguarderà tale algoritmo.

Apriori funziona solo con valori categoriali, pertanto, se un set di dati contiene attributi numerici, essi devono essere convertiti in nominali prima di applicare l'algoritmo.

Apriori è l'algoritmo che viene selezionato come predefinito da WEKA, ma risulta possibile modificare i parametri schiacciando sulla casella di testo immediatamente a destra del pulsante "Choose".

La finestra di dialogo che si apre, per modificare parametri come Confidence e Support, è la sottostante.

Facendo clic sul pulsante "More" è possibile visualizzare la sinossi dei vari parametri.



WEKA consente di ordinare le regole risultanti in base a parametri diversi, come confidence, leverage, and lift.

I valori predefiniti per Number of rules, decrease for Minimum support (delta factor) e minimum Confidence sono 10, 0,05 e 0,9.

Il Support è la percentuale di esempi coperti da LHS e RHS mentre la Confidence è la proporzione di esempi coperti da LHS che sono coperti anche da RHS.

Quindi se RHS e LHS di una regola coprono il 50% dei casi, allora la regola ha 0,5 di Support, se l'LHS di una regola copre 200 casi e di questi l'RHS copre 50 casi, la Confidence è 0,25.

In matematica LHS e RHS sono delle abbreviazioni che indicano rispettivamente il lato sinistro e destro di un'equazione.

Con le impostazioni predefinite Apriori tenta di generare 10 regole iniziando con un supporto minimo del 100%, diminuendo in modo iterativo il supporto del fattore delta fino a raggiungere il supporto minimo diverso da zero, o il processo finisce anche se è stato generato il numero richiesto di regole con almeno la minima confidenza.

Se esaminiamo l'output di Weka, un supporto minimo di 0.15 indica il supporto minimo ottenuto per generare 10 regole con il valore minimo del parametro specificato, come ad esempio 0.9 di confidence.

In generale, l'utilizzo di WEKA dalla riga di comando offre una maggiore flessibilità rispetto all'utilizzo della versione GUI.

Nel caso delle regole di associazione, la versione della GUI non consente di salvare i set di elementi frequenti (indipendentemente dalle regole generate).

Possiamo farlo usando la riga di comando.

Prendendo in considerazione un esempio, se osserviamo l'output del mining della regola di associazione, le opzioni effettive della riga di comando sono riportate in "Run information" nella parte superiore.

In un esempio specifico la riga di comando si presenta in questo modo:

weka.associations.Apriori -N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0

Possiamo usarlo direttamente utilizzando l'interfaccia "Simple CLI".

Nell'interfaccia principale di WEKA, fare clic sul pulsante "Simple CLI" per avviare l'interfaccia della riga di comando.

Il comando principale per generare le regole è:

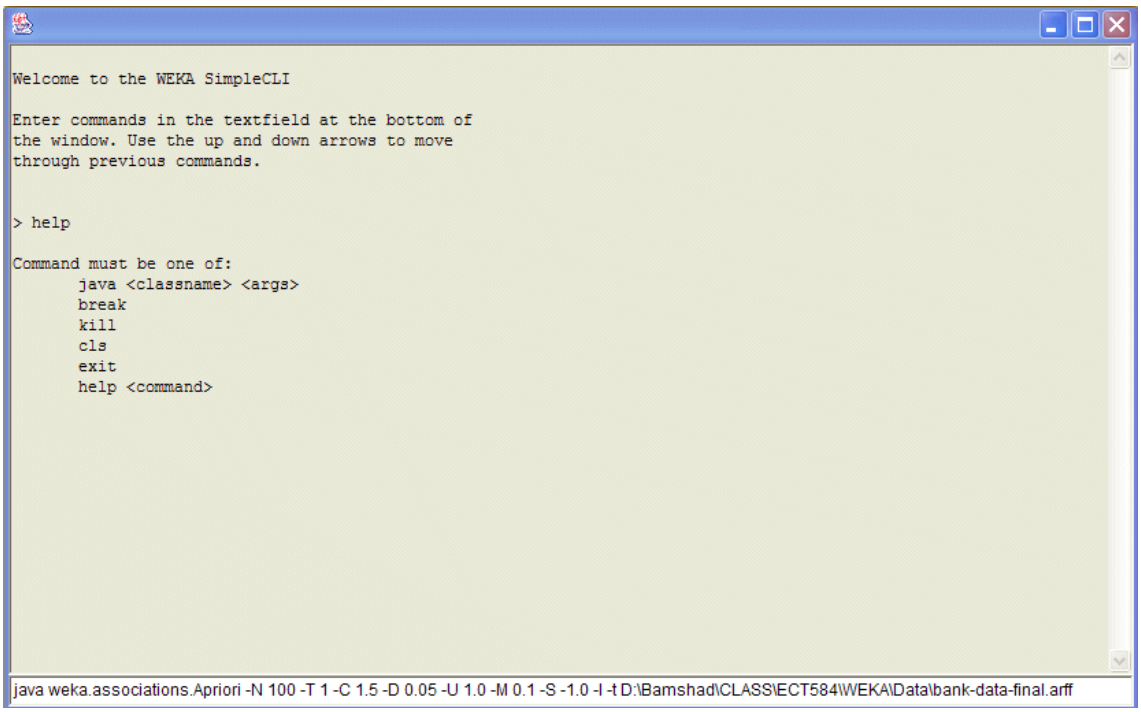
java weka.associations.Apriori options -t directory-path\bank-data-final.arff

dove la parola opzioni viene sostituita con le opzioni della riga di comando, che per questo particolare esempio sono:

-N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0

L'opzione "-t directory-path \ bank-data-final.arff" aggiuntiva dice a WEKA di utilizzare il file "bank-data-final.arff" come file di input (situato nella directory specificata).
Questo comando produrrà esattamente lo stesso output che si avrebbe utilizzando la GUI. Tuttavia, possiamo aggiungere un'opzione aggiuntiva ("-I") che si traduce nella generazione di tutti gli itemset frequenti:

java weka.associations.Apriori options -I -t directory-path\bank-data-final.arff



Quando è tutto pronto, premere Invio per eseguire il programma con le opzioni indicate. Il risultato di questo comando verrà visualizzato nel pannello superiore dell'interfaccia della “Simple CLI”.

4. Caso di studio

La finalità di questo studio è quella di analizzare il database excel fornitomi ed estrapolare informazioni utili relative all'OEE, un parametro molto importante per una linea di produzione come quella della Magneti Marelli, già trattato in precedenza.

Questo database è strutturato in modo da presentare sulle colonne voci che hanno un'incidenza più o meno elevata sull'OEE, ma di seguito verranno elencate solo quelle più significative:

- FERMI NON PROGRAMMATI \ RIGENERAZIONE UTENSILI
- FERMI PROGRAMMATI \ RIGENERAZIONE UTENSILI
- FERMI A CALENDARIO AM
- MANUTENZIONE PM PROGRAMMATA
- PULIZIE TECNICHE
- GUASTI - RISOLTI PM
- GUASTI - RISOLTI AM
- ATTESA INTERVENTO PM
- FERMO PER ESCLUSIONE AUTOMAZIONE
- SET-UP Cambio Tipo
- RIAVVIO
- TEMPO OPERATIVO
- MICROFERMATE
- INTERVENTO EMERGENZE
- FERMO CICLO INTENZIONALE
- TEMPO CICLO DIFFERENTE DALLO STANDARD
- MANCANZA PERSONALE
- MANCANZA MATERIALI DIRETTI
- MANCANZA ALIMENTAZIONE DA POSTAZIONE A MONTE
- MANCANZA ASSORBIMENTO DA POSTAZIONE A VALLE
- TEMPO OPERATIVO NETTO
- FERMO PROBLEMI QUALITA'

- RALLENTAMENTO PROBLEMI QUALITA'
- MINUTI PRODUZIONE SCARTO
- MINUTI PRODUZIONE REWORK
- TEMPO OPERATIVO A VALORE AGGIUNTO
- Pezzi totali prodotti
- Pezzi di scarto
- Pezzi rigettati
- "Tempo ciclo a pezzo (MIN/PZ) teorico"
- % scarto
- FTQ
- Tempo ciclo a pezzo (MIN/PZ) consuntivo
- TOTALE PEZZI BUONI
- TOT. TEMPO PROD. BUONA (PZ BUONI X TC)
- TOT. TEMPO PROD. (PZ TOTALI X TC)
- DISPONIBILITA'
- PERFORMANCE
- QUALITA'
- "Tempo ciclo a pezzo (MIN/PZ) Reale"
- Quadratura.

Sulle righe il database presenta i vari giorni (dal 02/01/2019 al 26/06/2019) in cui sono state eseguite queste misurazioni.

Successivamente ho ritenuto opportuno “pulire” il database, prima citato, da quelle colonne che non presentavano sufficienti dati tali da poterli ritenere influenti sulla valutazione dell’OEE.

Le colonne prese in considerazione dopo la “pulizia” sono le seguenti:

- GUASTI - RISOLTI PM
- GUASTI - RISOLTI AM
- TEMPO OPERATIVO
- TEMPO CICLO DIFFERENTE DALLO STANDARD
- TEMPO OPERATIVO NETTO

- MINUTI PRODUZIONE SCARTO
- MINUTI PRODUZIONE REWORK
- TEMPO OPERATIVO A VALORE AGGIUNTO
- DISPONIBILITA'
- PERFORMANCE
- QUALITA'.

Tali colonne sono state inserite in un nuovo file excel.

Per poter estrapolare delle informazioni più accurate, su questo nuovo file, ho classificato, tramite la funzione di excel “se annidati”, tutti i valori di ciascuna colonna in cinque gruppi: “basso”, “medio-basso”, ”medio”, ”medio-alto”, ”alto”.

Questa classificazione è stata eseguita anche per l’OEE.

In seguito, per analizzare più nel dettaglio quanto effettivamente una colonna influisse sulla classificazione dell’OEE, ho scelto di utilizzare la funzione excel “correlazione”.

La funzione “correlazione” restituisce il coefficiente di correlazione di due intervalli di celle.

Si utilizza il coefficiente di correlazione per stabilire la relazione tra due proprietà.

È possibile ad esempio esaminare la relazione tra la temperatura media di un ambiente e l'utilizzo di condizionatori d'aria.

Per quanto il coefficiente di correlazione sia più vicino a + 1 o -1, indica una correlazione positiva (+ 1) o negativa (-1) tra gli intervalli di celle.

Correlazione positiva significa che se i valori in un intervallo aumentano, anche i valori dell'altro intervallo aumentano.

Un coefficiente di correlazione più vicino a 0 indica una correlazione no o debole.

Per le colonne sopra citate ho ottenuto un valore di correlazione con l’OEE di:

- -0.490
- -0.112
- 0.839
- 0.287
- 0.899
- -0.009

- -0.368
- 0.982
- 0.870
- 0.135
- 0.617

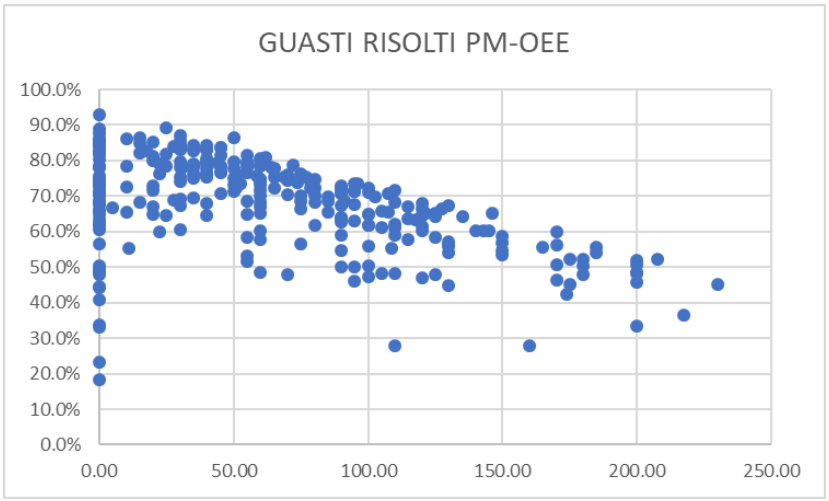
Come si evince dai risultati ottenuti, la colonna “GUASTI - RISOLTI PM” risulta essere inversamente proporzionale all’ OEE e con una discreta influenza; sia la colonna “GUASTI - RISOLTI AM” che la colonna “MINUTI PRODUZIONE SCARTO” sono, come la prima, inversamente proporzionali all’OEE ma con un’incidenza nettamente inferiore, mentre si ha una risalita con la colonna “MINUTI PRODUZIONE REWORK” ma comunque inferiore alla prima.

Tutte le altre colonne sono direttamente proporzionali all’OEE, in particolare “TEMPO OPERATIVO, “TEMPO OPERATIVO NETTO”, “TEMPO OPERATIVO A VALORE AGGIUNTO”, ”DISPONIBILITA'” e “QUALITA'” hanno un’importante correlazione con l’OEE, mentre “TEMPO CICLO DIFFERENTE DALLO STANDARD” e “PERFORMANCE” hanno una correlazione relativamente bassa.

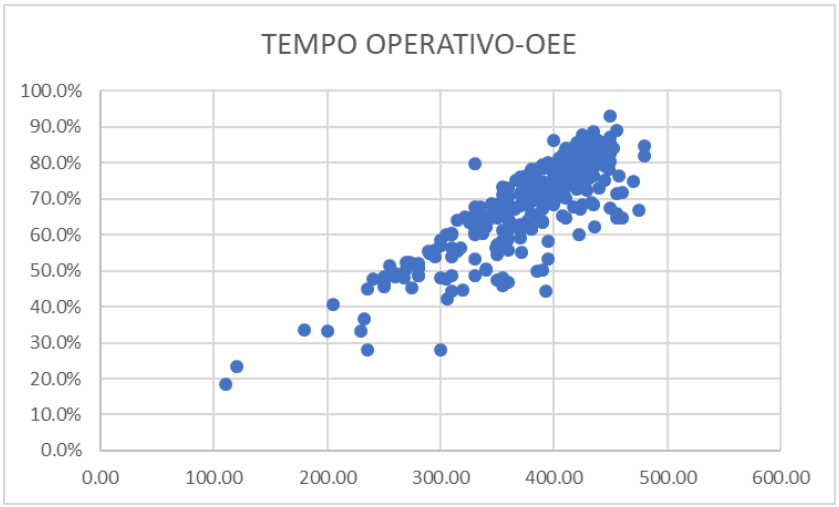
In seguito, ho deciso di trascurare quelle colonne che avevano una correlazione poco significativa, in modo da analizzare solo i valori che permettono una variazione più significativa dell’OEE.

Di conseguenza le voci prese in considerazione sono:

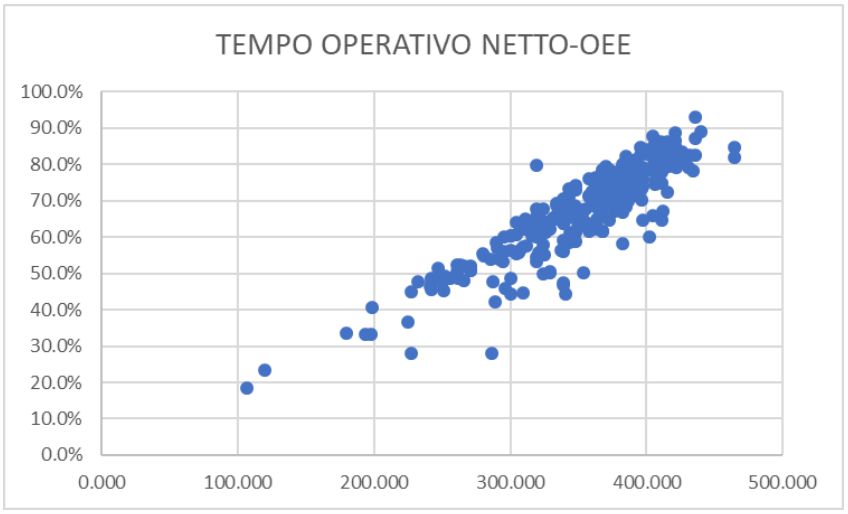
- GUASTI - RISOLTI PM (tra 0 e 46 “basso”, >46 “medio-basso”, >92 “medio”, >138 “medio-alto”, > 184 “alto”)



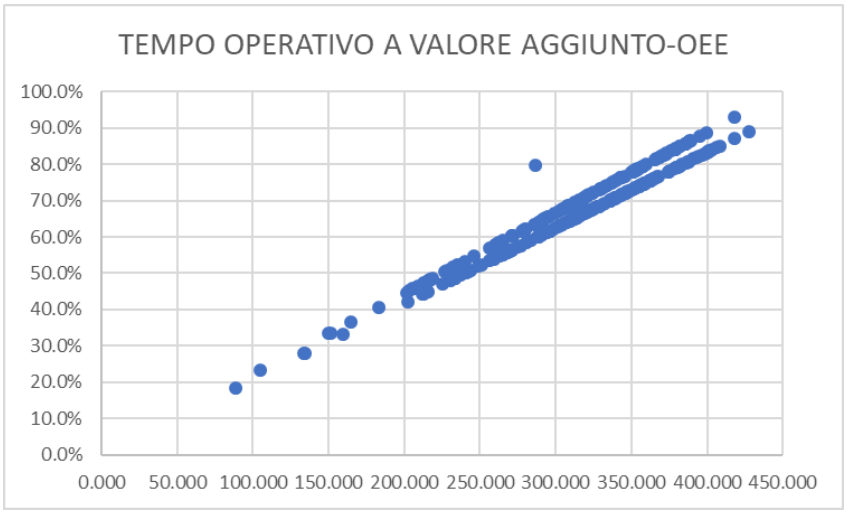
- TEMPO OPERATIVO (tra 0 e 96 “basso”, >96 “medio-basso”, >192 “medio”, >288 “medio-alto”, >384 “alto”)



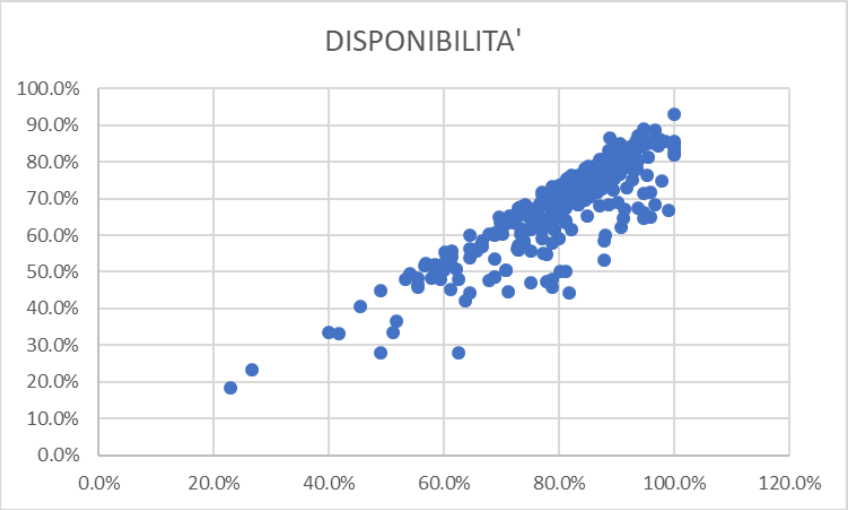
- TEMPO OPERATIVO NETTO (tra 0 e 92.9 “basso”, >92.9 “medio-basso”, >185.8 “medio”, >278.7 “medio-alto”, >371.6 “alto”)



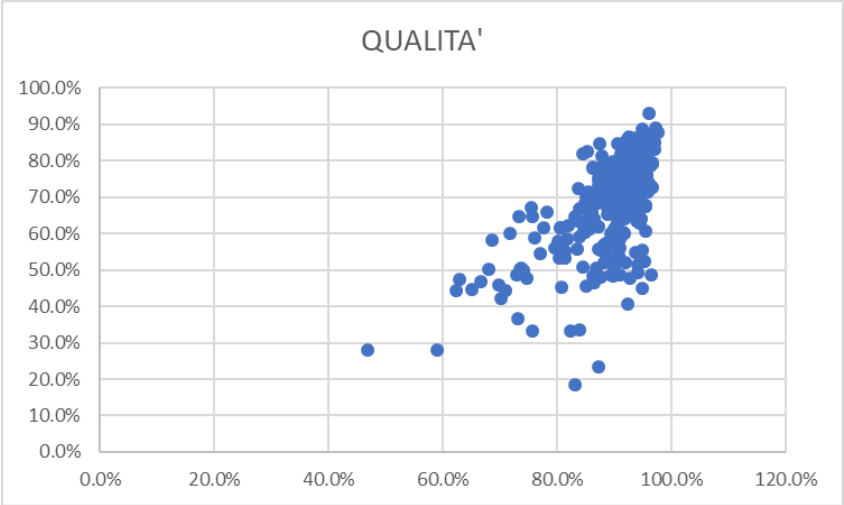
- TEMPO OPERATIVO A VALORE AGGIUNTO (tra 0 e 85.6 “basso”, >85.6 “medio-basso”, >171.1 “medio”, >256.7 “medio-alto”, >342.2 “alto”)



- DISPONIBILITA' (tra 0% e 20% “basso”, >20% “medio-basso”, >40% “medio”, >60% “medio-alto”, >80% “alto”)



- QUALITA' (tra 0% e 20% “basso”, >20% “medio-basso”, >40% “medio”, >60% “medio-alto”, >80% “alto”).



Lo stesso OEE è a sua volta suddiviso in: tra 0% e 20% “basso”, >20% “medio-basso”, >40% “medio”, >60% “medio-alto”, >80% “alto.

5. Conclusione

La media dell'OEE di tutta la linea di produzione della Magneti Marelli è pari al 67.7%, che in base alla nostra classificazione risulta essere "medio-alto", in particolare è così suddiviso:

- basso=1
- medio-basso=7
- medio=62
- medio-alto=181
- alto=45

L'obiettivo di questo studio è quello di capire in quali condizioni si verifica un OEE alto e con quale ripetitività.

Quindi ho isolato i 45 casi in cui l'OEE si è dimostrato essere "alto" e ho creato una tabella excel per poterli studiare più in dettaglio.

Come si può dedurre dalla tabella, un OEE "alto" si verifica in corrispondenza di:

- GUASTI - RISOLTI PM = basso
- TEMPO OPERATIVO = alto
- TEMPO OPERATIVO NETTO = alto
- TEMPO OPERATIVO A VALORE AGGIUNTO = alto
- DISPONIBILITA' = alto
- QUALITA' = alto.

In realtà si riesce ad ottenere una valutazione elevata dell'OEE anche nel caso in cui la voce GUASTI-RISOLTI PM risulti essere "medio-basso", che avviene per quattro volte.

Si può scendere ad un valore "medio-alto" dell'OEE o mantenendo costanti i parametri più influenti e facendo variare i parametri secondari oppure variando leggermente i parametri primari.

Quando i parametri principali subiscono una variazione importante, allora l'OEE scende ancora fino a valori classificabili come "medio", "medio-basso" e "basso".

In verità un "OEE" basso risulta essere una rarità, in quanto si raggiunge una sola volta su tutti i casi e si verifica in questa particolare condizione:

- GUASTI - RISOLTI PM = basso
- TEMPO OPERATIVO = medio-basso

- TEMPO OPERATIVO NETTO = medio-basso
- TEMPO OPERATIVO A VALORE AGGIUNTO = medio-basso
- DISPONIBILITA' = medio-basso
- QUALITA' = alto.

La stessa analisi può essere eseguita utilizzando le Association Rules tramite il software WEKA, il quale mi avrebbe potuto fornire anche risultati diversi da quelli ottenuti.

6. Sitografia

https://it.wikipedia.org/wiki/Magneti_Marelli

https://www.marelli.com/it/

https://it.wikipedia.org/wiki/Overall_Equipment_Effectiveness

https://www.leanmanufacturing.it/strumenti/oee.html

https://www.organizzazioneaziendale.net/oee-significato-definizione-calcolo/2671

https://www.produzioneagile.it/oee-overall-equipment-effectiveness/

https://www.toolsforsmartminds.com/it/insight/blog/176-perche-l-oee-puo-essere-fuorviante

https://enterprise.teamsystem.com/blog/industry40/oee_come_si_calcola

http://www.intelligenzaartificiale.it/machine-learning/

https://www.sas.com/it_it/insights/analytics/machine-learning.html

https://www.ai4business.it/intelligenza-artificiale/machine-learning/machine-learning-cosa-e-applicazioni/

https://it.wikipedia.org/wiki/Apprendimento_automatico

https://www.cwi.it/tecnologie-emergenti/intelligenza-artificiale/machine-learning-124626

https://it.wikipedia.org/wiki/Apprendimento_supervisionato

http://www.andreaminini.com/ai/machine-learning/apprendimento-supervisionato

https://it.wikipedia.org/wiki/Clustering

http://www.andreaminini.com/ai/machine-learning/apprendimento-senza-supervisione

http://www.andreaminini.com/ai/machine-learning/riduzione-dimensionalita-dati

https://it.wikipedia.org/wiki/Analisi_della_regressione

https://www.deeplearningitalia.com/a-general-introduction-to-learning-methods-2/

https://www.machine-learning.it/apprendimento-supervisionato/

https://www.dataskills.it/tecniche-di-clustering/#gref

https://it.wikibooks.org/wiki/Intelligenza_artificiale/Apprendimento_con_rinforzo

https://it.wikipedia.org/wiki/Processo_decisionale_di_Markov

https://it.wikipedia.org/wiki/Apprendimento_per_rinforzo

https://it.wikipedia.org/wiki/Logica_fuzzy
http://www.andreaminini.com/ai/machine-learning/apprendimento-con-rinforzo
https://it.wikipedia.org/wiki/Q-learning
https://www.ai4business.it/intelligenza-artificiale/machine-learning/modelli-di-apprendimento-automatico/
https://www.sas.com/it_it/insights/analytics/machine-learning.html#machine-learning-today-world
https://it.wikipedia.org/wiki/Regole_di_associazione
https://en.wikipedia.org/wiki/Association_rule_learning
https://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining
https://it.wikipedia.org/wiki/Collaborative_filtering
https://towardsdatascience.com/association-rules-2-aa9a77241654
https://it.wikipedia.org/wiki/Algoritmo_apriori
http://www-db.deis.unibo.it/courses/SID/old/Lezioni/02%20-%20Regole%20associative.pdf
https://www.spaghettiml.com/2017/08/22/algoritmo-a-priori-parte-2/
https://it.wikipedia.org/wiki/Weka
https://it.wikipedia.org/wiki/Grafico_di_dispersione
http://www.mokabyte.it/2007/03/weka-1/
https://dbgroup.ing.unimore.it/tesi/Laurenzi.pdf
https://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf
https://it.emcelettronica.com/weka-machine-learning-per-tutti-parte-i
http://facweb.cs.depaul.edu/mobasher/classes/ect584/WEKA/associate.html
Homework 3 Association Rule Mining Association Rule ... - Kmitlwww.ce.kmitl.ac.th
https://en.wikipedia.org/wiki/Sides_of_an_equation
https://support.microsoft.com/it-it/office/correlazione-funzione-correlazione-995dcef7-0c0a-4bed-a3fb-239d7b68ca92