# ***CAPITOLO 3: DATA MINING & MACHINE LEARNING NEL FASHION RETAIL***

## ***3.1 KNOWLEDGE DISCOVERY IN DATABASE (KDD)***

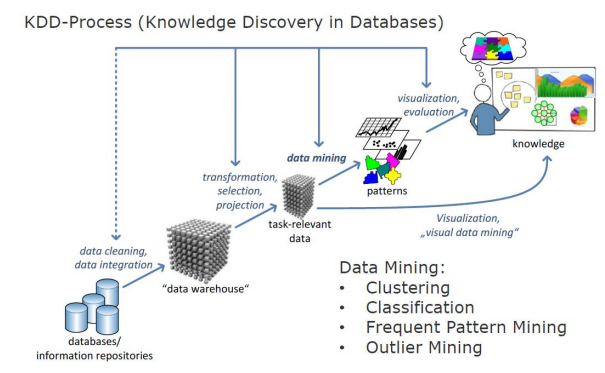


Figura 9: KDD Process

Il KDD è una procedura interattiva e iterativa, che cerca di estrarre dai dati informazioni implicite, sconosciute a priori e potenzialmente utili.

Andiamo ad analizzare ora le singole fasi:

* *Identificazione degli obiettivi*: l'oggetto di questa fase è l'individuazione dell'ambito di applicazione in cui deve essere considerato il KDD, inoltre bisogna individuare gli obiettivi da perseguire. Si tratta forse della fase più difficile sia in termini di allocazione risorse sia perché devono essere determinate, in modo preciso, le misure del successo e i criteri per misurare successi e fallimenti. Si può fare una lista solo parziale dei molteplici aspetti che vanno presi in considerazione, alcuni sono il costo stimato del progetto e la scelta degli strumenti di data mining da utilizzare.
* *Selezione*: In questa fase deve essere selezionato l'insieme iniziale dei dati, che devono essere sottoposti ad analisi. I dati grezzi vengono segmentati e selezionati secondo alcuni criteri al fine di pervenire ad un sottoinsieme di dati, che rappresentano il nostro target data o dati obiettivo. Se i dati originali sono collocati in un flat file, la creazione del target risulta molto semplice. I sistemi di gestione dei database immagazzinano e manipolano dati transazionali, ciò consente ai sistemi informatici, relativi a tali sistemi, di fare aggiornamenti e di estrarre informazioni in modo rapido. Ciò è dovuto alla strutturazione dei dati tramite modelli relazionali, il cui scopo è ridurre la ridondanza dei dati, tramite la decomposizione di singole tabelle in più strutture relazionali, ed accelerare l'accesso alle informazioni. Del resto lo scopo del DM è proprio utilizzare la ridondanza dei dati per reperire “conoscenza", ecco perché è necessario ricomporre le strutture relazionali. Si intuisce quindi che è stretto il legame tra DM e Data Warehouse, il cui scopo è proprio quello di mettere insieme i dati, e non scomporli, al fine di sfruttarne la ridondanza. Spesso è anche necessario mettere insieme informazioni estratte da più fonti, cosa che può rendere complessa la fase di selezione in quanto bisogna trasformare i dati in modo da assicurare l'omogeneità in quanto, ad esempio, la codifica dei dati deve essere uguale per tutti i record dei dati target, altrimenti l'analisi risulta di scarsa utilità.
* *Preelaborazione*: Generalmente il target data disponibile non deve essere analizzato interamente ma basta estrarne un campione opportuno, eseguendo poi un'analisi su base campionaria. Inoltre i dati devono essere preprocessati, cioè “puliti", trattando in maniera opportuna i dati anomali e mancanti. Vanno individuati i valori errati delle variabili; trovare gli errori nei dati categorici diventa un problema quando si analizzano dataset molto grandi. I dati vanno anche semplificati; queste tecniche di data smoothing sono mirate alla riduzione del numero di valori per una variabile numerica. Alcuni classificatori, come le reti neurali, utilizzano funzioni che effettuano la semplificazione durante il processo di classificazione, eseguendo così un data smoothing interno. Due semplici tecniche di semplificazione sono il calcolo e l'arrotondamento dei valori medi.
* *Trasformazione*: I dati, per essere utilizzati, spesso devono essere trasformati; questa fase può assumere varie forme e può essere necessaria per varie ragioni. Si possono convertire tipi di dati in altri o definirne di nuovi, ottenuti attraverso l'uso di operazioni matematiche e logiche sulle variabili, eseguire delle normalizzazioni (scalamento decimale, normalizzazione min-max o con lo z-score) o addirittura eliminare delle variabili. In genere infatti gli algoritmi di DM non lavorano in modo efficiente se i dati contengono una grande quantità di variabili che non sono in grado di prevedere la classe di appartenenza; si rende quindi utile una ricerca ed una successiva eliminazione delle variabili ridondanti e “inutili" per il problema in questione. A volte le variabili con poco potere previsivo possono essere combinate con altre per formare nuove variabili con un alto grado di capacità previsiva.
* *Data mining*: Ai dati trasformati vengono applicate una serie di tecniche in modo da poterne ricavare dell'informazione non banale o scontata. Sono gli obiettivi che si vogliono raggiungere a dare un'indicazione sul tipo di tecnica/algoritmo che deve essere applicata.
* *Interpretazione e valutazione*: Scopo di questa fase è determinare la validità del modello ottenuto con il DM; in sintesi non basta interpretare i risultati ma bisogna capire in che misura questo modello o risultato possa essere utile. Questo può essere fatto in vari modi sia attraverso un'analisi statistica che euristica o sperimentale.
* *Data Visualization*: L'ultimo obiettivo consiste nell'utilizzare ciò che è stato appreso, creando un report o un rapporto tecnico su ciò che è stato scoperto e cercando di capire in che modo sfruttare ciò che è stato scoperto.

Si capisce bene quindi che il processo di estrazione della conoscenza è lungo e piuttosto articolato, sono fondamentali le scelte che si fanno per il trattamento di anomalie o errori nei dati e l'identificazione chiara degli obiettivi che si vogliono perseguire.

### ***3.1.1 Data Mining vs Machine Learning***

Il data mining si riferisce all'estrazione di conoscenza da una grande quantità di dati ed è il processo per scoprire vari tipi di pattern che sono ereditati nei dati e che sono accurati, nuovi e utili. È un processo iterativo di creazione di un modello predittivo e descrittivo, attraverso la scoperta di tendenze e pattern precedentemente sconosciuti in grandi quantità di dati per supportare il processo decisionale. Può essere definito anche come il sottoinsieme dell'analisi aziendale, simile alla ricerca sperimentale. Le fonti del data mining sono i database e i metodi statistici.

Il Machine Learning indica un ambito di ricerca all’interno dell’Intelligenza Artificiale e, grazie all'esperienza basata sui dati, implica lo studio di algoritmi che sono in grado di estrarre informazioni automaticamente. Sono necessarie due fonti di dati: dati di addestramento e dati di test. Di solito, il Machine Learning utilizza tecniche di data mining e un altro algoritmo di apprendimento per costruire modelli di ciò che sta accadendo dietro alcuni dati in modo che possa prevedere i risultati futuri. Ma vediamo in tabella le varie differenze.

Tabella 2: Data Mining vs Machine Learning [13]

|  |  |  |
| --- | --- | --- |
|  | **Data mining** | **Machine learning** |
| Meaning | Extracting knowledge from a large amount of data | Introduce new algorithm from data as well as past experience |
| History | Introduce in 1930, initially referred as knowledge discovery in databases | introduce in near 1950, the first program was Samuel’s checker-playing program |
| Responsibility | Data mining is used to get the rules from the existing data. | Machine learning teaches the computer to learn and understand the given rules. |
| Origin | Traditional databases with unstructured data | Existing data as well as algorithms. |
| Implementation | We can develop our own models where we can use data mining techniques for | We can use machine learning algorithm in the decision tree, neural networks and some other area of artificial intelligence. |
| Nature | Involves human interference more towards manual. | Automated, once design self-implemented, no human effort |
| Application | used in cluster analysis | used in web search, spam filter, credit scoring, fraud detection, computer design |
| Abstraction | Data mining abstract from the data warehouse | Machine learning reads machine |
| Techniques involve | Data mining is more of a research using methods like machine learning | Self-learned and trains system to do the intelligent task. |
| Scope | Applied in the limited area | Can be used in a vast area. |

## ***3.2 ALGORITMI***

Esistono diversi algoritmi per raggiungere l'obiettivo del data mining che consiste nell'estrarre nuove informazioni dai dati esistenti. Come vedremo, ci sono due approcci per raggiungere questo obiettivo: apprendimento supervisionato e apprendimento non supervisionato [14].

* *Unsupervised learning:* a differenza del precedente, non utilizza dati classificati e etichettati in precedenza; non sappiamo, quindi, a quale categoria essi appartengano. Alla macchina viene chiesto, quindi, di estrarre una regola che raggruppi i casi presentati secondo caratteristiche che ricava dai dati stessi. Per questo è anche definito apprendimento di caratteristiche (feature learning). Gli algoritmi in questo caso cercano una relazione tra i dati per capire se e come essi siano collegati tra di loro. Non contenendo alcuna informazione preimpostata, l’algoritmo è chiamato a creare “nuova conoscenza” (knowledge discovery). Una delle applicazioni principali è il *clustering*, ovvero il raggruppamento dei dati in gruppi omogenei definiti *cluster*. L’apprendimento non supervisionato, quindi, serve generalmente ad estrarre informazioni non ancora note.
* ***Supervised learning***: metodologia di apprendimento automatico in cui alla macchina vengono passati degli esempi composti da una coppia di dati contenenti il dato originale e il risultato atteso. Compito della macchina è quello di trovare la regola (funzione o modello) con cui creare una relazione tra i due in modo tale che, al presentarsi di un esempio sconosciuto in precedenza, possa ottenere il risultato corretto. I dati sono precedentemente etichettati, ovvero assegnati a una certa categoria. L’apprendimento supervisionato è utilizzato principalmente per i problemi di *classificazione*, come ad esempio nel marketing per classificare i clienti potenziali e proporre i prodotti a cui potrebbero essere interessati sulla base del profilo e della storia degli acquisti. Un altro esempio sono i sistemi anti-spam delle email che, all’arrivo di un messaggio, riescono a decidere se una determinata email debba essere etichettata come spam o meno.

Immagine che contiene screenshot

Descrizione generata con affidabilità molto elevata

Figura 10: Data Mining Algoritms

Utilizzando alcune delle tecniche sopra citate possiamo creare modelli predittivi. Qualunque sia la loro applicazione, i modelli predittivi usano l'esperienza per assegnare punteggi e livelli di confidenza, ad alcuni risultati rilevanti in futuro.

Quindi, una delle chiavi del successo sta avendo abbastanza dati con il risultato già noto per addestrare il modello. In parole povere, ci sono davvero due cose da fare con i modelli predittivi:

La prima fase è la formazione, in cui il modello viene creato utilizzando i dati del passato, il secondo è il punteggio, in cui il modello creato viene testato con dati non visibili per vedere come ha segnato.

Non bisogna mai dimenticare che il più importante è quello di ottenere buoni risultati nei dati invisibili e non nei dati di allenamento. L'overfitting è la situazione che si verifica quando il modello spiega i dati dell'allenamento ma non può generalizzare per testare i dati.

Per applicare un modello predittivo, assumiamo che una parte dei dati sia un buon predittore dei dati rimanenti (o in serie temporali che il presente sia un buon predittore del futuro). Supponiamo anche che gli schemi osservati possano essere spiegati, almeno in parte, dagli attributi che stiamo considerando.

Le innovazioni che utilizzano intelligenza artificiale (AI) e Machine Learning (ML) sono tra le principali tendenze tecnologiche nel mondo del retail. Stanno avendo un grande impatto sul settore, in particolare nelle aziende di e-commerce che si affidano alle vendite online, dove l’uso di una qualche forma di Machine Learning nel retail è oggi molto comune.

Grandi retailer online come eBay, Amazon o Alibaba hanno integrato con successo le tecnologie AI nell’intero ciclo di vendita, dalla logistica di stoccaggio al servizio clienti post- vendita.

Le aziende che utilizzano i sistemi di raccomandazione ottengono aumenti delle vendite a seguito di offerte personalizzate e di una migliore esperienza del cliente. Le raccomandazioni in genere accelerano le ricerche e rendono più facile per gli utenti accedere ai contenuti a cui sono interessati e sorprenderli con offerte che non avrebbero mai cercato. Inoltre, sono in grado di acquisire e fidelizzare i clienti inviando e-mail con collegamenti a nuove offerte che soddisfano gli interessi dei destinatari e si adattano ai loro profili.

L’utente inizia a sentirsi conosciuto e compreso ed è più propenso ad acquistare prodotti aggiuntivi. Conoscendo ciò che un utente vuole e mostrandoglielo subito, è meno probabile che egli lasci la piattaforma. Ciò si traduce in una maggiore possibilità di acquisto e in una diminuzione della minaccia di perdere un cliente a favore di un concorrente.

Includendo l’offerta, la stagionalità, gli eventi esterni relativi alla tua attività (ad esempio un concerto, una partita, un festival) e la domanda e l’offerta del mercato, un sistema automatico di prezzi può adeguare in modo efficiente i prezzi.

Un sistema in grado di apprendere la maggior parte di ciò che accade nel mercato consente di avere informazioni migliori rispetto ai concorrenti per prendere decisioni corrette in breve tempo. Fornire un valore aggiunto agli utenti includendo raccomandazioni è allettante. Inoltre, consente alle aziende di posizionarsi davanti ai loro concorrenti e aumentare i loro guadagni.

Vediamo nel dettaglio i più comuni algoritmi usati dal Machine Learning per andare in contro al cliente.

### ***3.2.1 Clustering***

L’obiettivo della clusterizzazione è di organizzare gli oggetti esaminati in gruppi, i quali condividono proprietà simili. Il Clustering si può considerare uno dei più importanti metodi di apprendimento non supervisionato. Come ogni metodo appartenente a questa categoria, non fa uso di identificatori determinati a priori per intuire la possibile struttura dei dati. Un cluster può essere definito come una collezione di oggetti simili tra loro e dissimili da elementi negli altri cluster.

Esistono varie classificazioni delle tecniche di clustering, una prima categorizzazione dipende dalla possibilità che un elemento possa o meno essere assegnato a più cluster [15]:

1. *Clustering Esclusivo*: Ogni elemento può appartenere solamente ad un cluster, ossia le intersezioni tra i clusters sono sempre insiemi vuoti, questa procedura prende anche il nome di *Hard Clustering*.
2. *Clustering Inclusivo*: Ogni elemento può appartenere a più cluster contemporaneamente, con un indice che decreta il grado di appartenenza ad ogni cluster, procedura che prende il nome di *Soft* o *Fuzzy Clustering*.
3. *Clustering Partizionale*: Si utilizza il concetto di distanza tra gli elementi, i quali appartengono ad un particolare gruppo in base alla loro relazione con un punto significativo del dataset.
4. *Clustering Gerarchico*: Si costruisce una gerarchia di partizioni, costruita sia per aggregazione che per divisione, mediante una rappresentazione ad albero che prende il nome di *Dendogramma*. Esistono altre suddivisioni per quanto riguarda il Clustering Partizionale, più dettagliate, le quali si differenziano per la valutazione della distanza tra gli elementi e la relativa creazione dei cluster.

In seguito, sono descritte le principali strategie di Clustering ed algoritmi utilizzati.

#### **Clustering Basato su Centroidi (K-Means)**

Il clustering basato su centroidi è di tipo partizionale e ogni cluster è rappresentato da un prototipo chiamato *centroide* che tipicamente è la media tra le distanze dei punti del cluster. Uno dei più famosi algoritmi di clustering appartenenti a questa categoria è il K-Means che richiede di specificare il numero ***K*** di cluster che si vogliono ottenere (conferendo il nome *K* -Means). L’algoritmo iterativamente elegge i *K* centroidi del cluster, ed ogni elemento viene associato al centroide più vicino. L’algoritmo è il seguente:

|  |
| --- |
| **K-MEANS ALGORITHM** |
| 1: **function** K-Means(*clusters K*) |
| 2: *Elezione K Centroidi* |
| 3: **repeat** |
| 4: *Assegnamento di ogni elemento al punto K piu*´ *vicino* |
| 5: *Ricalcolo dei K Centroidi* |
| 6: **until** *I Centroidi non variano* |

Inizialmente i centroidi vengono scelti randomicamente, mentre nelle iterazioni successive dell’algoritmo essi consistono tipicamente nella media tra le distanze dei punti del cluster. Esistono differenti metodologie per calcolare tale distanza: *Distanza Euclidea*, *Cosine Similarity*, *Correlazione*. L’algoritmo converge per le misure di similitudine elencate, tale convergenza si manifesta principalmente nelle prime iterazioni, seguite da una fase di assestamento, infatti spesso la condizione di stop viene rilassata, ammettendo una soglia minima di cambiamento tra i centroidi.

La scelta dei centroidi è una fase molto sensibile, infatti vengono applicate le seguenti tecniche per risolvere, anche se non completamente, il problema:

* Si eseguono molteplici esecuzioni, stimando i centroidi in modi differenti oppure semplicemente randomicamente, in seguito si valuta la qualità del risultato ottenuto, per mezzo degli strumenti di validazione che saranno descritti in seguito.
* Si utilizza la procedura di *Clustering Gerarchico* (spiegata nella sezione successiva), si eseguono *K* suddivisioni e si calcolano i centroidi dei cluster ottenuti, questi saranno i punti di partenza per l’algoritmo K- Means.
* Si stima un numero di centroidi *N > K*, e vengono considerati solamente i K migliori.
* Tecniche di postprocessing, come eliminazione di piccoli clusters, unione di cluster molto simili tra di loro e suddivisione di cluster troppo grandi.
* Si utilizza l’algoritmo *Bisecting K-Means*, esso consiste in un approccio gerarchico attraverso il quale partendo da un unico cluster, si suddivide tramite algoritmo 2-Meansun numero arbitrario di volte, si prende l’iterazione che ha prodotto i migliori cluster e si applica ricorsivamente l’algoritmo, a ciascun cluster scelto fino a che non si ottengono i **K** cluster desiderati.

Immagine che contiene testo

Descrizione generata con affidabilità molto elevataImmagine che contiene testo, mappa

Descrizione generata con affidabilità molto elevata

Figura 11: K-Means Algorithm

La complessità dell’algoritmo è *O*(*n\* K \* I \* d*) dove *n* è il numero di punti, *K* il numero di cluster, *I* il numero di iterazioni e *d* il numero di attributi su cui si basa la funzione per il calcolo della distanza utilizzata.

In conclusione, l’algoritmo K-Means presenta difficoltà nella gestione di dati la cui presenza di outliers è troppo elevata, infatti sono spesso eseguite procedure di *Preprocessing* per attenuare la problematica, inoltre, come detto in precedenza la scelta dei centroidi è spesso problematica, soprattutto quando si ha a che fare con dati ad elevata densità. Tuttavia, K-Means risulta uno degli algoritmi più utilizzati soprattutto per quanto riguarda il problema della Customer Segmentation affrontato in questo progetto.

#### **Clustering Gerarchico**

Il clustering gerarchico produce un set di cluster annidati tra loro, organizzati come un albero gerarchico. Si può visualizzare come un Dendogramma e grazie ad esso non è necessario stabilire a priori il numero Ndi cluster da ricavare, poichè esso può essere scelto in seguito prendendo il livello *N* -esimo del dendogramma [19]. Questa tecnica si suddivide in due approcci:

* *Agglomerativo*: Il processo inizia considerando ogni punto come un cluster, ad ogni step si unificano i punti secondo una particolare funzione di similitudine arbitraria, fino ad ottenere un cluster unico ed il relativo dendogramma. Questo approccio si basa sullo sviluppo di una *Matrice di Prossimità* tra i cluster e risulta di fondamentale importanza la funzione per il calcolo della similitudine tra due cluster
* *Divisivo:* Caso complementare in cui si parte da un unico cluster e si suddivide ad ogni iterazione, fino ad ottenere un numero di cluster pari al numero di punti che costituiscono la base dati.

Entrambi gli approcci possono essere interrotti al raggiungimento di un numero di cluster *N* desiderato. Lo spazio richiesto è di *O*(*N* 2) a causa della matrice di prossimità, mentre la complessità è nell’ordine di *O*(*N* 3), dovuto dal fatto che ci sono *N* step ad ogni aggiornamento della matrice di prossimità che deve essere costantemente aggiornata. Come in K-Means, la presenza di outliers condiziona negativamente questo approccio, inoltre presenta difficoltà nella gestione di cluster le cui dimensioni sono molto variabili, per esempio tende a rompere cluster molto grandi a prescindere dalla loro validità.

#### **Density-Based Clustering**

Il density-based clustering si basa sul concetto di *Densità.* L’idea di base è trovare clusters definiti implicitamente da regioni ad alta densità separate da regioni a bassa densità. Uno degli algoritmi più famosi di questa categoria

è il DBSCAN che usa due parametri per identificare aree dense: un raggio *ε*, che serve a identificare un’area attorno ad un determinato punto, e un numero minimo di punti *MinPts* che devono essere presenti all’interno del raggio *ε*. Ogni punto viene etichettato secondo 3 differenti categorie:

* Core Point: tutti i punti che superano la soglia *MinPts* all’interno del raggio *ε*.
* Border Point: tutti i punti che non superano la soglia *MinPts* ma nel loro raggio *ε* hanno almeno un Core Point.
* Noise Point: tutti i punti che non sono Border o Core Point.

L’algoritmo parte da un punto casuale. Sono calcolati tutti i punti compresi nel raggio *ε* e se contiene un numero *MinPts* di punti, viene creato un nuovo cluster altrimenti viene etichettato come Noise-Point. Il punto potrebbe essere successivamente ritrovato in quanto incluso nel raggio *ε* di un vicino e di conseguenza essere inserito in un cluster.

Se un punto è associato ad un cluster, sono inseriti in esso anche i punti presenti all’interno del suo raggio *ε*, e di conseguenza anche i loro vicini all’interno sempre del raggio stabilito. Questo processo continua fino a quando non sono stati inseriti tutti vicini ottenuti, ogni punto a cui è associato un cluster viene marcato come visitato e l’algoritmo prosegue eseguendo la stessa procedura per un punto successivo che non è ancora stato visitato.

L’algoritmo ha complessità *O*(*n*2) che tuttavia può essere ridotta a *O*(*n* log *n*) tramite utilizzo di strutture indicizzate per l’interrogazione del vicinato.

Il punto di forza di questo approccio è dato dalla buona gestione di outliers e dalla conseguente capacità di riuscire a gestire cluster di forme e dimensioni molto differenti. Tuttavia, risulta inefficiente quando si ha a che fare con dati che son caratterizzati da densità troppo variabili

### ***3.2.2 Classificazione***

Con Classificazione si intende il processo che data una collezione di record, denominata Training Set, cerca di costruire un modello in grado di attribuire una caratteristica, denominata attributo Classe, basandosi sulla combinazione delle altre proprietà che caratterizzano il singolo individuo della popolazione. Una volta ottenuto il modello questo può essere usato per predire la classe di nuove istanze di record per cui la classe è sconosciuta. Dopo la costruzione del modello la fase più è la sua validazione, che avviene applicando il modello ad una partizione dei dati disposizione, denominata Test Set. Questo processo è fondamentale per determinare la qualità del modello. In seguito, sono riportati alcuni dei principali metodi presenti in letteratura per affrontare il processo di classificazione:

#### **Decision Trees**

Immagine che contiene testo

Descrizione generata con affidabilità elevata

Figura 12: Decision Tree

Descrive una struttura ad albero [15] dove i nodi foglia rappresentano le classificazioni e le ramificazioni l’insieme degli attributi che le determinano. Di conseguenza ogni nodo interno risulta essere una macro-classe costituita dall’unione delle classi associate ai suoi nodi figli. Il meccanismo si basa sul concetto di Split, ossia come ogni attributo viene suddiviso in base ai valori che può assumere. il tipo di attributo Nominale, Ordinale o Continuo influisce molto sulle tipologie di split, il quale può consistere in una suddivisione binaria oppure multipla.

Per quanto riguarda gli attributi nominali e ordinali, questi split avvengono semplicemente dividendo, in molteplici sottoinsiemi, tutti i valori assunti dall’attributo. Nel caso di attributi continui è opportuno effettuare procedure di discretizzazione per renderlo categorico oppure suddividerlo in sottoinsiemi espressi con disequazioni.

I differenti algoritmi esistenti in letteratura si differenziano in base alla strategia impiegata, ad ogni nodo, per la valutazione dello Split, esistono infatti differenti indici:

* *GINI INDEX* è usato in CART, SLIQ, SPRINT [16]. Considerando p(j t) la frequenza relativa della classe j al nodo t questo è definito come segue:

Immagine che contiene oggetto

Descrizione generata con affidabilità elevata

Nel caso il nodo p sia suddiviso in k partizioni allora la qualità dello split si calcola nel seguente modo, con ni il numero di record al figlio i ed n il numero di record al nodo p:



* *GAIN INDEX* si basa sul concetto di entropia, indice relativo alla omogeneità del nodo. Indicando con p(j t) la relativa frequenza della classe j al nodo t questo indice è definito come segue:



Il Gain, misura la riduzione dell’entropia ottenuta eseguendo un particolare split sul nodo p:



* *CLASSIFICATION ERROR* misura l’errore di classificazione commesso al nodo t. Indicando con p(j t) la relativa frequenza della classe j al nodo t questo indice `e definito come segue:



#### **Classificatori Basati Su Istanze**

Consiste in una famiglia di algoritmi i quali, anzi che eseguire generalizzazioni esplicite, confrontano nuove istanze direttamente con i record analizzati e opportunamente memorizzati dal training set. Degna di nota è la procedura Nearest-Neighbor che utilizza una particolare ed arbitraria metrica per il calcolo della distanza ed un parametro k rappresentante il numero minimo di vicini da estrarre [15].

Per ogni record che deve essere classificato, si calcola la distanza dal training set identificando i k record ritenuti più vicini e si usano i valori assunti dai loro attributi per classificare il record in esame.

#### **Support Vector Machine (SVM)**

La classificazione viene eseguita trovando l’iperpiano che massimizza il margine tra due classi. I vettori (possibili attributi della classe) che definiscono l’iperpiano, sono definiti vettori di supporto. Il vantaggio di questo metodo consiste nel fatto che se i dati sono linearmente separabili, allora esiste un minimo globale unico. Una SVM ideale dovrebbe produrre un iperpiano che separa completamente i vettori di due classi non sovrapposte. In genere, la completa separazione non è sempre possibile, spesso si arriva ad ottenere un modello con troppi possibili casi che comporta una classificazione non corretta [18].

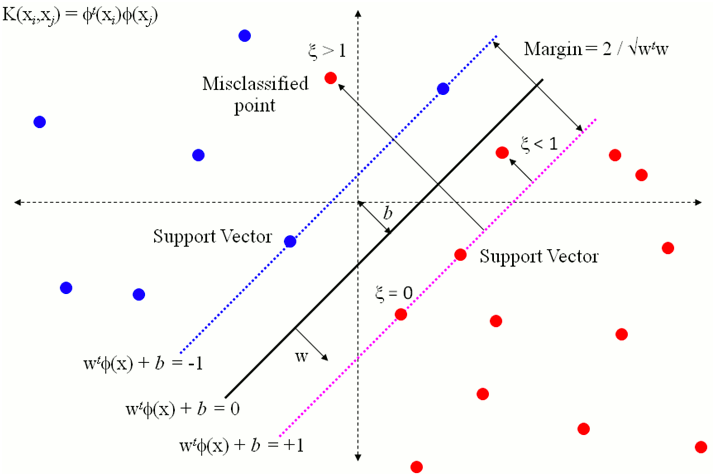


Figura 13. SVM

#### **Classificatori Byesiani**

Consiste in un framework probabilistico per risolvere il problema della classificazione. Si basa fortemente sul concetto di Probabilità Condizionata:

 Ne segue il Teorema di Bayes: 

Si considerano gli attributi e la classe come variabili casuali. Dato un record con attributi (A1, A2, ..., An), l’obiettivo è quello di prevedere la classe C, ossia vogliamo trovare il valore di C che massimizza la probabilità P (C A1, A2, ..., An). Grazie al teorema di Bayes, si ottiene un problema di ottimizzazione equivalente che consiste nel trovare C che massimizza: P (A C) = P(A,C) . Esistono differenti modi per la stima di tale probabilità basandosi sui dati, come distribuzione normale, stima di densità, m-estimate, Laplace [15].

#### **Validazione**

Questo processo è di fondamentale importanza, in quanto permette di valutare le prestazioni del modello costruito e di poterlo confrontare con altre possibili modellazioni. Le misure di valutazione si basano sul Test-Set, partizione dei dati su cui applicare il modello predittivo.

L’applicazione del modello sul Test-Set produce la Matrice di Confusione, ossia matrice indicante l’incidenza tra le classi predette ed il loro valore reale dei record nel Test-Set. Si possono quindi determinare le seguenti tipologie di previsione:

* *True Positive*: Predizioni Positive Corrette
* *False Positive*: Predizioni Negative Corrette
* *True Negative*: Predizioni Positive Errate
* *False Negative*: Predizioni Negative Errate

Queste possono essere applicate a qualsiasi tipologia di attributo, non solamente alle classi binarie. Le metriche più utilizzate sono: Accuracy, Precision, Recall, F-Measure.

**Immagine che contiene screenshot

Descrizione generata con affidabilità molto elevata**

Figura 14: Confusion Matrix

### ***3.2.3 Association Rules***

La base di partenza di un algoritmo per l’estrazione di regole associative è costituita da un insieme di Transazioni. Ogni transazione consiste in un insieme di item. Estrarre le *Regole di Associazione* consiste nel prevedere l’occorrenza di un item in base all’occorrenza di altri item compresi anch’essi nelle transazioni a disposizione.

Risulta importante definire alcuni concetti alla base di questa tecnica:

* *Itemset*: Collezione di uno o più elementi generalmente definito per mezzo del parametro *k*, indicativo della sua dimensione nella forma *k*-*Itemset*.
* *Supporto Itemset*: Dato un itemset *I*, il supporto è la frazione delle transazioni che contengono *I* e si indica con *supp*(*I*).
* *Itemset Frequente*: Tutti gli itemset che superano un’arbitraria soglia minima di supporto.
* Una *Regola di Associazione* è un’implicazione espressa nella forma: *X* → *Y* con *X, Y itemset* dove *X* prende il nome di *Premessa* ed *Y Conseguenza* della regola.

Oltre al supporto, visto precedentemente, esiste un’altra forma di validazione della regole che tiene conto sia della premessa che della conseguenza: la *Confidenza*. Essa indica quanto spesso una particolare regola è verificata, consiste nella proporzione tra il numero delle transazioni che contengono l’intera regole e le transazioni che contengono la premessa:

*conf* (*X* → *Y*) = *supp*(*X* ∪ *Y* ) */ supp*(*X*)

Formalmente il supporto *supp*(*X* ∪*Y* ) può essere riscritto come la probabilità congiunta *P* (*EX EY* ), dove *EX* e *EY* sono tutte le transazioni che contengono *X* o *Y* rispettivamente. Quindi possiamo esprimete la confidenza come la probabilità condizionata *P* (*EY* | *EX*).

∩

|

Dato un set di transazioni, l’obiettivo consiste nell’estrazione di tutte le regole che rispettano le soglie arbitrarie di supporto e confidenza. La loro estrazione non può essere eseguita con un approccio Brute-Force, a causa dell’elevato numero di regole che possono essere generate. Per ridurre il numero di possibili regole, si sfrutta il Principio Apriori.

#### **Principio Apriori**

Questo principio si basa sulla proprietà *anti-monotona* del supporto, che ci permette di stabilire con certezza che se un itemset non risulta frequente, allora nemmeno tutti gli itemset che lo contengono risulteranno frequenti. Tale propriet`a `e cosi formalizzata, con *X* e *Y* itemset:

∀*X, Y* : (*X* ⊆ *Y* ) ⇒ *supp*(*X*) ≥ *supp*(*Y* )

Questa proprietà è alla base dell’algoritmo *Apriori*, procedura che par- tendo da tutti i possibili item con cardinalità 1, costruisce tutti i gli itemset di dimensione *n* + 1 con *n* la dimensione dell’itemset di partenza e ad ogni iterazione verifica se l’itemset generato è frequente o meno.

La proprietà anti-monotona permette di escludere itemset non frequenti e di conseguenza tutti possibili itemset derivanti da essi. Gli step da cui è costituita la procedura sono i seguenti:

|  |
| --- |
| **ALGORITHM APRIORI** |
| 1: **function** Apriori(*set transazioni T* , *minSupport*) |
| 2: *k* = 1 |
| 3: *Generazione itemset con cardinalit*à 1 |
| 4: **repeat** |
| 5: *Generazione itemset di cardinalità* *k* + 1 |
| 6: *Eliminazione itemset contenenti non frequenti* |
| 7: *Calcolo support itemset generati* |
| 8: *Eliminazione itemset non frequenti* |
| 9: **until** *Non sono generati ulteriori itemset frequenti* |

Segue l’algoritmo implementato nel dettaglio:

|  |
| --- |
| **ALGORITHM APRIORI** |
| 1: **function** Apriori(*T, s*) |
| 2: *L*1 ← {*large* 1 − *itemsets*} |
| 3: k=2 |
| 4: **while** *Lk*−1 ƒ= ∅ **do** |
| 5: *Ck* ← *Generate*(*Lk*−1) |
| 6: **for** *transaction t* ∈ *T* **do** |
| 7: *Ct* ← *Subset*(*Ck*1*, t*) |
| 8: **for** *candidates c* ∈ *Ct* **do** |
| 9: *count*[*c*] ← *count*[*c*] + 1 |
| 10: *Lk* ← {*c* ∈ *Ck*|*count*[*c*] ≥ *s*} |
| 11: *k* ← *k* + 1 |
| 12: **return** U*k Li* |

Al termine di questa procedura, otteniamo tutti gli itemset che hanno superato la soglia supporto. Bisogna procedere con l’estrazione delle regole di associazione dagli itemset ottenuti. Le regole generate saranno valutate in base alla loro Confidenza (soglia arbitraria), e quest’ultima generalmente non gode della proprietà *anti-monotona*, ma la confidenza delle regole generate dal

solito itemset possiede la seguente proprietà, indicando con *Conf* (*X* ⇒ *Y*) la confidenza della regola *X* ⇒ *Y*:

*Conf* (*ABC* ⇒ *D*) ≥ *Conf* (*AB* ⇒ *CD*) ≥ *Conf* (*A* ⇒ *BCD*)

Si evince che la confidenza è a*nti-monotona* rispetto al numero di item che compongono la premessa della regola.

Si procede quindi generando le regole che possiedono solo un item nella conseguenza, vengono eliminate tutte le regole che non superano la soglia minima di confidenza. Sulla base delle regole rimaste, si procede generando e valutando le regole con un item addizionale nella conseguenza, procedendo fino a che non sono state generate tutte le possibili regole.

#### **Indici Di Validazione Di una Association Rule**

Le regole estratte sono sottoposte ad un’ulteriore fase di post-processing, in quanto la confidenza può alle volte essere fuorviante come indice di validità per una regola. Questo aspetto emerge per itemset che fanno parte della premessa di una regola, caratterizzati da alto supporto.

Un itemset molto frequente tende ad alzare l’indice di confidenza delle regole di cui esso costituisce la premessa, indipendentemente dal fatto che la regola sia contestualmente valida.

Per avere un’ottima validazione ci si basa sui seguenti indici:

Immagine che contiene oggetto

Descrizione generata con affidabilità molto elevata

Figura 15: Indici Di Validazione Di Una Regola Di Associazione

### ***3.2.4 Artificial Neural Networks & Deep Learning***

Il Deep Learning è un metodo specifico di apprendimento automatico che incorpora reti neurali in strati successivi per imparare dai dati in modo iterativo. È particolarmente utile quando si cerca di imparare i pattern da dati non strutturati e sono progettati per emulare come funziona il cervello umano in modo che i computer possano essere addestrati per trattare problemi che non sono ben definiti [21].

Le reti neurali e l'apprendimento profondo sono spesso usati nelle applicazioni di riconoscimento immagini, parlato e computer vision.

Una rete neurale consiste di tre o più livelli: uno strato di input, uno o più livelli nascosti e un livello di output. I dati sono ingeriti attraverso il livello di input. Quindi i dati vengono modificati nel nascosto livello e i livelli di output in base ai pesi applicati a questi nodi. La tipica rete neurale può essere composta da migliaia o anche milioni di nodi di elaborazione semplici che sono densamente interconnessi.

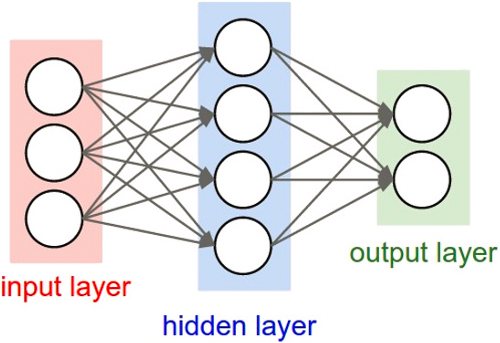


Figura 16: Artificial Neural Network Schema

In una rete neurale gli input inviati ai nodi di input sono costituiti dal valore degli attributi dell’istanza che deve essere analizzata. L’output di questo primo livello della rete rimane invariato, poiché in uscita dai nodi di input sono presenti gli stessi valori che vengono forniti in input. In ogni nodo appartenente ai livelli successivi al primo, nodi nascosti e nodi di output, avviene l’effettiva computazione. Infatti, gli input di questi livelli corrispondono agli output dei livelli precedenti in cui però bisogna considerare il peso associato al collegamento tra i due nodi ed un valore caratteristico del nodo, l’offset. Considerando un nodo 𝑛 tra i nodi nascosti o tra quelli di output, il suo input 𝐼𝑛 è dato dalla seguente relazione:

Immagine che contiene oggetto, orologio

Descrizione generata con affidabilità elevata

dove 𝑤𝑖,𝑛 è il peso del collegamento tra il nodo 𝑖 del livello precedente e il nodo 𝑛 preso in considerazione, 𝑂𝑖 è l’output del nodo 𝑖 del livello precedente e 𝑜𝑓𝑓𝑠𝑒𝑡𝑛 è l’offset associato al nodo 𝑛 considerato.

Ogni nodo, inoltre, applica poi una funzione di attivazionesul valore che riceve in input ed invia il suo output al livello successivo. Infine, quando viene generato l’output dai nodi di output, se durante la fase di apprendimento si verifica un errore tra il valore della classe calcolato e quello previsto per un’istanza, viene calcolato l’errore da ogni nodo di output e viene propagato ai livelli precedenti, dove vengono sistemati i valori dei pesi e degli offset di tutti i nodi di tutti i livelli che costituiscono la rete neurale.

Il termine Deep Learning viene utilizzato quando ci sono più livelli nascosti all'interno di una rete neurale. Usando un iterativo approccio, una rete neurale si adatta e fa continuamente inferenze fino al raggiungimento di un punto di arresto specifico. Praticamente, è una tecnica di apprendimento automatico che utilizza la gerarchia reti neurali per imparare da una combinazione tra algoritmi non supervisionati e algoritmi supervisionati imparando da dati non etichettati e non strutturati. Spesso viene chiamato Deep Learning una sotto-disciplina dell'apprendimento automatico.

Ci sono molte aree in cui l'Deep Learning avrà un impatto imprese. Ad esempio, il riconoscimento vocale avrà applicazioni in molti ambiti, dalle automobili alla gestione dei clienti ma soprattutto viene usato nelle applicazioni dell’Internet of Things (IoT), il deep learning può essere usato per prevedere quando una macchina funzionerà male.

## ***3.3 UTILIZZO NEL FASHION RETAIL: FURLA***

Negli ultimi decenni, lo sviluppo di informazioni e comunicazioni delle tecnologie hanno danno nuova vitalità al marketing aziendale. Ad esempio, la tecnologia del codice e l'emergere di negozi online hanno migliorato notevolmente l’efficienza dell'impresa. I dati stanno aumentando a un ritmo molto rapido, probabilmente 1000 volte rispetto a cinque anni fa. Tuttavia, i dati e gli utili aziendali non lo sono direttamente proporzionale. Sfortunatamente, il cervello umano non può gestire così tanto dati, perciò, la tecnologia di data mining inizia ad assume una posizione molto importante.

La tecnologia Data mining nel marketing è un'applicazione relativamente universale. Queste applicazioni sono riferite a una Boundary Science, perché impostate su una varietà di teorie scientifiche basate principalmente su due discipline di base: Information Technology e Marketing [20].

Molto importante è lo studio dei metodi Statistici che sta alla base di ogni possibile algoritmo. Inoltre, il data mining fa riferimento anche a discipline letterarie e comportamentali per valutare meglio le caratteristiche di un cliente, come la psicologia e la sociologia.

In generale, attraverso l’estrazione, il trattamento e lo smaltimento del una grande quantità di informazioni che riguardano il comportamento del consumatore sono utili per identificare l’interesse di specifici gruppi di consumatori o individuali, le abitudini di consumo, le preferenze dei consumatori e soprattutto la domanda, orientando la vendita ai gruppi di consumatori per un marketing dal contenuto specifico. Questa è l'idea di base.

Poiché l'automazione è popolare in tutto il settore, le imprese che gestiscono i processi devono avere molti dati operativi. I dati non sono raccolti allo scopo di analisi, ma provengono da operazioni commerciali. L'analisi di questi dati conferisce al decision-maker il valore reale informazioni, al fine di ottenere profitti. Le informazioni commerciali provengono dal mercato attraverso vari canali come, ad esempio, il processo di acquisto tramite credito carta dove possiamo raccogliere i dati di consumo del cliente, come ora, luogo, beni o servizi interessanti interessati, prezzi voluti e il livello di capacità di ricezione. Inoltre, le imprese possono anche acquistare una varietà di informazioni sui clienti da altri società di consulenza.

Il marketing basato sul data mining di solito può creare sulle vendite delle promozioni specifiche per il cliente secondo i suoi precedenti record di acquisto.  Le più comuni applicazioni nel settore bancario, assicurativo, sistema di traffico, vendita al dettaglio e in campo commerciale.

Come già descritto nello Stato dell’arte le tecnologie e le analisi del marketing sono basate sull’analisi del mercato, come il Database Marketing, Segmentazione e classificazione del cliente, analisi del profilo e cross-selling. Possono essere utilizzate anche per valutazioni di valutazione del credito e frode.

***Immagine che contiene testo

Descrizione generata con affidabilità molto elevata***

Figura 17: Applicazione del data mining per il marketing

Il processo di base del data mining nel marketing mostra come segue:

* *Preparare i dati primitivi*. Include il carattere individuale informazioni (età, sesso, hobby, background, professione, indirizzo, codice postale e reddito), la precedente esperienza di acquisto e la relazione all'interno dei clienti. La preelaborazione dei dati primitivi è molto importante per selezionare i potenziali clienti.
* *Stabilire un determinato modello*. Questo modello può utilizzare molto tecnologie tradizionali di data mining e molte tecnologie da altri argomenti correlati. Tuttavia, il problema che tali tecnologie dovrebbe risolvere è quello di individuare il mercato migliore o accettabile, all'interno fonte di dati limitata, tempo limitato e spese limitate.
* Utilizzare i dati di test per ottenere ciascuno modello o parametro. In definitiva, utilizzare questo modello per selezionare i clienti e decidere il piano di marketing.

### ***~~3.3.1 Clustering: K-Means~~***

### ***~~3.3.2 Classificazione: Decison Tree~~***

### ***~~3.3.3 Association Rules: Apriori~~***