

POLITECNICO DI TORINO

Corso di Laurea in Engineering & Management

Master's Degree Thesis

*Creation Of A Data Mart For Evaluate The Closing
Reasons And The Best Geo-Locations For A Fashion
Retail Store*



Supervisor

Prof. Marco Cantamessa

Dott. Vincenzo Scinicariello

Candidate

Luca Bregata

Accademic Year 2018/2019

POLITECNICO DI TORINO

Corso di Laurea in Engineering & Management

Master's Degree Thesis

*Creation Of A Data Mart For Evaluate The Closing
Reasons And The Best Geo-Locations For A Fashion
Retail Store*



Supervisor

Prof. Marco Cantamessa

Dott. Vincenzo Scinicariello

Candidate

Luca Bregata

Accademic Year 2018/2019

Alla Mia Famiglia e
alla mia ragazza
per il supporto e l'aiuto
ricevuto durante
questo viaggio.

SOMMARIO

<i>SOMMARIO</i>	7
<i>INDICE DELLE FIGURE</i>	9
<i>INDICE DELLE TABELLE</i>	10
<i>ABSTRACT</i>	11
<i>INTRODUZIONE</i>	13
METODOLOGIA	16
TOP-DOWN ANALISI DEL PROBLEMA	17
OBIETTIVO	19
PROJECT SCHEDULING	20
<i>CAPITOLO 1: STATO DELL'ARTE</i>	21
1.1 BUSINESS INTELLIGENCE	21
1.2 DATAWAREHOUSE	23
1.2.1 Architettura Di Un Data Warehouse	25
1.2.2 Extraction, Transformation & Loading (ETL)	27
1.2.2.1 Extraction.....	28
1.2.2.2 Trasformation	28
1.2.2.3 Loading	29
1.2.2.4 Possibili Problemi Nell' ETL E Come Risolverli	30
1.3 OLTP vs OLAP	31
1.4 COSA SI INTENDE CON IL TERMINE BIG DATA	33
1.4.1 Benifici e Barriere Sull'utilizzo Dei Big Data	35
1.4.2 Tecniche Per L'analisi Dei Big Data	37
1.5 PROGETTI DI BIG DATA NEL MARKETING	38
1.5.1 Direct E Digital Marketing	39
1.5.2 Customer Micro-Segmentation	40
1.5.3 Price Optimization	40
1.5.4 Location-Based Marketing	41
1.5.5 In-Store Analysis	42
1.5.6 Cross-Selling / Up-Selling.....	43
1.6 KNOWLEDGE DISCOVERY IN DATABASE (KDD)	44
1.6.1 Data Mining vs Machine Learning.....	46
1.7 ALGORITMI DI DATA MINING	48
1.7.1 Clustering	51
1.7.1.1 Clustering Basato su Centroidi (K-Means).....	52
1.7.1.2 Density-Based Clustering.....	54
1.7.2 Classification And Regression Trees (CART).....	55
1.7.2.2 Altri Tipi Di Classificatori.....	57
1.7.3 Predizione: Association Rules	59
1.7.3.1 Principio Apriori	60
1.7.4 Artificial Neural Networks & Deep Learning	62
1.7.5 Regressione Lineare	64
<i>CAPITOLO 2: TRADITIONAL ETL PER LA CREAZIONE DELLA DATA MART</i>	68
2.1 TALEND OPEN SOURCE	70
2.2 CREAZIONE DELLA DATA MART	72
2.2.2 Storicizzazione	75

2.2.3 Il Modello Multidimensionale - Dimensional Fact Model	76
2.3 LEVEL L0 - DATA INGESTION	78
2.3.1 I Metadati	79
2.4 LEVEL L1 - DATA OPERATION.....	84
2.4.1 Data Quality	85
2.4.2 Tmap Component In Talend Open Source	87
2.5 LEVEL L2 – ETL DATA MART BEST PRACTICE	89
2.6.1 Snowflake Schema	94
2.6 LEVEL L2 – VISUALIZATION DATA MART BEST PRACTICE.....	96
2.6.1 Star Schema	98
2.7 FULL LOAD ETL.....	99
2.7.1 Auditing ETL.....	100
CAPITOLO 3: ALGORITMI DI DATA MINING PER IL FASHION RETAIL.....	102
3.1 PREDIZIONE DELLA BEST GEO - LOCATION PER APRIRE UN NUOVO NEGOZIO	104
3.1.1 Regressione dei dati ISTAT	106
3.1.2 Processo ETL dei dati ISTAT.....	109
3.1.3 Best Geo-Locations	110
3.2 CLASSIFICAZIONE: CART.....	112
3.2.1 Training & Test Set.....	113
3.2.2 Classificazione E Previsione Delle Cause Della Chiusura Dei Negozi.....	114
CAPITOLO 4: DATA VISUALIZATION.....	118
4.1 MICROSOFT POWER BI.....	119
4.2 STRUMENTI UTILIZZATI E RISULTATI OTTENUTI TRAMITE LA DATA VISUALIZATION.....	120
4.2.1 Visualizzazioni	121
4.2.4 Report	123
4.2.3 Dashboard	125
4.2.4 Dataset	127
CONCLUSIONI.....	128
RESULTS.....	128
SVILUPPI FUTURI: REAL-TIME BUSINESS INTELLIGENCE	129
APPENDICE.....	132
A1. SQL - CREAZIONE DELLE SURROGATE KEY.....	132
A2. MATRICE DI SELEZIONE DATI ISTAT	133
A3. R-CODE: PREDIZIONE DATI ISTAT	134
A4. SQL-CODE: BEST GEO-LOCATIONS.....	136
A5. SQL-CODE: AGGREGATE FACT SALES PER LA CREAZIONE DEL MODELLO CART	137
A6. R-CODE: CALSSIFICATION AND REGRESION TREE (CART)	140
A7. DAX-CODE: VARIABILI DI POWER BI	144
REFERENCES.....	146

INDICE DELLE FIGURE

<i>FIGURA 1: PROJECT SCHEDULING</i>	20
FIGURA 2: DATAWAREHOUSE	24
FIGURA 3: MODELLO DI INMON	26
FIGURA 4: MODELLO DI KIMBALL	26
FIGURA 5: LE 5V DEI BIG DATA	34
FIGURA 6: KDD PROCESS	44
FIGURA 7: DATA MINING ALGORITHMS	49
FIGURA 8: K-MEANS ALGORITHM	54
FIGURA 9: CONFUSION MATRIX	59
FIGURA 10: INDICI DI VALIDAZIONE DI UNA REGOLA DI ASSOCIAZIONE	62
FIGURA 11: ARTIFICIAL NEURAL NETWORK SCHEMA	63
FIGURA 12: LINEAR REGRESSION	66
FIGURA 13: LEVEL TREE	73
FIGURA 14: HYPERCUBE OLAP	77
FIGURA 15: CONNESSIONE AL DATABASE	80
FIGURA 16: CARICAMENTO DA FILE DELIMITED	82
FIGURA 17: CARICAMENTO DA FILE EXCEL	82
FIGURA 18: MULTI-CARICAMENTO	83
FIGURA 19: MONO-CARICAMENTO IN L0	84
FIGURA 20: PRE-LOADING L1	85
FIGURA 21: JOIN & MAPPING IN TMAP	87
FIGURA 22: TRASFORMATION IN TMAP	88
FIGURA 23: SNOWFLAKEDB DIMENSION	92
FIGURA 24: SNOWFLAKEDB FACT	93
FIGURA 25: SNOWFLAKE SCHEMA	95
FIGURA 26: JOB PRODUCT STAR SCHEMA	96
FIGURA 27: TMAP PRODUCT STAR SCHEMA	97
FIGURA 28: STAR SCHEMA	98
FIGURA 29: JOB STG ANAGRAFICHE	99
FIGURA 30: FULL LOAD ETL & AUDITING	100
FIGURA 31: APPLICAZIONE DEL DATA MINING PER IL MARKETING	103
FIGURA 32: JOB ISTAT EXCEL	105
FIGURA 33: JOB ISTAT WITH PREDICTION	109
FIGURA 34: STAR SCHEMA FACT ISTAT	110
FIGURA 35: CART	116
FIGURA 36: RESULTS OF PREVISION	117
FIGURA 37: VISUALIZZAZIONE 1Q 2019	122
<i>FIGURA 38: MAPPA DELLA CLASSIFICAZIONE DEI NEGOZI</i>	124
<i>FIGURA 39: TABELLA DELLA CLASSIFICAZIONE DEI NEGOZI</i>	124
FIGURA 40: DEFAULT DASHBOARD VIEW	125
<i>FIGURA 41: VISTE IN PROFONDITÀ DI UNA DASHBOARD</i>	126
FIGURA 42: DATASET RELATIVO AL CONTO ECONOMICO AZIENDALE	127
FIGURA 43: PLOT AG EXPENSES ABOUT LIGURIA	135

INDICE DELLE TABELLE

TABELLA 1: OLTP VS OLAP	32
TABELLA 2: DATA MINING VS MACHINE LEARNING [13]	47
TABELLA 3: DATAWAREHOUSE VS DATA MART	72
TABELLA 4: LIVELLI ETL	74
TABELLA 5: METADATI	81
TABELLA 6: DATA QUALITY	86
TABELLA 7: RANKING DELLE REGIONI	111
TABELLA 8: RANKING DEI COMUNI	112
TABELLA 9: MATRICE DATI ISTAT	133

ABSTRACT

L'evoluzione della Business Intelligence è iniziata decenni fa con i primi report mainframe, chiamati output di sistema. Essi venivano principalmente stampati su carta, per poi essere periodicamente distribuiti ai manager. Le prime query hanno velocizzato il processo e hanno consentito ai manager tecnicamente esperti di creare report personalizzati ad hoc, ma pochi manager avevano il tempo e le competenze per farlo. L'emergere del data warehouse ha dato un grande impulso alla BI aggregando tutti i dati in un'unica posizione, dove potrebbe essere interrogato in modo interattivo senza impatto sulle applicazioni tramite l'uso di Query e rapporti online con interfacce grafiche sempre più facili da utilizzare.

L'avvento delle data warehouse, delle data mart e gli strumenti di analisi analitica hanno reso la BI accessibile a più gestori e hanno permesso ai manager di ottenere informazioni e risposte critiche in modo efficiente e rapido.

Il progetto proposto sarà dedicato alla descrizione in dettaglio della creazione di una data mart dedicata alle vendite di una azienda del fashion attraverso una soluzione ottimale di best practice di un processo ETL ottenendo come risultato lo Snowflake schema e lo Star schema, ideale per la data. Inoltre, utilizzando il processo di classificazione comprendente sia i dati aziendali che gli open data, si ha avuto la possibilità di localizzare le area più efficaci per aprire un nuovo negozio e di offrire una spiegazione sul perché alcuni negozi sono stati chiusi in un recente passato.

Le conclusioni, saranno visualizzate in Power BI, software Microsoft per la data visualization.

INTRODUZIONE

Man mano che diventiamo una società digitale, la quantità di dati creati e raccolti cresce e accelera in modo significativo. L'analisi di questi dati diventa una sfida per gli strumenti analitici tradizionali che fanno sempre più fatica a stare al passo. È necessaria quindi una costante innovazione per colmare il divario tra i dati generati e i dati che possono essere analizzati in modo efficace.

I grandi strumenti e le tecnologie di dati offrono opportunità e sfide nel poterli studiare in modo proficuo per comprendere meglio le preferenze dei clienti, ottenere un vantaggio competitivo sul mercato e far crescere il loro business.

Le architetture di gestione dei dati si sono evolute dal tradizionale modello di data warehousing ad architetture più complesse che soddisfano requisiti differenti, come l'elaborazione in tempo reale e in batch, dati strutturati e non strutturati, transazioni ad alta velocità, etc.

Mediamente Consulting s.r.l., nata nel 2012, è una società di consulenza e progettazione specializzata in sistemi a supporto delle decisioni. In particolare, si occupa di progetti di business intelligence, data warehouse e advanced analytics in ambito Big Data.

Grazie a tali sistemi, il cliente ha la possibilità di visualizzare le informazioni e conseguentemente di prendere una decisione più consapevole basata su fatti. Pertanto, Mediamente Consulting s.r.l. supporta il cliente nella conoscenza delle proprie performance e lo aiuta ad incrementarle attraverso decisioni migliori.

Per lo svolgimento del mio progetto di tesi, sono stato inserito in un team che segue un'importante azienda di moda, che risponde alle esigenze analitiche in ambito fashion.

Il cliente si occupa di gestire diversi settori in ambito "Fashion Retail". Si tratta di una holding multinazionale responsabile delle vendite di prodotti in tutto il mondo che sta mettendo a punto una forma innovativa di gestione delle operazioni di analisi strategiche e operative che consente di cogliere appieno il potenziale di crescita del suo brand,

all'interno di un mercato globale considerevole e molto competitivo, in cui il segmento di mercato sta godendo di una crescita sostanziale.

Di seguito andrò a presentare brevemente e schematicamente i capitoli in cui si compone il mio elaborato. L'illustrazione dell'intero lavoro svolto, è stato possibile grazie all'ausilio di concetti teorici fondamentali e all'aiuto fornito dai miei colleghi in ambito lavorativo.

Il primo capitolo può essere considerato un'introduzione ai concetti fondamentali dei Big Data e in generale della Business Intelligence, che rappresentano lo Stato Dell'arte. Inoltre, si spiegheranno i meccanismi usati e i concetti chiave del data mining e del machine learning.

La parte centrale del mio elaborato, che corrisponde ai capitoli 2-3, sarà dedicato alla descrizione delle analisi condotte e della metodologia usata da cui sono partito per lo sviluppo del mio progetto di tesi. Sarà spiegato in dettaglio il processo ETL aziendale ottenendo come risultato lo Star Schema, ideale per la data visualization, e lo Snowflake schema, ideale per la struttura di un processo Extraction, Trasformation & Loading. Utilizzando il processo di classificazione comprendente sia i dati aziendali che i dati ISTAT ricavati dall'omonimo sito, si ha avuto la possibilità di localizzare le area più efficaci per aprire un nuovo negozio e di offrire una spiegazione sul perché alcuni negozi sono stati chiusi.

In questo modo si potrà avere una panoramica generale dei concetti applicati per svolgere il lavoro.

Infine, nell'ultimo capitolo, descriverò l'intero processo del progetto per poi concludere andando a delineare nel dettaglio i dati ottenuti tramite l'utilizzo di Power BI sul contesto applicativo in cui si è svolto il progetto, osservando e analizzando le caratteristiche e scelte implementative ottenendo così una visione strategica e decisionale che l'azienda potrà decidere di implementare o meno.

Questo capitolo risulta il più importante, in quanto verrà descritto il mio contributo al progetto, ovvero lo sviluppo di un metodo per fare reportistica su dati live, attraverso una analisi top-down, partendo dalle esigenze dell'utente e una datawarehouse già completa, per costruire le Query e Fact Table per la realizzazione di data mart formando

un Database contenuto, che sarà essenziale per la progettazione e visualizzazione dei report finali.

METODOLOGIA

In questo paragrafo saranno illustrate le fasi principali del progetto aziendale.

1. **Top-Down Analysis** per comprendere e analizzare in modo completo e efficace tutte le caratteristiche rilevanti del progetto, ponendosi domande e provando a capire come ottenere risposte da esse, per poi arrivare ad ottenere una base i dati che serviranno per raggiungere l'obiettivo finale;
2. **Costruzione della Data Mart** tramite l'utilizzo di Talend Open Source, software usato per il Traditional ETL, evidenziando tutti i processi aziendali dalla Staging area della datawarehouse, dove si importeranno i file di origine (csv, Excel...);
3. **Creazione dell'ETL Best Practice Model e dello Snowflake schema** grazie alla creazione di surrogate key per le relazioni tra le tabelle, ottimizzandone le prestazioni;
4. **Creazione della Visualization Data Mart Best Practice Model e dello Star schema**, entrambi utili per essere efficienti nella rappresentazione di vari report utili per le finali decisioni aziendali del cliente;
5. **Creazione di un modulo per velocizzare il processo di aggiornamento** delle tabelle, raggruppando i job appartenenti alla stessa famiglia (Anagrafiche e Movimenti) in modo che si eseguano in contemporanea e grazie ad un sistema di avvisi di Warning/Error con conseguente invio di Email, di avere subito un Audit del problema;
6. **Utilizzo di software di Data Mining e Machine Learning** che per mezzo dell'intelligenza artificiale, implementano autonomamente diversi algoritmi in grado di eseguire un processo sempre più ottimizzato unendo i dati del datawarehouse e i dati ISTAT italiani, precedentemente estratti;
7. **Creazione di una Dashboard in Power BI** per avere un controllo e una serie di dati effettivi su tutto quello che riguarda le vendite. Grazie a questo processo, è possibile estrarre risultati da fornire ai clienti in modo chiaro ed esaustivo, aiutandoli a comprendere i problemi dei loro sistemi di produzione e commercio e migliorando il loro Data Driven Strategy, identificato come la scelta di efficaci e nuove azioni manageriali.

TOP-DOWN ANALISI DEL PROBLEMA

Attributi aziendali di successo nel campo della moda:

- posizione strategica;
- qualità/prezzo;
- diversificazione dei prodotti;
- innovazione;

Obiettivo aziendale:

- Guadagnare il più possibile avendo il minimo inventario in magazzino = costi;
- Aumentare le vendite.

Competitors:

Aziende della moda che usufruiscono della stessa tipologia di mercato.

Data Mining – Classificazione e regressione tramite algoritmi che usano l'intelligenza artificiale:

- Country;
- Tipo di negozio (Retail, Outlet, ...);
- Causa di una chiusura;
- Rimarrà aperto?
- Ranking per la geo-positioning per aprire un nuovo negozio.

Questions

- Meglio una vetrina elegante o stravagante per attirare i clienti?
- Dopo quanto devo spostare la merce da un negozio retail ad uno outlet?
- Come posso effettuare il mio riassortimento?
- Quante volte un mio cliente ritorna a comprare? Quante volte invece torna a comprare ma attraverso l'e-commerce?
- Che rischio può permettersi questo negozio? Quanta merce posso rischiare di lasciare invenduto?
- Quale particolare mi differenzia e mi fa subito riconoscere da competitors?

- Meglio aprire negozi di nicchia nelle grandi città o aprire negozi negli outlet in punti strategici?
- Quando devo creare una promozione e per chi? Meglio a breve o lungo termine? Mi conviene fare promozioni su un singolo prodotto o basandomi sulle regole di associazione? Quando il mio numero delle vendite è sotto la media?
- Quanto cresce/diminuisce il mio marchio rispetto al settore?
- Come si stanno muovendo i miei competitors?
- Mi conviene puntare più su una clientela puramente femminile o anche maschile?
- Quanto un periodo di saldi incide sul numero di vendite? Quanto incide, invece, sul Margine complessivo aziendale?

Dati Necessati Per L'analisi

- Prodotto (id, descrizione, categoria, data di uscita, stagionalità, caratteristiche, sesso);
- Negozi (id, posizione geografica, descrizione, tipo/canale);
- Scontrini (id, id_data, prodotto, id_cliente, valore, sconto, quantità);
- Date (id, giorno, mese, quadrimestre, semestre, anno);
- Dati pubblici commerciali: bilancio, Dati ISTAT su popolazione, turismo ,indicatori economici.

OBIETTIVO

Il panorama della moda è stato ampiamente studiato per migliorare capire come i leaders influenzano i loro seguaci e, in cambio, fornire informazioni preziose ai decisori sul lancio e sul marketing del prodotto strategie. L'obiettivo del nostro studio è ottenere tramite il processo ETL, del Best Practice per la creazione di una data warehouse o di una data mart per un Cliente, sfruttando i dati ottenuti per ottenere delle analisi di business sulle vendite, con risultati chiari ed efficaci.

In particolare, Il progetto verterà in una analisi approfondita sul territorio italiano e principalmente sullo studio dei negozi, evidenziando tramite l'utilizzo di algoritmi di machine learning e data mining, le cause principali della chiusura di essi e la zona geografica ideale per aprire un nuovo store.

PROJECT SCHEDULING

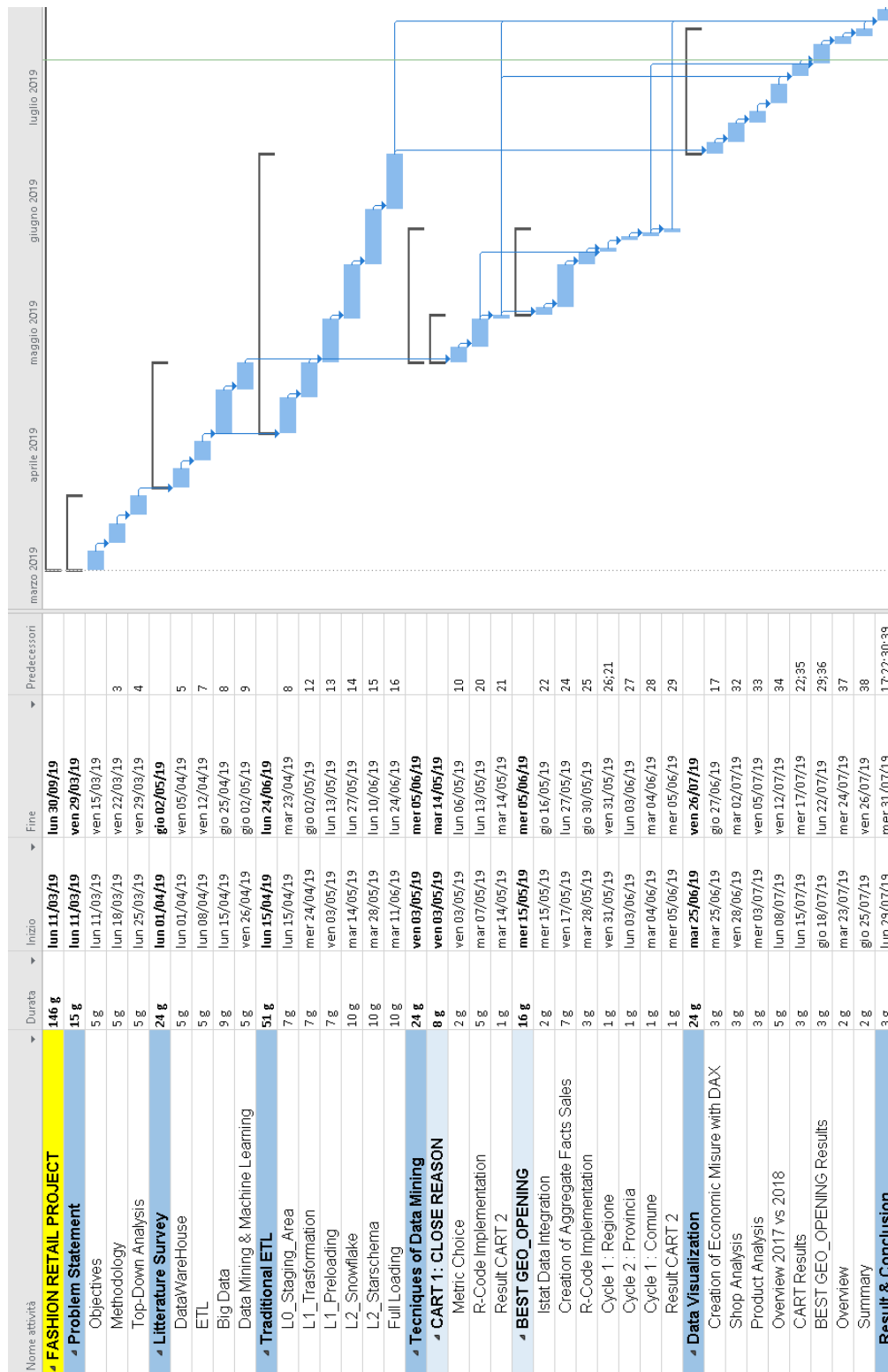


Figura 1: Project Scheduling

CAPITOLO 1: STATO DELL'ARTE

1.1 BUSINESS INTELLIGENCE

Con il termine Business Intelligence (BI) ci si riferisce ad una serie di processi aziendali che ruotano attorno ai dati, con operazioni di raccolta, elaborazione, analisi, cui scopo è quello di produrre informazioni al servizio del management strategico e tattico, che trova supporto analitico, storico e previsionale della Data Driven Strategy. La BI è stata collocata altresì nel sottoinsieme operativo, poiché sta assumendo un ruolo sempre più importante anche nelle normali attività giornaliere delle aziende.

Nell'ambito lavorativo moderno, il cui scopo principale è fare business e diventare leader di mercato, le aziende si trovano sempre più frequentemente a confrontarsi con realtà differenti dalla propria. Ciò avviene tramite l'analisi del comportamento dei competitors facenti parte dello stesso settore e lo studio del mercato in cui si trovano.

L'adozione della BI da parte delle imprese permette una conoscenza più approfondita non solo di loro stesse ma anche del mercato di riferimento.

Nel periodo attuale, il “cambiamento” è all'ordine del giorno, pertanto saper leggere in anticipo le tendenze dei mercati è un fattore competitivo a cui non si può e non si deve rinunciare.

Data l'elevata mole di dati generata ogni giorno, diventa necessario trovare un metodo che:

- Permetta di raccogliere e processare dati ad alta velocità (sempre più spesso si parla di processi real-time);
- Fornisce un servizio di pulizia del dato stesso, eliminando dati sporchi, ridondanti o errati tramite i processi di ETL “Extraction, Transformation & Loading” che prelevano i dati dai sistemi alimentanti (ERP, fogli Excel etc.) e li portano nel DWH certificandoli

attraverso processi di data quality. Questo processo sarà spiegato in maniera specifica nel cap.2;

- Definisce un sistema consolidato e stabile di memorizzazione per i dati certificati (data warehouse);
- Trasforma l'informazione in fonte di conoscenza attraverso analisi di business sui dati stessi, determinando nuovi KPI.

I Big Data provengono da diverse fonti, sia interne che esterne, spesso sono in formati differenti e risiedono in posizioni multiple in numerosi sistemi legacy e altre applicazioni. I dati possono essere strutturati (dati conservati in Database relazionali, organizzati secondo schemi e tabelle rigide), non strutturati (dati conservati senza alcuno schema come forme libere di testo tra cui articoli e parti di e-mail, audio senza tag, immagini e video) o semi-strutturati (dati che presentano caratteristiche sia di quelli strutturati che di quelli non strutturati; un esempio è rappresentato dai file compilati con sintassi XML per i quali non ci sono limiti strutturali all'inserimento dei dati, ma le informazioni vengono organizzate secondo logiche strutturate e interoperabili). Dopo che i dati sono stati uniti, questi hanno bisogno di essere processati o trasformati, essendo in uno stato grezzo.

Il passo successivo consiste nella scelta della piattaforma e della tecnologia da utilizzare per le applicazioni di Big Data analytics che includono queries, reports, OLAP e data mining e alla visualizzazione, compresa in tutte queste applicazioni [22].

Un ruolo centrale in quest'ambito viene svolto dai Big Data analytics e tecnologie di business intelligence basate su come:

- CRM & Customer Analytics: soluzioni e tecnologie che raccolgono, organizzano e sintetizzano i dati dei clienti per aiutare le organizzazioni a risolvere i problemi di business riguardanti i consumatori attraverso tool, dashboard, portali e altri metodi negli ambiti di Marketing, Sales e Customer Service; i consumatori vengono poi segmentarli in gruppi sulla base dei comportamenti adottati, implementare azioni di Marketing personalizzate e determinare trend generali;
- Predictive Analytics: Analytics avanzati che implementano tecniche quali la regressione, i modelli predittivi e la statistica per analizzare i dati e i contenuti e rispondere alle domande "Cosa succederà" o "Cosa accadrà molto probabilmente?";

- Social Analytics: tools che estraggono, analizzano e sintetizzano automaticamente i contenuti generati dagli utenti online. Questa tecnologia verrà descritto in modo approfondito nel successivo capitolo;
- Text Analytics: processo di estrazione delle informazioni dai testi, utilizzato per diversi scopi, tra cui il *riepilogo*, ovvero il tentativo di trovare i contenuti chiave in un grande insieme di informazioni, la *sentiment analysis*, già spiegate o per determinare cosa ha guidato un determinato commento di una persona e quindi per un fine esplicativo;
- Web Analytics: applicazioni analitiche utilizzate per capire e migliorare l'esperienza online del consumatore, l'acquisizione di utenti e l'ottimizzazione del digital Marketing e delle campagne pubblicitarie. Questi offrono reporting, segmentazione, gestione delle campagne e integrazione con altre fonti dati e processi;
- Prescriptive analytics: utilizzano la tecnologia di ottimizzazione per risolvere decisioni complesse con un numero molto elevato di variabili decisionali, vincoli e compromessi, fornendo alle organizzazioni suggerimenti per eseguire azioni ottimali per il raggiungimento degli obiettivi aziendali.

1.2 DATAWAREHOUSE

I Data Warehouse (DWH) sono il principale strumento a supporto della Business Intelligence. Essi permettono di collezionare dati integrati, consistenti e certificati, afferenti a tutti i processi di business dell'azienda e provenienti dalle fonti operazionali. Questi dati vengono in seguito opportunamente trasformati attraverso procedure ETL e controllati attraverso il sistema di data quality.

La qualità dei dati è un requisito fondamentale per l'intero sistema informativo, in quanto, se i dati risultano sporchi, possono oltre che causare un peggioramento delle performance aziendali, portare a prendere decisioni inopportune, comportando costi aggiuntivi e perdita di opportunità.

L'obiettivo di un DWH è pertanto quello di supportare il “*knowledge Worker*” (dirigente, amministratore, gestore, analista) per aiutarlo a condurre analisi finalizzate all'attuazione di processi decisionali e al miglioramento del patrimonio informativo, e

fornire un unico punto di accesso per tutti i dati dell'azienda resi consistenti e affidabili attraverso i processi di ETL. Il datawarehouse garantisce inoltre una profondità storica completa dei dati, poiché in esso viene persistito anche lo stato passato delle informazioni permettendo così un'analisi temporale.

Dovranno quindi essere attentamente progettati per gestire in maniera efficiente ed efficace le caratteristiche dei Big Data.

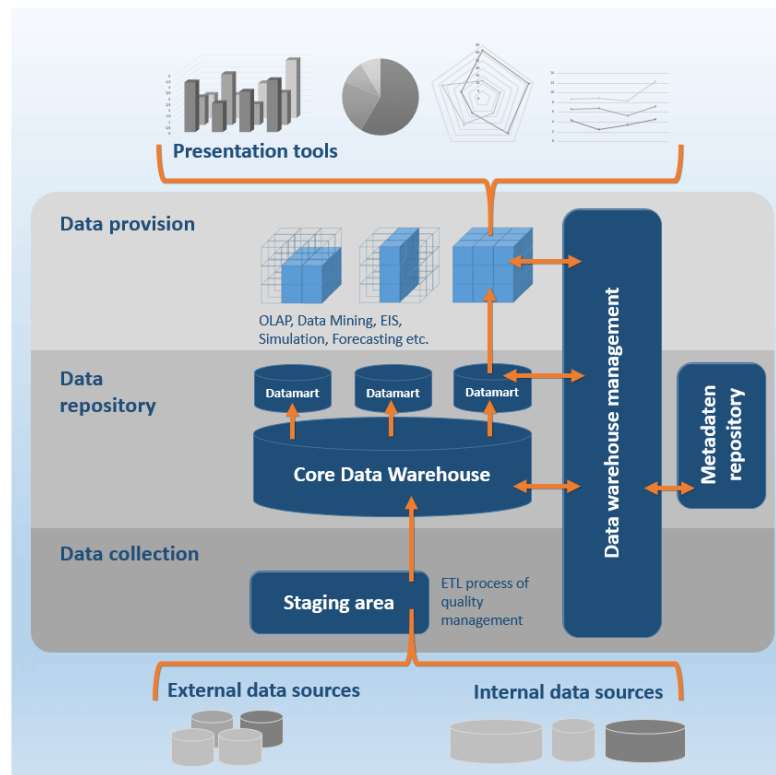


Figura 2: Datawarehouse

I Datawarehouse sono realizzati come principale base per il Decision Support System (DSS), cioè un *sistema* di supporto alle decisioni è un sistema in grado di fornire chiare informazioni agli utenti, in modo che essi possano analizzare dettagliatamente una situazione e prendere le opportune decisioni sulle azioni da intraprendere in modo facile e veloce [12]. Il DSS si appoggia su dati di uno o più database, spesso organizzati in strutture diverse con dati non omogenei.

In altre parole, un sistema di questo tipo deve supportare le attività di analisi e controllo manageriale di routine, le attività di ricerca delle cause di un problema (*focused search*) e le attività di gestione manageriale complessa (*decision making*), permettendo inoltre

un facile utilizzo ad un'utenza con un tempo disponibile ridotto e riluttante verso nuove tecnologie (soprattutto nei casi in cui non riesce a percepire in breve tempo i benefici).

Andiamo a descriverne in dettaglio le caratteristiche:

- *Orientato al soggetto*: nel data warehouse i dati sono organizzati per soggetti rilevanti come, per esempio, i prodotti, i clienti, i fornitori e il periodo di tempo, al fine di offrire tutte le informazioni inerenti una specifica area;
- *Integrato*: il data warehouse deve essere in grado di integrarsi perfettamente con la moltitudine di standard utilizzati nelle diverse applicazioni. I dati devono essere ricodificati, per risultare omogenei dal punto di vista semantico, e devono utilizzare le stesse unità di misura;
- *Variabile nel tempo*: a differenza dei dati operazionali, quelli di un data warehouse hanno un orizzonte temporale molto ampio (anche 5-10 anni), risultando riutilizzabili in diversi istanti temporali;
- *Non volatile*: i dati operazionali sono aggiornati in modo continuo; nel data warehouse i dati sono caricati inizialmente con processi integrali e successivamente aggiornati con caricamenti parziali; i dati, una volta caricati, non vengono modificati e mantengono la loro integrità nel tempo.

È possibile che un Datawarehouse sia suddiviso in diversi data mart, ognuno dei quali specifico per un solo processo di business fra quelli presenti all'interno dell'azienda (ordini, vendite, clienti, marketing, etc.). Nel capitolo 2 vedremo, per l'appunto, come una data mart relativa ad un fashion retailer.

1.2.1 Architettura Di Un Data Warehouse

In fase di progettazione risulta fondamentale stabilire quali tipologie di architettura adottare. Chiaramente, da quando sono stati idealizzati, i modelli (descritti successivamente) si sono evoluti e, di conseguenza, un DWH deve essere costruito secondo i principi moderni [10].

I pattern descritti in questo paragrafo rimangono comunque delle basi da cui partire.

Modello di Inmon - Corporate Information Factory: I Datawarehouse si costruiscono nella loro totalità fin dal principio come un unico blocco monolitico; non è possibile vederli come la composizione dei DM. Viene adottata una visione Top-Down.

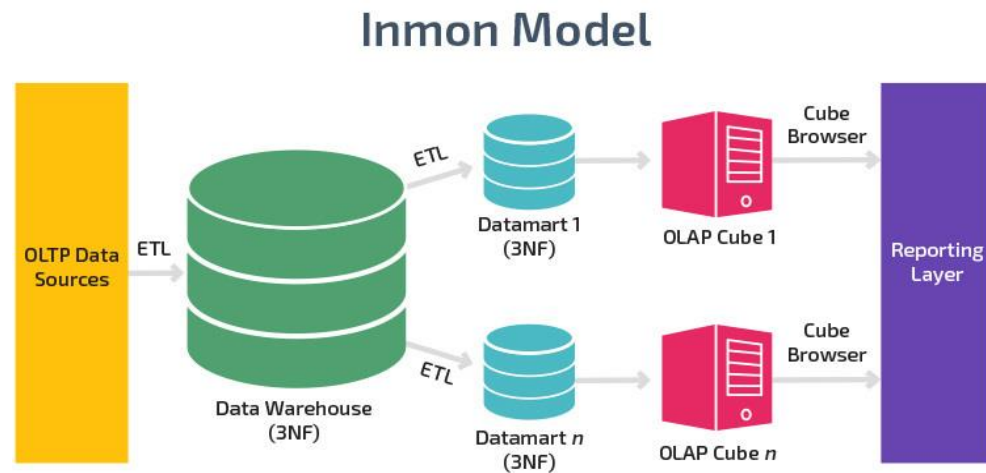


Figura 3: Modello di Inmon

Modello di Kimball - Dimensional Model: adotta un approccio Bottom-up in cui il Datawarehouse nasce dall'unione dei vari data mart che riferiscono ognuno ad una specifica area di business.

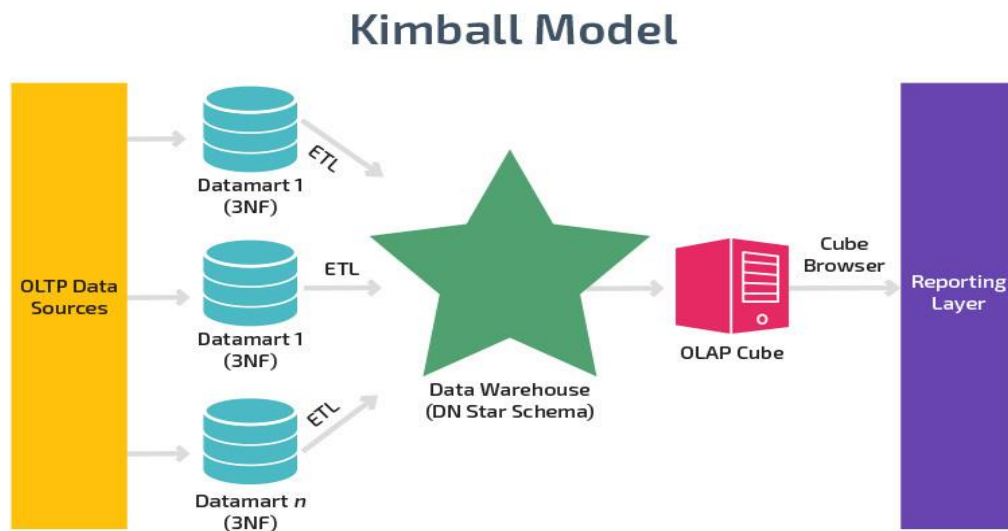


Figura 4: Modello di Kimball

È stato dimostrato che gli approcci di Inmon e Kimball funzionano per consegnare con successo i data warehouse. Esistono però organizzazioni ottimali in cui è stata

implementata una combinazione di entrambi, in un modello ibrido: il data warehouse viene creato utilizzando il modello Inmon e, oltre al data warehouse integrato, i data mart orientati ai processi aziendali vengono creati utilizzando lo schema a stella per la creazione di report. Non possiamo generalizzare e affermare che un approccio è migliore dell'altro; entrambi hanno i loro vantaggi e svantaggi, ed entrambi funzionano bene in diversi scenari. L'architetto deve selezionare un approccio per il data warehouse in base ai diversi fattori;

Infine, affinché qualsiasi approccio abbia successo, deve essere attentamente studiato, discusso in dettaglio e progettato per soddisfare le esigenze di reporting della BI dell'organizzazione e dovrebbe anche integrarsi con la cultura dell'organizzazione.

1.2.2 Extraction, Transformation & Loading (ETL)

Il ruolo degli strumenti di ETL è quello di alimentare una sorgente dati singola, dettagliata, esauriente e di alta qualità che possa a sua volta alimentare il data warehouse. Le operazioni da essi svolte vengono spesso indicate con il termine *riconciliazione* che, durante il processo di alimentazione del data warehouse avviene in due occasioni: quando il DW viene popolato per la prima volta e periodicamente quando viene aggiornato. La riconciliazione consiste di quattro distinti processi detti rispettivamente:

- Extraction o Capture;
- Cleaning o Scrubbing;
- Trasformation;
- Loading.

In linea generale il confine tra pulizia e trasformazione è abbastanza nebuloso quindi per semplicità si assume che l'operazione di pulizia sia essenzialmente mirata alla correzione dei valori dei dati, mentre la trasformazione si occupa più propriamente del loro formato.

1.2.2.1 Extraction

La Data Integration è composta da due sottofasi chiamate estrazione e pulitura.

Durante la prima sottofase i dati rilevanti vengono estratti dalle sorgenti e questa operazione può essere di tipo:

- Statico: viene effettuata quando il DW deve essere popolato per la prima volta e consiste concettualmente in una fotografia dei dati operazionali;
- Incrementale: viene usata per l'aggiornamento periodico del DW, e cattura solamente i cambiamenti avvenuti nelle sorgenti dall'ultima estrazione. L'idea alla base è quella di utilizzare i cambiamenti registrati a livello dei dati per aggiornare il DWH. I benefici derivabili sono il volume molto piccolo dei dati coinvolti di volta in volta nell'operazione rispetto all'estrazione statica, e che la maggior parte dei dati nel Datawarehouse restano invariati e vengono analizzati solo i dati che hanno subito modifiche. Vengono usate tecniche CDC (Change Data Capture) che permettono di monitorare le sorgenti dati con l'obiettivo di individuare i cambiamenti avvenuti a livello dei dati. Queste tecniche sono particolarmente importanti per la data warehouse maintenance grazie alla propagazione dei cambiamenti rilevati a livello della sorgente.

La Pulitura, invece, è la sottofase che si occupa di migliorare la qualità dei dati andando ad eliminare dati "sporchi" dovuti a duplicazioni, inconsistenze, dati mancanti, valori errati etc.

Le principali funzionalità di pulitura dei dati riscontrabili negli strumenti ETL sono la correzione e l'omogeneizzazione, che utilizzano dizionari appositi per correggere errori di scrittura e riconoscere sinonimi, e la pulitura basata su regole, che applica regole proprie del dominio per stabilire le corrette corrispondenze tra i valori.

1.2.2.2 Trasformation

È la fase centrale del processo di riconciliazione e ha l'obiettivo di convertire i dati dal formato operativo sorgente a quello del DW. Tra le funzionalità di questo livello per l'alimentazione del livello dei dati riconciliati si hanno:

- Conversione e normalizzazione: operano sia a livello di formato di memorizzazione sia a livello di unità di misura al fine di uniformare i dati;
- Matching: che stabilisce corrispondenze tra campi equivalenti in sorgenti diverse;
- Selezione: che riduce, se necessario, il numero di campi e record rispetto alle sorgenti.

Nella fase di alimentazione del DW si hanno invece due sostanziali differenze: La normalizzazione viene sostituita dalla denormalizzazione e si introduce l'aggregazione che realizza le opportune sintesi dei dati.

1.2.2.3 Loading

In questa fase avviene il caricamento dei dati sul Datawarehouse attraverso due modalità alternative:

- refresh: i dati vengono riscritti integralmente sostituendo completamente quelli precedenti. In generale questa tecnica viene utilizzata solo durante la fase iniziale di popolamento;
- update: vengono aggiunti al Datawarehouse solo i cambiamenti avvenuti sui dati senza sovrascrivere ad ogni iterazione tutti i dati. Questa tecnica viene utilizzata, in abbinamento all'estrazione incrementale per l'aggiornamento periodico.

Un modo per ridurre il tempo di caricamento è quello di parallelizzare il processo ETL. Questo può verificarsi in due modi: più passaggi eseguiti in parallelo e un singolo passaggio in esecuzione in parallelo.

- Passi di carico multipli. Il flusso di lavoro ETL è diviso in più indipendenti lavori presentati insieme. È necessario riflettere attentamente su ciò che accade ogni lavoro; l'obiettivo principale è creare posti di lavoro indipendenti. Molto più sicuro per la gestione di eventuali errori;

- Pipeline. Il database stesso può anche identificare determinati compiti che può eseguire in parallelo. Ad esempio, la creazione di un indice può essere in genere parallela attraverso tutti i processori disponibili sulla macchina.

1.2.2.4 Possibili Problemi Nell' ETL E Come Risolverli

Dopo che il sistema ETL è in produzione, i guasti possono verificarsi per innumerevoli motivi.

Le Cause comuni di guasti alla produzione di ETL includere:

- Errore di rete;
- Errore del database;
- Errore del disco;
- Errore di memoria;
- Errore nella qualità dei dati;
- Aggiornamento di sistema senza preavviso.

Per proteggersi da questi guasti, è necessario un solido sistema di backup e un sistema compagno di ripristino e riavvio. Devi pianificare per errori irreversibili durante il caricamento perché accadrà. Il sistema dovrebbe anticipare questo e fornire funzionalità di recupero, arresto e riavvio di arresto anomalo.

Ad esempio, Per un processo di caricamento dovrebbe impegnare serie relativamente piccole di record alla volta e tenere traccia di ciò che è stato commesso. La dimensione del set dovrebbe essere regolabile perché le dimensioni della transazione hanno implicazioni di prestazioni su diversi DBMS.

Il sistema di ripristino e riavvio viene utilizzato, ovviamente, per riprendere un lavoro che è entrato in errore e si è fermato o per far riportare l'intero lavoro indietro tramite backup e riavviarlo. Questo sistema è significativo dipende dalle capacità del sistema di backup. Quando si verifica un errore, la reazione iniziale istintiva è tentare di salvare qualsiasi cosa sia stata elaborata e riavviare il processo da quel punto. Ciò richiede uno strumento ETL solido e affidabile funzionalità di checkpoint, in modo che possa determinare perfettamente cosa ha elaborato e cosa non deve riavviare il lavoro esattamente nel

punto giusto. In molti casi, potrebbe essere meglio uscire da tutte le righe che sono state caricate come parte del processo e riavviare dall'inizio.

Per questo motivo è consigliato di progettare tabelle dei fatti con un surrogato primario a singola colonna chiave. Questa chiave surrogata è un numero intero semplice che viene assegnato in sequenza come le righe vengono create per essere aggiunte alla tabella dei fatti. Con la chiave surrogata della tabella dei fatti, puoi facilmente riprendere un carico che viene fermato o estrarre tutte le righe nel carico limitando un intervallo di chiavi surrogate.

Quanto più un processo ETL è lungo, tanto più devi essere consapevole delle vulnerabilità a causa di un errore. La progettazione di un sistema ETL modulare composto da processi efficienti, resistenti agli arresti anomali e alle interruzioni impreviste, può ridurre il rischio di un guasto con conseguente notevole recupero. Un'attenta considerazione di quando mettere fisicamente i dati scrivendoli su disco, insieme a punti di recupero accuratamente predisposti e caricamento di data / ora o di tabelle sequenziali dei fatti consente di specificare la logica di riavvio appropriata.

1.3 OLTP vs OLAP

On-Line Transaction Processing (OLTP)

A livello di database, gli On-Line Transaction Processing si basano su query multi-access veloci ed efficaci. Le principali operazioni svolte sono INSERT, DELETE e UPDATE in quanto modificano direttamente i dati. Questi ultimi vengono costantemente aggiornati e, di conseguenza, richiedono un efficiente supporto alle operazioni di riscrittura. Una caratteristica fondamentale di questi sistemi è la normalizzazione, la quale fornisce un modo rapido ed efficace per effettuare scrittura nel database.

On-Line Analytical Processing (OLAP)

L'On-Line Analytical Processing è un insieme di tecniche software per l'analisi accelerata e interattiva di grandi moli di dati, con la possibilità di farlo da punti di vista differenti. Questi sistemi si riveleranno molto utili per l'ottenimento di informazioni di

sintesi, che avranno lo scopo di supportare e migliorare i processi decisionali aziendale. Esempi di strumenti OLAP sono i data warehouse, i cubi multidimensionali.

Le maggiori differenze fra i due sistemi sono riportati in tabella [10]:

Tabella 1: OLTP VS OLAP

	OLTP	OLAP
Finalità	Supporto all'operatività	Supporto al processo decisionale
Modalità di utilizzo	Guidata, per processi e stati successivi	Interrogazione ad hoc
Quantità di dati per operazione elementare	Bassa: centinaia di record per ogni transazione	Alta: milioni di record per ogni query
Qualità	In termini di integrità	In termini di consistenza
Orientamento	Per processo/applicazione	Per Soggetto
Frequenza di aggiornamento	Continua, tramite azioni	Sporadica, tramite funzioni esplicite
Copertura temporale	Dati correnti	Storica
Ottimizzazione	Per accessi in lettura e scrittura su una porzione di dati	Per accessi in sola lettura su tutta la base di dati

In base alla memorizzazione dei dati, si avranno diverse architetture OLAP, ognuna delle quali con i propri pro e contro [10]:

- Relational OLAP (ROLAP): i dati vengono memorizzati in un database relazionale come supporto al motore OLAP. Le analisi multidimensionali vengono tradotte in query, restituendo risultati in forma multidimensionale;

- Multidimensional OLAP (MOLAP): si ha sia il database che il motore multidimensionale. Per le operazioni di Drill-Down non è il sistema ideale, in quanto, può generare errori;
- Hybrid OLAP (HOLAP): unisce i vantaggi dei due sistemi precedenti. In particolare, pre-aggrega i dati in sistemi multidimensionali per un'analisi efficiente e veloce, mentre vengono ricercate in un database relazionale in caso di Drill-Down;
- Desktop OLAP (DOLAP): i dati vengono caricati in un sistema client e vengono calcolati dal motore in locale.

1.4 COSA SI INTENDE CON IL TERMINE BIG DATA

Il termine Big Data indica una raccolta di dati estesa in termini di *volume*, *velocità* e *varietà* che richiede tecnologie e metodi analitici specifici per l'estrazione di valore o conoscenza. Il termine è utilizzato in riferimento alla capacità di analizzare ovvero estrapolare e mettere in relazione un'enorme mole di dati eterogenei, strutturati e non strutturati, allo scopo di scoprire i legami tra fenomeni diversi e prevedere quelli futuri.

Le dimensioni variano nei diversi settori, da dozzine di terabyte a centinaia di petabyte (1000 terabyte), in base anche agli svariati strumenti software a disposizione. Esse aumenteranno sicuramente nel tempo grazie i continui avanzamenti tecnologici.

In questa definizione emergono le cosiddette 5V che caratterizzano i Big Data, ovvero il volume, velocità, varietà, veridicità e valore [2].

Il *volume* fa appunto riferimento all'enorme massa di dati generata attraverso numerosi canali.

La *velocità* si riferisce alla rapidità con cui i dati vengono acquisiti e utilizzati grazie a transazioni sempre più frequenti e veloci: le aziende non solo raccolgono i dati più velocemente, ma cercano di sfruttarli il prima possibile, spesso in real-time.

La *veridicità* riguarda la questione relativa alla qualità dei dati e al loro livello di sicurezza, la cui garanzia rappresenta una sfida molto importante. Per poter sfruttare i Big Data è

necessario saper agire per poter estrarne il *valore* e, quindi, incrementare la produttività e la competitività delle aziende e creando un surplus economico per i consumatori.

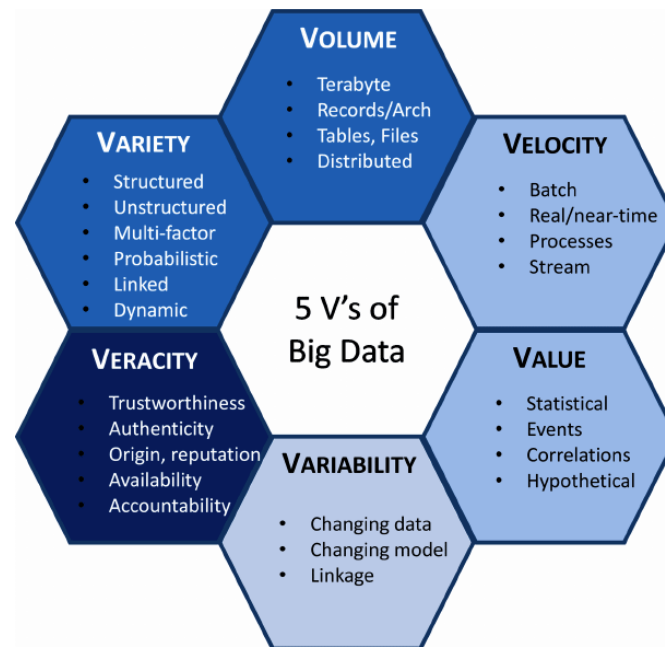


Figura 5: Le 5V dei Big Data

La *varietà* è legata alle differenti tipologie di dati disponibili provenienti da un numero crescente di fonti di dati sia strutturati sia non strutturati; in particolare è possibile identificare cinque categorie di informazioni che costituiscono i Big Data:

- Dati generati da smartphone e altri dispositivi mobile relativi a persone, attività e localizzazione, tra cui dati RFID (radio-frequency identification), dispositivi che tracciano il prodotto, e dati da dispositivi di controllo come i contatori per il monitoraggio dell'acqua o del gas;
- Dati di vendita e pricing, dati generati dall'attività delle carte fedeltà e degli eventi promozionali;
- Computer log Data, come i click streams dai siti web;
- Informazioni dai social media come Twitter e Facebook;
- Social multimediali e altre informazioni da YouTube e siti simili.

La capacità di memorizzare, aggregare i dati e di utilizzare i risultati per svolgere analisi di business profonde, è in continuo miglioramento grazie alla disponibilità di strumenti

software e tecniche sempre più sofisticate combinate a una crescente potenza di calcolo. Stiamo assistendo ad un enorme cambiamento della capacità di generare, comunicare, condividere e accedere ai dati dovuto all'aumento del numero di persone, strumenti e sensori ora connessi da reti digitali. Per capire la grandezza del fenomeno, basta osservare la figura sottostante che mostra quanti dati vengono generati in un minuto.

1.4.1 Benifici e Barriere Sull'utilizzo Dei Big Data

I Big Data rappresentano una grande opportunità per le aziende e per le economie nazionali in quanto consentono di ottenere diversi benefici significativi:

- Rivelare le variabilità delle performance e migliorare le prestazioni: La creazione e la memorizzazione di dati transazionali in forma digitale consente alle aziende di avere dati più accurati e dettagliati su svariate performance, dallo stato dei magazzini ai giorni di malattia del personale, tutto in tempo reale o quasi. Inoltre, esse utilizzando i dati per analizzare la variabilità delle prestazioni e per capirne le cause più profonde;
- Personalizzare le azioni: I Big Data consentono di creare specifici segmenti di clienti chiamati cluster e di personalizzare prodotti e servizi sulla base delle loro esigenze per realizzare promozioni e pubblicità adatte ad esse;
- Migliorare le previsioni e Supportare le persone nel processo di decision making: Utilizzando Analytics sofisticati su interi Dataset è possibile automatizzare e migliorare i processi decisionali tramite le previsioni dei Key Performance Indicators (KPI), minimizzare i rischi e scoprire preziosi insight; Questi benefici naturalmente non possono essere perseguiti con l'analisi e la gestione di piccoli campioni di dati tramite i fogli di calcolo. I rivenditori per esempio possono utilizzare algoritmi che consentono la messa a punto automatica e l'ottimizzazione degli inventari e dei prezzi a partire dai dati in tempo reale relativi alle vendite nei negozi e a quelle online.
- Creare trasparenza: Un accesso facile e tempestivo ai Big Data rende disponibile una maggiore quantità di informazione e facilita la condivisione dei dati tra le diverse unità organizzative di un'impresa;

- Profiling dei consumatori: La disponibilità quasi in real-time di dati da smartphone fornisce caratteristiche dettagliate sui clienti e sul loro complesso processo decisionale quando fanno acquisti: i Big Data permettono infatti di identificare i modelli comportamentali dei consumatori e far luce sulle loro intenzioni;
- Creare nuovi prodotti e servizi, nuove tipologie di aziende e innovativi modelli di business. Le società possono sfruttare i Big Data per migliorare lo sviluppo di modelli futuri e per creare servizi post-vendita innovativi;
- Incrementare la produttività e la profittabilità delle aziende. Lo sfruttamento dei Big Data può portare ad un aumento dell'efficacia e dell'efficienza delle imprese, le quali potranno realizzare più output utilizzando meno input e migliorare il livello di qualità dell'output stesso.

Questo elenco di benefici mette in evidenza come l'investimento nei Big Data porti alla creazione di valore per le aziende e quindi all'ottenimento di vantaggio competitivo nel lungo termine. Risulta quindi fondamentale per loro sviluppare competenze in questo ambito.

Nonostante le opportunità offerte dai Big Data siano enormi, c'è ancora un certo scetticismo all'interno delle aziende sui reali benefici apportati a causa degli scarsi risultati ottenuti in pratica [3].

Esistono quindi una serie di barriere da considerare, che possono essere classificate in 6 categorie:

- Barriere tecniche: Difficoltà di integrazione dei dati, Basso grado di influenza del business, Scarsa qualità dei dati;
- Barriere legate alle competenze: Difficoltà di comprensione degli strumenti analitici e di quantificazione dei benefici, Carenza di talenti, Difficoltà nella scelta del tool adatto;
- Barriere organizzative/gestionali: Mancanza di commitment dei top manager che non sono coinvolti nelle iniziative di Big Data, verso le quali mostrano poco interesse, risultando inefficaci;
- Barriere culturali: La maggior parte delle aziende non è ancora pronta e del tutto aperta alle innovazioni che i Big Data potrebbero portare, in quanto il loro sfruttamento richiederebbe significativi cambiamenti culturali e organizzativi: Inerzia.

- Barriere economiche: Le iniziative Big Data richiedono ingenti spese in termini di tecnologie implementate e di nuove figure professionali da assumere o di consulenza;
- Barriere legate alle privacy: I consumatori non vogliono che le loro informazioni personali, come i personal location Data e i dati elettronici generati dal loro uso di Internet, vengano utilizzate dalle aziende, soprattutto perchè non sanno dove e come queste verranno sfruttate dalle organizzazioni, le quali devono considerare anche le leggi relative alle privacy dei diversi Paesi. Tools che consentono di tracciare ogni movimento dei dipendenti e di misurare continuamente le loro performance fanno gli interessi delle organizzazioni e non dei singoli individui, che vedono minacciata la loro privacy.

1.4.2 Tecniche Per L'analisi Dei Big Data

Fino ad ora abbiamo parlato della ideologia e del valore aggiunto che possono portare i Big Data ad una azienda. Di seguito, invece, saranno elencate le principali tecniche e le tecnologie utilizzate per aggregare, manipolare, gestire e analizzare i dati.

- A/B testing: tecnica in cui un gruppo di controllo viene confrontato con gruppi di test al fine di determinare quali modifiche e azioni miglioreranno una data variabile obiettivo, come il tasso di risposta a una campagna di Marketing;
- Crowdsourcing: tecnica utilizzata per raccogliere dati, sottoposta a un grande gruppo di persone o a una comunità, attraverso, per esempio, il Web;
- Data integration: insieme di tecniche che integrano e analizzano dati provenienti da diverse fonti al fine di sviluppare insight più efficienti e accurati rispetto a quelli ottenuti esaminando una singola fonte;
- Modelli predittivi: tecniche in cui viene creato o scelto un modello matematico per prevedere la probabilità di un risultato;
- Data mining: insieme di tecniche di classificazione, cluster analysis, regole associative e regressione, che permette di estrarre modelli da grandi dataset combinando metodi statistici e di machine learning con la gestione dei database;

- Machine Learning: parte della computer science riguardante la progettazione e lo sviluppo di algoritmi che consentono ai computer di identificare i comportamenti basandosi su dati empirici e, in particolare, di riconoscere schemi complessi e prevedere decisioni per mezzo della intelligenza artificiale;
- Natural language processing (NLP): insieme di tecniche di computer science e linguistica che si ricorrono ai computer per analizzare il linguaggio umano;
- Regressione: set di tecniche che permettono di determinare come il valore di una variabile dipendente cambia quando una o più variabili indipendenti vengono modificate;
- Ottimizzazione: insieme di tecniche numeriche utilizzate per riprogettare sistemi e processi complessi al fine di migliorare le performance relativamente a uno o più aspetti, tra cui costi, velocità e affidabilità;
- Sentiment Analysis: applicazione del processing natural language e di altre tecniche analitiche per identificare ed estrarre informazioni soggettive dai testi, per esempio la “polarità” (positiva, negativa o neutra) delle caratteristiche o dei prodotti su cui le persone hanno espresso un’opinione e il grado e la forza dell’opinione stessa;
- Statistica: scienza della raccolta, organizzazione e interpretazione dei dati, utilizzata per esprimere giudizi sulle relazioni tra variabili che potrebbero essersi verificate per caso (ipotesi nulla) e su quelle causali (statisticamente significative);
- Data Visualization: tecniche di creazione di immagini, grafici, diagrammi o animazioni che consentono di comunicare, capire e migliorare i risultati dell’analisi dei Big Data.

1.5 PROGETTI DI BIG DATA NEL MARKETING

Lo sfruttamento dei Big Data in ambito Marketing rappresenta un enorme potenziale, tanto che le aziende si stanno dedicando e hanno un grande interesse verso progetti che prevedono il loro utilizzo in quest’area. Oltre ai Social Analytics che affronteremo in seguito, possiamo classificarne altri 6: il Direct e il Digital Marketing, la Customer Micro-Segmentation, il Location-based Marketing, Price Optimization, l’In-store Analysis e il Cross-Selling/Up-Selling.

1.5.1 Direct E Digital Marketing

Il Direct Marketing comprende tutte le tecniche di Marketing che consentono alle aziende di comunicare in modo mirato e personalizzato direttamente con il cliente o l'utente finale. La continua e significativa crescita di internet e della sua importanza ha comportato il rapido sviluppo del Digital Marketing, che assume la forma di display advertising, contenuti su Facebook, video clip su Youtube, e-mail personalizzate e molto altro. Le aziende per fare Digital Marketing oggi possono contare sull'enorme ammontare di informazioni degli utenti, che trascorrono ore e ore al giorno su Internet, relative ai loro interessi, ai contenuti delle loro comunicazioni, agli acquisti che fanno e molto altro [4].

Il Direct Marketing si serve di molte tecniche di Big Data, oltre che per identificare i clienti più profittevoli e quelli che risponderanno con maggiore probabilità, soprattutto per profilare i clienti, in modo da prevedere anche il comportamento di quelli sconosciuti. Vengono utilizzate sia tecniche di apprendimento supervisionato, come i modelli di ottimizzazione, le reti neurali bayesiane e gli alberi decisionali sia quelle non supervisionate, tra cui il clustering. Per ottenere risultati migliori l'ideale è combinare diverse tecniche [5].

I vantaggi apportati dai Big Data al Direct Marketing sono, oltre alla personalizzazione del messaggio, la visione a 360° del cliente, l'identificazione dei contenuti, del timing e del canale più appropriato per inviare il messaggio e la possibilità di fare questo in real time.

Da ciò deriva un incremento del tasso di conversione, ovvero del numero di visitatori che decidono di cliccare su un certo contenuto casuale o di visitare un sito web come risultato di un'azione guidata, e quindi la massimizzazione del Digital ROI, l'acquisizione di nuovi clienti e la fidelizzazione di quelli che già si rivolgono all'azienda.

1.5.2 Customer Micro-Segmentation

La molteplicità di nuove tipologie di dati e lo sviluppo di Analytics avanzati permette di avere dettagli granulari e un numero maggiore di informazioni sui consumatori e, quindi, di generare micro-segmenti molto precisi, costituiti da un piccolo numero di persone. Molti Retailer affermano addirittura di essere impegnati nella personalizzazione e non più nella semplice segmentazione [1]. I tradizionali segmenti B2C (Business to Customers) e B2B (Business to Business) basati rispettivamente su dati demografici, psicografici e comportamentali e sulle dimensioni delle aziende o sui criteri di acquisto adottati sono ormai superati.

Sfruttando quindi:

- Activity-Based Data: click-stream data dal web, le storie degli acquisti, i dati dei call center, i dati mobile;
- Profili dei social network: la storia lavorativa e l'appartenenza a gruppi;
- Sentiment Data: associazioni a prodotti e aziende (like o follows) e commenti online.
- Dati tradizionali: dati delle ricerche di mercato e quelli transazionali;

È possibile costruire segmenti molto più stretti. Gli uomini di Marketing possono quindi creare offerte, prodotti e servizi personalizzati e su misura per ciascun cluster, con ovvi benefici sui ritorni. Questi dati possono inoltre essere aggiornati in real-time, riuscendo quindi ad identificare i cambiamenti dei clienti e delle loro preferenze.

1.5.3 Price Optimization

Le aziende possono sfruttare la crescente granularità dei dati sulle vendite e i potenti Analytics per ottimizzare i prezzi. L'ammontare di informazioni a loro disposizione è enorme, dalle serie storiche della domanda, ai dati relativi alle scorte, a quelli riguardanti i competitor, fino al livello delle vendite attuali. Questa base di dati è in continuo aumento considerata l'esplosione di nuovi canali di vendita online dove i consumatori possono

confrontare i prezzi, incrementando la competizione tra le varie firme presenti nel mercato, andando incontro alle esigenze del cliente [6].

Da queste ingenti quantità di dati, attraverso opportuni tools, i pricing manager sono in grado di estrarre insight per definire quasi in real-time il prezzo ottimale che un consumatore è disposto a pagare per ciascun prodotto, basandosi sulle sue caratteristiche.

La price optimization può considerare, per esempio, l'elasticità della domanda al prezzo, con specifici modelli che analizzano i dati delle vendite storiche per ricavare insight sul pricing di ciascuna unità, che possono poi essere utilizzati per fare promozioni o per ridurre i prezzi, valutando i costi conseguenti. I benefici che le aziende riescono a conseguire in questo modo sono un aumento dei ricavi, dei margini e della quota di mercato.

Tuttavia, per riuscire a sfruttare i Big Data in quest'area è necessario costruire una fiducia verso il cliente, Identificare le opportunità più promettenti, che comprendono determinare quanto il consumatore vuole pagare esattamente per un dato prodotto attraverso la customer segmentation e le promozioni personalizzate, e Ascoltare i dati su cui le organizzazioni devono saper far leva. Particolare attenzione è focalizzata nel corretto utilizzo di adeguati Analytics per identificare elementi che spesso vengono trascurati e per determinare i fattori guida per ciascun cliente e prodotto che porterà alla scelta finale di prezzo [7].

1.5.4 Location-Based Marketing

Il Location-based Marketing si basa sull'adozione crescente di smartphone e di altri mobile device che generano i personal location Data, i quali permettono di conoscere posizione e comportamento delle persone in real-time sfruttando il GPS o il WI-Fi, favorendo lo sviluppo di una strategia di Marketing che considera le abitudini lavorative e di divertimento e non solo le preferenze dei consumatori. Altre fonti utilizzate sono i segnali delle torri di triangolazione cellulare e i pagamenti tramite carte di credito e di debito, le quali, attraverso il terminale del punto di vendita, rendono disponibili i dati di identificazione personale.

Quello che le aziende fanno solitamente prende il nome di *Geo-Targeted Advertising*, ovvero effettuare azioni di advertising in tempo reale in base alla localizzazione dei propri clienti. Infatti, per ottenere enormi vantaggi, le aziende ricorrono alle push notifications, cioè delle offerte customizzate e aggiornate per un determinato cliente mentre, per esempio, cammina con in mano lo smartphone all'interno del negozio. Pertanto, lo sfruttamento dei dati di geolocalizzazione può portare ad un aumento delle vendite, ad un incremento dei profitti e ad un miglioramento della customer experience e perciò alla fidelizzazione della clientela.

Tuttavia, relativamente a questo progetto le aziende si trovano a dover affrontare due sfide: La privacy ed un trade-off, cioè se gli utenti desiderano ricevere offerte mobile quando si trovano in prossimità dello store stesso.

1.5.5 In-Store Analysis

L'in-store analysis prevede l'analisi dei dati real-time relativi al comportamento dei consumatori tramite la posizione e il percorso dei clienti all'interno dello store vengono tracciati attraverso svariate tecnologie: video camere, Wi-Fi, strumenti Bluetooth, sistemi dei punti di vendita, carte di pagamento, trasponder dei carrelli, applicazioni degli smartphone, Path Intelligence e tag RFID sulle carte d'acquisto.

Tools e Analytics, quali web dashboard, app mobile, real-time alert e strumenti di data mining, vengono utilizzati per organizzare, analizzare e visualizzare questo grande ammontare di dati, identificare trend e confrontare le prestazioni dei diversi periodi. Così facendo vengono estratti insight relativi ai comportamenti dei consumatori all'interno dello store, con l'obiettivo ultimo di migliorare la customer experience.

In particolare, gli insight ottenuti sono relativi a quanti clienti entrano nel negozio, a come si comportano gli shopper all'interno dello store e per conoscere il consumatore tramite gli attributi sesso, età, se è la prima volta che entra nel negozio, se ritorna spesso, da dove viene e quali sono i suoi interessi

Le aziende si servono di questi insight per migliorare efficacemente l'organizzazione, ovvero per ottimizzare il layout dello store, le sue caratteristiche, il posizionamento sugli

scaffali e il mix di prodotti offerti per trasformare i clienti una tantum in clienti abituali, per incrementare la frequenza delle loro visite e delle loro spese migliorando la store experience, per aumentare la dimensione media della transazione e attirare un numero sempre maggiore di consumatori. Le organizzazioni mettono in atto aggiustamenti in tempo reale per ottimizzare l'intero processo d'acquisto.

1.5.6 Cross-Selling / Up-Selling

I Big Data offrono grandi opportunità per aumentare la dimensione media dell'acquisto di un consumatore sia mettendo a disposizione prodotti o servizi collegati con la scelta d'acquisto iniziale sia offrendo qualcosa di maggior valore rispetto a questa, ovvero per migliorare le azioni di Cross-Selling e di Up-Selling. Dati quali le caratteristiche demografiche dei clienti, la posizione real-time, le preferenze, la storia degli acquisti passati vengono utilizzati a tal fine. Gli algoritmi come le regole di associazione, si basano su questi dati per prevedere il comportamento dei consumatori in vari scenari di vendita ed estrarre insight per capire molto prima cosa vogliono e determinare quindi il miglior approccio [9].

I benefici che le aziende traggono sono un aumento delle vendite e dei profitti e la fidelizzazione dei clienti.

Caso esemplare è quello di Amazon che raccoglie i dati da tutti gli utenti, riconoscendo i trend nelle persone che fanno acquisti simili attraverso tools di Analytics, in modo da cogliere potenziali opportunità e in base a ciascun prodotto o servizio visitato dall'utente sul sito suggerisce "potresti anche volere" ed è proprio in questo modo che riesce ad incrementare significativamente le vendite.

1.6 KNOWLEDGE DISCOVERY IN DATABASE (KDD)

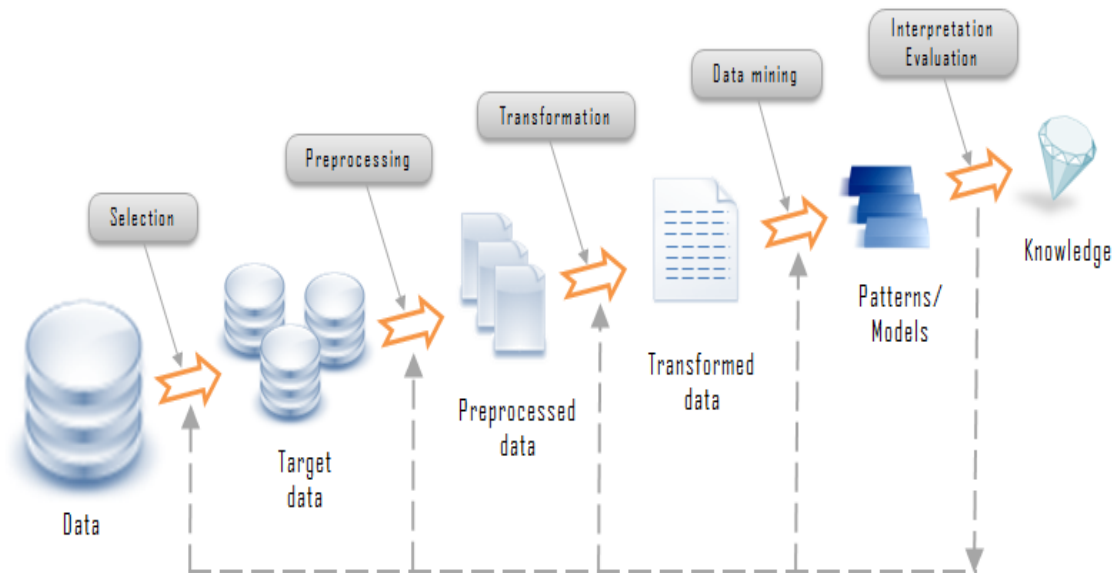


Figura 6: KDD Process

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, individuando 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, da sottoporre all'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. 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 DWH, 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 pre-processati, 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, 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, perciò, 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.

1.6.1 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 con 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
Definizione	Estrarre Knowledge da una grande quantità di dati	Introdurre un nuovo algoritmo da dati e esperienza passata
Storia	Introdotta nel 1930	Introdotta nel 1950
Responsabilità	Il data mining viene utilizzato per ottenere le regole dai dati esistenti.	L'apprendimento automatico insegna al computer a imparare e comprendere le regole date.
Origine	Database tradizionali con dati non strutturati	Dati esistenti e algoritmi.
Implementazione	We can develop our own models where we can use data mining techniques for	Possiamo usare l'algoritmo di machine learning nell'albero decisionale, nelle reti neurali e in qualche

		altra area di intelligenza artificiale.
Natura	Manuale	Automatico
Applications	Usato nella cluster analysis	Usato in web search, spam filter, credit scoring, fraud detection, computer design
Tecniche	Il data mining è più di una ricerca che utilizza metodi come l'apprendimento automatico	Auto apprendimento e insegnamento fatto da task intelligenti.
Scopo	Area limitata	Vasta area.

1.7 ALGORITMI DI DATA MINING

L'obiettivo del data mining consiste nell'estrarre nuove informazioni dai dati esistenti. Come vedremo, esistono due approcci per raggiungerlo: l'apprendimento supervisionato e l'apprendimento non supervisionato [14].

- Supervised learning: metodologia di apprendimento automatico in cui vengono passati alla macchina 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 ad una certa categoria. L'apprendimento supervisionato è utilizzato principalmente per i problemi

di *classificazione*, come, ad esempio, si usa 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;

- *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.

	<u>Unsupervised</u>	<u>Supervised</u>
<u>Continuous</u>	<ul style="list-style-type: none"> • Clustering & Dimensionality Reduction <ul style="list-style-type: none"> ○ SVD ○ PCA ○ K-means 	<ul style="list-style-type: none"> • Regression <ul style="list-style-type: none"> ○ Linear ○ Polynomial • Decision Trees • Random Forests
<u>Categorical</u>	<ul style="list-style-type: none"> • Association Analysis <ul style="list-style-type: none"> ○ Apriori ○ FP-Growth • Hidden Markov Model 	<ul style="list-style-type: none"> • Classification <ul style="list-style-type: none"> ○ KNN ○ Trees ○ Logistic Regression ○ Naive-Bayes ○ SVM

Figura 7: Data Mining Algorithms

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. Per far ciò, bisogna dividere il processo in due fasi:

La prima fase è la formazione, in cui il modello viene creato utilizzando i dati del passato, mentre la seconda 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.

Le innovazioni che utilizzano l'intelligenza artificiale e il Machine Learning 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 è oggi molto comune, soprattutto nei retail.

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 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.

Quando l'utente inizia a sentirsi compreso, è più propenso ad acquistare prodotti aggiuntivi. Conoscendo ciò che un cliente 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), la domanda e l'offerta del mercato, un sistema automatico di prezzi può adeguare in modo efficiente i prezzi.

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

1.7.1 Clustering

L'obiettivo della clusterizzazione è di organizzare gli oggetti esaminati in gruppi, che condividono proprietà simili. Il Clustering si può considerare uno dei più importanti metodi di apprendimento non supervisionato e, come ogni metodo appartenente a questa categoria, non fa uso di identificatori determinati a priori per intuire la possibile struttura dei dati.

Esistono varie forme di clustering [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 [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 dendrogramma. 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.

La complessità è nell'ordine di $O(N^3)$, e come in K-Means, la presenza di outliers condiziona negativamente questo approccio.

D seguito, sono descritte le principali strategie di Clustering ed algoritmi utilizzati nell'ambito Fashion.

1.7.1.1 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. 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(<i>clusters</i> K)
2: <i>Elezione K Centroidi</i>
3: repeat
4: <i>Assegnamento di ogni elemento al punto K piu' vicino</i>
5: <i>Ricalcolo dei K Centroidi</i> until <i>I Centroidi non variano</i>

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. In essa, 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 per eseguire 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, tramite;
- 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-Means un numero arbitrario di volte. Si prende l'iterazione che ha prodotto i migliori cluster e si applica ricorsivamente l'algoritmo fino a che non si ottengono i K cluster desiderati.

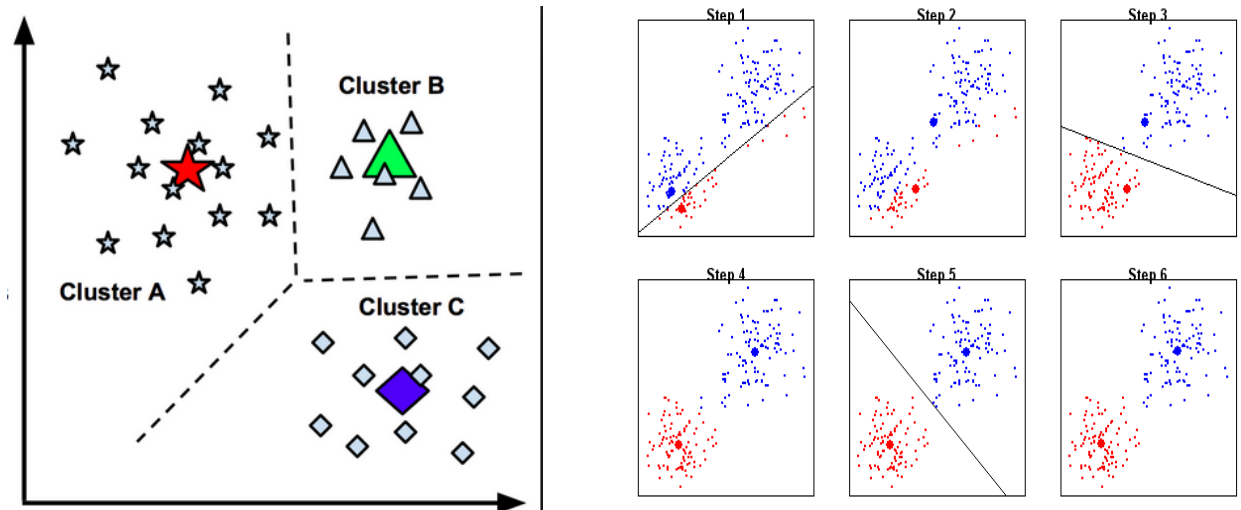


Figura 8: K-Means Algorithm

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 difficoltosa, 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.

1.7.1.2 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 i vicini. 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 sono caratterizzati da densità troppo variabili. È molto usato per clusterizzare tramite la Geo-localizzazione.

1.7.2 Classification And Regression Trees (CART)

CART è una procedura non parametrica dove non è necessario pre-testare la normalità o altre assunzioni che riguardano la distribuzione statistica dei dati. L'albero finale include solo le variabili indipendenti che risultano essere predittive della variabile

dipendente; le altre variabili indipendenti non predittive non hanno effetto sul risultato finale; anche sotto questo aspetto CART si differenzia dalle altre procedure statistiche tradizionali. Con il termine 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, esso può essere usato per predire la classe di nuove istanze di record per cui la classe è sconosciuta.

La struttura di un classification tree include i nodi non terminali (*parent nodes*), i quali hanno due discendenti diretti (*child nodes*), ed i nodi terminali che non subiscono ulteriori bipartizioni (*terminal nodes*). Il primo nodo radice (*root node*) contiene tutte le osservazioni. Dal nodo radice discendono due “nodi figli”. Ogni child node, che indichiamo con la lettera t contiene un sottocampione del campione originale, in cui i membri condividono le stesse caratteristiche, che influenzano la variabile dipendente di interesse. Ogni t, a sua volta, costituisce un potenziale parent node che può essere ancora suddiviso in due child node. Il processo continua fino a che l'albero non termina la sua crescita. I nodi terminali sono i nodi finali dell'albero decisionale e contengono insiemi di osservazioni che vanno a formare classi molto omogenee al loro interno e il più possibile eterogenee tra loro.

Vi sono alcuni step importanti da seguire quando si costruisce un albero decisionale con la procedura CART; gli step includono: adottare un criterio di bontà della tecnica con i cui i nodi vengono suddivisi da parent nodes a child nodes (split criterion); stabilire una regola di arresto di crescita dell'albero (stopping rule).

Per scegliere le split criterion si utilizza generalmente una tecnica di *Recursive Binary Splitting*.

Il metodo è binario e ricorsivo: binario, poiché ogni parent node si divide in due discendenti diretti, e ricorsivo, poiché i nodi (non terminali) nati dalla suddivisione del parent node in due discendenti diretti possono diventare, a loro volta, parent node e suddivisi in due nodi successivi.

Gli alberi decisionali con molti nodi e numero di divisioni possono portare a un sovradattamento dei dati (definito più propriamente dal termine *overfitting*). Ciò

significa che il modello risulta di difficile interpretazione in quanto diventa inaccurato per previsioni successive ed ha bisogno delle stopping rule. I metodi per evitare questo problema sono impostare un numero minimo di dati di allenamento da utilizzare su ciascun nodo foglia o impostare la profondità massima del modello, che si riferisce alla lunghezza del percorso più lungo dal nodo radice al nodo foglia.

I differenti algoritmi esistenti si differenziano in base alla strategia impiegata sui singoli nodi, per la valutazione dello Split. Esistono infatti differenti indici per la validazione di una classificazione:

- GINI INDEX identifica la qualità dello split. Considerando $p(i)$ la frequenza relativa della classe i al nodo:

$$Gini = 1 - \sum_{i=1}^c (p_i)^2$$

- GAIN INDEX: si basa sul concetto di *entropia*, indice relativo alla omogeneità del nodo, ottenuta eseguendo un particolare split sul nodo p_i :

$$Entropy = \sum_{i=1}^c -(p_i) * \log_2(p_i)$$

1.7.2.2 Altri Tipi Di Classificatori

- Basata 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;
- Classificatore Bayesiano: Consiste in un framework probabilistico per risolvere il problema della classificazione. Si considerano gli attributi e la classe come variabili

casuali Si basandoci fortemente sul concetto di Probabilità Condizionata. 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 $P(C|A_1, A_2, \dots, A_n)$. Ne segue il Teorema di Bayes:

$$P(C|A) = \frac{P(A|C) * P(C)}{P(A)}$$

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];

- 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, ma spesso si arriva ad ottenere un modello con troppi possibili casi che comporta una classificazione non corretta [18].

La *Validazione* di questi 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 una matrice indicante l'incidenza tra le classi predette e 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.

		True class			
		p	n		
Hypothesized class	Y	True Positives	False Positives	fp rate = $\frac{FP}{N}$	tp rate = $\frac{TP}{P}$
	N	False Negatives	True Negatives	precision = $\frac{TP}{TP+FP}$	recall = $\frac{TP}{P}$
Column totals:		P	N	accuracy = $\frac{TP+TN}{P+N}$	
				specificity = $TN / N = 1 - FP \text{ Rate}$	

Figura 9: Confusion Matrix

1.7.3 Predizione: 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 definiti 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: $X \rightarrow Y$ con X , Y itemset dove X prende il nome di *Premessa* e 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 regola e le transazioni che contengono la premessa:

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

Formalmente il supporto $supp(X \cup Y)$ può essere riscritto come la probabilità congiunta $P(E_X \cap E_Y)$, dove E_X e E_Y sono tutte le transazioni che contengono X o Y rispettivamente. Quindi, possiamo esprimere la confidenza come la probabilità condizionata $P(E_Y | E_X)$.

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.

1.7.3.1 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 proprietà è così formalizzata, con X e Y itemset:

$$\forall X, Y: (X \subseteq Y) \Rightarrow supp(X) \geq supp(Y)$$

Questa proprietà è alla base dell'algoritmo *Apriori*, dove, partendo da tutti i possibili item con cardinalità 1, si 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	DESCRIPTION
1: function Apriori(T, s)	(<i>set transazioni T, minSupport</i>)
2: $L_1 \leftarrow \{large\ 1 - itemsets\}$	$k = 1$ e <i>Generazione itemset con cardinalità 1</i>
3: $k=2$	
4: while $L_{k-1} \neq \emptyset$ do	<i>Generazione itemset di cardinalità $k + 1$.</i>
5: $C_k \leftarrow Generate(L_{k-1})$	
6: for transaction $t \in T$ do	<i>Eliminazione itemset contenenti non frequenti.</i>
7: $C_t \leftarrow Subset(C_k, t)$	
8: for candidates $c \in C_t$ do	<i>Calcolo support itemset generati.</i>
9: $count[c] \leftarrow count[c] + 1$	
10: $L_k \leftarrow \{c \in C_k count[c] \geq s\}$	Eliminazione itemset non frequenti
11: $k \leftarrow k + 1$	
12: return $\bigcup_k L_k$	

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*.

In questo caso specifico, invece, la confidenza delle regole generate dal solito itemset è *anti-monotona* rispetto al numero di item che compongono la premessa della stessa regola possiede la seguente proprietà.

Indicando con $Conf(X \Rightarrow Y)$ la confidenza della regola $X \Rightarrow Y$ si otterrà:

$$Conf(ABC \Rightarrow D) \geq Conf(AB \Rightarrow CD) \geq Conf(A \Rightarrow BCD)$$

Si procede quindi generando le regole che possiedono solo un item nella conseguenza, eliminando 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 aggiuntivo

nella conseguenza, procedendo fino a che non sono state generate tutte le possibili regole.

Le regole estratte sono sottoposte ad un'ulteriore fase di post-processing, in quanto la confidenza a volte può 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:

The diagram illustrates the calculation of three validation indices for an association rule $Rule: X \Rightarrow Y$. Three blue arrows originate from the rule and point to the respective formulas:

- Support**: $Support = \frac{freq(X, Y)}{N}$
- Confidence**: $Confidence = \frac{freq(X, Y)}{freq(X)}$
- Lift**: $Lift = \frac{Support}{Supp(X) \times Supp(Y)}$

Figura 10: Indici Di Validazione Di Una Regola Di Associazione

1.7.4 Artificial Neural Networks & Deep Learning

Il Deep Learning è un metodo specifico di apprendimento automatico che incorpora un numero elevato di reti neurali in vari strati successivi tra loro per apprendere dai dati in modo iterativo.

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

Una rete neurale è particolarmente utile quando si cerca di studiare i pattern da dati non strutturati e sono progettati per emulare come, attraverso l'intelligenza artificiale, i computer possano essere addestrati per trattare problemi che non sono ben definiti [21].

Essa consiste in 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 ed elaborati nel livello nascosto, ottenendo diversi livelli di output in base ai pesi applicati ai singoli nodi nascosti.

La tipica rete neurale può essere composta da migliaia o anche milioni di nodi di elaborazione semplici che sono densamente interconnessi.

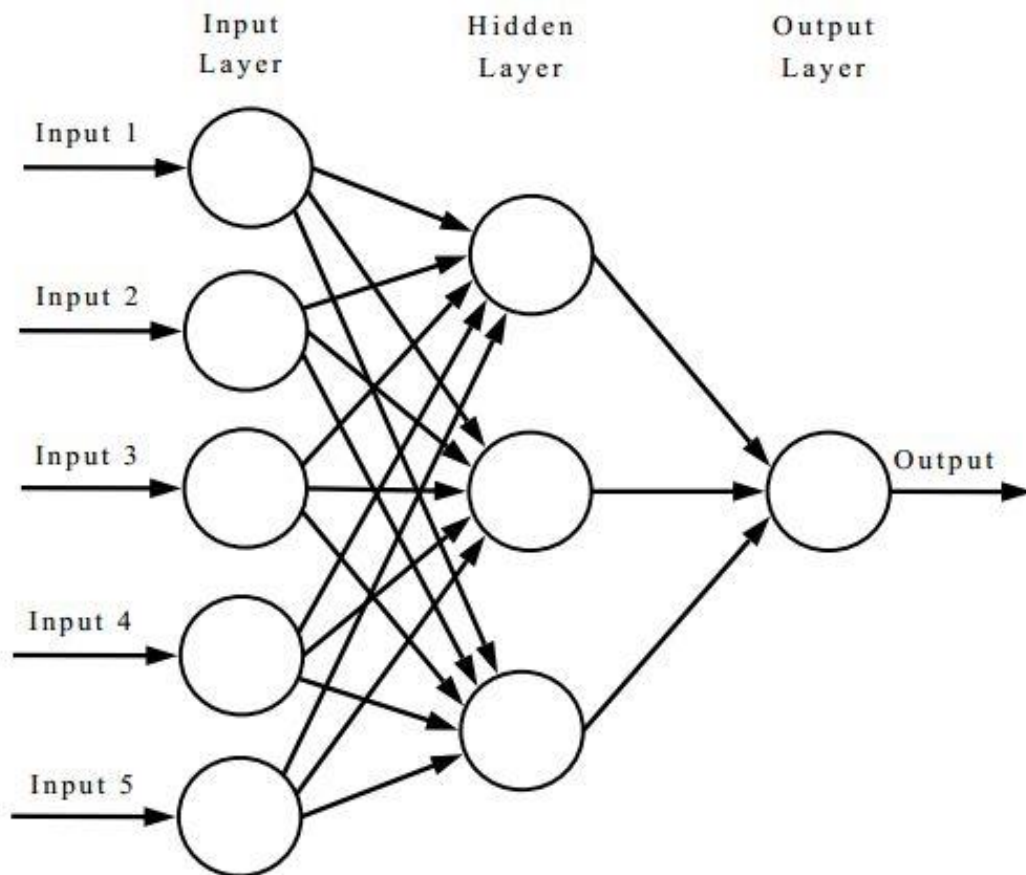


Figura 11: Artificial Neural Network Schema

In una rete neurale, gli input inviati ai *input layer* 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 per l'analisi. In ogni nodo appartenente ai livelli successivi al primo, *hidden layer* e *output layer*, 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 I_n è dato dalla seguente relazione:

$$I_n = \sum_{i=1}^n (w_{i,n}) * O_i + \text{offset}_n$$

dove $w_{i,n}$ è il peso del collegamento tra il nodo i del livello precedente e il nodo n preso in considerazione, O_i è l'output del nodo i del livello precedente e offset_n è l'offset associato al nodo n considerato.

Ogni nodo, inoltre, applica poi una funzione di attivazione sul 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 delle reti neurali per imparare da dati non etichettati e non strutturati tramite una combinazione tra algoritmi non supervisionati e algoritmi supervisionati. Spesso viene chiamato Deep Learning una sotto-disciplina del Machine Learning.

Il Deep Learning viene usato nelle applicazioni dell'Internet of Things (IoT) o per prevedere quando una macchina funzionerà male.

1.7.5 Regressione Lineare

L'analisi di regressione è una tecnica statistica utilizzata per determinare una relazione tra una variabile dipendente e un insieme di fattori esplicativi. La variabile dipendente,

indicata come variabile Y , è il valore che stiamo cercando di determinare in base ai fattori esplicativi.

I fattori esplicativi, indicati come variabili X , vengono anche chiamati fattori indipendenti, variabili predittive o semplicemente fattori modello. L'analisi di regressione aiuta gli analisti a scoprire la sensibilità della variabile dipendente ai cambiamenti nei fattori esplicativi. Queste sensibilità sono essenziali per una corretta gestione del rischio.

Esistono tre tipi di dati comunemente utilizzati nell'analisi di regressione: serie temporali, sezioni trasversali e dati raggruppati.

- Serie temporali: dati raccolti in un periodo di tempo. Nelle serie economiche e finanziarie questi dati si riferiscono spesso a rendimenti di mercato, rendimenti dell'indice, prezzi e valori delle attività, PIL, disoccupazione, tassi di interesse, ecc. Questi dati vengono raccolti a intervalli di tempo uguali come giornaliero, mensile, trimestrale, ecc;
- Sezione trasversale: dati raccolti per una famiglia di variabili nello stesso momento. Ad esempio, nell'analisi fondamentale raccogliamo spesso informazioni specifiche dell'azienda come il rapporto prezzo / utili, il rapporto prezzo / valore contabile, il rapporto debito / capitale netto o il volume medio giornaliero degli scambi;
- Dati raggruppati: dati che sono una combinazione di serie temporali e dati cross section.

Nel caso in cui abbiamo più fattori esplicativi, l'analisi è denominata modello di regressione multipla e ha la forma:

$$Y = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon$$

dove Y è la variabile dipendente (quello che stiamo cercando di prevedere), X è il fattore esplicativo (quello che stiamo usando per predire), e ε è il rumore casuale (errore). Inoltre, la variabile dipendente Y , i fattori esplicativi x e il termine di errore ε sono vettori di colonne di valori.

Simple Linear Regression Model

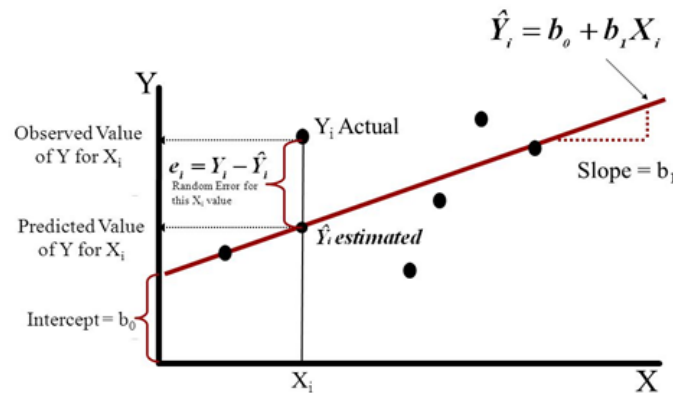


Figura 12: Linear Regression

Nell'equazione precedente, b_0 e b_1 sono i parametri del modello attuale che definiscono l'esatta sensibilità della variabile dipendente ai fattori esplicativi, e ε è la quantità di variabilità che non è spiegata dal modello.

In pratica, questi valori esatti non sono noti con certezza e devono essere stimati dai dati. Per fare ciò si utilizzano la

$$\text{Varianza} = \text{Var} [\varepsilon] = \sigma_{\varepsilon}^2$$

$$\text{Expected Value} = E [b_0] = b_0 * E [b_1] = b_1 * E [\varepsilon] = 0$$

L'obiettivo dell'analisi di regressione è determinare l'insieme di fattori esplicativi e sensibilità corrispondenti che spieghino il più possibile i valori dipendenti osservati.

Metriche di valutazione e Presupposti del modello.

Nell'effettuare l'analisi di regressione, le metriche importanti per analizzare l'analisi sono:

- b_k = Parametro del modello e si riferisce alla sensibilità stimata di Y al fattore k;
- R^2 = bontà di adattamento (la percentuale della varianza complessiva spiegata dal modello). L'indice di determinazione lineare si definisce quale rapporto di composizione tra devianza di regressione e devianza totale, misurando nell'intervallo $[0,1]$ quanta parte della devianza totale è spiegata dai regressori del

modello. Se si considera la decomposizione della devianza totale SST (Sum of Squares for Total Variation) in devianza di regressione SSR (Sum of Squares due to Regression) e devianza residua SSE (Sum of Squares due to Residual), si dimostra che, al crescere del numero delle variabili esplicative, la devianza dei residui diminuisce e quindi l'indice di determinazione lineare aumenta. Pertanto, un alto valore di R^2 non è un indicatore di buon adattamento in quanto esso dipende anche dal numero di regressori inclusi nel modello. Affinché si possano confrontare due regressioni su Y con un diverso numero di regressori si dovrà considerare l'indice corretto che tiene conto dei gradi di libertà delle singole espressioni di variabilità;

- F-stat: valore critico per l'intero modello. La maggior parte dei test F si pone considerando una scomposizione della variabilità in una raccolta di dati in termini di somme di quadrati. La statistica del test in un test F è il rapporto tra due somme di quadrati scalati che riflettono diverse fonti di variabilità. Queste somme di quadrati sono costruite in modo tale che la statistica tende ad essere maggiore quando l'ipotesi nulla non è vera. Affinché la statistica per seguire la F-distribuzione sotto l'ipotesi nulla, cioè se i valori dei dati sono indipendenti e normalmente distribuiti con una varianza comune, la somma dei quadrati dovrebbero essere statisticamente indipendenti, e ciascuno dovrebbero seguire una scala χ^2 distribuzione [29];
- T-stat: valore critico per il parametro stimato. La statistica T viene utilizzata in un test T quando si decide se supportare o rifiutare l'ipotesi nulla. Maggiore è la T, maggiore è la prova che i valori sono significativamente diversi dalla media. Viceversa, un valore T più basso indica che non è significativamente diverso dalla media [30].

CAPITOLO 2:

TRADITIONAL ETL PER LA CREAZIONE DELLA DATA MART

Il manifestarsi delle prime necessità di dati integrati ha portato le aziende ad affrontare il problema internamente in quanto il mercato non sapeva offrire soluzioni sufficientemente flessibili ed affidabili. Per questo il primo approccio per rispondere alle necessità di avere dati integrati fu quello di sviluppare internamente all'azienda software ad hoc soprattutto per eseguire le fasi di estrazione, trasformazione e caricamento dei dati in un ambiente unico e integrato. Nonostante i recenti progressi dei prodotti di data Integration ancora oggi la maggior parte delle aziende utilizza soluzioni ETL personalizzate per rispondere alle necessità di integrazione.

Tuttavia, le più recenti evoluzioni del mercato hanno portato ad un aumento della domanda di prodotti completi di data Integration, portando al 60% la percentuale delle imprese che utilizza uno dei pacchetti di prodotti di integrazione offerti sul mercato, con lo scopo di effettuare attività di business intelligence [8].

La recente crisi economica ha portato inoltre ad una diminuzione dei budget assegnati allo sviluppo dell'Information Technology nelle aziende, determinando un incremento dell'adozione di soluzioni di integrazione open source.

Si può quindi affermare che il mercato della data Integration è oggi caratterizzato dalla convivenza di tre tipologie di prodotti:

- Software personalizzati: l'emergere delle prime necessità di integrazione dei dati molte imprese svilupparono internamente prodotti ad hoc in grado di rispondere alle esigenze specifiche del proprio ambito di business. Con la maturazione del mercato

dei prodotti di data Integration questo tipo di approccio è divenuto sempre meno conveniente. Inoltre, l'emergere di architetture SOA e applicazioni SaaS sta decretando la fine dei prodotti sviluppati in casa. Oggi le suite di data Integration presenti sul mercato offrono sicuramente funzionalità e affidabilità migliori;

- Software proprietari: lo sviluppo di applicativi di data Integration ha contribuito ad aumentare la produttività delle attività collegate all'integrazione dei dati. I prodotti di integrazione dei dati sono maturati costantemente negli anni garantendo un ventaglio di funzionalità sempre più ricco e variegato, rendendo tali applicativi idonei a supportare la grande maggioranza degli scenari di business che richiedono l'utilizzo di dati integrati. Il numero di applicativi sul mercato è oggi elevato, si va dalle suite di prodotti in grado di coprire la quasi totalità delle necessità aziendali a prodotti specializzati in particolari contesti di business o specifiche problematiche;
- Software open source: il limite dei maggiori prodotti proprietari presenti sul mercato sono i costi necessari per la loro implementazione. Per venire in contro alle necessità delle aziende più piccole e con risorse limitate si sono da poco affacciate sul mercato i primi prodotti open source, prodotti in grado di supportare una discreta quantità di funzioni ma con un costo decisamente minore rispetto ai prodotti proprietari (costi di licenza nulli, costi di infrastruttura ridotti, servizi pagati in base all'utilizzo).

Per la mia Tesi ho deciso di utilizzare un Software Open Source che offre tutte le funzionalità necessarie per lo svolgimento, minimizzando al massimo i costi: Talend.

2.1 TALEND OPEN SOURCE

L'approccio open source di Talend prevede la disponibilità di due prodotti:

- Talend Open Studio: suite gratuita scaricabile gratuitamente con licenza open source (GPL). Talend Open Studio si presenta come prodotto di data integration completo e contraddistinto da un'ampia gamma di funzionalità, sufficienti per la maggior parte delle necessità;
- Talend Integration Suite: è una versione potenziata del prodotto gratuito che aggiunge funzionalità avanzate come lo sviluppo collaborativo, monitoraggio avanzato del progetto e il Data Masking.

Per chi non possiede l'hardware necessario per supportare il sistema c'è una terza opzione costituita da Talend On Demand, ovvero una offerta di tipo Software ad a Service (SaaS).

I prodotti di Talend offrono ad oggi le seguenti funzionalità:

- Ambiente di sviluppo user-friendly (basato sulla piattaforma Eclipse);
- Elevato numero di connessioni preimpostate;
- Deposito comune dei metadati;
- Supporto allo sviluppo collaborativo.
- Servizi di trasformazione di dati;
- Funzionalità di monitoraggio dell'andamento dell'integrazione;
- Data Profiling e Data Quality.

Vediamo quindi quali sono i punti di forza dell'approccio di open source di Talend [7]:

- Nessuna barriera all'adozione: la disponibilità gratuita del prodotto di base rende praticamente immediata l'installazione del software. Talend supporta il cliente attraverso tutorial sull'utilizzo di base, inoltre è possibile fare affidamento ad una vasta comunità di utilizzatori;
- Curva di apprendimento veloce: il prodotto si presenta graficamente user-friendly l'interfaccia grafica è intuitiva e l'utilizzo delle funzionalità di base non richiede particolari addestramenti;

- Modello di prezzi stabile e prevedibile: i prodotti proprietari prevedono spesso costi elevati man mano che si espandono le funzionalità e le capacità del prodotto, con costi di licenza che aumentano all'aumentare delle macchine installate. Questo rende spesso difficile una corretta previsione dei costi nelle fasi iniziali del progetto, soltanto a lavoro ultimato è possibile rendersi conto del costo effettivo della soluzione adottata. Talend prevede un modello di costo basato sul numero di sviluppatori e sull'utilizzo del servizio, indipendente da licenze, hardware e quantità di dati da integrare;
- L'importanza di una comunità a supporto: la comunità online di esperti ed utilizzatori del prodotto è già oggi molto vasta ed è un fattore di grande importanza per facilitare l'implementazione e il mantenimento delle soluzioni offerte da Talend. Forum, wiki, guide e contributi gratuiti degli utenti rappresentano un valore aggiunto che solo un prodotto di questo tipo può offrire;
- Ampio supporto a tipologie di dati differenti: con oltre 400 connessioni preimpostate la soluzione di Talend garantisce la compatibilità con un grande numero di sistemi, database, pacchetti di software, applicazioni gestionali, servizi web, ecc. Nessun'altra soluzione sul mercato vanta un numero di possibili connessioni così elevato;
- Flessibilità, versatilità e riuso del prodotto: Talend non si limita ad un supporto alle tecniche standard di ETL ma permette l'implementazione di diverse strategie di integrazione. La possibilità di riuso di progetti già perfezionati costituisce inoltre un altro punto di forza dell'approccio open source;
- Funzionalità e performance: il livello di funzionalità offerto è paragonabile a quello dei prodotti proprietari. Tuttavia, si registrano alcune lacune nel campo della modellazione dei dati, data quality e data mining. Un team di ricerca e sviluppo dedicato permette al prodotto di essere sempre aggiornato alle ultime esigenze del mercato e di proporre funzionalità innovative;
- Costi e tempi ottimizzati: le soluzioni offerte da Talend risultano da un 50% a un 80% più economiche rispetto ai prodotti tradizionali, essendo meno costose da acquisire e mantenere e permettono uno sviluppo più rapido del sistema di integrazione.

2.2 CREAZIONE DELLA DATA MART

Nello specifico, un Data Mart è un database analitico progettato per incontrarsi con le esigenze specifiche di un'impresa. Essendo sottoinsieme logico o fisico di un data warehouse di dimensioni maggiori, segue le stesse regole di progettazione con dati aggregati a vari livelli di dettaglio, anche se, talvolta può essere costituito anche in assenza di un sistema di dati integrato [11].

Tabella 3: Datawarehouse Vs Data Mart

Definition	Data Warehouses	Data Marts
Finalità	Application-neutral Centralizzati e condivisi Intera impresa	Applicazioni specifiche Dipartimenti o aree
Dati	Bassa denormalizzazione	Alta denormalizzazione
Soggetti utilizzatori	Soggetti di molte aree	Soggetti di una singola area
Sorgenti dei dati	Molte Dati esterni operazionali	Poche Dati esterni operazionali
Caratteristiche	Flessibile, estensibile Lunga vita Data-oriented	Ristretto, non estensibile Vita breve Project-orientation
Tempo d'implementazione	9-18 mesi per il primo stadio	4-12 mesi

L'implementazione può essere di due tipi: *Top-Down*, costruzione del DWH, e conseguente aggregazione ed esportazione nei vari data mart, e *Bottom-Up*, concentrandosi su aree specifiche del business si costruiranno i vari data mart per poi giungere alla costruzione del DWH. In questo modo si avrà un approccio scalabile.

La fase di ETL del progetto Fashion_Retail è basata sulla creazione di una data mart con implementazione *Top-Down* data mart con l'obiettivo di ottenere una tabella Fatto

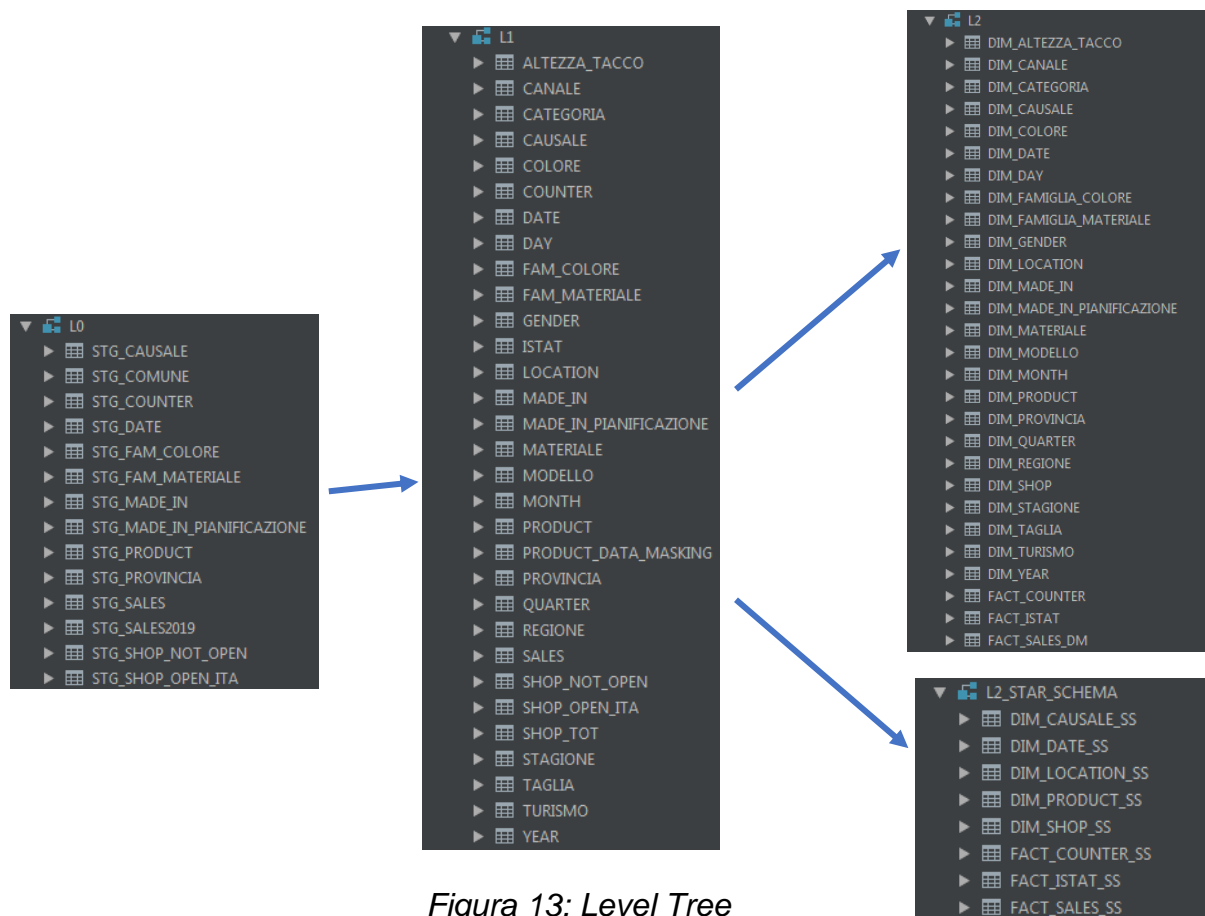


Figura 13: Level Tree

delle vendite con dati specifici e elaborati per ricavarne successivamente informazioni che attualmente il cliente non conosce e che potrebbero portarlo a produrre un potenziale vantaggio competitivo ed economico. Ogni dato segue rigorosamente il sistema ACID (Atomicità, Coerenza, Isolamento e Durabilità).

Per facilitare il riconoscimento dei file, delle tabelle e per eventuali necessità di ricarico utilizzeremo una nomenclatura ferrea a livelli che renderà più semplice il riconoscimento del singolo file. Tali livelli dovranno poi risiedere in una struttura ad albero che consente la navigazione storica e concettuale all'interno delle cartelle.

Tabella 4: Livelli ETL

SCHEMA	L0	L1	L2	L2_STARSCHEMA
Definizione	Estraggo i dati da vari tipi di file senza trasformazioni.	Principali trasformazioni e le operazioni di data quality.	Molto veloce e tabelle snelle. uso di surrogate key per collegare i vari attributi.	Poche tabelle ma corpose, per avere tutti i dati necessari per i report.
Area	Staging/Extraction Area.	Trasformation Area.	ETL Area.	Visualization Area.
Primary key	NO.	YES.	YES.	YES.
Surrogate key	NO.	YES.	YES.	NO.
Azione sulla Tabella	Truncate table, but maintain schema	Nothing.	Nothing.	Nothing.
Azione sui dati	Insert.	Update/Insert.	Update/Insert.	Update/Insert.

2.2.3 Delta Dei Dati

L'alimentazione del DWH normalmente inizia con lo scatenare il processo FULL_LOAD o Initial Load che prevede il popolamento iniziale di una situazione consistente di dati a una certa data da cui si può poi proseguire con i carichi delta.

Il carico della porzione delta dei dati può contemplare diverse situazioni:

- CDC (Change Data Capture): le tabelle vengono sottoposte a un meccanismo di change data capture che intercetta automaticamente per ogni tabella o meno il delta dei dati rispetto all'estrazione precedente e il processo nostro di replica verso L0

ottiene i dati già da caricare (consigliato per replicare le tabelle molto grandi in modo da non dover scaricare moli di dati inutili),

- **MINUS**: le tabelle sorgenti non vengono sottoposte a nessun meccanismo di rilevazione di cambiamenti e dobbiamo autonomamente estrarre la mole di dati e trovare il delta effettuando una minus dei dati che possediamo con quelli nuovi (consigliato per le tabelle di anagrafica che contengono una mole di dati nota e non prevedono cambiamenti eccessivi nella numerosità dei record);
- **FULL**: replica per intero giornaliera dei dati della tabella date le dimensioni contenute e la variabilità bassa del contenuto.

2.2.2 Storicizzazione

Le tabelle DLT possono aver una storicizzazione necessaria al ricalcolo del DWH, per evitare una perdita di dati o carico di dati parziali dovuta a problemi, di server per esempio.

Per aver una storicizzazione del dato abbiamo due scelte:

- creare tabelle ombra delle DLT partizionate per JOB_ID di estrazione chiamate HIS. Le tabelle Delta DLT saranno non partizionate e in truncate/insert (troncare la tabella mantenendo lo schema iniziale per poi inserire i nuovi dati) e conterranno solo i dati del JOB_ID attuale mentre, le HIS, saranno in insert con il partizionamento per JOB_ID, velocizzando il processo di estrazione.
- avere lo storico direttamente sulle DLT in insert con il partizionamento per JOB_ID, sempre per velocizzare le estrazioni.

È possibile utilizzare una sola delle due modalità in modo da uniformare l'architettura delle tabelle a un modello unico e, a seconda della modalità utilizzata, sarà necessario differenziare il codice di un eventuale ricalcolo.

Nel progetto preso in considerazione, le tabelle Delta e la Storicizzazione non si utilizzeranno perché i dati derivano da file csv o Excel locali con nessuna possibilità di ricalcolo, non essendo collegati ad una sorgente con un costante aggiornamento giornaliero/mensile.

2.2.3 Il Modello Multidimensionale - Dimensional Fact Model

Il modello E-R (Modello Entità–Relazione), diffuso per progettare sistemi informativi relazionali, non è adatto per esprimere e analizzare in modo dettagliato grandi moli di dati [10].

Il modello multidimensionale o DFM (Dimensional Fact Model) è un modello concettuale dove è possibile rappresentare i dati all'interno di un ipercubo i cui spigoli rappresentano le dimensioni di analisi, che successivamente verrà suddiviso in tanti “cubetti”, ciascuno dei quali è identificato da una terna di coordinate. ogni cubetto contiene idealmente i valori assunti dalle misure per quella data terna e viene comunemente denominato “fatto” in quanto rappresenta l'accadimento di un evento di interesse per il dominio di business.

Un modello multidimensionale si basa principalmente su 4 concetti chiave:

- Fatto: concetto rilevante per il processo di Decision-Making. Tipicamente modella una specifica area di business (Vendite, Ordini, Produzione, etc.), ed è caratterizzato da una a più misure;
- Misura: rappresenta l'aspetto quantitativo del fatto che risulta di elevata importanza per l'analisi. Proprio dalle Misure vengono estratti dei KPI (Key Performance Indicator) che guideranno le imprese nelle proprie strategie di business. Alcuni esempi possono essere la Quantità prodotta, il Profitto, e il Prezzo;
- Dimensione: rappresenta le coordinate di analisi del Fatto. Tra queste possiamo trovare Data, Prodotto, Negozio;
- Attributo Dimensionale: è un raggruppamento logico di alcuni elementi di una stessa dimensione. Classi di elementi che consentono all'utente di selezionare i dati per specifiche caratteristiche.

Per navigare all'interno del cubo multidimensionale esistono differenti operazioni che permettono di organizzare i dati al suo interno, attraverso diverse prospettive [10].

La prima è il Pivoting che permette di modificare rapidamente la visualizzazione dei dati girando gli assi del cubo e ha lo scopo di cambiare il punto di vista da cui si analizza i dati del cubo. La seconda, invece, è lo Slice & Dice che seleziona e proietta i dati del cubo. Nello specifico si estrarranno sotto-cubi filtrando su una (Slice) o più (Dice)

dimensioni. Infine, abbiamo il Roll-Up & Drill-Down che consentono di spostarsi all'interno di una gerarchia, scegliendo il livello di aggregazione secondo il quale l'utente desidera analizzare i dati. Nello specifico si salirà di un livello gerarchico con il roll-up, mentre si scenderà di un livello con il drill-down.

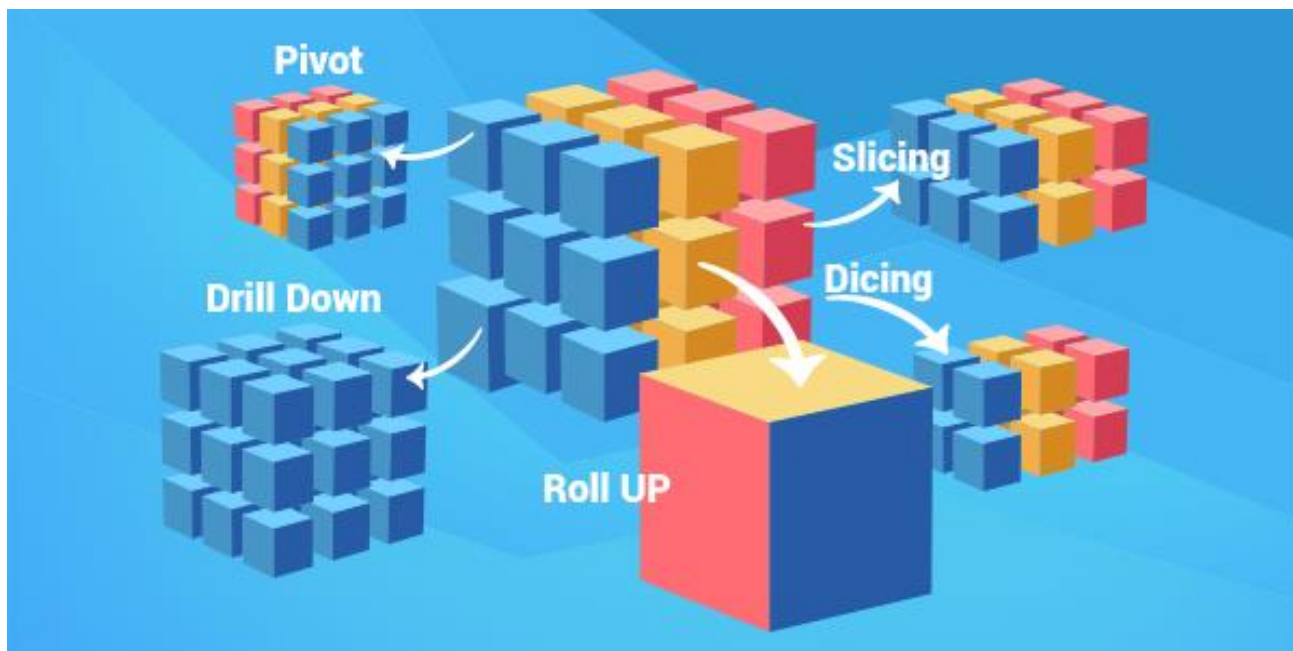


Figura 14: Hypercube OLAP

Questo sistema è stato idealizzato per portare a determinati scopi come, per esempio, fornire supporto al design concettuale, creare un ambiente dove gli utenti possano fare query in maniera intuitiva e formale per interrogare in modo efficace i report forniti, favorire la comunicazione tra designer e utenti al fine di formalizzare i requisiti di progetto, costruire una stabile piattaforma di design logico e infine, creare e pubblicare una documentazione chiara e efficace.

2.3 LEVEL L0 - DATA INGESTION

Il livello L0 rappresenta la fase iniziale chiamata Staging or Extraction Area del DWH dove avviene lo scarico delle informazioni dai sistemi sorgenti.

I sistemi sorgente possono essere di diverso genere ma i più comuni sono i sistemi operazionali su database o i file prodotti dal fornitore:

- Da tabella: lettura via rete del set di dati giornaliero e replica dell'intero set di dati;
- Da file: lettura delle informazioni contenute nell'estrazione giornaliera.

La data ingestion è il processo di acquisizione e importazione di dati per l'uso o l'archiviazione immediata in un database.

I dati possono essere trasmessi in streaming in tempo reale o ingeriti in lotti:

Quando i dati vengono ingeriti in tempo reale, ogni elemento di dati viene importato mentre viene emesso dalla sorgente. Quando i dati vengono importati in batch, gli elementi di dati vengono importati in blocchi discreti a intervalli di tempo periodici. Un processo efficace di acquisizione dei dati inizia dando la priorità alle origini dati, convalidando i singoli file e indirizzando gli elementi di dati alla destinazione corretta.

Se esistono numerose fonti di dati di grandi dimensioni in diversi formati, può essere difficile per le aziende acquisire dati a una velocità ragionevole e elaborarli in modo efficiente al fine di mantenere un vantaggio competitivo. A tal fine, i fornitori offrono programmi software su misura per specifici ambienti di elaborazione o applicazione. Quando l'importazione dei dati è automatizzata, il software utilizzato per eseguire il processo può anche includere funzionalità di preparazione dei dati per strutturare e organizzare i dati in modo che possano essere analizzati dalla Business Intelligence (BI) e dalla Business Analytics (BA).

Le Tabelle che andremo a creare in questo livello saranno tutte precedute dal prefisso "STG", da Staging Area, corrispondente all'importazione totale del documento di partenza con nessuna modifica allo schema sorgente, e con solo piccole trasformazioni dovute alla capienza delle variabili del Database di SQL Server usato per il progetto.

2.3.1 I Metadati

Il termine “metadati” si applica ai dati usati per descrivere altri dati. Nel contesto del data warehousing, in cui giocano un ruolo sostanziale, essi indicano le sorgenti, il valore, l’uso e le funzioni dei dati memorizzati nel DWH e descrivono come i dati vengono alterati e trasformati durante il passaggio attraverso i diversi livelli dell’architettura.

La o le tabelle di metadati sono strettamente collegate al DWH vero e proprio e le applicazioni ne fanno un intenso uso sia dal lato dell’alimentazione che da quello dell’analisi.

È possibile distinguere due categorie di metadati, parzialmente sovrapposte, in base ai diversi utilizzi che ne fanno l’amministratore del sistema e gli utenti finali:

- Metadati interni: di interesse per l’amministratore, descrivono, le sorgenti, le trasformazioni, le politiche di alimentazione, gli schemi logici e fisici, i vincoli e i profili degli utenti;
- Metadati esterni: di interesse per gli utenti, riguardano, per esempio, le definizioni, la qualità, le unità di misura e le aggregazioni significative.

I metadati vengono memorizzati in un apposito contenitore al quale possono accedere tutti gli altri componenti dell’architettura.

Si possono classificare inoltre riguardo il livello in cui vengono considerati:

- Globali: contengono metadati relativi a tutti i livelli e a tutti i processi, e servono per sincronizzare le varie fasi su un livello comune temporale o di dettaglio;
- Processo: distinguiamo i metadati a seconda del sistema alimentante e del processo in cui vengono coinvolti. I metadati che descrivono il singolo processo relativo o meno a un determinato sistema (punto di sincronia interno tra le tabelle, percentuale propria del sistema di tolleranza errori) devono esser proprie per ogni sistema.

È possibile definire una reportistica sui metadati in quanto sono un punto ottimale per leggere e aver chiara la situazione in ogni istante per ogni processo. È utile avere un cruscotto dove è possibile leggere la sincronia tra i processi e i sistemi sorgente o di estrazione.

Il concetto di metadato è molto critico all'interno della gestione del DWH e viene spesso dibattuto se tenerlo interno al progetto o gestirlo con logiche esterne che svincolino la tecnologia e il prodotto utilizzato dallo scopo poi effettivo del metadato.

Normalmente registrato all'interno di una tabella, per motivi di fruibilità da parte dei più eterogenei sistemi, esso ha l'utilità di descrivere univocamente e in modo preciso un'informazione su che stato si trova un processo, al fine di evitare lanci di più istanze, lanciare un processo in un momento sbagliato, dire se il processo è terminato in modo corretto o con errori, fornire l'intervallo temporale per cui quel processo terminato o in esecuzione ha estratto i dati.

Tramite le funzionalità del software ETL Talend Open Studio è stato possibile importare vari tipi di file per costruire un nuovo Database (Data Mart) in SQL Server più semplice sulla macchina 192.168.2.14 chiamato FASHION_RETAIL, Ma utile per valutare tutto il reparto vendite.

Per usare il database su Talend ho bisogno di creare una connessione al database per ogni livello, dove poi implementerò le Anagrafiche (insieme delle tabelle Dimensioni) e i Movimenti (insieme delle tabelle Fatto) tramite importazione dei metadati.

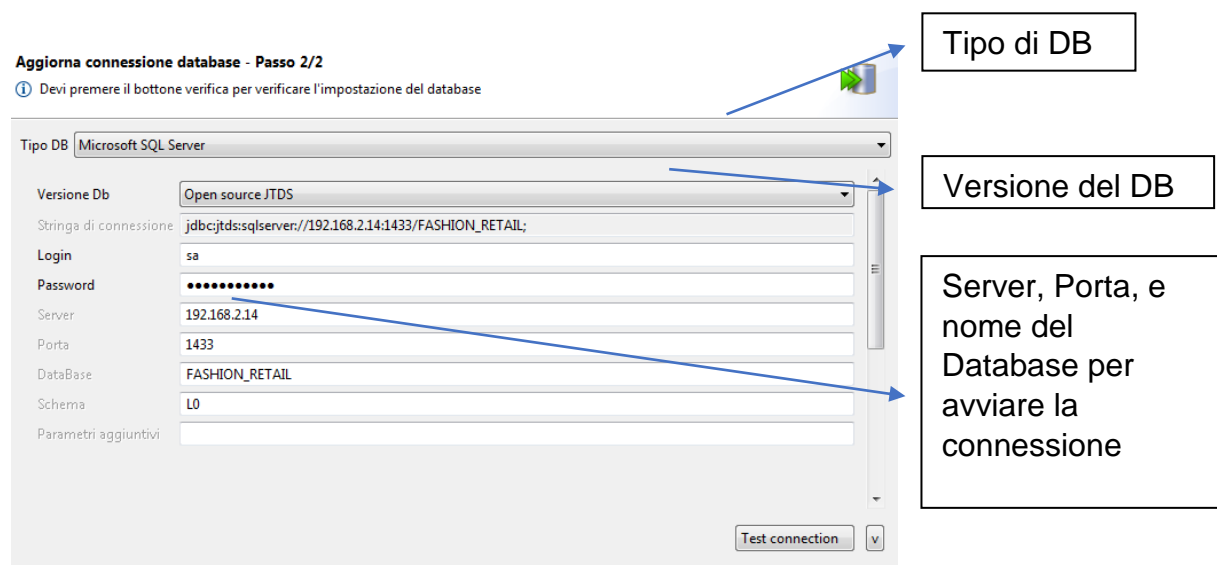


Figura 15: Connessione al Database

Nella tabella sottostante sono elencate le tabelle create a livello L0-Data Ingestion con i file sorgente estratti per il progetto, che sono principalmente di tipo Excel o Csv (Delimited file).

Tabella 5: Metadati

Database Table	Metadata Type	Metadata Name
STG_Causale	File: .Csv	Causali
STG_Made_In	File: .Csv	Made_In
STG_Made_In_Pianificazione	File: .Csv	Made_In_Pianificazione
STG_Product	File: .Csv	ProdottiS15, Prodotti<S15, Prodotti<S15n2
STG_Shop_Open	File: .Excel	Negozi aperti
STG_Shop_Closed	File: .Excel	Negozi chiusi
STG_Fam_Colore	File: .Csv	Colori
STG_Fam_Materiale	File: .Csv	Materiali
STG_Sales	File: .Csv	Scontrini, Scontrini2019
STG_Provincia	File: .Csv	Provincia
STG_Counter	File: .Csv	Contapersone, Contapersone2019

Avendo una serie di file in Csv ed in Excel comporta una fase di importazione a livello di archiviazione nel software di sviluppo del processo ETL, che li ingloberà in cartelle, una per ogni tipo, sottoforma di metadati.

Nel progetto implementato in Talend, si avranno questi due tipi di visualizzazioni di caricamento:

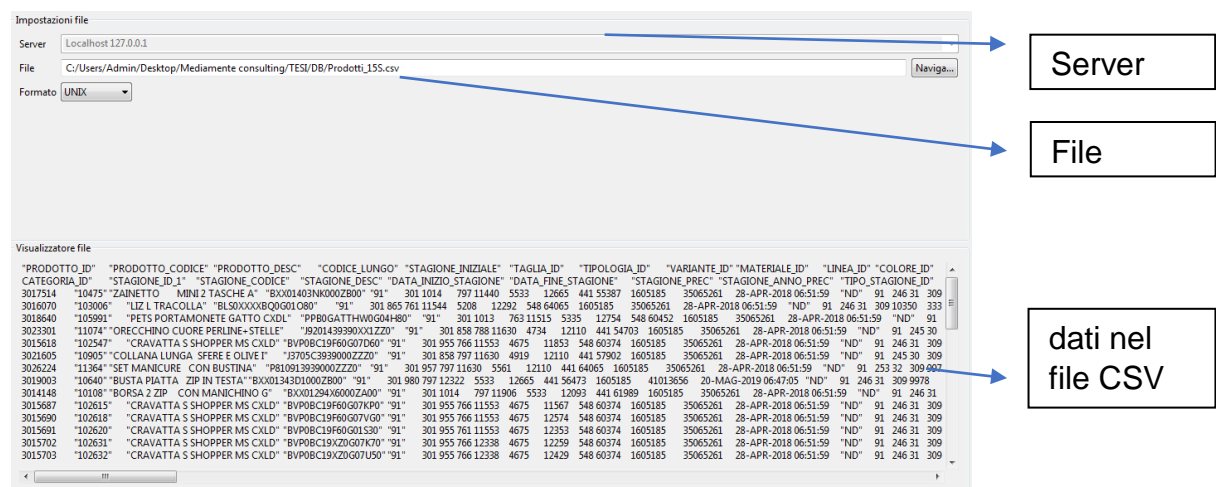


Figura 16: Caricamento da file Delimited

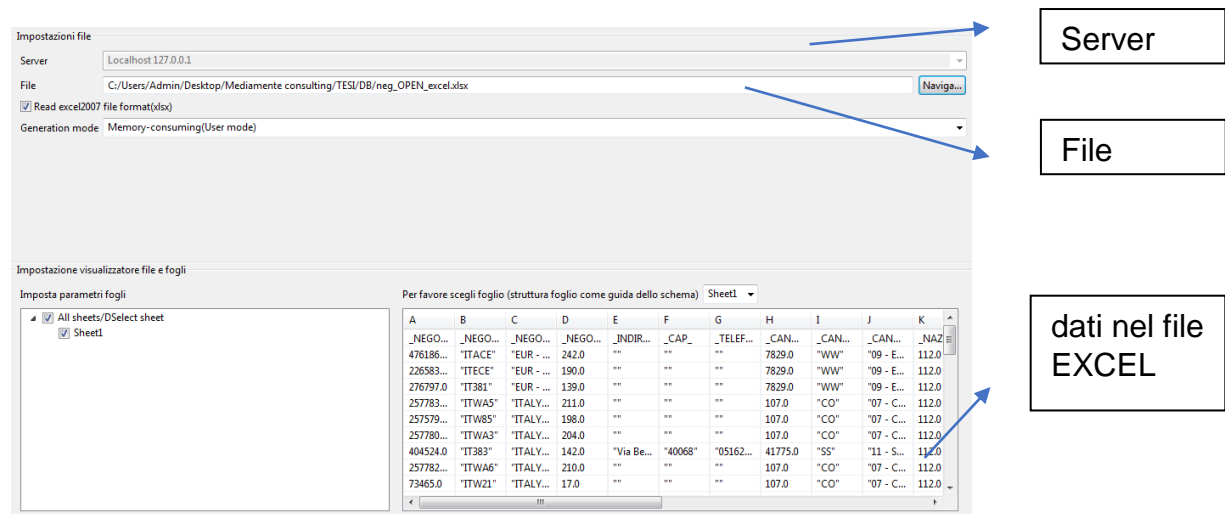


Figura 17: Caricamento da file Excel

I Metadati, una volta importati, devono essere rielaborati per la creazione del database sul server. Concettualmente per ogni job creato in Talend, sono state effettuate le seguenti quattro operazioni:

- Importazione file csv o Excel;
- Unione dei file tramite lo strumento tUnite, se necessario;
- Cambiamento e mappatura dei nomi, lunghezza o tipo degli attributi o unione di tabelle grazie alle chiavi primarie tramite lo strumento tMap;
- Creazione della tabella in SQL SERVER tramite lo strumento tDBOutput (tMSSQLOutput).

Per svolgere le seguenti operazioni è necessario creare un nuovo JOB, che conterrà i vari metodi di importazione.

Un esempio molto significativo è rappresentato dalla creazione dello STG_PRODUCT, la tabella di staging area della dimensione Prodotto.

In essa si può osservare come l'importazione di vari file vengono uniti tramite una Palette denominata tUnite, che cattura gli schemi dei file sorgenti per creane uno che si adatta a tutti; se gli schemi sorgenti sono differenti te lo segnala con un Warning, anche se il processo continua a funzionare regolarmente.

Il tool tMap vedi (2.2.4 data quality) permette di trasformare gli schemi di input per ottimizzare gli output (fase L1) e per creare delle relazioni di join tra le varie tabelle.

In questo specifico caso, aiuta ad identificare quali prodotti sono accettati secondo lo schema definito precedentemente e quali rigettati, con rispettivamente la creazione della staging area del prodotto e un file excel con i prodotti rifiutati.

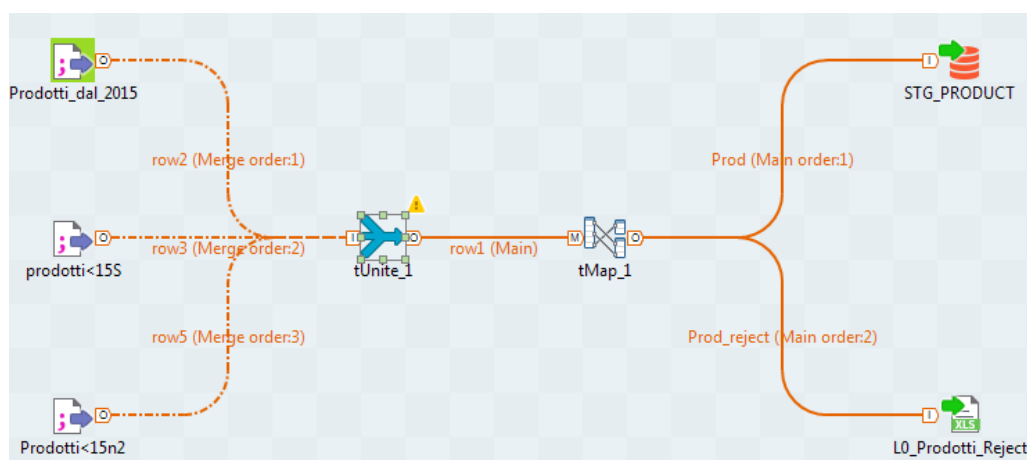


Figura 18: Multi-Caricamento

Lo stesso procedimento è stato svolto per le altre tabelle, con il caso più comune la diretta importazione del file nel database.

È importante osservare che ogni file occupa un Job (area di lavoro) diverso. Questo è necessario per prevenire errori durante il caricamento dei dati, isolando il problema.

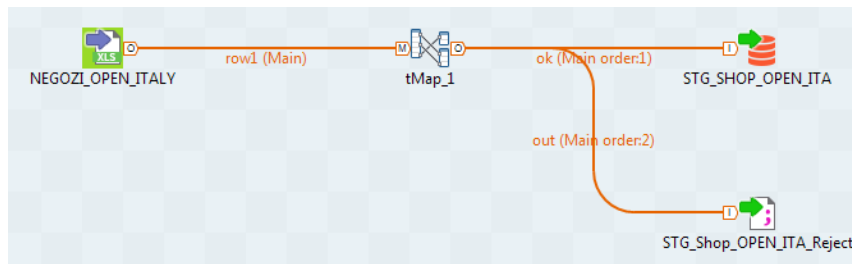


Figura 19: Mono-Caricamento in L0

2.4 LEVEL L1 - DATA OPERATION

Il livello L1, comprende la data quality, la normalizzazione dei dati, e tutte le trasformazioni dei file sorgenti, e rappresenta sicuramente il cuore di tutto il processo di ETL.

A differenza del livello di estrazione L0, i dati sono estratti esclusivamente dalle tabelle create nella Staging Area elencate precedentemente, per poi essere successivamente trasformate e caricate nello stesso Database FASHION_RETAIL, ma in uno schema differente, per valorizzarne il processo e avere un continuo controllo sulle attività che si svolgono.

Per le tabelle già presenti nella Staging Area il processo è molto semplice, in quanto si dovrà solo selezionare gli attributi nel tMap che mi interessano ed eseguire le trasformazioni adeguate per la Normalizzazione dei dati. Nel trasferimento, per un primo check di integrità, si creano le Primary Key, che andranno ad identificare univocamente l'attributo specifico di ogni tabella, spesso rappresentato da un campo ID.

Importante fase è il Pre-Loading, cioè l'estrazione delle dimensioni secondarie da altre tabelle presenti nel livello L0.

Usando come input la tabella STG_PRODUCT, per esempio, possiamo osservare la possibile formazione di nuove dimensioni e rispettivamente eseguire le queries per ciascuno di essi, estraendo univocamente, grazie alla funzione "SELECT DISTINCT" del linguaggio SQL, i campi dalla tabella sorgente relativi alla Categoria.

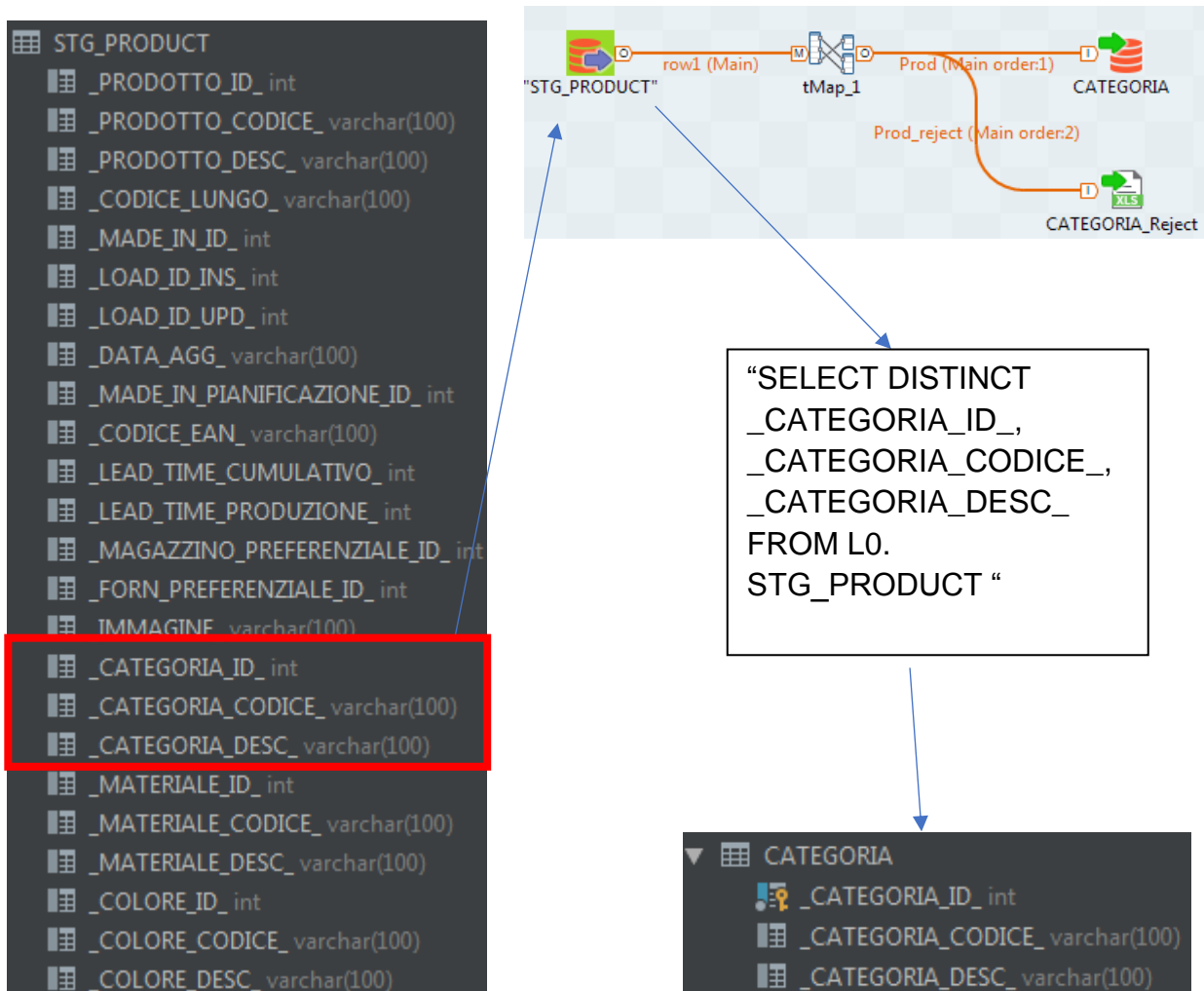


Figura 20: Pre-Loading L1

2.4.1 Data Quality

La qualità dei dati è una percezione o una valutazione dell'idoneità dei dati stessi utili per uno scopo in un determinato contesto. La qualità dei dati è determinata da fattori quali accuratezza, completezza, affidabilità, rilevanza e quanto è attuale. Poiché i dati sono diventati più strettamente collegati alle operazioni delle organizzazioni, l'enfasi sulla qualità dei dati ha guadagnato maggiore attenzione.

I Check si dividono principalmente in due tronconi:

- Di integrità referenziale: controlli tramite la verifica delle foreign key. Viene effettuato tramite join con le tabelle di L1 contenenti i padri di cui verificare le relazioni;
- Di validazione record: i dati devono subire controlli per scartare record che non soddisfano i requisiti stabiliti: Not Null e le Condizioni Simple/Complex (es.: data compresa in intervalli o campi testo di lunghezza definita ecc...)

La tabella seguente illustra (uno per tipo) tutte le operazioni di data quality eseguite durante la progettazione della DataWareHouse, eseguite tutte ne tool tMap o direttamente nei metadati:

In alcuni casi, sono state volute ulteriori trasformazioni a livello di metadati, in particolare nella dimensione dei campi. Queste modifiche sono state eseguite per evitare la data truncation, onde evitare successive incongruenze nei dati finali.

I dati di scarsa qualità sono spesso considerati come la fonte di rapporti non accurati e strategie mal concepite in una varietà di società. Il danno economico dovuto a problemi di qualità dei dati può variare da spese varie aggiunte quando i pacchi vengono spediti a indirizzi sbagliati, fino a multe salate di conformità normativa per rapporti finanziari impropri.

Tabella 6: Data Quality

Data Type Sorgente - Destinazione	Sorgente	Trasformation for Data Normalization	Destinazione
Datetime- Date	“dd-MMM- yyyy hh:mm:ss”	Change in the variable type directly in tMap options	dd-MM-yyyy
String- String	“Piemonte”	Regione.toUpperCase()	“PIEMONTE”
String- String	“Trentino- Alto-Adige”	Regione.replace ("-", " ")	“Trentino Alto Adige”
String- String	Bolzano/ Bolzen	Provincia.replace ("/Bolzen", "")	“Bolzano”

String- String	"Piemonte"	Regione.trim()	"Piemonte"
Integer- Integer	null	Totale_ Arrivi_2018 ==null ? 0 :	0
Integer- Integer	23	Totale _Arrivi_2018	23
String-double	"23"	Double.parseDouble (SCONTO)	23.0

2.4.2 Tmap Component In Talend Open Source

Il componente tMap [24] è uno dei componenti principali di processing di Talend Open Souce, ed è utilizzato principalmente per mappare i dati di input ai dati di output, ovvero mappare uno schema sorgente su uno di destinazione.

Collego gli attributi del file con gli attributi che creerò nella tabella del DB. La visualizzazione completa di essi è presente nella parte bassa della rappresentazione

INNER JOIN tramite Chiave primaria

Creazione Output contenete i record rigettati

Mapping

Figura 21: Join & Mapping in Tmap

Oltre a svolgere funzioni di mappatura, il tMap può anche essere utilizzato per unire più tabelle di input unendo i dati in un'unica tabella di destinazione.

Tutte le trasformazioni precedentemente elencate nella tabella riferita alla Data Quality, sono svolte in questa componente, con una ulteriore possibilità di filtrare i dati.

Un'espressione di mappatura può fare riferimento a qualsiasi numero di colonne da ciascuno degli schemi di input. Nell'editor è possibile utilizzare qualsiasi metodo di classe Java disponibile e le routine Talend, con la restrizione di inserire un'espressione di mappatura per ciascuna colonna in ogni schema di output.

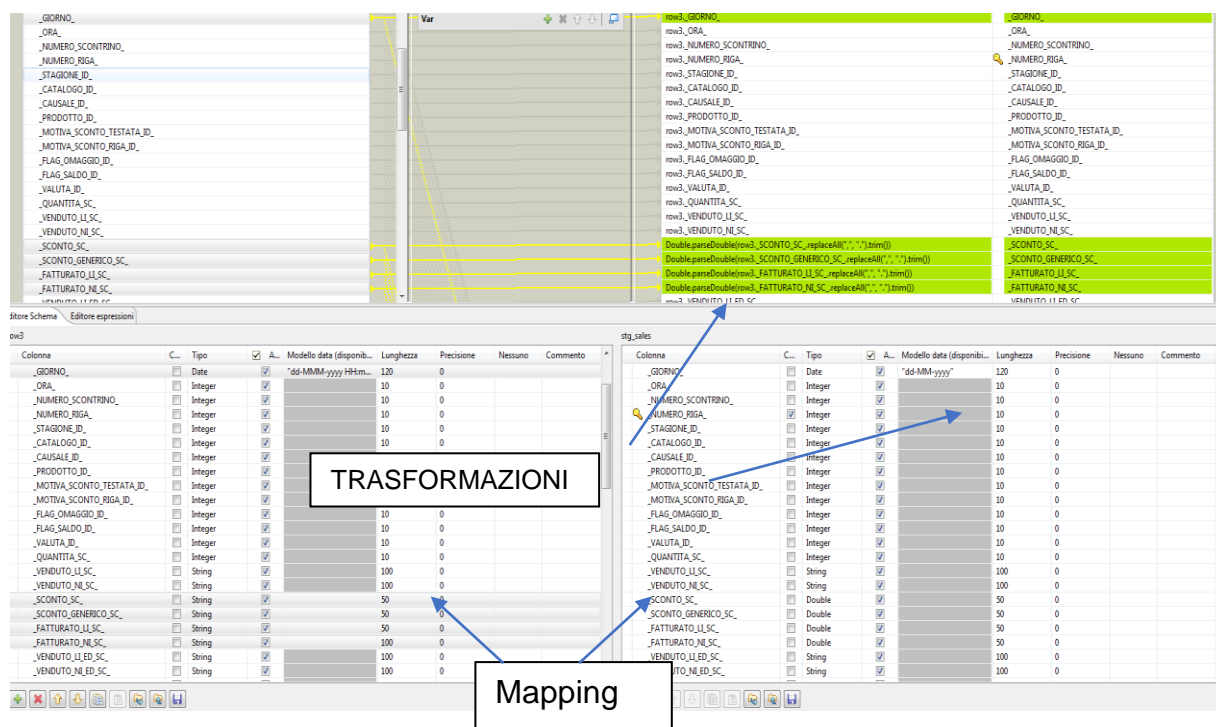


Figura 22: Trasformation in Tmap

Come si può vedere dagli esempi sopra, ogni mappatura può avere una complessità diversa in base alle nostre esigenze.

Nella best practice bisognerebbe sempre fare in modo che i dati e i modelli creati possano essere sempre accessibili per il riutilizzo. Infatti, è molto probabile che in un recente futuro, si possa estendere la nostra capacità di uno specifico attributo di una tabella, ad esempio controllandone il formato. Perciò, per avere un codice ed un processo performante è meglio cambiare la logica in una singola routine, piuttosto che operare più volte nelle singole espressioni di mappatura.

2.5 LEVEL L2 – ETL DATA MART BEST PRACTICE

Nei nuovi progetti di data analytics una delle migliori pratiche di implementazione di un data warehouse o di un data mart è lo SnowflakeDB un Database molto simile ma, allo stesso tempo, molto diverso dagli altri database relazionali, e sviluppato principalmente per avere un processo di ETL efficace e veloce.

È definito molto diverso perché è costruito sui principi del cloud: è veramente elastico, è praticamente a zero manutenzione, è quasi real-time e ha un supporto nativo per dati strutturati e semi-strutturati (JSON). Allo stesso tempo, però, è simile perché è un database relazionale colonnare memorizzato, infatti, i fondatori di SnowflakeDB provenivano da Oracle, stessa azienda fondatrice del database relazionale.

In particolar modo, il mio obiettivo è stato quello di portare un Database relazionale come SQL Server, ad un livello di progettazione in stile Snowflake:

- I dati devono essere classificati e contrassegnati in modo appropriato, in particolare, se altamente protetti;
- I dati devono conservare la propria storia attraverso audit e verifiche dei dati che dovranno essere convalidati alla fonte, ove possibile;
- I dati devono essere elaborati in micro-batches (lotti);
- I dati devono essere ingeriti e caricati e quindi trasformati come richiesto, ovvero in base alle regole di ELT o ETL;
- Elaborare i dati CDC end-to-end per evitare problemi prestazionali;
- Creare istanze di data mart separatamente in base ai requisiti specifici del Business aziendale, creando dei sistemi di governance per ogni vista semantica.

Detto questo, ci sono alcune best practice che si applicano alla implementazione di un Database Snowflake, specifiche per le sue differenze architettoniche uniche con altri database relazionali o piattaforme di big data:

- Utilizzare i DWH indipendenti di Multi-Cluster con funzionalità di storage e con scalabilità dei dati condivise per ottimizzare le esigenze di elaborazione dei vari

carichi di lavoro. Ad esempio, la Staging Area (Level L0) può trovarsi su un altro database rispetto al Core Layer (Level L1);

- Assegna un DWH virtuale separato per ogni data mart per avere un'esperienza ottimizzata di consumo dei dati;
- Mantenere, se possibile, i dati semi-strutturati nel formato originale per aumentare le prestazioni dell'elaborazione dei dati. Spesso i dati JSON vengono elaborati più velocemente di quelli convertiti in tabelle relazionali. Quando si memorizzano i dati semi-strutturati, SnowflakeDB ottimizza lo storage in base agli elementi ripetuti all'interno delle stringhe di semi-struttura;
- Carica i dati in piccoli blocchi invece di un file di grandi dimensioni e caricarli in parallelo utilizzando più nodi. Ad un cliente siamo stati in grado di caricare 24 mesi di dati di telemetria degli eventi in meno di 2 settimane con cluster di nodi piccoli;
- Assegnare cluster virtuali separati agli schemi di una data warehouse per ottimizzare le prestazioni e considerare il clustering di tabelle di grandi dimensioni per migliorare le prestazioni della query. Recluster se le prestazioni diminuiscono. A volte il cluster/reclustering può ridurre le prestazioni della query, anche se è consigliato di analizzare i dati della tabella prima di apportare tali modifiche;
- SnowflakeDB memorizza i metadati (valori min e max, valori distinti, ecc.) in modo che possa efficacemente sfoltire le micro partizioni necessarie per eseguire la scansione di una query;
- Si consiglia di eseguire l'ingestione dei dati basata su eventi per consentire l'ordine cronologico dei dati. Creare dei pipeline di dati che utilizzano la potenza di elaborazione di Snowflake. Le pipeline basate su framework flessibili di essere completamente separati dalla struttura di elaborazione effettiva. Questa operazione viene fatta tramite la creazione di chiavi surrogate (SK_KEY) di tipo INTEGER per migliorare l'efficienza permettendo un rapido collegamento e caricamento di dati per le tabelle di grandi dimensioni, dove sarà riportata solo la chiave e non tutti gli attributi. Ciò consente modifiche minime del codice se sono presenti modifiche nello schema di destinazione o nello schema di origine, potendo comunque eseguire tutte le operazioni di caricamento, trasformazione, aggregazione e elaborazione dei dati;

- Costruire un robusto framework di controllo del bilancio di audit che traccia non solo la discendenza dei dati e la qualità dei dati, ma consente l'ottimizzazione delle prestazioni del database rispetto ai costi di elaborazione;
- Crea cloni a zero copie per creare database di test o di convalida per evitare la duplicazione dei dati.

SnowflakeDB è un potente database e ha delle caratteristiche uniche che consentono una rapida implementazione dei progetti di analisi dei dati, ma come tutti gli altri database richiede un'attenta progettazione. Seguendo la best practice appena elencata, la prima parte da svolgere nel progetto è la creazione delle Surrogate Key per tutte le tabelle Anagrafiche del livello L1, creando delle relazioni padre e figlio tra le tabelle Fatto e le tabelle Dimensioni.

Per esempio, per quanto riguarda le pre-loading della tabella prodotto, le surrogate key saranno create con il seguente codice SQL:

- ***ALTER TABLE L1.PRODUCT ADD _PRODOTTO_DM_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.ALTEZZA_TACCO ADD _ALTEZZA_TACCO_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.CATEGORIA ADD _CATEGORIA_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.COLORE ADD _COLORE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.GENDER ADD _GENDER_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.FAM_COLORE ADD _FAMIGLIA_COLORE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.FAM_MATERIALE ADD _FAMIGLIA_MATERIALE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.MADE_IN ADD _MADE_IN_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.MADE_IN_PIANIFICAZIONE ADD _MADE_IN_PIANIFICAZIONE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.MATERIALE ADD _MATERIALE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.MODELLLO ADD _MODELLO_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.STAGIONE ADD _STAGIONE_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.TAGLIA ADD _TAGLIA_SK INT IDENTITY(1,1) NOT NULL;***
- ***ALTER TABLE L1.TURISMO ADD _TURISMO_SK INT IDENTITY(1,1) NOT NULL;***

A questo punto, tramite la creazione di una connessione allo schema L1 del Database Fashion_Retail in forma di metadato, procedo con l'estrazione delle tabelle aventi la

Surrogate Key in esso presenti, e tramite l'elaborazione nello strumento di Talend tMap [24], diventeranno le chiavi primarie delle nuove Tabelle Dimensioni.

Nello specifico, si vede come l'imput "row1" definito come la Tabella Categoria del livello L1 avente come chiave primaria l'Id, viene riportata a livello L2 con uno schema identico al precedente, ma con chiave è primaria la chiave surrogata:

In generale:

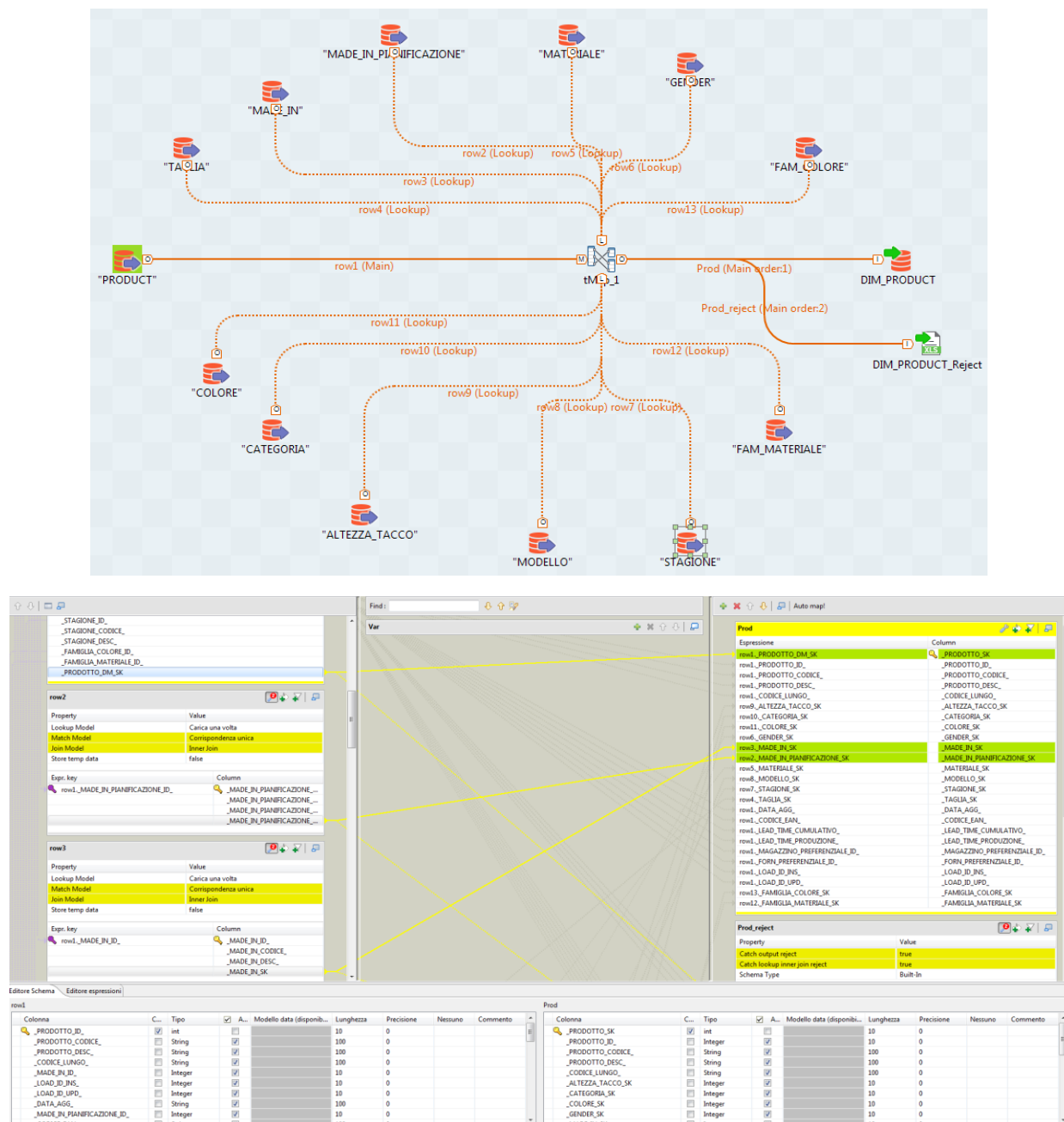


Figura 23: SnowflakeDB Dimension

Inoltre, è molto importante fare attenzione nella costruzione delle tabelle di fatto, in quanto, essendo tabelle padre delle tabelle dimensioni, non avranno una propria chiave surrogata. In esse, sarà riportata solo la surrogata delle tabelle dimensioni che la caratterizzano, tralasciando tutte le informazioni come, per esempio, l'Id e la descrizione, in quanto, in questo processo non ha alcun bisogno di scendere nel dettaglio in un'unica tabella ma è un sistema strutturato ad albero con un sistema di gerarchie, definito appunto tramite il comando Join tra le chiavi surrogate delle tabelle dimensioni.

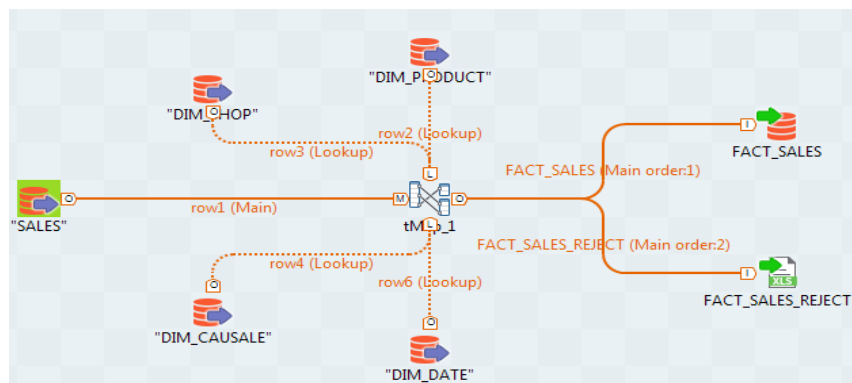


Figura 24: SnowflakeDB FACT

2.6.1 Snowflake Schema

Una base dati è in 3NF (*terza forma normale*) se tutti gli attributi non-chiave dipendono dalla chiave soltanto, ossia non esistono attributi non-chiave che dipendono da altri attributi non-chiave. Tale normalizzazione elimina la dipendenza transitiva degli attributi dalla chiave e prende il nome di schema Snowflake.

Il nome Snowflake Schema deriva dal fatto che le tabelle delle dimensioni si ramificano e assomigliano, per l'appunto, ad un fiocco di neve. Osservando il modello, si evidenzia come una tabella dei fatti è circondata da delle tabelle dimensionali, con i quali si creerà la suddetta ramificazione. A differenza dello schema a stella, le tabelle di dimensioni nello schema a fiocco di neve possono avere le proprie categorie. L'idea dominante dietro lo schema fiocco di neve è che le tabelle delle dimensioni sono completamente normalizzate. Ogni tabella delle dimensioni può essere descritta da una o più tabelle di ricerca o ancora da più tabelle di ricerca aggiuntive. Questo viene ripetuto finché il modello non è completamente normalizzato.

Ovviamente, la normalizzazione crea una maggiore complessità nell'eseguire le query dello schema snowflake, in quanto, per esempio, dovremo scavare più in profondità per ottenere il nome del tipo di prodotto o il comune di un negozio. La struttura si basa su una serie di JOIN annidati, dove ad un semplice JOIN, bisognerà aggiungerne un altro JOIN per ogni nuovo livello all'interno della stessa dimensione. Naturalmente, non esiste un numero di annidazioni standard, ma dipende dal livello del dato che si vuole estrarre. Più il dato è in profondità, più il processo di scrittura delle query sarà complesso [23].

Fondamentalmente, una query eseguita su un data mart basato su schema Snowflake verrà eseguita più lentamente rispetto ad uno su Starschema. Nella maggior parte dei casi, questo non rappresenta un problema: non importa molto se otteniamo il risultato in un secondo o in un millisecondo.

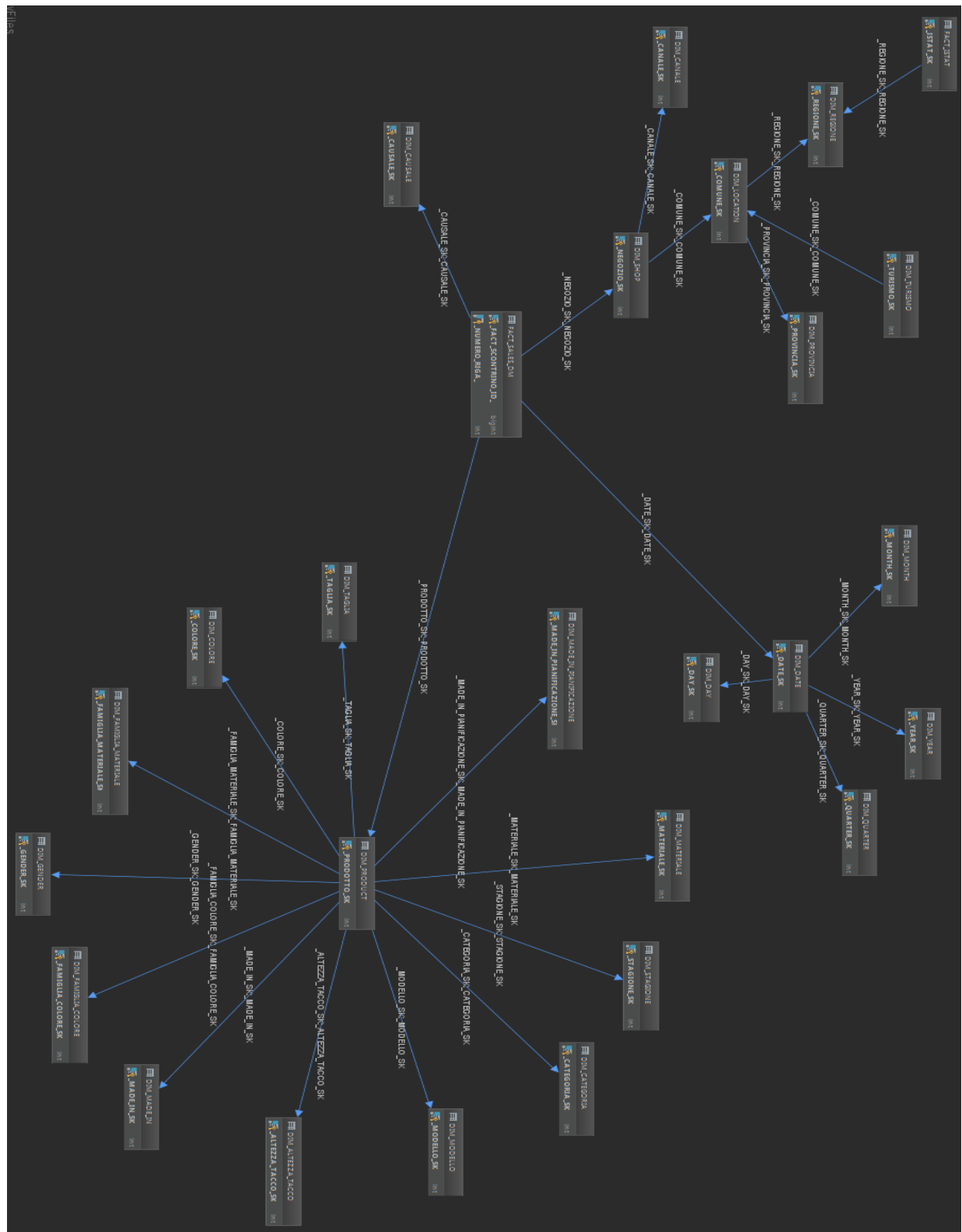


Figura 25: Snowflake Schema

2.6 LEVEL L2 – VISUALIZATION DATA MART BEST PRACTICE

I Database Relazionali, i più usati, rispetto ai SnowflakeDB presentano gli stessi livelli di Staging Area e Trasformation Area ma con una sostanziale differenziazione nel livello finale L2.

In questo caso, lo scopo non è quello di avere un processo di ETL super performante, ma di avere un minor numero di tabelle finali di grandi dimensioni aventi più informazioni possibili per facilitare, tramite software di data visualization, la creazione di report utili ai fini di decisioni strategiche future o per un semplice audit sull'andamento economico finanziario della azienda.

Lo schema del Job non cambia rispetto a quello dello SnowflakeDB, ma a cambiare è la mappatura delle variabili nel tMap [24].

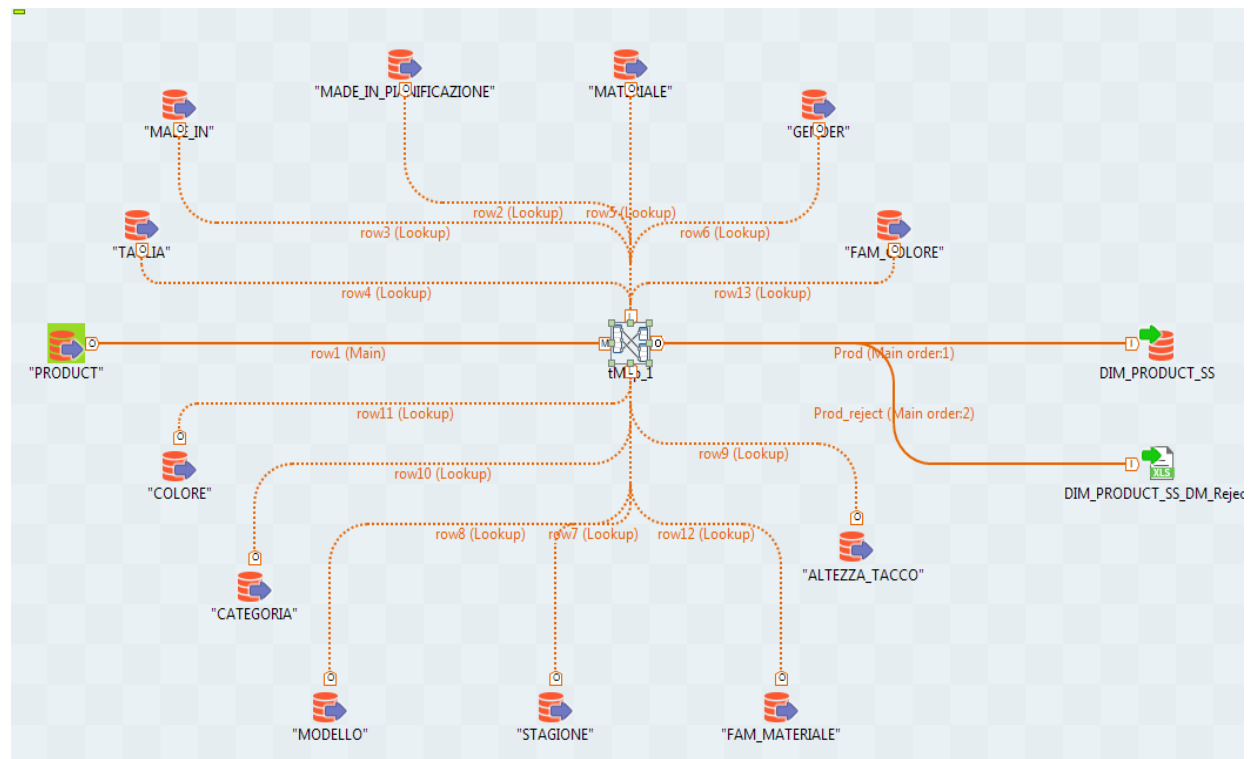


Figura 26: Job Product Star Schema

Infatti, le tabelle del Livello L2 anche in questo caso derivano dal livello L1 ed è possibile ricavarle con una serie di JOIN tra le tabelle, non più collegate alla Surrogate Key, ma direttamente alla Primary Key, con i dovuti controlli di integrità.

Gli attributi della tabella finale possono derivare anche da tabelle differenti, in quanto ogni attributo di una tabella è collegato all'attributo della tabella finale tramite l'azione i JOIN.

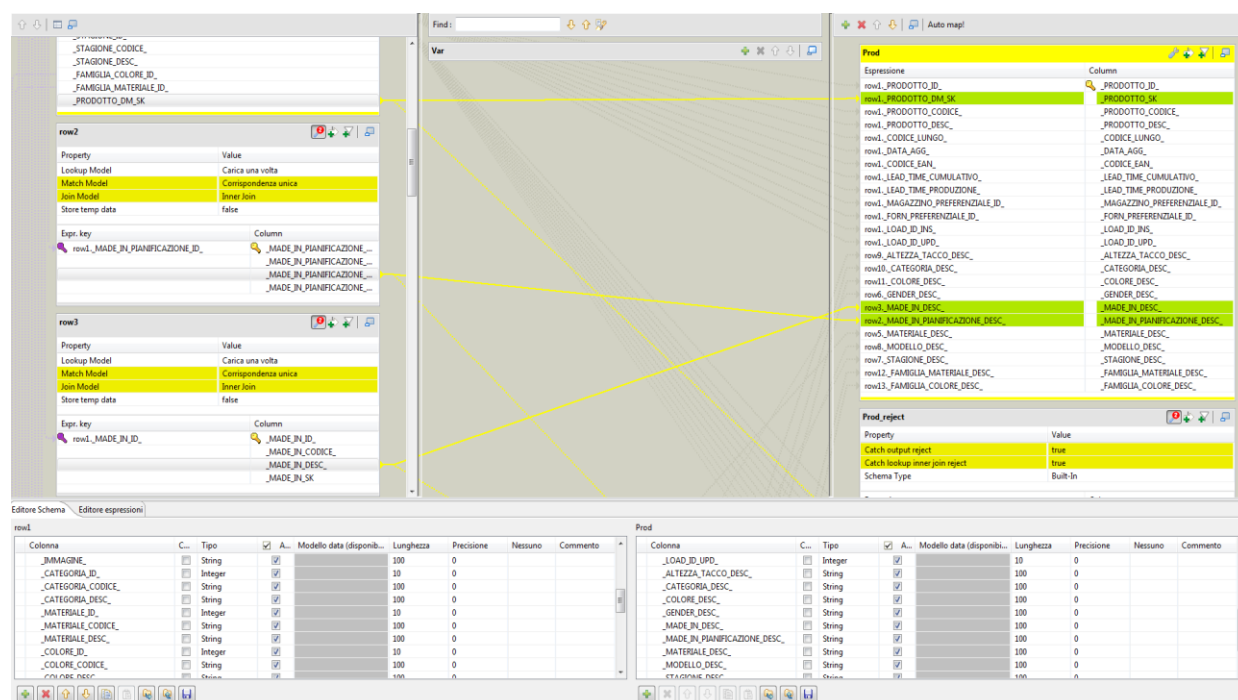


Figura 27: tMap Product Star Schema

Lo stesso procedimento si è svolto per le tabelle dei fatti, ricordandosi di fare le JOIN non con le chiavi surrogate delle tabelle dimensioni, ma farli direttamente con le chiavi primarie.

2.6.1 Star Schema

Una volta costruito il Data Fact Model, viene implementato lo schema logico. Esso viene rappresentato secondo uno Star Schema, il cui centro è costituito da una tabella dei fatti; le punte della stella rappresentano invece le tabelle delle dimensioni che si diramano dal centro. Le caratteristiche principali di uno Star Schema sono le seguenti:

- Struttura semplice di facile comprensione;
- Query molto performanti, perché riducono i join da effettuare tra tabelle;
- Tempo di caricamento dei dati relativamente lungo, perché la ridondanza dei dati dovuta alla de-normalizzazione, provoca l'aumento delle dimensioni della tabella;
- Ampiamente supportato da un gran numero di strumenti di business intelligence;
- Le tabelle dei fatti in uno Star Schema sono in terza forma normale, mentre le tabelle dimensionali sono denormalizzate [10].

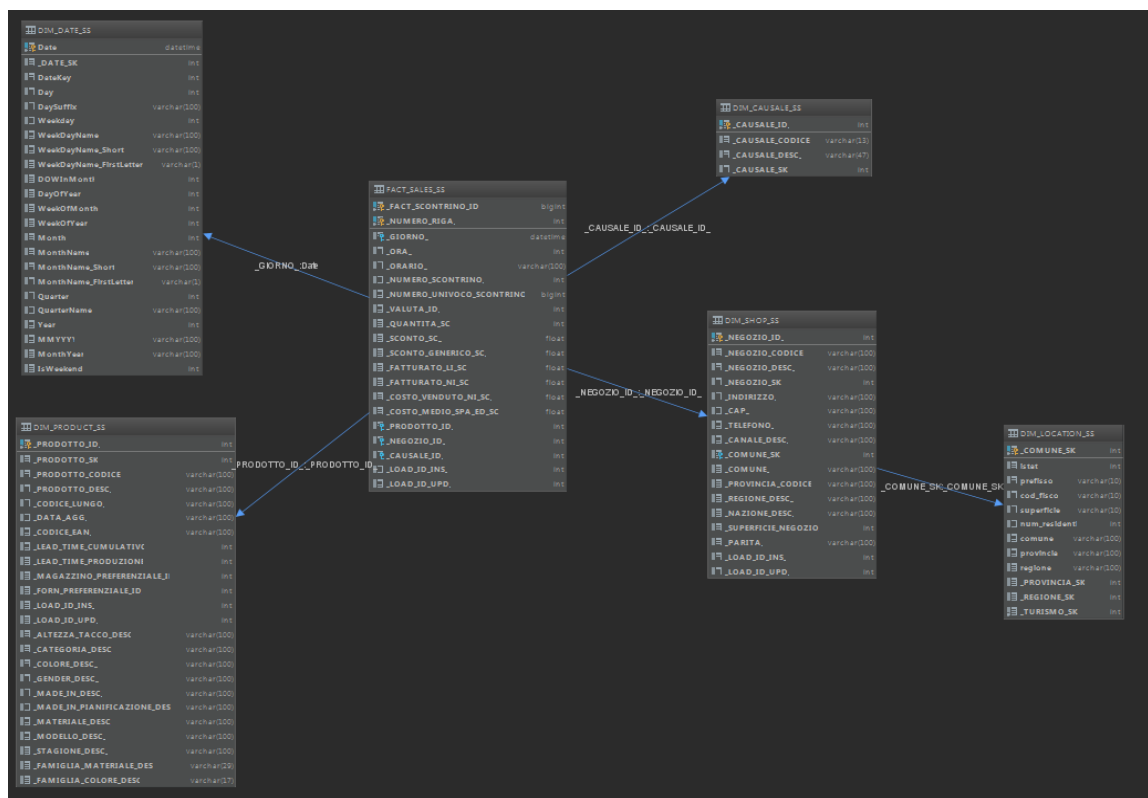


Figura 28: Star Schema

2.7 FULL LOAD ETL

Per velocizzare tutto il processo spiegato in precedenza, il metodo migliore è quello di creare dei job che contengano altri job. In questo modo, posso racchiudere in sottogruppi per ogni livello le anagrafiche e i movimenti per poi eseguirli tutti allo stesso tempo.

Per esempio, nel livello L0 ho il Job STG_Anagrafiche, dove raccolgo tutti i job del livello L0 riguardanti tutte le anagrafiche, come visualizzato nell'immagine seguente.

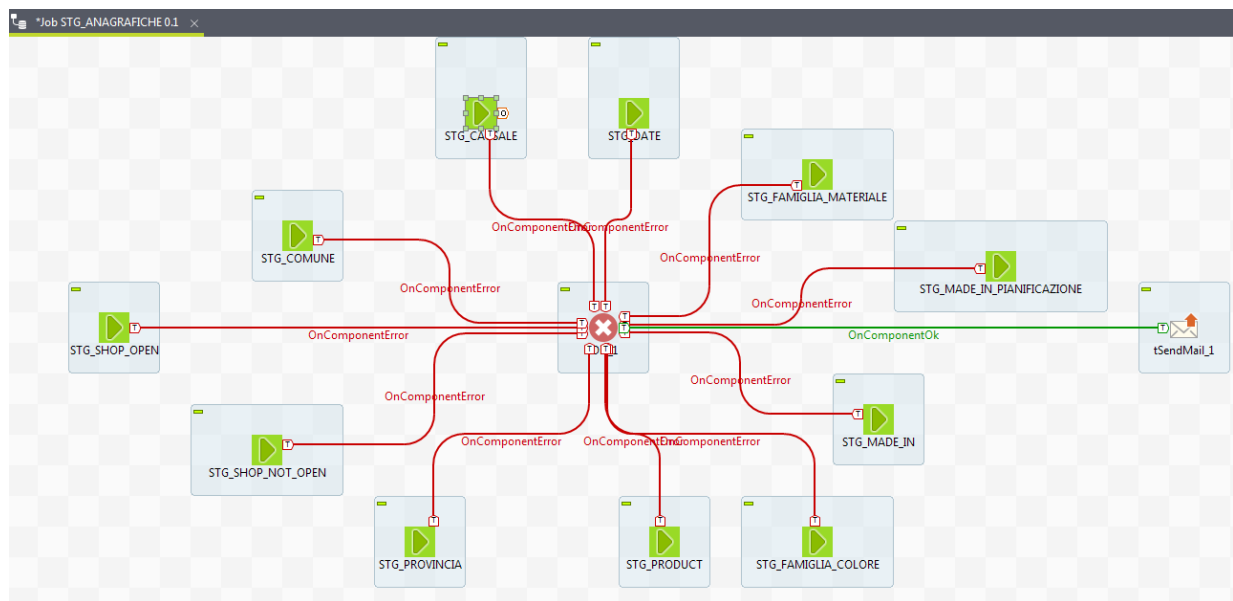


Figura 29: Job STG Anagrafiche

Lo stesso procedimento è svolto per le Anagrafiche e i movimenti di ciascun livello, arrivando al livello L2 con solo due job da unire, rispettivamente uno relativo alle anagrafiche e un ai movimenti.

Come ultimo step del processo ETL, non resta che unire i due tipi di configurazione dato con tutta la loro gerarchia in un unico job finale. Il vantaggio del Full Load ETL è la capacità tramite un univoco comando di caricare interamente tutto un database.

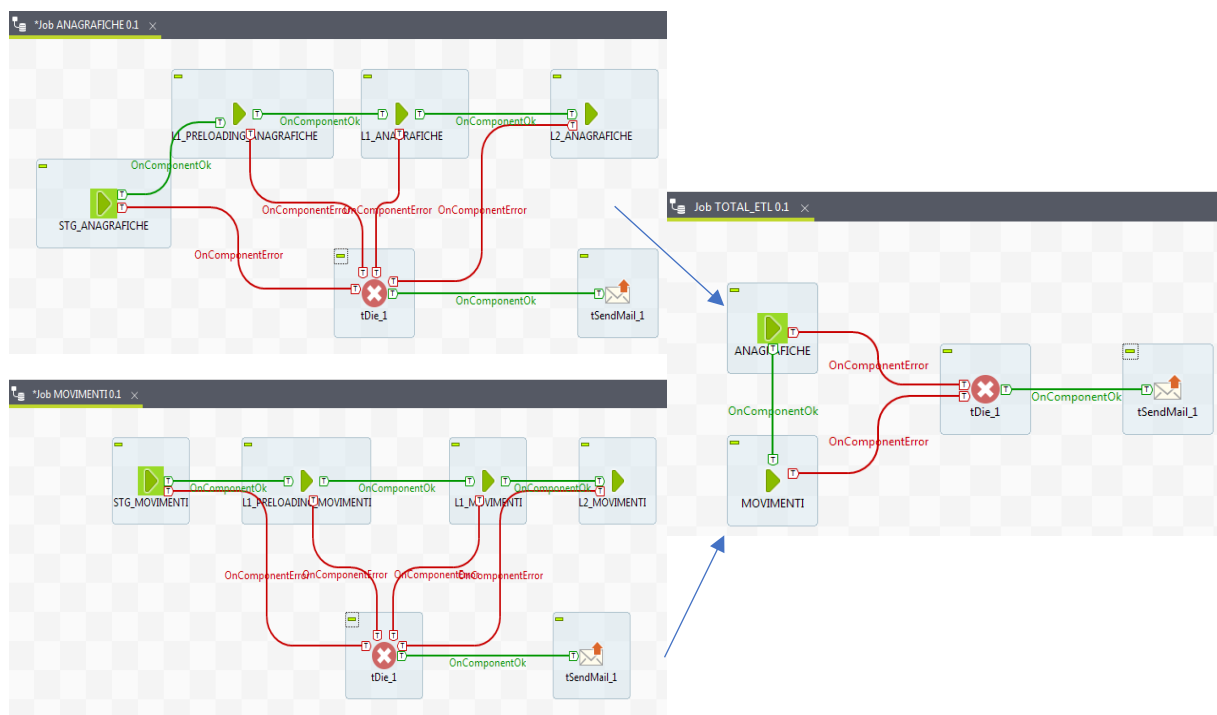


Figura 30: Full Load ETL & Auditing

È molto importante controllare il flusso dei dati nelle varie fasi del processo. Nella figura sopra stante, oltre che all'ultimo step è importante focalizzarsi su due tools: tDie e tSendEmail. Essi funziona insieme, e quando il processo va in Errore, viene mandato un segnale al tDie, che tramite l'aiuto dello strumento tSendEmail, manda una Email, con l'allegato in forma di Script dell'errore, sia al proprietario che al gestore del Database.

Un esempio riguardante tutto il processo esplicativo è spiegato nell'appendice A5.

2.7.1 Auditing ETL

Il controllo in un processo di estrazione, trasformazione e caricamento ha lo scopo di soddisfare i seguenti obiettivi:

- Verificare le anomalie dei dati oltre a controllare semplicemente gli errori gravi;

- Catturare e archiviare una traccia elettronica di eventuali modifiche materiali apportate ai dati durante la trasformazione.

L'auditing ETL aiuta a confermare che non ci sono anomalie nei dati anche in assenza di errori. Un meccanismo di auditing ben progettato aggiunge anche l'integrità del processo ETL eliminando l'ambiguità nella logica di trasformazione, intrappolando e tracciando ogni modifica apportata ai dati lungo il percorso. Anche nelle architetture ETL più rudimentali, è possibile controllare alcune metriche di alto livello per confermare che i dati caricati sono quelli previsti.

In generale, i processi ETL di auditing dovrebbero verificare quanto segue per confermare che gli input corrispondono agli output:

- Conteggio generale delle righe;
- Totali aggregati (che potrebbero includere importi finanziari o altri dati di riepilogo).

Alcuni processi richiedono un audit più esaustivo. In altri casi, potrebbe essere necessario verificare se i dati siano entro limiti ragionevoli o se supportano tali valori. Altro aspetto che non bisogna dimenticare di verificare i casi in cui non è stato caricato alcun dato. Purtroppo, succede spesso, e le due maggiori cause sono dovute ad un file sorgente che non contiene dati, una query configurata in modo errato che non restituisce righe o una directory di origine vuota destinata a contenere uno o più file potrebbero portare al corretto completamento del processo ETL ma a caricare esattamente zero righe di dati. Tuttavia, se un determinato processo dovrebbe sempre comportare un numero di file caricato diverso da zero, assicurarsi di aggiungere una fase di controllo per verificarlo.

L'auditing ETL è raramente l'elemento più visibile nell'architettura, ma è una polizza assicurativa necessaria per proteggere l'integrità dei dati e del processo.

CAPITOLO 3: ALGORITMI DI DATA MINING PER IL FASHION RETAIL

Negli ultimi decenni, lo sviluppo di informazioni e comunicazioni delle tecnologie hanno danno nuova vitalità al marketing aziendale. I dati da immagazzinare ed analizzare 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.

La tecnologia Data Mining nel marketing è un'applicazione relativamente universale. Queste applicazioni sono riferite a una Boundary Science, una varietà di teorie scientifiche basate principalmente sulle discipline di base dell'Information Technology, del Marketing e dello 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 [20].

In generale, attraverso l'estrazione, il trattamento e lo smaltimento di una grande quantità di informazioni per identificare l'interesse, le preferenze e i comportamenti di specifici gruppi o dei singoli consumatori, le abitudini di consumo, ma soprattutto, la domanda, orientando le vendite per un marketing dal contenuto specifico.

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 delle 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 altre 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 la predizione, la segmentazione e la classificazione del cliente, il profiling e il cross-selling.

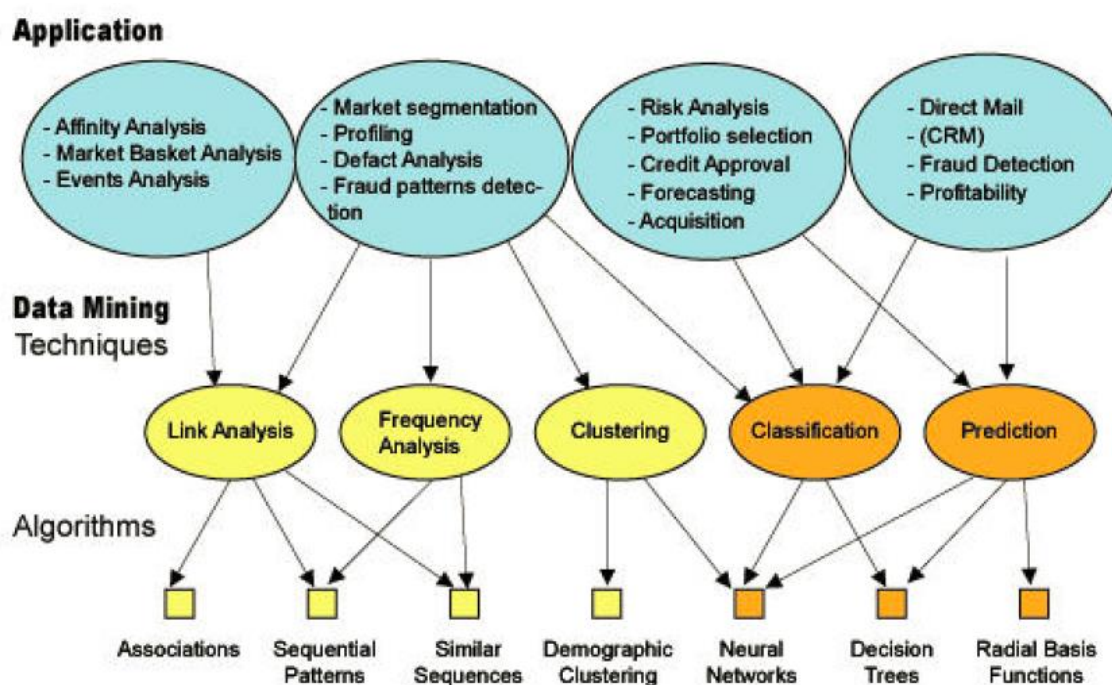


Figura 31: Applicazione Del Data Mining Per Il Marketing

Esse possono essere utilizzate anche per operazioni di valutazione del credito e frode.

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

- Preparare i dati primitivi: Include informazioni di carattere individuale (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 molte tecnologie tradizionali di data mining e molte tecnologie da altri argomenti correlati. Tuttavia, il problema che tali tecnologie dovrebbero risolvere è quello di individuare il mercato migliore o accettabile, all'interno di fonti di dati limitate, tempo limitato e spese limitate.

In definitiva, utilizzare questo modello per selezionare i clienti e decidere il piano di marketing.

Nel nostro progetto andremo a idealizzare una possibile predizione dei dati ISTAT italiani del 2018 tramite la regressione lineare, in quanto adattabili solo fino al 2017 dal sito omonimo [25]. Inoltre, si svilupperà una classificazione CART per capire le future aspettative dei negozi attualmente aperti, e per trovare una possibile ideale locazione per aprire un nuovo store.

3.1 PREDIZIONE DELLA BEST GEO - LOCATION PER APRIRE UN NUOVO NEGOZIO

Per una analisi completa e affidabile, è molto importante l'integrità e la completezza di dati. Dopo una ricerca in vari siti dedicati, per il progetto si è scelto di analizzare e prevedere i dati dell'Istituto Nazionale di Statistica, comunemente denominato ISTAT [25].

I dati effettivi hanno un orizzonte che parte dal 2004 al 2017. Essendo nel 2019, i dati forniti non si possono considerare completi. Perciò, si è deciso di affrontare il problema tenendo conto di un orizzonte temporale di dieci anni, considerando i dati dal 2007 al 2017 per fare una previsione riferita al 2018.

La ricerca per decidere gli indicatori adatti all'analisi, si è svolta seguendo una procedura matriciale, facendo un mapping diviso in aree geografiche e serie temporale coperta, inserendo il tutto in una tabella esplicativa mostrata nell'appendice A2.

La preferenza a livello regionale si sono espresse con un totale di cinque indicatori, uno o massimo due per tipo:

- Settore Trasporti: Rete ferroviaria in esercizio;
- Settore Famiglia: Spesa media mensile familiare per beni e servizi non alimentari, reddito medio;
- Settore Macro-Economia: Pil Pro Capite;
- Settore Lavoro: Tasso di disoccupazione.

Mentre, a livello comunale, si è evidenziato la popolazione residente e il turismo.

Prima dell'implementazione del codice, è stato necessario creare dei fogli di lavoro univoci per ogni indicatore, favorendo una predizione efficace dei dati, data la diversità di origine degli stessi. La suddivisione, onde evitare meccanismi complicati e macchinosi ed eventuali errori di copiatura, si è svolta tramite Talend Open Studio [28], software di ETL già ampiamente discusso nel capitolo precedente.

Il job prende i dati completi precedentemente caricati nella Staging Area e tramite l'utilizzo di Query, Ogni tabella viene interrogata in modo da estrarre nel foglio Excel dedicato, solo i dati regionali inerenti alla nostra analisi.

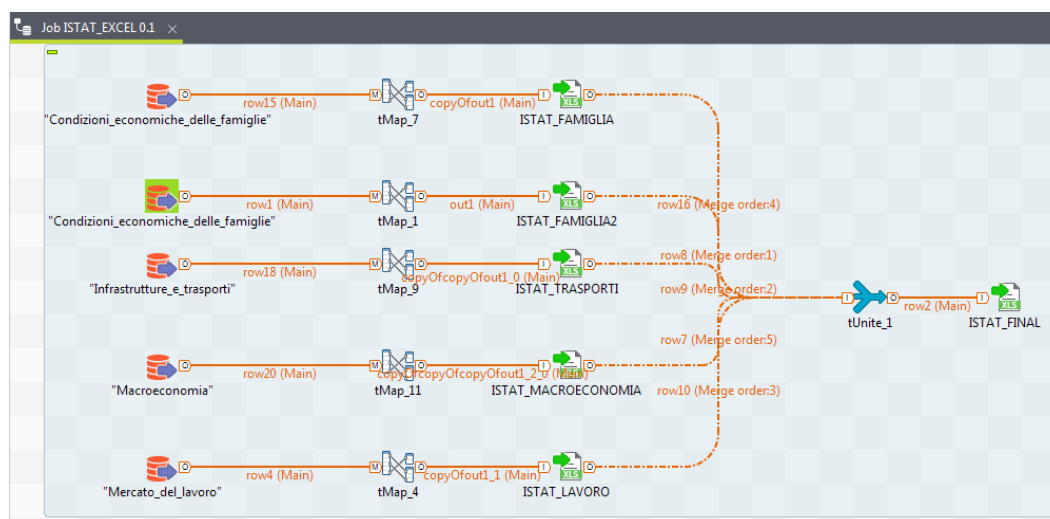


Figura 32: Job ISTAT Excel

Per esempio, per estrarre solo i dati relativi al '*Spesa media mensile familiare per beni e servizi non alimentari*' delle famiglie, si è considerato la tabella dedicata ai dati delle condizioni economiche delle famiglie dell'Istituto Nazionale di Statistica Italiano e si è

estratto Il reddito Medio, evidenziandolo con la clausola SQL IN.. La stessa cosa è stata fatta per escludere i dati nazionali o relativi alle zone di appartenenza escludendo gli altri campi (Centro Nord, Nord-Est, ...) con un NOT IN.

Naturalmente, le stesse operazioni sono state svolte per gli altri indicatori.

La Query usata nel caso della Spesa media mensile è la seguente:

"

```
SELECT *  
FROM OPEN_DATA_ITALY.Condizioni_economiche_delle_famiglie  
WHERE Indicatore IN ('Spesa media mensile familiare per beni e servizi non  
                        alimentari')  
AND Territorio NOT IN ('Nord-ovest',  
                        'Bolzano/Bozen',  
                        'Trento',  
                        'Nord-est',  
                        'Nord',  
                        'Centro',  
                        'Centro-Nord',  
                        'Mezzogiorno',  
                        'Italia')"
```

Creando i File Excel univoci per ogni indicatore è ora possibile svolgere la Regressione Lineare.

3.1.1 Regressione dei dati ISTAT

Per eseguire la predizione, si è utilizzato un "R", un linguaggio e un ambiente dedicato per il calcolo statistico e grafico. È un progetto GNU e fornisce un'ampia varietà di modelli statistici (modellazione lineare e non lineare, test statistici classici, analisi di serie temporali, classificazione, clustering, ...) e tecniche grafiche ed è altamente estensibile. Il linguaggio R fornisce un'opzione Open Source per la partecipazione a tale attività, con il supplemento di R. Uno dei punti di forza di R è la facilità con cui è possibile produrre trame di qualità di pubblicazione ben progettate, compresi simboli matematici e formule dove necessario. È stata prestata grande attenzione alle impostazioni predefinite per le scelte di progettazione minori nella grafica, ma l'utente mantiene il controllo completo [27].

R-studio è un ambiente di sviluppo integrato per R, con una console, un editor di evidenziazione della sintassi che supporta l'esecuzione diretta del codice e strumenti per il tracciamento, la cronologia, il debug e la gestione dello spazio di lavoro [26].

L'obiettivo è quello di estrarre un valore predittivo Y riferito all'anno 2018, condizionato dalle variabili X_i identificate come gli anni dal 2007 al 2017.

La risultante è la forma della regressione lineare multi-variabile:

$$Y = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon$$

Un esempio di codice R usato per la regressione lineare è quello mostrato nell'appendice A4, dove l'indicatore preso in considerazione è per la spesa media mensile.

Per ottenere i risultati effettivi della nostra regressione è necessario utilizzare il comando "summary(reg)", che creerà un output come quello sottostante dove sono visualizzati tutti gli indicatori più importanti per valutarne la reale bontà del modello:

summary(reg)

Coefficients:

	<i>Estimate Std.</i>	<i>Error</i>	<i>t value</i>	<i>Pr(> t)</i>
<i>(Intercept)</i>	-64.51987	67.42781	-0.957	0.36363
<i>Istat_Famiglie\$Anno_2007</i>	0.36274	0.15502	2.340	0.04402 *
<i>Istat_Famiglie\$Anno_2008</i>	0.07351	0.10744	0.684	0.51106
<i>Istat_Famiglie\$Anno_2009</i>	0.23755	0.12889	1.843	0.09843 .
<i>Istat_Famiglie\$Anno_2010</i>	-0.31284	0.20341	-1.538	0.15842
<i>Istat_Famiglie\$Anno_2011</i>	-0.04140	0.12748	-0.325	0.75278
<i>Istat_Famiglie\$Anno_2012</i>	-0.82588	0.32808	-2.517	0.03292 *
<i>Istat_Famiglie\$Anno_2013</i>	1.65055	0.38821	4.252	0.00214 **
<i>Istat_Famiglie\$Anno_2014</i>	-0.88867	0.37681	-2.358	0.04271 *
<i>Istat_Famiglie\$Anno_2015</i>	0.29580	0.34225	0.864	0.40987
<i>Istat_Famiglie\$Anno_2016</i>	0.51256	0.18976	2.701	0.02435 *

Residuals:

Min	1Q	Median	3Q	Max
-49.282	-17.568	2.277	18.774	37.037

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37.12 on 9 degrees of freedom

Multiple R-squared: 0.9957

Adjusted R-squared: 0.9908

F-statistic: 206.5 on 10 and 9 DF

p-value: 2.185e-09

Lo stesso procedimento è stato svolto per gli altri indicatori, ad eccezione del reddito, dove è stato necessaria una predizione sia dei dati del 2017 che un'ulteriore analisi per fornire i dati del 2018, in quanto mancanti entrambi.

Come rispecchia l'output del comando `summary`, i risultati ottenuti sono molto accettabili. Infatti, I risultati della stima di un modello di regressione lineare potrebbero e dovrebbero riportare:

- Un numero adeguato di osservazioni;
- I valori delle stime dei parametri β accettabili;
- I valori delle statistiche dei test t di Student associati a ciascun parametro, onde valutarne la significatività; tali statistiche sono spesso accompagnate dall'indicazione dell'errore *standard* associato, nonché del *p-value* che è considerato accettabile solo se inferiore a 0,10, 0,05 o 0,01;
- Statistiche atte a valutare la bontà complessiva del modello; queste possono essere a seconda dei casi limitate a misura di bontà del *fitting* quali R^2 e R^2 Adjustment per i gradi di libertà. R^2 varia tra 0 ed 1: quando è 0 il modello utilizzato non spiega per nulla i dati; quando è 1 il modello spiega perfettamente i dati;
- Statistiche dei test quali il *test F*, ossia la statistica F di Fisher associata all'ipotesi nulla che tutti gli elementi di β , Per verificare la significatività dell'intero modello si utilizza il test F. Si vuole verificare l'ipotesi $H_0: \beta_1 = 0, \dots, \beta_k = 0$ contro l'alternativa

che almeno uno dei parametri sia diverso da zero. Sotto l'ipotesi che gli errori siano $N(0, \sigma^2)$, la devianza totale ammette sempre la scomposizione $SST = SSE + SSR$.

3.1.2 Processo ETL dei dati ISTAT

Dopo aver affrontato il problema della completezza dei dati fornita dai dati ISTAT, risolta con la predizione, si possono caricare i dati ottenuti con un semplice processo di ETL, come svolto in precedenza nel capitolo 2.

Il principio è lo stesso. Dopo aver trascritto i dati ottenuti dalla regressione nel foglio di calcolo fornito dal file ISTAT, si passerà alla creazione del metadato in Talend [24].

Questo, sarà inserito in un job e caricato a Livello L0 nella Staging Area, senza svolgere nessuna operazione. La data quality sarà svolta dalla tMap [25] nel Livello L1, per poi essere collegati tra di loro tramite la Surrogate Key a livello L2. Negli ultimi due processi si lavorerà direttamente sul database, senza intaccare il foglio Excel originale.

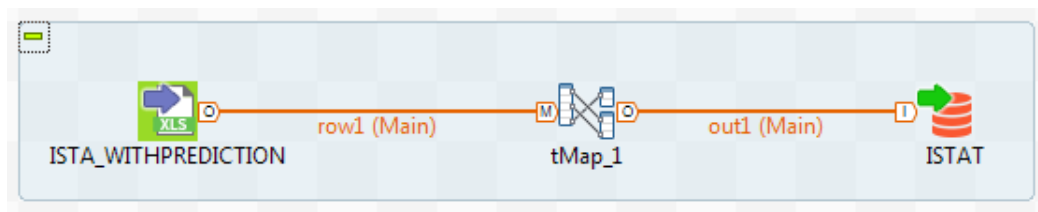


Figura 33: Job ISTAT with Prediction

Naturalmente, i collegamenti saranno svolti tramite la dimensione riferita alle regioni, che a sua volta sarà legata alle provincie che saranno legate ai comuni, creando una gerarchia a livello relazionale. Invece, per quanto riguarda la tabella Comune, già compresa dell'attributo riferito alla popolazione, sarà collegata ulteriormente con la tabella con i dati relativi al turismo. Molto importante è annotare come la tabella con i relativi dati Istat, sarà considerata come una tabella di fatto, e non come una semplice dimensione.

Una semplice visualizzazione dei collegamenti tra le varie tabelle è mostrata dalla figura seguente:

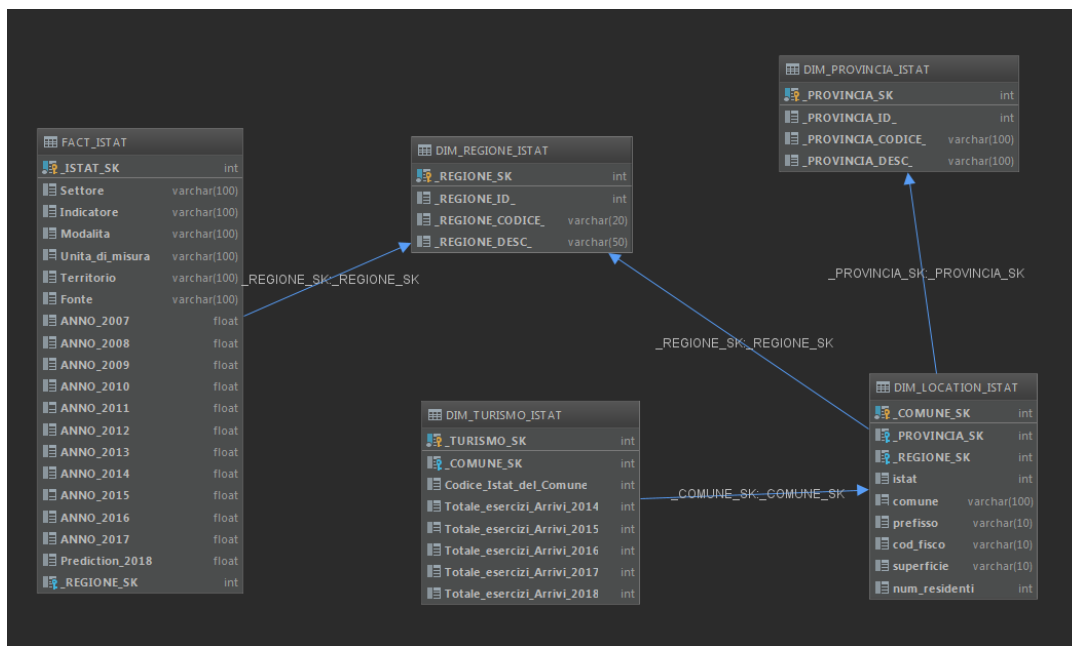


Figura 34: Star Schema Fact ISTAT

3.1.3 Best Geo-Locations

per svolgere una analisi completa e ottenere un risultato attendibile partendo dai dati ISTAT [25], evidenziati e trasformati tramite il processo di data quality, sono state necessarie principalmente tre query, tutte estremamente collegate tra loro, con l'obiettivo finale di trovare quali, tra le migliaia di città sono le più appetibili economicamente per aprire un nuovo negozio.

Inizialmente, le query saranno relative agli indicatori regionali presi dal sito dell'ISTAT (spesa media mensile familiare per beni e servizi non alimentari, reddito medio, rete ferroviaria in esercizio, tasso di disoccupazione, PIL Pro Capite), mentre l'ultima, sarà estesa in profondità a livello comunale, filtrando per le migliori tre regioni ottenute tramite la soluzione delle precedenti interrogazioni del database.

La prima query è una visualizzazione completa delle migliori e peggiori due regioni per ogni indicatore illustrando, inoltre, il relativo settore e l'unità di misura che lo caratterizza.

Per implementare il modello, verrà creato un rank tramite una funzione analitica di partizione, che può essere decrescente, se si trattano di dati economici positivi, o crescente, se si tratta di dati non molto favorevoli allo sviluppo come il tasso di disoccupazione.

Nella seconda Query sarà invece creato un fattore di somma dei vari rank raggruppati per regione e ordinati per rank decrescente. In questo modo, si otterrà come risultato un ordine di potenzialità di sviluppo delle regioni Italiane. Ad esso, si sommeranno i dati aziendali relativi allo storico del numero di negozi chiusi e ancora aperti per ogni regione e, il valore finale su cui sarà basata la nostra classifica, prenderà il nome di Rank.

Come è mostrato nella tabella seguente, le tre regioni migliori su cui investire un capitale economico sono Trentino Alto Adige, Valle D'Aosta e Friuli Venezia Giulia.

Tabella 7: Ranking delle Regioni

Rank	Territorio	#Close_Shop	#Open_Shop	Istat_Rank	Final_Rank
1	Trentino Alto Adige	6	1	14	21
2	Valle D'aosta	0	0	26	26
3	Friuli Venezia Giulia	4	1	37	42
4	Liguria	1	0	46	47
5	Piemonte	7	5	37	49
6	Emilia Romagna	19	7	23	49
7	Veneto	11	7	34	52
8	Umbria	2	1	51	54
9	Toscana	12	5	38	55
10	Lombardia	14	17	32	63
11	Marche	3	1	59	63
12	Molise	1	0	65	66
13	Abruzzo	7	1	58	66
14	Basilicata	1	0	66	67
15	Sardegna	6	0	76	82
16	Lazio	23	12	47	82
17	Calabria	4	0	83	87
18	Puglia	7	4	80	91
19	Campania	11	3	87	101
20	Sicilia	13	5	90	108

L'ultimo step da svolgere per trovare la posizione ideale per aprire un nuovo negozio è quella di selezionare i comuni presenti nelle tre regioni precedentemente scelte come le più appetibili e andare in profondità analizzando i dati relativi al numero di turisti riferiti all'anno scorso e il numero di residenti per ogni città.

La soluzione è rappresentata dalla creazione di un nuovo indicatore ottenuto tramite il rapporto dei turisti diviso per il numero dei residenti.

Le città con l'indice più alto saranno quelle più interessanti.

Tabella 8: Ranking dei Comuni

Rank	Comune	Numero Residenti	Turismo_2018	Index
1	Andalo	1036	163938	158
2	Corvara In Badia	1344	197682	147
3	Gressoney-La-Trinitè	305	40569	133
4	Campitello Di Fassa	734	92091	125
5	Valsavarenche	179	21665	121
6	Rhemes-Notre-Dame	107	12926	120
7	Lignano Sabbiadoro	6616	691154	104
8	Canazei	1921	187328	97
9	Avelengo	725	69218	95
10	Mezzana	884	81943	92
11	Selva Di Val Gardena	2657	245928	92
12	Molveno	1123	99145	88
13	Stelvio	1184	98287	83
14	Sesto	1913	156239	81
15	La Thuile	791	58552	74

Il risultato ottenuto evidenzia come Andalo, Corvara In Badia e Gressoney-La-Trinitè sono le migliori città dove poter aprire un negozio nel 2019.

3.2 CLASSIFICAZIONE: CART

L'algoritmo di classificazione CART, come precedentemente introdotto nello Stato dell'arte, è una procedura non parametrica che costruisce un albero decisionale con l'obiettivo di etichettare un attributo; Infatti, con il termine 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, esso può essere usato per predire la classe di nuove istanze di record per cui la classe è sconosciuta.

Gli step importanti da seguire quando si costruisce un albero decisionale con la procedura CART sono principalmente due: adottare un criterio di bontà della tecnica con

i cui i nodi vengono suddivisi da parent nodes a child nodes (split criterion) e stabilire una regola di arresto di crescita dell'albero (stopping rule).

Per scegliere le split criterion si utilizza generalmente una tecnica di *Recursive Binary Splitting*. Per la stopping rule, bisogna fare attenzione al tipo di albero decisionale che si è considerato. Infatti, alberi con molti nodi e split possono portare ad un sovra-adattamento dei dati (definito più propriamente dal termine *overfitting*). Ciò significa che il modello risulta di difficile interpretazione, in quanto, diventa inaccurato per previsioni successive ed ha bisogno delle stopping rule. I metodi per evitare questo problema sono impostare un numero minimo di dati di allenamento da utilizzare su ciascun nodo foglia o impostare la profondità massima del modello, che si riferisce alla lunghezza del percorso più lungo dal nodo radice al nodo foglia.

3.2.1 Training & Test Set

Il primo vero passo da fare quando si parla di classificazione è quello di creare un training set adeguato alla etichettatura che vorremmo prevedere. Nel progetto implementato, utilizzeremo il processo CART per definire e predire quali negozi continueranno ad esercitare e quali negozi chiuderanno nel 2019, avendo come training set i negozi che hanno chiuso nel 2018, aventi i dati dal 2017 al primo semestre 2019.

La tabella che caratterizzerà la nostra classificazione sarà una tabella aggregata per il primo quadrimestre di ogni anno, ma soprattutto per negozi, dove si andranno ad analizzare le vendite, il costo del venduto e il margine operativo, per quanto riguarda l'aspetto economico finanziario, e saranno considerati anche il numero di scontrini effettuati nell'anno e infine l'etichetta reale rappresentante lo stato del negozio Chiuso/Aperto nell'anno seguente, che rappresenta il nostro split.

Per creare la tabella aggregata, si è dovuto far riferimento alle tabelle dimensione create precedentemente nel modello ETL (Shop e Date) e la tabella fatto Sales, svolgendo una query dettagliata per ricavare gli attributi sopra citati.

La totalità della query è mostrata nell'appendice A5.

3.2.2 Classificazione E Previsione Delle Cause Della Chiusura Dei Negozi

Create le tabelle di riferimento, si passa alla creazione del Classification And Regression Tree, più comunemente chiamato CART. L'idea, come spiegato in precedenza, è quella di prendere come dati di partenza i negozi dal 2017 e verificarne se sono rimarranno aperti nell'intero 2019, estraendo direttamente la tabella dal database tramite i comandi esposti nell'appendice A5.

Questi dati saranno ulteriormente divisi in training and test set grazie ad una partizione randomica 80/20 sul 100% degli elementi analizzati, dove verrà creato il modello partendo dai Training set di dati per testarlo sui dati di test successivamente.

Una volta eseguita l'operazione di test, sarà verificata la sua disuguaglianza distribuzione tramite il Gini index, dove 0 rappresenta la perfetta uguaglianza, mentre un indice di 100 implica una perfetta disuguaglianza e Inoltre, sarà studiata l'accuratezza tramite la media.

> summary(CART)

Call:

```
rpart(formula = FlagOpen ~ ., data = TRAININGSet, method = "class",  
      control = rpart.control(minsplit = 5))
```

n= 78

	CP	nsplit	rel error	xerror	xstd
1	0.250	0	1.00	1.0	0.1928198
2	0.100	1	0.75	0.8	0.1783112
3	0.025	3	0.55	0.9	0.1860521
4	0.010	8	0.40	1.0	0.1928198

Variable importance

GAIN	COGS	MARGIN	nRECEIPT
28	26	25	21

view results

```
> print(CART)
```

n= 78

node), split, n, loss, yval, (yprob)

* denotes terminal node

```

1) root 78 20 OPEN (0.25641026 0.74358974)
2) MARGIN< 1235.842 5 0 CLOSE (1.00000000 0.00000000) *
3) MARGIN>=1235.842 73 15 OPEN (0.20547945 0.79452055)
6) MARGIN< 17461.73 19 7 OPEN (0.36842105 0.63157895)
12) nRECEIPT>=113 8 2 CLOSE (0.75000000 0.25000000) *
13) nRECEIPT< 113 11 1 OPEN (0.09090909 0.90909091) *
7) MARGIN>=17461.73 54 8 OPEN (0.14814815 0.85185185)
14) nRECEIPT>=459 43 8 OPEN (0.18604651 0.81395349)
28) MARGIN< 39091.27 3 1 CLOSE (0.66666667 0.33333333) *
29) MARGIN>=39091.27 40 6 OPEN (0.15000000 0.85000000)
58) COGS< 75358.5 6 2 OPEN (0.33333333 0.66666667) *
59) COGS>=75358.5 34 4 OPEN (0.11764706 0.88235294)
118) COGS>=161920.7 18 4 OPEN (0.22222222 0.77777778)
236) nRECEIPT< 2059 2 0 CLOSE (1.00000000 0.00000000) *
237) nRECEIPT>=2059 16 2 OPEN (0.12500000 0.87500000) *
119) COGS< 161920.7 16 0 OPEN (0.00000000 1.00000000) *
15) nRECEIPT< 459 11 0 OPEN (0.00000000 1.00000000) *

```

#Inserisco TESTset nell'ALGORITMO CART

```
> prevision <- predict(CART, TESTSet, type="class")
```

#accuratezza previsione

```
> Gini(prevision)
```

```
0.09428571
```

```
> mean (prevision == TESTSet$FlagOpen)
```

```
0.8571429
```

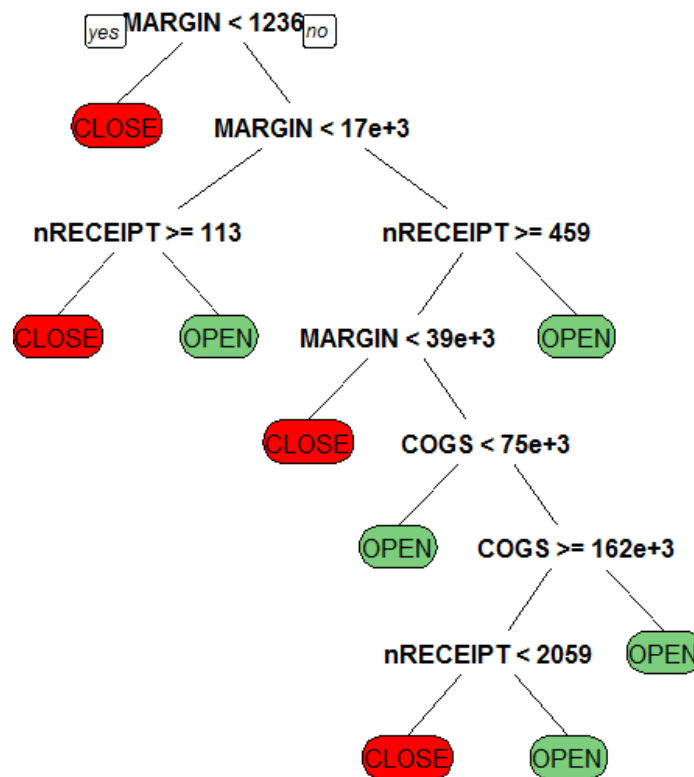


Figura 35: CART

Nell' albero ottenuto, sono evidenziati 4 fattori: Il valore di split e nelle caselle colorate, l'etichetta finale, la probabilità che con i valori di input si verifichi l'etichetta finale e, infine, la probabilità in cui si è verificata quella condizione sul totale dei dati di partenza.

Particolare attenzione è da dedicare agli alberi decisionali con molti nodi e un elevato numero di divisioni, in quanto, possono portare a un sovra-adattamento dei dati (definito più propriamente dal termine overfitting) che fa risultare il modello di difficile interpretazione dovuta alla inaccuratezza per le previsioni successive.

Un modo per evitare questo problema è impostare un numero minimo di dati di allenamento da utilizzare su ciascun nodo foglia. Ad esempio, possiamo utilizzare un livello minimo di 5 per prendere una decisione e ignorare qualsiasi nodo foglia che richieda meno di 5 livelli.

Infatti, nella linea di codice creata per la costruzione del CART sottostante si evidenzia il comando di controllo "rpart.control", che evidenzia un minimo numero di split da eseguire:

```
CART <- rpart (FlagOpen ~ . , data=TRAININGSet ,
method="class", control=rpart.control (minsplit=5))
```

Un altro modo è impostare la profondità massima del modello, che si riferisce alla lunghezza del percorso più lungo dal nodo radice al nodo foglia.

Per migliorare ulteriormente le prestazioni dell'albero è possibile utilizzare anche la tecnica Pruning. Semplicemente, si tratta di rimuovere i rami che fanno uso di caratteristiche che hanno poca importanza. In questo modo, riduciamo la complessità dell'albero e quindi aumentiamo il suo potere predittivo (fatto che riduce a sua volta il sovra-adattamento).

Il metodo di Pruning più semplice inizia dalle foglie e rimuove ogni nodo con la classe più popolare in quella foglia, fermandosi prima di ridurre la precisione.

Una volta verificato e ottimizzato al massimo il modello, sono stati introdotti i dati del database da analizzare e predire nel modello creato, ottenendo la previsione tramite il comando dedicato: *prevision2019 <- predict (CART, DATASET18, type="class")*

Successivamente, il modello è stata ulteriormente valutata tramite l'indice di Gini [0.1705464] e l'accuratezza [0.8701299], e infine, confrontato con i risultati iniziali, per capirne la vera efficienza. Il risultato ha portato ad una previsione di 4 negozi a rischio chiusura su un totale complessivo di 29.

	SHOP_ID	MARGIN	nRECEIPT	GAIN	COGS	FlagOpen	prevision2019
1	4064196	194247.68	5505	681708.61	487460.93	OPEN	OPEN
2	73638	20007.68	337	44479.96	24472.28	OPEN	OPEN
3	2237727	630648.10	7019	1225036.95	594388.85	OPEN	OPEN
4	2286770	175389.83	1656	330031.18	154641.35	OPEN	OPEN
5	3253865	31478.99	473	67379.97	35900.98	OPEN	CLOSE
6	4063154	69378.27	2699	288114.85	218736.58	OPEN	OPEN
7	73647	103459.19	3064	403503.95	300044.76	OPEN	OPEN
8	73569	56877.92	813	116076.54	59198.62	OPEN	OPEN
9	73539	32635.11	525	69497.32	36862.21	OPEN	CLOSE
10	4596364	102839.96	1222	201342.97	98503.01	OPEN	OPEN
11	3694385	292708.76	2945	564322.08	271613.32	OPEN	OPEN
12	3145981	40576.74	1484	172534.32	131957.58	OPEN	OPEN
13	1489952	206888.62	2081	399424.69	192536.07	OPEN	CLOSE
14	2265838	485280.72	8978	1033463.06	548182.34	OPEN	OPEN
15	73830	25486.93	405	56269.82	30782.89	OPEN	OPEN
16	73801	39345.60	568	80020.96	40675.36	OPEN	OPEN
17	73716	32220.88	475	68861.94	36641.06	OPEN	CLOSE
18	2264278	46636.65	701	97573.92	50937.27	OPEN	OPEN
19	2726114	26713.93	402	54045.17	27331.24	OPEN	OPEN
20	73773	12420.89	211	28334.43	15913.54	OPEN	OPEN
21	4064198	336480.19	7787	1196995.75	860515.56	OPEN	OPEN
22	175674	77269.68	1035	162405.14	85135.46	OPEN	OPEN
23	73740	36069.38	529	75713.85	39644.47	OPEN	OPEN
24	73825	72161.05	946	150774.37	78613.32	OPEN	OPEN
25	2022028	4178.04	88	10211.89	6033.85	OPEN	OPEN
26	2167235	445591.73	4230	846926.01	401334.28	OPEN	OPEN
27	2203462	188159.04	5098	659833.14	471674.10	OPEN	OPEN
28	876293	63376.15	891	134117.87	70741.72	OPEN	OPEN
29	2312294	141998.28	4191	549655.07	407656.79	OPEN	OPEN

Figura 36: Results of Prevision

CAPITOLO 4: DATA

VISUALIZATION

La data visualizzazione è un termine generico che descrive qualsiasi tentativo di aiutare le persone a comprendere il significato dei dati analizzati posizionandoli in un contesto visivo. Modelli, tendenze e correlazioni che potrebbero non essere rilevati nei dati basati su testo possono essere esposti e riconosciuti più facilmente tramite report utilizzando dei software di visualizzazione dei dati come per esempio Microsoft Power BI.

I sistemi di reportistica vengono sviluppati in ambiti complessi che hanno previsto una soluzione di data warehouse. Una delle finalità di un processo di DW è proprio quella di strutturare un contesto informativo hardware-software capace di rispondere alle esigenze dello scenario organizzativo.

Col crescere dei dati accumulati a disposizione delle organizzazioni, i vantaggi di un'elaborazione centralizzata dei documenti si rivelano nei tempi di esecuzione dei singoli report: la particolare configurazione hardware delle postazioni su cui vengono ospitate le risorse a livello fisico del sistema permette l'ottimizzazione delle richieste al sistema e ne diminuisce il carico di attività rispetto alla situazione in cui singoli utenti ricercano informazioni sul sistema individualmente.

Il documento prodotto viene chiamato *report* e si presenta come una combinazione di tabelle e grafici che presentano le misure di rilievo per i vari fenomeni analizzati, disaggregate e destrutturate secondo le esigenze. Tali misure costituiscono una base comune per le analisi successive. Una volta elaborato e generato, viene validato dalle strutture preposte e viene distribuito (ed aggiornato con cadenza periodica) ai clienti che ne sfrutteranno le potenzialità.

Un processo di implementazione di un sistema di reportistica è genericamente composto dalle seguenti fasi, che possono essere ampliate o ridotte in conseguenza dei particolari

ambienti di sviluppo e dei differenti contesti macroeconomici di attività dell'organizzazione:

- Identificazione delle esigenze informative e di visualizzazione;
- Identificazione del contesto informativo e delle fonti;
- Identificazione della configurazione del sistema hardware/software;
- Fase di integrazione hardware/software delle risorse informative;
- Preparazione del report;
- Validazione del report;
- Fase di collaudo del sistema;
- Fase di esercizio del sistema di reportistica.

Queste fasi non sono da intendersi necessariamente come consecutive in quanto alcune possono anche svolgersi in concomitanza.

4.1 MICROSOFT POWER BI

BI è una suite di Business Intelligence completa e integrata che aiuta a ridurre la complessità dell'interazione e organizzazione delle informazioni e ad ottenere vantaggi competitivi per l'azienda attraverso decisioni migliori e più chiare.

Microsoft fornisce una serie di strumenti di data warehouse e analisi dei dati per la creazione di report per consentire agli utenti di accedere, comprendere, analizzare, collaborare e agire sulle informazioni quando vogliono e ovunque si trovino. Essi sono utilizzati per, in primo luogo, fornire dati di qualità. Il secondo obiettivo è ottenere una visione più approfondita e migliorare il processo decisionale e infine, consentire alle organizzazioni di adottare decisioni agili per raggiungere gli obiettivi e la strategia aziendale.

Nello svolgimento della tesi, utilizzerò Microsoft Power BI, una suite di strumenti di analisi aziendale per analizzare dati e condividere informazioni [31].

Le dashboard di Power BI forniscono una vista a 360 gradi per gli utenti aziendali con le metriche più importanti in un unico posto, aggiornate in tempo reale e disponibili su tutti i

loro dispositivi. Con un clic, gli utenti possono esplorare i dati dietro il loro cruscotto utilizzando strumenti intuitivi che facilitano la ricerca di risposte. La creazione di un dashboard risulta molto semplice, grazie alle centinaia di connessioni con le più diffuse applicazioni aziendali e ai template precostruiti per aiutarti a metterti subito in funzione. Inoltre, puoi accedere ai tuoi dati e rapporti ovunque per mezzo dell'app di Power BI Mobile, che si aggiornano automaticamente dopo qualsiasi modifica ai dati.

Per facilitare l'utilizzo della applicazione si sono create delle variabili grazie all'uso del DAX, acronimo di Data Analysis Expressions e indica il linguaggio delle formule usato in Power BI, anche in background.

Tutte le formule create per la data visualization in Power BI del progetto sono reperibili all'appendice A7.

4.2 STRUMENTI UTILIZZATI E RISULTATI OTTENUTI TRAMITE LA DATA VISUALIZATION

Power BI può essere semplicemente definito come un sistema di visualizzazione suddiviso in blocchi che possono espandere il loro campo di definizione dinamicamente, passando da una analisi generale ad una dettagliata con un semplice click, con l'obiettivo di creare report elaborati e complessi adatti alle esigenze del cliente. I blocchi che ne caratterizzano l'uso sono principalmente 4:

- Visualizzazioni;
- Report;
- Dashboard;
- Dataset.

4.2.1 Visualizzazioni

Quando si crea o si modifica un report di Power BI, è possibile usare vari tipi di oggetti visivi. Le icone di questi oggetti visivi vengono visualizzate nel riquadro Visualizzazioni.

Gli sviluppatori creano gli oggetti visivi personalizzati tramite l'SDK relativo. Questi oggetti visivi consentono agli utenti aziendali di visualizzare i dati nel modo che meglio si adatta alle loro attività. Gli utenti possono importare i file degli oggetti visivi personalizzati nei report e usarli come qualsiasi altro oggetto visivo di Power BI, assumendo una posizione di primaria importanza, e potendo inoltre essere filtrati, evidenziati, modificati, condivisi a piacere.

Gli oggetti visivi personalizzati vengono distribuiti in tre modi:

- File di oggetti visivi personalizzati;
- Oggetti visivi organizzazione;
- Oggetti visivi del Marketplace.

In alcune organizzazioni, gli oggetti visivi personalizzati sono ancora più importanti, in quanto, potrebbero essere necessari per comunicare dati o informazioni approfondite e specifiche dell'organizzazione o semplicemente, avere requisiti speciali per i dati da mettere in evidenza ai privati. Perciò, queste organizzazioni hanno la necessità di sviluppare oggetti visivi personalizzati, condividerli in tutto il cloud e accertarsi che vengono gestiti correttamente.

In sintesi, il termine visualizzazione può essere definito come una rappresentazione visiva dei dati. Questa rappresentazione può essere sotto forma di grafico, mappa o qualsiasi altro strumento utile per rappresentare i tuoi dati.

Power BI ha un buono numero di visualizzazioni che possono aiutare a rappresentare i dati in vari modi creativi e ad effetto.

L'immagine sottostante mostra alcune delle visualizzazioni presenti in Power BI, fornendo una overview generica del primo quadrimestre 2019:

- La visualizzazione di sinistra (Scheda Multiriga), è stata creata per mettere in relazione una misura e un attributo della tabella prodotto. Come si può osservare, infatti, per ogni

paese di origine manifatturiera è stato associato un numero equivalente al totale delle unità vendute nel primo periodo dell'anno.

- Al centro, troviamo un veloce ma efficace valore corrispondente del Fatturato, dello sconto, del costo del venduto e del margine fino ad oggi nel 2019. Ciascuno di questi valori è accompagnato da un grafico ad area dettagliato settimanalmente che mostra il confronto con il primo quadrimestre dell'anno precedente. Questo grafico è molto utile per capire dove si è guadagnato e dove si è perso denaro, e confrontandoli con altri dati azienda, è possibile anche arrivare alla causa del fenomeno mutativo.
- A destra, infine, troviamo due istogrammi, uno verticale e uno orizzontale, che mostrano il fatturato totale filtrato per il periodo in analisi rispettivamente per il tipo di negozio e per il flag weekend, ottenuto mediante creazione di una nuova funzione di Power BI con linguaggio DAX. L'ultima visualizzazione rappresenta la percentuale del flusso di clienti nei punti di vendita in base giornaliera.

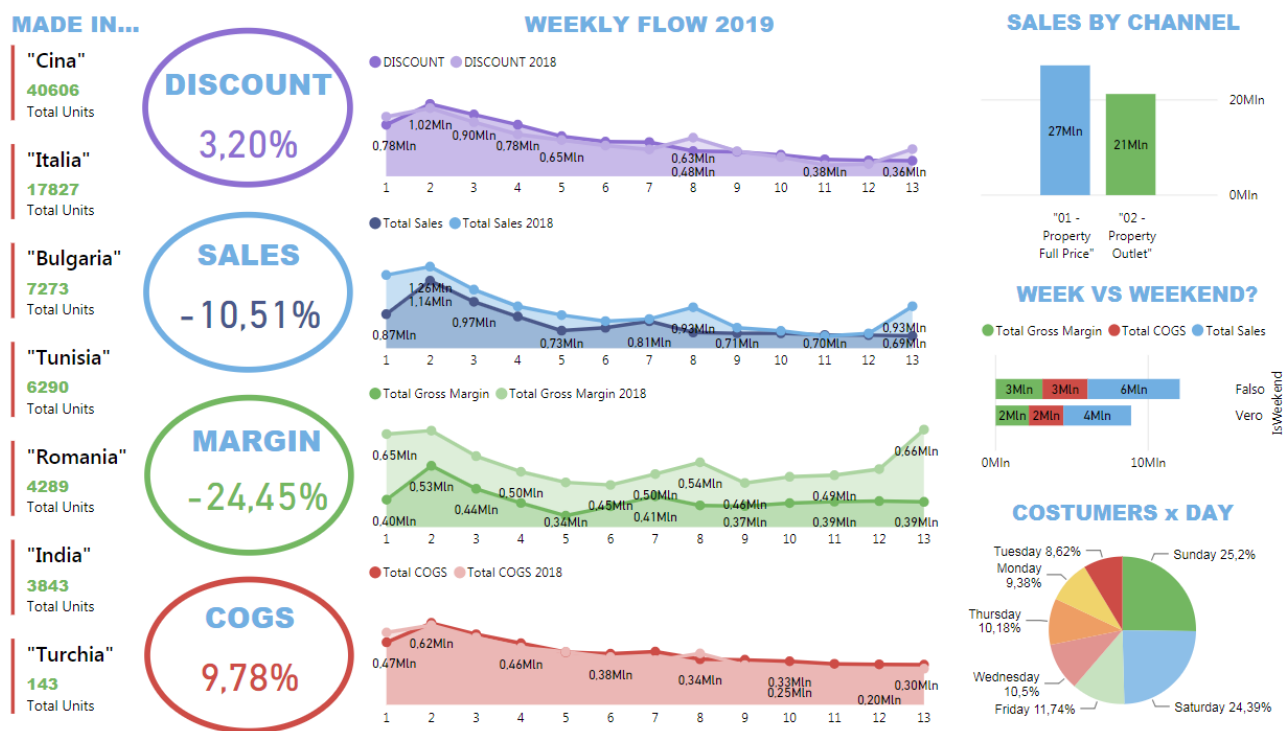


Figura 37: Visualizzazione 1Q 2019

Da questo tipo di visualizzazioni, i risultati sono ben chiari e difficile da confondere. Possiamo dire, per esempio che in questo quadrimestre l'andamento generale della azienda è sotto le aspettative, che la maggior parte dei prodotti venduti è made in Cina, che il canale di vendita

con maggiore incasso è il Property Full Price e, per concludere, che il migliore afflusso di gente si verifica nel weekend.

Tramite una piccola dimostrazione, è stato già possibile intuire l'enorme potenziale che può portare la data visualization, e in generale alla Business Analytics, alle decisioni manageriali di una azienda, portando tutto ciò ad un miglioramento nel corso degli anni della distribuzione della merce e del fatturato.

4.2.4 Report

Un report è una raccolta di visualizzazioni visualizzate insieme su una o più pagine. I report ti aiutano a organizzare le tue visualizzazioni in un modo che racconta la storia dei tuoi dati, nel modo che preferisci. Ad esempio, se si desidera mostrare le vendite dei della tua azienda all'interno dei diversi punti vendita del tuo paese, puoi avere un rapporto composto da un numero di grafici (torta, linea o barra), mappe e grafici che visualizzano le informazioni che voglio passare. Nell' esempio seguente sono mostrati i risultati ottenuti grazie ad una analisi dei negozi chiusi in Italia dal 2017 ad oggi dell'azienda cliente del progetto analizzato, identificando la causa della chiusura di essi, classificata come una chiusura per un cattivo margine di profitto, o una chiusura per un mercato coperto negli anni o la nascita di un nuovo negozio nei dintorni. Come dati di partenza, sono stati utilizzati i dati anagrafici implementati nella fase di costruzione della datawarehouse, spiegati precedentemente nel capitolo 2, e i dati ottenuti dalla classificazione CART illustrata nel capitolo 3.

Il risultato è visibile attraverso la mappa sottostante che racchiude due esempi per ciascun tipo di chiusura. I negozi di Serravalle e Milano mostrano una chiusura di tipo new shop, mentre il negozi di padova e modena di un margine non adeguato.

La grandezza dei pallini in figura è proporzionata al fatturato complessivo per ciascun punto vendita.

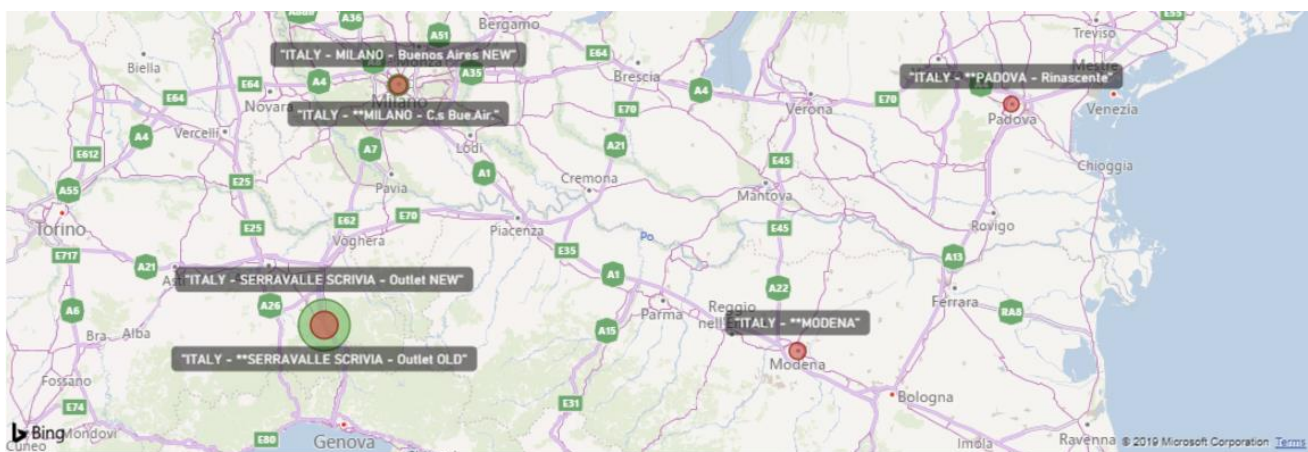


Figura 38: Mappa Della Classificazione Dei Negozi

In relazione con la mappa è stata creata una tabella esplicativa dove viene evidenziato il flag di chiusura e la descrizione dei negozi.

CLOSE FROM 2017				CLOSE FROM 2018			
_NEGOZIO_DESC_	TotalSales2017	Total Gross Margin LY	CloseREASON2017	_NEGOZIO_DESC_	TotalSales2018	Total Gross Margin TY	CloseREASON2018
"ITALY - **MILANO - C.s Bue.Air."	721.213,20	315.535,58	COVERED/NEW SHOP	"ITALY - **ROMA - Via Nazionale"	456.627,20	287.662,39	COVERED/NEW SHOP
"ITALY - **ROMA - Outlet"	1.365.206,15	637.587,15	COVERED/NEW SHOP	"ITALY - **CATANIA - C.so Italia"	25.093,47	7.790,47	BAD MARGIN
"ITALY - **SERRAVALLE SCRIVIA - Outlet OLD"	3.628.541,00	1.751.273,21	COVERED/NEW SHOP	"ITALY - **MILANO - Coin"	146.627,97	95.369,69	BAD MARGIN
"ITALY - **ASCOLI PICENO - C.C. Battente"	895,00	532,00	BAD MARGIN	"ITALY - **MODENA"	205.376,50	117.997,02	BAD MARGIN
"ITALY - **BARI"	89.788,52	15.294,87	BAD MARGIN	"ITALY - **REGGIO CALABRIA"	8.422,30	2.609,96	BAD MARGIN
"ITALY - **CHIETI - C.C. Megal?"	39.346,65	26.852,65	BAD MARGIN	"ITALY - **ROMA - P.zza Balduina"	252.484,64	85.815,82	BAD MARGIN
"ITALY - **COMO - La Borsetta"	347.498,50	127.343,25	BAD MARGIN	"ITALY - **CERVIA"	157.051,15	71.301,60	BAD MARGIN
"ITALY - **ERCOLANO - Donadio"	540,00	324,00	BAD MARGIN	"ITALY - **PALERMO - Via Libert? NEW"	238.727,45	158.288,42	BAD MARGIN
"ITALY - **FORLI - Virgili"	14.244,00	5.626,00	BAD MARGIN	"ITALY - **RAVENNA"	240.810,85	115.031,40	BAD MARGIN
"ITALY - **L'AQUILA "	19.755,50	10.071,50	BAD MARGIN	"ITALY - **ROMA - Coin"	53.853,69	37.427,34	BAD MARGIN
"ITALY - **NAPOLI - Vomero"	67.064,00	16.168,50	BAD MARGIN				
"ITALY - **PADOVA - Rinascente"	102.357,50	46.144,50	BAD MARGIN				
"ITALY - **PESARO"	21.213,00	12.325,00	BAD MARGIN				
"ITALY - **RIMINI - Virgili"	21.683,40	10.548,40	BAD MARGIN				
"ITALY - **TERAMO - c.c. Val Vibrata Colonnella"	3.641,40	1.735,40	BAD MARGIN				
"ITALY - **TERMOLI"	9.864,10	4.061,10	BAD MARGIN				

Figura 39: Tabella Della Classificazione Dei Negozi

4.2.3 Dashboard

Una dashboard è una raccolta di visualizzazioni su una singola pagina, che è possibile condividere con altri. Benché visivamente simile a un report, una dashboard deve adattarsi a una singola pagina e può essere condivisa con altri utenti che saranno in grado di interagire con i dati presentati in essa. Creando e condividendo una dashboard per il responsabile delle vendite, ad esempio, lui o lei dovrebbe essere in grado di interagire con essa e visualizzare nuove informazioni diverse da quella che è chiaramente visibile sulla dashboard di partenza, in base ai dati presenti.

Le immagini seguenti mostrano un esempio in Power BI di cosa si intende per una dashboard. La sottostante, comprende una analisi generale dei canali di vendita con associati i relativi guadagni, dei prodotti, e del made in. Sarà la visualizzazione di default per il cliente.

Il primo grafico (BarChart) comprende la visualizzazione del fatturato totale del 2019 riferente al canale del negozio, suddiviso in Property Full Price e Property Otlet. Il secondo grafico è la rappresentazione 80/20 di Pareto, dove è evidente che gran parte delle vendite deriva dalle borse. L'ultimo grafico, invece, mostra la percentuale sul totale della provenienza e fattura dei prodotti.

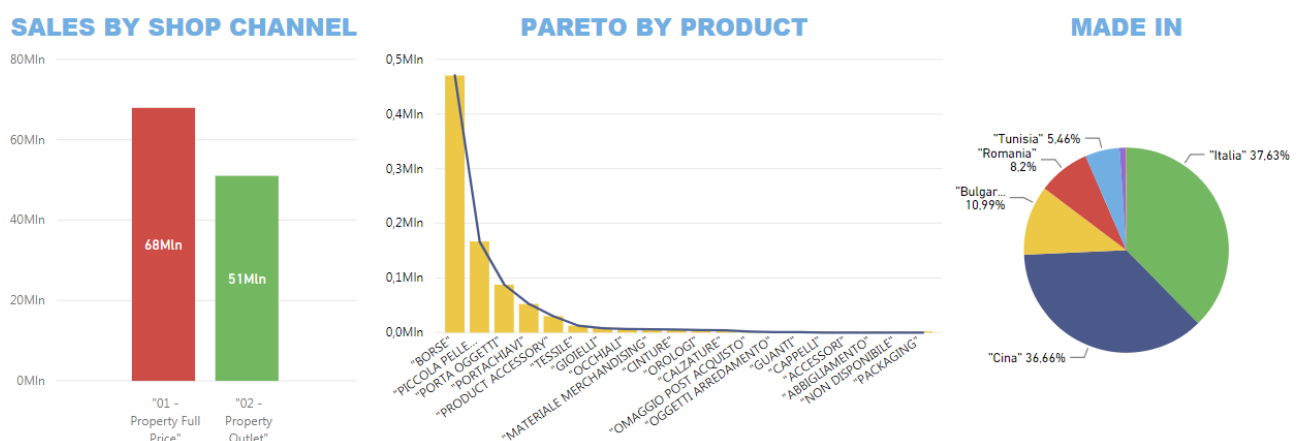


Figura 40: Default Dashboard View

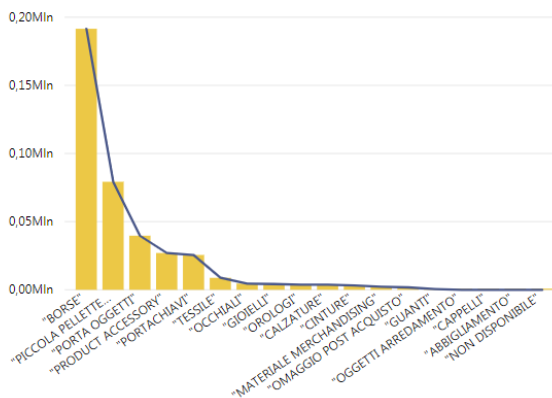
Tramite l'utilizzo di filtri o semplice navigazione è possibile passare da una analisi molto generale ad una analisi molto più dettagliata in ogni singolo particolare. Di seguito, saranno spiegate brevemente tre diverse viste, ognuna derivante dalla dashboard di partenza.

L'immagine iniziale e la seguente spiegano le variazioni di prodotti venduti e made in grazie ad una selezione del tipo di negozio che si vuole prendere in considerazione. L'ultima usa la stessa idea ma la selezione avviene per il prodotto borse. È molto importante osservare come i grafici interagiscono tra loro portando ad una analisi rapida ed efficace, scelta in base all'esigenza del cliente.

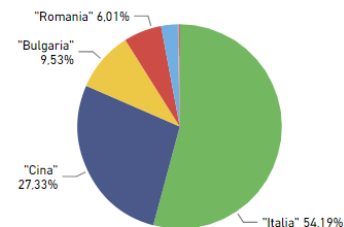
SALES BY SHOP CHANNEL



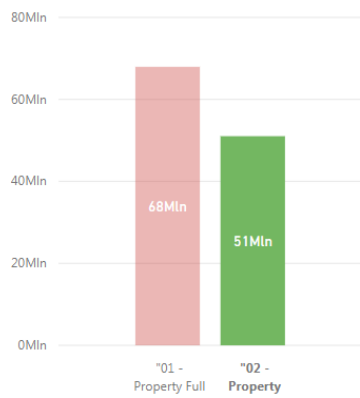
PARETO BY PRODUCT



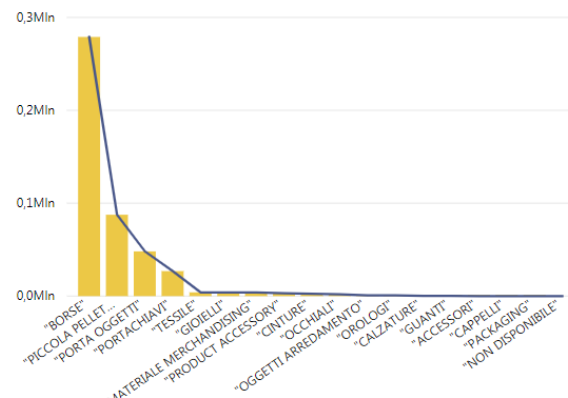
MADE IN



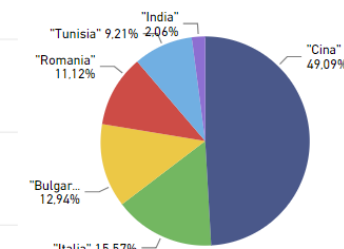
SALES BY SHOP CHANNEL



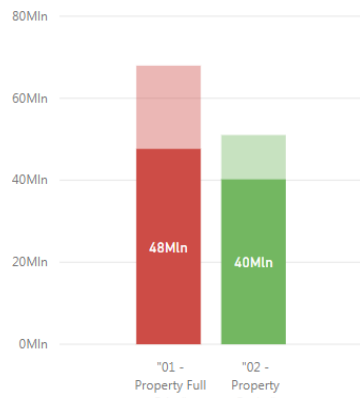
PARETO BY PRODUCT



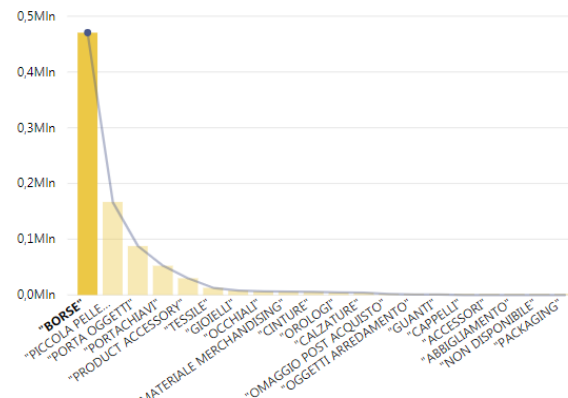
MADE IN



SALES BY SHOP CHANNEL



PARETO BY PRODUCT



MADE IN

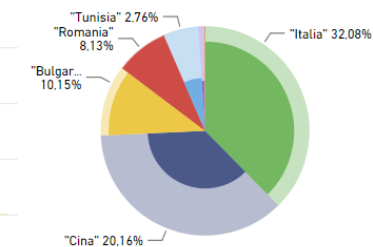


Figura 41: Viste In Profondità Di Una Dashboard

4.2.4 Dataset

Un set di dati, comunemente chiamato Dataset, è una raccolta di dati che viene utilizzata da Power BI per creare delle visualizzazioni. In poche parole, sono i dati dietro un grafico, o una mappa nel tuo rapporto. Ad esempio, se hai un grafico che visualizza il venduto in ciascun mese dell'anno, i dati utilizzati per produrre quel grafico sono noti come set di dati. È importante notificare, che non necessariamente i dataset devono provenire da una singola fonte. Quasi sempre, è una raccolta filtrata di dati combinati tra più fonti diverse con l'obiettivo di produrre una collezione unica degli stessi dati di origine che può essere utilizzata in Power BI per mostrare una peculiarità che può essere utili al fine delle decisioni aziendali di una azienda. Con l'impressionante numero di connettori inclusi in Power BI, è possibile estrarre dati da qualsiasi fonte: Excel, Social Network, Oracle, SQL Server ecc.

L'immagine seguente mostra un set di dati di esempio in Power BI, dove vengono considerati tutti valori economici e finanziari del cliente, riguardante le vendite, il costo de venduto e il margine lordo del 2017 e del 2018 raggruppati per ogni mese.

Una ulteriore, e conseguente, visualizzazione della variazione in termini percentuali del valore 2018 con quello dell'anno passato è applicata ai dati di partenza. Il colore verde e il colore rosso indicano una formattazione condizionale con valori rispettivamente di guadagno o perdite. In fondo è indicato anche il totale dei due anni, con relativa variazione per ogni fattore. Questo Dataset illustra complessivamente i dati utili per calcolare un conto economico parziale dell'azienda.

Month	TotalSalesTY	TotalSalesLY	Total Sales Var %	Total COGS TY	Total COGS LY	COGS Var %	Total Gross Margin TY	Total Gross Margin LY	Gross Margin Var %
1	6.345.706,06	6.381.205,06	-0,56%	3.116.733,50	3.593.597,10	-13,27%	3.228.972,56	2.787.607,96	15,83%
2	4.317.459,62	4.929.981,33	-2,42%	1.999.986,59	2.616.713,96	-23,57%	2.317.473,03	2.313.267,37	0,18%
3	4.120.016,30	4.529.293,17	-9,04%	1.305.991,03	1.978.735,79	-34,00%	2.814.025,27	2.550.557,38	10,38%
4	4.753.446,01	5.526.732,97	-13,99%	1.452.637,74	2.327.580,91	-37,59%	3.300.808,27	3.199.152,06	3,18%
5	4.931.287,17	5.469.303,02	-9,84%	1.343.074,00	2.157.069,57	-37,74%	3.588.213,17	3.312.233,45	8,33%
6	4.837.644,26	5.449.215,32	-11,22%	1.669.463,31	2.427.601,25	-31,23%	3.168.180,95	3.021.614,07	4,85%
7	7.058.923,82	8.484.647,39	-16,80%	3.935.805,15	4.512.366,04	-12,78%	3.123.118,67	3.972.281,35	-21,38%
8	5.117.565,21	5.808.951,24	-11,90%	2.814.755,43	2.677.724,47	5,12%	2.302.809,78	3.131.226,77	-26,46%
9	4.684.542,74	5.133.199,22	-8,74%	2.166.303,97	1.510.705,96	43,40%	2.518.238,77	3.622.493,26	-30,48%
10	4.265.956,80	5.021.728,09	-15,05%	1.957.960,64	1.866.856,54	4,88%	2.307.996,16	3.154.871,55	-26,84%
11	5.300.641,81	4.813.581,63	10,12%	3.415.161,00	2.466.275,59	38,47%	1.885.480,81	2.347.306,04	-19,67%
12	5.453.336,76	5.958.834,88	-8,48%	2.871.214,81	2.533.819,78	13,32%	2.582.121,95	3.425.015,10	-24,61%
Totale	61.186.526,56	67.506.673,32	-9,36%	28.049.087,17	30.669.046,96	-8,54%	33.137.439,39	36.837.626,36	-10,04%

Figura 42: Dataset Relativo Al Conto Economico Aziendale

CONCLUSIONI

RESULTS

Lo scopo del progetto era quello di implementare una data mart per avere un dato finale pulito e di qualità dedicata alle vendite di una azienda di fashion, su cui sono state svolte delle analisi di business significative per future decisioni aziendali tramite algoritmi di machine learning.

Nello specifico:

- 1) Creazione di una data mart ETL best practice per avere le migliori prestazioni possibili nell'esecuzione delle interrogazioni e della data ingestion, e una ideale per la data visualization con dati finali puliti e di qualità per svolgere delle analisi di business significative;
- 2) Utilizzo di software di Data Mining e Machine Learning che per mezzo dell'intelligenza artificiale, implementano autonomamente diversi algoritmi in grado di eseguire un processo sempre più ottimizzato unendo i dati del datawarehouse creato e gli open data, studiando le cause delle chiusure dei negozi e prevedendo la posizione ideale per aprirne di nuovi;
- 3) Creazione di una Dashboard in Power BI per avere un controllo e una serie di dati effettivi su tutto quello che riguarda le vendite.

Il primo obiettivo è stato sicuramente raggiunto in quanto il data mart è attualmente funzionante. L'obiettivo prestazionale è invece stato raggiunto solo in parte. Certamente, rispetto alle analisi multidimensionali condotte sui dati operazionali, che venivano in precedenza svolte dall'azienda, si è di fronte ad un sostanziale miglioramento. Il processo è ottimale, ma i software utilizzati, essendo open source, e il fatto di lavorare in un server locale, hanno rallentato di molto il livello di prestazione finale del processo ETL.

Gli algoritmi di machine learning e predizione hanno raggiunto un ottimo livello di analisi. Tramite l'albero decisionale CART del paragrafo 3.2, si è raggiunta una dimostrazione delle cause principali delle chiusure dei negozi del cliente, ottenendo come valore ottimale di Gini index 0.07807417, rappresentante la disuguaglianza della distribuzione dei dati, e una

accuratezza dal valore ottimale di 0.8275862, tramite lo studio della media. Entrambi sono valori accettabili, in quanto, il Gini index e l'accuratezza ideali sono rispettivamente uguale a 0 e a 1. Il risultato ottenuto ha identificato 4 negozi su 29 a rischio chiusura entro la fine del 2019.

Invece, per quanto riguarda la previsione dei dati ISTAT 2018, come visto in precedenza nel paragrafo 3.1.1, i valori di bontà dell'analisi sono ottimi. Grazie all'uso delle query e la previsione degli open data, si è potuto concludere che le migliore location per aprire un nuovo negozio, sono Andalo, Corvara In Badia e Gressoney.

In generale, possiamo concludere che il progetto è riuscito in maniera soddisfacente, ottenendo risultati efficaci per delle future strategie aziendali, tutti visualizzabili in modo smart ed intuitivo tramite l'utilizzo delle dashboard create in Power BI.

SVILUPPI FUTURI: REAL-TIME BUSINESS INTELLIGENCE

Mentre l'analisi in tempo reale e i big data sono entrambi di tendenza, l'analisi dei big data in tempo reale, che è la loro combinazione, è il futuro.

Il real-time viene spesso confuso con l'istantanea. Infatti, il motore di elaborazione in tempo reale non è sempre in grado di importare i dati di streaming ma può essere progettato per estrarre nuovi dati appena inseriti nei file sorgenti. Il tempo tra queste query dipende molto dalle esigenze aziendali e può variare da millisecondi a ore. Per esempio, il sistema analitico di una banca consentirebbe diversi minuti per valutare l'affidabilità creditizia di un richiedente e il prezzo dinamico di un rivenditore può richiedere fino a un'ora per l'aggiornamento. Tuttavia, tutti questi esempi sono considerati in tempo reale.

A differenza dei modelli tradizionali che esaminano i dati storici per i modelli, l'analisi in tempo reale si concentra sulla comprensione delle informazioni non appena vengono create per aiutare a prendere decisioni più rapide e migliori [33].

La business intelligence in tempo reale è l'utilizzo di strumenti di analisi e altri strumenti di elaborazione dati per consentire alle aziende di accedere ai dati e alle visualizzazioni più recenti e pertinenti. Per fornire con successo i dati migliori, gli strumenti di BI in tempo reale

utilizzano una combinazione di analisi server-less (in cui i dati vengono trasmessi direttamente a una dashboard o visualizzazione) e di una data warehouse, consentendo alle dashboard di mostrare i dati storici e in tempo reale in modo complementare.

Per le organizzazioni che producono gigabyte o terabyte di dati, molte di queste informazioni perdono la loro rilevanza una volta che sono immagazzinate. Le informazioni sui livelli di inventario, le esigenze dei clienti, i servizi in corso e altro ancora, possono essere incredibilmente utili, ma ancor più se analizzate non appena vengono generate.

L'analisi e la BI in tempo reale consentono inoltre agli utenti delle organizzazioni di eseguire delle ricerche personalizzate e utilizzare i dati disponibili, includendo la possibilità di eseguire analisi ad-hoc sui dati esistenti o creare visualizzazioni specifiche per i nuovi flussi. Infine, la BI in tempo reale aiuta a comprendere meglio le tendenze e a creare modelli predittivi più accurati per le organizzazioni.

Esistono diversi settori in cui l'utilizzo della BI può ottimizzare un'organizzazione:

- *Costumer Relathionship Management*: Le suite di gestione delle relazioni con i clienti (CRM) possono utilizzare i dati in tempo reale per fornire un servizio migliore ai consumatori. Ciò include un migliore coinvolgimento dei servizi e delle conversazioni alle preferenze conosciute dei consumatori al momento. Un esempio significativo è l'azienda Disney che, dopo anni di test a Disney World, ha lanciato il suo innovativo programma MyMagicPlus. Ora, ogni ospite riceve il proprio braccialetto MagicBand, che funge da chiave di identificazione, carta di credito e pass. I clienti passano semplicemente la fascia sui sensori situati intorno al parco per ottenere l'ingresso alle attrazioni o per pagare i souvenir, per esempio, dando a Disney una grande quantità di dati su dove sono i suoi ospiti, cosa stanno facendo e di cosa potrebbero aver bisogno. Questi dati consentono alla Disney di anticipare ogni esigenza degli ospiti e offrire un'esperienza incredibile e personalizzata o comprendere dove aggiungere altro personale nelle aree congestionate, aumentando l'efficienza del parco [34].
- *Location Analytics*: I dati geografici e di localizzazione spesso nascondono molte informazioni utili e l'estrazione di queste può aiutare un'impresa a ottimizzare i propri processi aziendali e ad aumentare i profitti con una migliore gestione delle risorse. Ad esempio, i sensori, come i sistemi di localizzazione GPS collegati ai veicoli di una flotta, emettono periodicamente dati sulla sua posizione. L'analisi della posizione può

trasformare queste informazioni di localizzazione periodiche per rilevare la congestione, prevedere ritardi, rilevare veicoli inattivi, suggerire percorsi alternativi, violare le regole e le linee guida del trasporto (ad esempio i controlli della velocità), identificando le rotte più redditizie.

- Service transformation: Le aziende potendo raccogliere i dati in tempo reale dalle macchine e dalle catene di produzione e vedere come si comportano, migliorando sia l'efficienza che la produttività e risolvendo eventuali problemi di manutenzione prima che diventino un'emergenza completa. Un caso di service transformation abilitata dalle tecnologie è quello di Rolls Royce (motori aeronautici e marittimi) che ha potenziato sempre di più l'utilizzo di prodotti connessi, big data e analytics in modo sistematico e avanzato in tre aree di attività: Design del prodotto, manufacturing e processi di post-vendita. Da oltre un decennio, l'azienda ha rivoluzionato il proprio business model, sviluppando piani di manutenzione e assistenza innovativi in grado di collegarsi direttamente alle performance dei propri prodotti. L'utilizzo sempre più massiccio di queste tecnologie ha portato Rolls Royce al lancio degli Intelligent Insights, cioè l'insieme di dati raccolti dal funzionamento dei motori degli aerei dei clienti. Essi, vengono trasferiti sul cloud e analizzati in modo automatico tramite l'algoritmo di data mining Natural Processing Language con l'obiettivo di realizzare in modo automatico connessioni e inferenze tra dati catturati da fonti diverse, consentendo azioni preventive e predittive [35].

Oggi, La business intelligence in tempo reale sta diventando un aspetto sempre più centrale del processo decisionale delle organizzazioni e, implementando la giusta soluzione, comprendendo e raccogliendo i dati corretti, creando una infrastruttura solida e autorizzando i tuoi team a usarla, crea un vero e proprio vantaggio competitivo rispetto ai competitors.

Con lo sviluppo di nuove tecnologie si avrà una conseguente crescita di tutto il mondo Big Data, arrivando al punto di diventare indispensabili per una qualsiasi azienda.

APPENDICE

A1. SQL - CREAZIONE DELLE SURROGATE KEY

```
ALTER TABLE L1.CANALE ADD _CANALE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.ALTEZZA_TACCO ADD _ALTEZZA_TACCO_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.CATEGORIA ADD _CATEGORIA_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.CAUSALE ADD _CAUSALE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.COLORE ADD _COLORE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.COMUNI_ISTAT ADD _COMUNE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.LOCATION ADD _COMUNE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.GENDER ADD _GENDER_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.FAM_COLORE ADD _FAMIGLIA_COLORE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.FAM_MATERIALE ADD _FAMIGLIA_MATERIALE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.DATE ADD _DATE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.ISTAT ADD _ISTAT_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.MADE_IN ADD _MADE_IN_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.MADE_IN_PIANIFICAZIONE ADD _MADE_IN_PIANIFICAZIONE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.MATERIALE ADD _MATERIALE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.MODELLLO ADD _MODELLO_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.PROVINCIA ADD _PROVINCIA_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.REGIONE ADD _REGIONE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.STAGIONE ADD _STAGIONE_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.TAGLIA ADD _TAGLIA_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.TURISMO ADD _TURISMO_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.SHOP ADD _NEGOZIO_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.SHOP_OPEN_ITA ADD _NEGOZIO_OPEN_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.SHOP_NOT_OPEN ADD _NEGOZIO_NOT_OPEN_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.SHOP_TOT ADD _NEGOZIO_TOT_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.PRODUCT_DATA_MASKING ADD _PRODOTTO_DM_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.PRODUCT ADD _PRODOTTO_DM_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.DAY ADD _DAY_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.YEAR ADD _YEAR_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.MONTH ADD _MONTH_SK INT IDENTITY(1,1) NOT NULL;
ALTER TABLE L1.QUARTER ADD _QUARTER_SK INT IDENTITY(1,1) NOT NULL;
```

A2. MATRICE DI SELEZIONE DATI ISTAT

Tabella 9: Matrice Dati ISTAT

TABELLA DATI ISTAT	INDICATORI / TITOLI	AREA					ANNI	
		Italia	zone	regione	provinci	comune	Dati fino al 2017	2018
Famiglie	Incidenza della povertà assoluta							
	Reddito familiare netto medio (esclusi i fitti imputati)							
	Spesa media mensile familiare per beni e servizi non alimentari							
	Spesa media mensile familiare totale							
Trasporti	Autobus circolanti							
	Autovetture							
	Rete autostradale							
	Rete ferroviaria in esercizio							
	Trasporto di merci su strada							
Lavoro	Tasso di disoccupazione							
	Tasso di disoccupazione giovanile							
Macro- Economia	Consumi finali interni							
	Investimenti fissi lordi							
	Pil pro capite							
Territorio	Densità della popolazione dei comuni							
	Densità della popolazione dei comuni con superficie da 1.001 a 2.000							
	Densità della popolazione dei comuni con superficie da 2.001 a 6.000							
	Densità della popolazione dei comuni con superficie da 6.001 a 25.000							
	Densità della popolazione dei comuni con superficie fino a 1.000							
	Densità popolazione comuni con superficie superiore ai 25.000							
	Permessi di costruire - abitazioni in nuovi fabbricati residenziali							
	Permessi di costruire - m2 utili abitabili in nuovi fabbricati residenziali							
	Popolazione residente media							
Turismo	Totale arrivi							

A3. R-CODE: PREDIZIONE DATI ISTAT

Importo le Library

```
library(readxl)
```

Importo file Excel

```
Istat_Famiglie <- read_excel("C:/Users/Admin/ ISTAT_FAMIGLIE.xlsx")
```

Creo vettore con gli anni

```
x <- c(Istat_Famiglie$Anno_2007, Istat_Famiglie$Anno_2008, Istat_Famiglie$Anno_2009,  
Istat_Famiglie$Anno_2010, Istat_Famiglie$Anno_2011, Istat_Famiglie$Anno_2012,  
Istat_Famiglie$Anno_2013, Istat_Famiglie$Anno_2014, Istat_Famiglie$Anno_2015,  
Istat_Famiglie$Anno_2016, Istat_Famiglie$Anno_2017)
```

Creo matrice con le colonne degli anni e numero le righe

```
m1 <- matrix(x, ncol=11)
```

Nomino righe con le regioni (territorio)

```
y <- ISTAT_FAMIGLIE$Territorio
```

```
dimnames(m1) <- list(c(ISTAT_FAMIGLIE$Territorio), NULL)
```

Nomino colonne con anni dal 207 al 2017

```
t <- array(2007:2017)
```

```
dimnames(m1)[[2]] <- c(t)
```

Stampo la matrice creata

```
m1
```

Regressione lineare (lm) dei dati dal 2007 al 2017 relativi alla 'Spesa media mensile familiare per beni e servizi non alimentari'

```
reg <- lm(Istat_Famiglie$Anno_2017~ Istat_Famiglie$Anno_2007+ Istat_Famiglie$Anno_2008+  
Istat_Famiglie$Anno_2009+ Istat_Famiglie$Anno_2010+  
Istat_Famiglie$Anno_2011+Istat_Famiglie$Anno_2012+Istat_Famiglie$Anno_2013+  
Istat_Famiglie$Anno_2014+ Istat_Famiglie$Anno_2015+Istat_Famiglie$Anno_2016)
```

Stampo la predizione 2018

```
predict(reg)
```

Aggiungo la Colonna della previsione ad una nuova matrice

```
m2 <- cbind(m1, predict(reg))
```

Stampo la previsione

```
m2[,12]
```

PIEMONTE	VALLE D'AOSTA	LIGURIA	LOMBARDIA	TRENTINO	VENETO
2171.792	2363.691	1978.892	2553.094	2572.198	2305.999
FRIULI	EMILIA ROMAGNA	TOSCANA	UMBRIA	MARCHE	LAZIO
2135.363	2530.087	2393.346	1894.259	1871.358	2257.226

ABRUZZO	MOLISE	CAMPANIA	PUGLIA	BASILICATA	CALABRIA
1779.977	1669.947	1665.452	1707.722	1516.702	1351.419
SICILIA	SARDEGNA				
1481.153	1637.064				

plot about Liguria

```
plot(m2[3,], main="Avarage Expenses of Liguria", type="l", ylab="Euro€", xlab="Years", col="black")
```

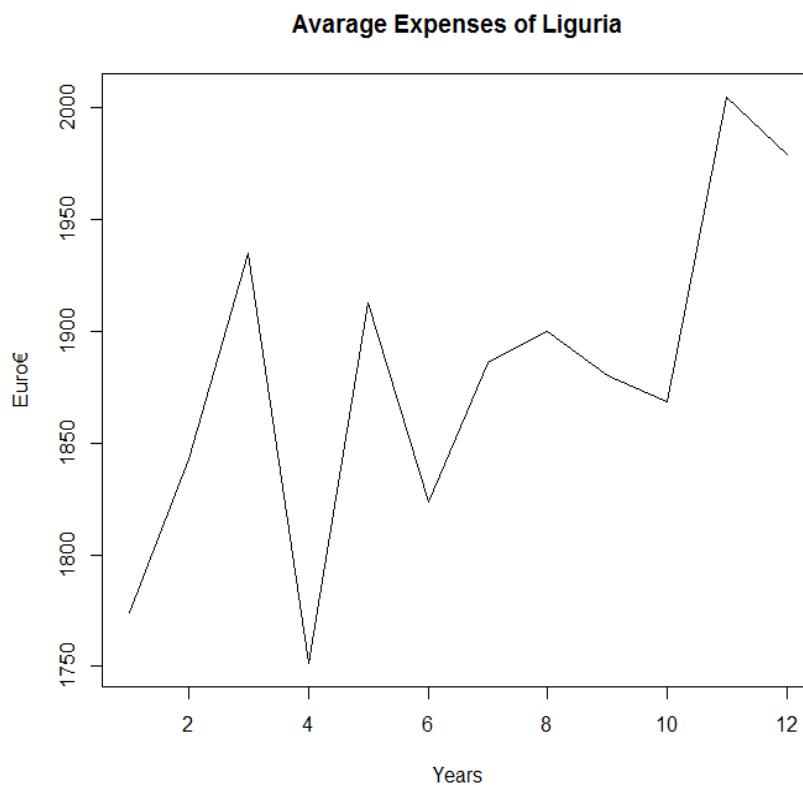


Figura 43: Plot Ag Expenses About Liguria

A4. SQL-CODE: BEST GEO-LOCATIONS

-----BEST REGION BY INDICATORS-----

```
select settore, indicatore, unita_di_misura, territorio, Prediction_2018, rnk
from (Select *,
      case
        when Indicatore <> 'Tasso di disoccupazione'
        then RANK() over (partition by Indicatore order by Prediction_2018 desc)
        else RANK() over (partition by Indicatore order by Prediction_2018 ) end rnk
      from L2_ISTAT.FACT_ISTAT ) t
where rnk < 3 or rnk > 18
```

-----BEST REGIONS-----

```
select t3.territorio, isnull(t2.c, 0) as #closing_shop, isnull(t4.c, 0) as #opening_shop, sum(t3.rank1) lstat_rank,
       sum(t3.rank1 + isnull(t2.c, 0) + isnull(t4.c, 0)) as Final_Rank
from (select t.Territorio, sum(t.rnk) rank1
      from (Select *,
            case when FACT_ISTAT.Indicatore <> 'Tasso di disoccupazione'
            then RANK() over (partition by FACT_ISTAT.Indicatore
                           order by FACT_ISTAT.Prediction_2018 desc)
            else RANK() over (partition by FACT_ISTAT.Indicatore order by FACT_ISTAT.Prediction_2018
                           ) end rnk
            from L2_ISTAT.FACT_ISTAT
          ) t
      Group by t.Territorio) t3
left join
(Select DIM_LOCATION_SS.regione, COUNT(DISTINCT DIM_SHOP_SS._negozio_desc_) as c
 From L2_STAR_SCHEMA.DIM_SHOP_SS,
      L2_STAR_SCHEMA.DIM_LOCATION_SS
 where L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_ like '%/*%' escape '/'
 and L2_STAR_SCHEMA.DIM_SHOP_SS._COMUNE_SK =
L2_STAR_SCHEMA.DIM_LOCATION_SS._COMUNE_SK
 group by DIM_LOCATION_SS.regione) as t2
on t3.Territorio = t2.regione
left join
(Select DIM_LOCATION_SS.regione, COUNT(DISTINCT DIM_SHOP_SS._negozio_desc_) as c
 From L2_STAR_SCHEMA.DIM_SHOP_SS,
      L2_STAR_SCHEMA.DIM_LOCATION_SS
 where L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_ not like '%/*%' escape '/'
 and L2_STAR_SCHEMA.DIM_SHOP_SS._COMUNE_SK =
L2_STAR_SCHEMA.DIM_LOCATION_SS._COMUNE_SK
 group by DIM_LOCATION_SS.regione) as t4
on t4.regione = t2.regione
Group by t3.Territorio, t2.c, t4.c
order by Final_Rank;
```

-----BEST CITIES-----

```
select t.comune,
       t.num_residenti,
       t2.Totale_esercizi_Arrivi_2018,
       t2.Totale_esercizi_Arrivi_2018 / t.num_residenti measure
from (select *, RANK() over ( order by num_residenti ) rnk
```



```

from L2_ISTAT.DIM_LOCATION_ISTAT
Where L2_ISTAT.DIM_LOCATION_ISTAT.regione in ('TRENTINO ALTO ADIGE', 'FRIULI VENEZIA
GIULIA')
or L2_ISTAT.DIM_LOCATION_ISTAT.regione like '%AOSTA%') as t,
(select *, RANK() over ( order by Totale_esercizi_Arrivi_2018 ) rnk2
from L2_ISTAT.DIM_TURISMO_ISTAT) as t2,
L2_ISTAT.DIM_LOCATION_ISTAT
Where
t._COMUNE_SK=t2._COMUNE_SK
group by
t.num_residenti, t.comune, rnk, t2.Totale_esercizi_Arrivi_2018
order by measure desc;

```

A5. SQL-CODE: AGGREGATE FACT SALES PER LA CREAZIONE DEL MODELLO CART

CREATION OF AGGREGATE TABLES

```

create table AGGREGATE.CART_YEAR_SHOP_SALES_1718
(
SHOP_ID int not null,
SHOP Varchar(1000) not null,
DISCOUNT float not null,
MARGIN float not null,
nRECEIPT int not null,
GAIN float not null,
COGS float not null,
FlagOpen Varchar(1000) not null
)
create table AGGREGATE.CART_QUARTER_SHOP_SALES_1718
(
SHOP_ID int not null,
SHOP Varchar(1000) not null,
MARGIN float not null,
nRECEIPT int not null,
GAIN float not null,
COGS float not null,
FlagOpen Varchar(1000) not null
)
create table AGGREGATE.CART_QUARTER_SHOP_SALES_19
(
SHOP_ID int not null,
SHOP Varchar(1000) not null,
MARGIN float not null,
nRECEIPT int not null,
GAIN float not null,
COGS float not null,
FlagOpen Varchar(1000) not null
)

```

QUERIES

```
INSERT INTO AGGREGATE.CART_YEAR_SHOP_SALES_1718
select Distinct L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_           as SHOP_ID,
                L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_        as SHOP,
                SUM(_SCONTO_GENERICO_SC_ + _SCONTO_SC_)           as DISCOUNT,
                (SUM(_FATTURATO_NI_SC_) - SUM(_COSTO_VENDUTO_NI_SC_)) as MARGIN,
                count(distinct _NUMERO_SCONTRINO_)                as nRECEIPT,
                SUM(_FATTURATO_NI_SC_)                             as GAIN,
                SUM(_COSTO_VENDUTO_NI_SC_)                         as COGS,
                case
                  when L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_ not like '%*%' then 'OPEN'
                  else 'CLOSE' end                               as FlagOpen
from L2_STAR_SCHEMA.FACT_SALES_SS,
      L2_STAR_SCHEMA.DIM_DATE_SS,
      L2_STAR_SCHEMA.DIM_SHOP_SS
Where L2_STAR_SCHEMA.FACT_SALES_SS._GIORNO_ = L2_STAR_SCHEMA.DIM_DATE_SS.Date
and L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_ =
L2_STAR_SCHEMA.FACT_SALES_SS._NEGOZIO_ID_
and L2_STAR_SCHEMA.DIM_DATE_SS.Year in (2018,2017)
Group by
L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_,L2_STAR_SCHEMA.DIM_SHOP_SS._CANALE_DE
SC_,L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_
Order by FlagOpen;
```

```
INSERT INTO AGGREGATE.CART_QUARTER_SHOP_SALES_1718
select Distinct L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_           as SHOP_ID,
                L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_        as SHOP,
                (SUM(_FATTURATO_NI_SC_) - SUM(_COSTO_VENDUTO_NI_SC_)) as MARGIN,
                count(distinct _NUMERO_SCONTRINO_)                as nRECEIPT,
                SUM(_FATTURATO_NI_SC_)                             as GAIN,
                SUM(_COSTO_VENDUTO_NI_SC_)                         as COGS,
                case
                  when L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_ not like '%*%' then 'OPEN'
                  else 'CLOSE' end                               as FlagOpen
from L2_STAR_SCHEMA.FACT_SALES_SS,
      L2_STAR_SCHEMA.DIM_DATE_SS,
      L2_STAR_SCHEMA.DIM_SHOP_SS
Where L2_STAR_SCHEMA.FACT_SALES_SS._GIORNO_ = L2_STAR_SCHEMA.DIM_DATE_SS.Date
and L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_ =
L2_STAR_SCHEMA.FACT_SALES_SS._NEGOZIO_ID_
and L2_STAR_SCHEMA.DIM_DATE_SS.Year in (2018, 2017)
and L2_STAR_SCHEMA.DIM_DATE_SS.Quarter = 1
Group by L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_,
L2_STAR_SCHEMA.DIM_SHOP_SS._CANALE_DESC_,
L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_
Order by FlagOpen;
```

```

INSERT INTO AGGREGATE.CART_QUARTER_SHOP_SALES_19
select Distinct L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_          as SHOP_ID,
                L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_        as SHOP,
                (SUM(_FATTURATO_NI_SC_) - SUM(_COSTO_VENDUTO_NI_SC_)) as MARGIN,
                count(distinct _NUMERO_SCONTRINO_)                as nRECEIPT,
                SUM(_FATTURATO_NI_SC_)                            as GAIN,
                SUM(_COSTO_VENDUTO_NI_SC_)                        as COGS,
                case
                  when SUM(_FATTURATO_NI_SC_) >0 then 'OPEN'
                  else 'CLOSE' end                               as FlagOpen
from L2_STAR_SCHEMA.FACT_SALES_SS,
      L2_STAR_SCHEMA.DIM_DATE_SS,
      L2_STAR_SCHEMA.DIM_SHOP_SS
Where L2_STAR_SCHEMA.FACT_SALES_SS._GIORNO_ = L2_STAR_SCHEMA.DIM_DATE_SS.Date
and L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_ =
L2_STAR_SCHEMA.FACT_SALES_SS._NEGOZIO_ID_
and L2_STAR_SCHEMA.DIM_DATE_SS.Year in (2019)
--and
and L2_STAR_SCHEMA.DIM_DATE_SS.Quarter = 1
Group by L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_DESC_,
L2_STAR_SCHEMA.DIM_SHOP_SS._CANALE_DESC_,
L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_
Order by FlagOpen;

```

```

-----
TRUNCATE
-----

```

```

truncate table AGGREGATE.CART_YEAR_SHOP_SALES_1718;
truncate table AGGREGATE.CART_QUARTER_SHOP_SALES_1718;
truncate table AGGREGATE.CART_QUARTER_SHOP_SALES_19;

```

```

-----
CHECK
-----

```

```

Select COUNT(DISTINCT _negozio_desc_) as OPEN_SHOP, year
From L2_STAR_SCHEMA.DIM_SHOP_SS,
      L2_STAR_SCHEMA.FACT_SALES_SS,
      L2_STAR_SCHEMA.DIM_DATE_SS
where L2_STAR_SCHEMA.DIM_SHOP_SS._NEGOZIO_ID_ =
L2_STAR_SCHEMA.FACT_SALES_SS._NEGOZIO_ID_
and L2_STAR_SCHEMA.FACT_SALES_SS._GIORNO_ = L2_STAR_SCHEMA.DIM_DATE_SS.Date
and L2_STAR_SCHEMA.FACT_SALES_SS._FATTURATO_NI_SC_ > 0
group by Year;

```

```

-----
OPEN_SHOP year
93    2017
77    2018
29    2019
-----

```

A6. R-CODE: CLASSIFICATION AND REGRESSION TREE (CART)

```
library(dbConnect)
library(odbc)
library(DBI)
library(rpart.plot)
library(rpart)
library(tree)
library(gplots)
library(ineq)

#Estraggo tabella 17_18

con <- dbConnect(odbc::odbc(),DRIVER="SQL Server", SERVER="192.168.2.14", DATABASE =
"FASHION_RETAIL", UID="", PWD=" ")

query <- dbSendQuery(con, "select * from AGGREGATE.CART_QUARTER_SHOP_SALES_1718 ")
#query <- dbSendQuery(con, "select * from AGGREGATE.CART_YEAR_SHOP_SALES_1718 ")

DATA <- dbFetch(query)

DATASET <- DATA [3:ncol(DATA)]    #elimino colonne 1,2 riferite ai negozi (SHOP_ID & SHOP)

#Partizione dei dati

pd17 <- sample(2,nrow(DATASET),replace=TRUE, prob= c(0.8,0.2))

TRAININGSet <- DATASET[pd17==1,]
TESTSet<- DATASET[pd17==2,]

#CART

CART <- rpart(FlagOpen ~ ., data=TRAININGSet, method="class",control=rpart.control(minsplit=5))

#CART<- rpart(FlagOpen ~ DISCOUNT + MARGIN + nRECEIPT + GAIN+ COGS, data=DATASET17,
method="class",rpart.control(minsplit = 5))

#Color CART

boxcols <- c("red", "palegreen3")[CART$frame$yval]

#Plot CART

rpart.plot(CART, fallen.leaves=TRUE,box.col = boxcols, uniform=TRUE )
prp(CART,box.col = boxcols)

# view results

print(CART)

#Inserisco TESTset nell'ALGORITMO CART

prevision <- predict(CART, TESTSet, type="class")
```

#accuratezza previsione

```
Gini(prevision)
```

```
mean (prevision == TESTSet$FlagOpen)
```

#Estraggo tabella 19

```
con2 <- dbConnect(odbc::odbc(),DRIVER="SQL Server", SERVER="192.168.2.14", DATABASE =  
"FASHION_RETAIL", UID=" ", PWD=" ")
```

```
query19 <-dbSendQuery(con2,"select * from AGGREGATE.CART_QUARTER_SHOP_SALES_19 ")
```

```
DATA19 <- dbFetch(query19)
```

#elimino colonna riferita alla descrizione negozi

```
DATASET19 <- DATA19 [-2]
```

#Inserisco la tabella nell'ALGORITMO CART

```
prevision2019 <- predict(CART, DATASET19, type="class")
```

```
prevision2019
```

#accuratezza previsione

```
Gini(prevision2019)
```

```
mean (prevision2019 == DATASET19$FlagOpen)
```

#Statistics

```
summary(prevision2019)
```

```
cbind(DATA19, prevision2019)
```

OUTPUT

view results

```
> print(CART)
```

```
n= 78
```

```
node), split, n, loss, yval, (yprob)
```

```
    * denotes terminal node
```

```
1) root 78 20 OPEN (0.25641026 0.74358974)
```

```
2) MARGIN< 1235.842 5 0 CLOSE (1.00000000 0.00000000) *
```

```
3) MARGIN>=1235.842 73 15 OPEN (0.20547945 0.79452055)
```

```
6) MARGIN< 17461.73 19 7 OPEN (0.36842105 0.63157895)
```

```
12) nRECEIPT>=113 8 2 CLOSE (0.75000000 0.25000000) *
```

```
13) nRECEIPT< 113 11 1 OPEN (0.09090909 0.90909091) *
```

```
7) MARGIN>=17461.73 54 8 OPEN (0.14814815 0.85185185)
```

```

14) nRECEIPT>=459 43 8 OPEN (0.18604651 0.81395349)
28) MARGIN< 39091.27 3 1 CLOSE (0.66666667 0.33333333) *
29) MARGIN>=39091.27 40 6 OPEN (0.15000000 0.85000000)
58) COGS< 75358.5 6 2 OPEN (0.33333333 0.66666667) *
59) COGS>=75358.5 34 4 OPEN (0.11764706 0.88235294)
118) COGS>=161920.7 18 4 OPEN (0.22222222 0.77777778)
236) nRECEIPT< 2059 2 0 CLOSE (1.00000000 0.00000000) *
237) nRECEIPT>=2059 16 2 OPEN (0.12500000 0.87500000) *
119) COGS< 161920.7 16 0 OPEN (0.00000000 1.00000000) *
15) nRECEIPT< 459 11 0 OPEN (0.00000000 1.00000000) *

```

> summary(CART)

Call:

```

rpart(formula = FlagOpen ~ ., data = TRAININGSet, method = "class",
      control = rpart.control(minsplit = 5))
n= 78

```

	CP	nsplit	rel error	xerror	xstd
1	0.250	0	1.00	1.0	0.1928198
2	0.100	1	0.75	0.8	0.1783112
3	0.025	3	0.55	0.9	0.1860521
4	0.010	8	0.40	1.0	0.1928198

Variable importance

GAIN	COGS	MARGIN	nRECEIPT
28	26	25	21

#Inserisco TESTset nell'ALGORITMO CART

```
> prevision <- predict(CART, TESTSet, type="class")
```

#accuratezza previsione

```
> Gini(prevision)
```

```
[1] 0.09428571
```

```
> mean (prevision == TESTSet$FlagOpen)
```

```
[1] 0.8571429
```

> prevision2019

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
OPEN	OPEN	OPEN	OPEN	CLOSE	OPEN	OPEN	OPEN	CLOSE	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN	OPEN
OPEN	OPEN	CLOSE	OPEN	OPEN	CLOSE	OPEN														

22 23 24 25 26 27 28 29

OPEN CLOSE OPEN OPEN OPEN OPEN OPEN OPEN

Levels: CLOSE OPEN

#accuratezza previsione

> Gini(previson2019)

[1] 0.07807417

> mean (previson2019 == DATASET1819\$FlagOpen)

[1] 0.8275862

>

Statistics

> summary(previson2019)

CLOSE OPEN

4 25

	SHOP_ID	MARGIN	nRECEIPT	GAIN	COGS	FlagOpen	previson2019
1	4064196	194247.68	5505	681708.61	487460.93	OPEN	OPEN
2	73638	20007.68	337	44479.96	24472.28	OPEN	OPEN
3	2237727	630648.10	7019	1225036.95	594388.85	OPEN	OPEN
4	2286770	175389.83	1656	330031.18	154641.35	OPEN	OPEN
5	3253865	31478.99	473	67379.97	35900.98	OPEN	CLOSE
6	4063154	69378.27	2699	288114.85	218736.58	OPEN	OPEN
7	73647	103459.19	3064	403503.95	300044.76	OPEN	OPEN
8	73569	56877.92	813	116076.54	59198.62	OPEN	OPEN
9	73539	32635.11	525	69497.32	36862.21	OPEN	CLOSE
10	4596364	102839.96	1222	201342.97	98503.01	OPEN	OPEN
11	3694385	292708.76	2945	564322.08	271613.32	OPEN	OPEN
12	3145981	40576.74	1484	172534.32	131957.58	OPEN	OPEN
13	1489952	206888.62	2081	399424.69	192536.07	OPEN	CLOSE
14	2265838	485280.72	8978	1033463.06	548182.34	OPEN	OPEN
15	73830	25486.93	405	56269.82	30782.89	OPEN	OPEN
16	73801	39345.60	568	80020.96	40675.36	OPEN	OPEN
17	73716	32220.88	475	68861.94	36641.06	OPEN	CLOSE
18	2264278	46636.65	701	97573.92	50937.27	OPEN	OPEN
19	2726114	26713.93	402	54045.17	27331.24	OPEN	OPEN
20	73773	12420.89	211	28334.43	15913.54	OPEN	OPEN
21	4064198	336480.19	7787	1196995.75	860515.56	OPEN	OPEN
22	175674	77269.68	1035	162405.14	85135.46	OPEN	OPEN
23	73740	36069.38	529	75713.85	39644.47	OPEN	OPEN
24	73825	72161.05	946	150774.37	78613.32	OPEN	OPEN
25	2022028	4178.04	88	10211.89	6033.85	OPEN	OPEN
26	2167235	445591.73	4230	846926.01	401334.28	OPEN	OPEN
27	2203462	188159.04	5098	659833.14	471674.10	OPEN	OPEN
28	876293	63376.15	891	134117.87	70741.72	OPEN	OPEN
29	2312294	141998.28	4191	549655.07	407656.79	OPEN	OPEN

A7. DAX-CODE: VARIABILI DI POWER BI

Total Sales = CALCULATE (sum ('L2_STAR_SCHEMA FACT_SALES_SS_DM'[_FATTURATO_LI_SC_]))

Total Sales LY = CALCULATE ([Total Sales]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2017)

Total Sales TY = CALCULATE ([Total Sales]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2018)

Total Sales Var = [Total Sales TY]-[Total Sales LY]

Total Sales Var % = IF ([Total Sales LY]<>0; [Total Sales Var]/[Total Sales LY]; BLANK())

Total Units = CALCULATE (sum ('L2_STAR_SCHEMA FACT_SALES_SS_DM'[_QUANTITA_SC_]))

Total Units Last Year = CALCULATE ([Total Units]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2017)

Total Units This Year = CALCULATE ([Total Units]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2018)

Total COGS = CALCULATE (SUM ('L2_STAR_SCHEMA FACT_SALES_SS_DM'[_COSTO_VENDUTO_NI_SC_]))

Total COGS LY = CALCULATE ([Total COGS]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2017)

Total COGS TY = CALCULATE ([Total COGS]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2018)

Total Gross Margin = [Total Sales] - [Total COGS]

Total Gross Margin LY = [Total Sales LY] - [Total COGS LY]

Total Gross Margin TY = [Total Sales TY] - [Total COGS TY]

Gross Margin This Year % = [Total Gross Margin TY] / [Total Sales TY]

Gross Margin Last Year % = [Total Gross Margin LY] / [Total Sales LY]

Total Gross Margin var = [Total Gross Margin TY] - [Total Gross Margin LY]

Gross Margin Var % = IF ([Total Gross Margin LY] <>0; [Total Gross Margin var] / [Total Gross Margin LY]; BLANK())

Avg \$/Unit TY = IF ([Total Units This Year] <>0; [Total Sales TY] / [Total Units This Year]; BLANK ())

Avg \$/Unit LY = IF ([Total Units Last Year] <>0; [Total Sales LY] / [Total Units Last Year]; BLANK ())

Average Unit Price = [Avg \$/Unit TY]

Store Count = DISTINCTCOUNT ('L2_STAR_SCHEMA DIM_SHOP_SS'[_NEGOZIO_ID_])

Sales Per Sq Ft = ([Total Sales TY] / (DISTINCTCOUNT ('L2_STAR_SCHEMA DIM_DATE_SS'[Month]) * SUM('L2_STAR_SCHEMA DIM_SHOP_SS'[_SUPERFICIE_]))) * 12

REASON2017 = if ([Total Gross Margin LY] <=200000;"BAD MARGIN";"COVERED/NEW")

REASON2018 = if ([Total Gross Margin TY] <=200000;"BAD MARGIN";"COVERED/NEW")

FLG APERTO = IF (CONTAINSSTRING ('L2_STAR_SCHEMA DIM_SHOP_SS'[_NEGOZIO_DESC_];"~*") = TRUE;0;1)

Count Aperti = sum ('L2_STAR_SCHEMA DIM_SHOP_SS'[FLG APERTO])

TotSHOP = COUNT ('L2_STAR_SCHEMA DIM_SHOP_SS'[_NEGOZIO_ID_])

#OPENshop = sum ('L2_STAR_SCHEMA DIM_SHOP_SS'[OPEN])

#CLOSEshop = COUNT ('L2_STAR_SCHEMA DIM_SHOP_SS'[_NAZIONE_DESC_]) -
('L2_STAR_SCHEMA DIM_SHOP_SS'[#OPENshop])

Total Discount = sum ('L2_STAR_SCHEMA FACT_SALES_SS_DM'[_SCONTO_GENERICO_SC_]) +
sum ('L2_STAR_SCHEMA FACT_SALES_SS_DM'[_SCONTO_SC_])

TOT DISCOUNT TY= Calculate ([Total Discount];'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2018)

TOT DISCOUNT LY= Calculate ([Total Discount];'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2017)

Total Sales 2019 = CALCULATE ([Total Sales]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2019)

Total Sales Var 2019% = IF ([Total Sales TY] <>0; ([Total Sales 2019] - [Total Sales TY]) / [Total Sales TY];
BLANK ())

Total COGS 2019 = CALCULATE ([Total COGS]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2019)

COGS Var 2019 %= IF ([Total COGS TY] <>0; ([Total COGS 2019] - [Total COGS TY]) / [Total COGS TY];
BLANK ())

Total Gross Margin 2019 = [Total Sales 2019] - [Total COGS 2019]

Gross Margin Var 2019% = IF ([Total Gross Margin TY] <>0; ([Total Gross Margin 2019] - [Total Gross
Margin TY]) / [Total Gross Margin TY]; BLANK ())

DISCOUNT 2019 = CALCULATE ([Total DISCOUNT]; 'L2_STAR_SCHEMA DIM_DATE_SS'[Year]=2019)

DISCOUNT Var 2019 % = IF ([Total DISCOUNT] <>0; ([DISCOUNT 2019] - [DISCOUNT TY]) / [DISCOUNT
TY]; BLANK ())

REFERENCES

- 1) Manyika J., C. M. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*.
- 2) Opresnik D., T. M. (2015). The value of Big Data in servitization. *International Journal of Production Economics*.
- 3) Court D. (2015). Getting big impact from big data. *McKinsey Quarterly*.
- 4) Hazan E., B. F. (2013). Leveraging big data to optimize digital marketing. *McKinsey Quarterly*.
- 5) Casali A. (2015). *Digital Marketing. Mobile, video, big data e social: internet trasforma la pubblicità*.
- 6) Lühr P., M. R. (2013). Name your price: The power of Big Data and analytics. *McKinsey Quarterly*.
- 7) Talend (2009) The Top 10 Reasons for Choosing Open Source Data Integration.
- 8) Mark R. Madsen (2009) The Role of Open Source in Data Integration, Third Nature Technology Report.
- 9) Baumgartner T., H. H. (2011). Find Big Growth in Big Data. In H. H. Baumgartner T., *Sales Growth. Five proven strategies from world's sales leader*.
- 10) Adamson C. (2010), *Star Schema: The Complete Reference*, New York, McGraw-Hill.
- 11) Rezzani A. (2012), *Business Intelligence – Processi, metodi, utilizzo in azienda*, Milano, Feltrinelli Editore.
- 12) Vidette P., Patricia K. e Stephen B. (1998), "Building a Data Warehouse for Decision Support", Prentice-Hall.
- 13) <https://www.educba.com/data-mining-vs-machine-learning/>
- 14) <https://www.simonefavarolo.it/2017/04/07/introduzione-machine-learning/>
- 15) Michael J. Berry, Gordon Linoff, (2004), "Introduction to Data Mining".
- 16) Wray Buntine, (1992), "Learning classification trees".
- 17) <https://en.wikipedia.org/wiki/Bayes%27theorem> , "Bayes' theorem".
- 18) Dr. Saed Sayad, "Support Vector Machine - Classification (SVM)".
http://www.saedsayad.com/support_vector_machine.htm .

- 19) https://en.wikipedia.org/wiki/Hierarchical_clustering , “Hierarchical clustering” 2016.
- 20) Xiaoshan Du.(2016). Master’s Thesis: Data Mining Analysis and Modeling for Marketing Based on Attributes of Customer Relationship.
- 21) Judith Hurwitz, Daniel Kirsch.(2018). Machine Learning For Dummies, IBM Limited Edition.
- 22) Raghupathi W., R. V. (2014). Big data analytics in healthcare: promise and potential. Health Information Science and Systems.
- 23) Emil Drkušić. Star Schema vs. Snowflake Schema 2016
<https://www.vertabelo.com/blog/technical-articles/data-warehouse-modeling-star-schema-vs-snowflake-schema> .
- 24) Talend Help Center, tMap.
<https://help.talend.com/reader/wDRBNUuxk629sNcl0dNYaA/mxzKD~8eLuNFSXH6LMi7qg> .
- 25) Dati dell’Istituto Nazionale di Statistica Italiano, ISTAT. <http://dati.istat.it/> .
- 26) Sito R-Studio. <https://www.rstudio.com/> .
- 27) Definizione codice R. <https://www.r-project.org/about.html> .
- 28) Talend Open Studio. <https://help.talend.com/home> .
- 29) F-Stat. Wikipedia. <https://en.wikipedia.org/wiki/F-test> .
- 30) T-Stat. <https://www.statisticshowto.datasciencecentral.com/t-statistic/> .
- 31) Microsoft Power BI. <https://powerbi.microsoft.com/it-it/> .
- 32) Alberto Ferrari and Marco Russo. Introducing Microsoft Power BI.
- 33) Everything you need to know about Real Time Business.
<https://www.sisense.com/blog/everything-you-need-to-know-about-real-time-business-intelligence/>
- 34) Disney MagicBand.
<https://disneyworld.disney.go.com/en-eu/plan/my-disney-experience/bands-cards/>
- 35) Amit Roy Choudhury and Jim Mortleman. How IoT is turning Rolls-Royce into a data-fuelled business. January 2018.