UNIVERSITÀ DEGLI STUDI DI TORINO
POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Realizzazione di un Datawarehouse e sviluppo di metodologie di Advanced Analytics a supporto delle strategie aziendali

Relatore:
prof. Paolo Garza

Candidato:
Donato Chiarello
matricola: 209988

Business Analytics & Big Data
Simone Morassutto

Anno accademico 2019-2020

† A mio nonno Biagio

Indice

Introduzione

I processi di marketing aziendali sono caratterizzati da un alto livello di complessità. L'importanza di modelli strutturati sta crescendo grazie allo sviluppo di sempre più efficienti meccanismi di registrazione delle transazioni, che forniscono informazioni sul comportamento dei consumatori. L'abilità di sfruttare queste informazioni raccolte rappresenta una forza competitiva per l'azienda. Infatti saper direzionare il marketing, in base ai dati raccolti, si possono raggiungere vantaggi rispetto ai competitors.

Nello specifico la "market basket analysis" ha il compito di trovare le relazioni tra prodotti, all'interno del carrello della spesa. Conoscendo queste informazioni, un punto vendita può rispondere in maniere accurata ai bisogni dei consumatori.

Il classico esempio di correlazione tra birra e pannolini è un esempio di relazioni inaspettate ma che hanno una spiegazione. Le informazioni tratte possono essere utilizzate ad esempio posizionando nei punti vendita i prodotti vicini, promuovendo offerte sui prodotti se acquistati insieme etc.

Caso di Studio All'interno del team di Mediamente Consulting, azienda di consulenza in ambito Data Analytics e Big Data, si è realizzato una metodologia per l'analisi del "market basket analysis" e tutte le analisi derivanti. Nell'ambito di tesi mi soffermerò principalmente su come è stato costruito il DWH e le derivanti analisi avanzate effettuate a partire dai dati del DWH. Questa metologia è stata utilizzata in un sistema esistente. La piattaforma prevede la lettura dei dati da diversi sistemi sorgenti tramite un software di *ETL*. Le informazioni vengono archiviate nel *DWH*, successivamente analizzate tramite tecniche di *Data Mining* per ricavare i *KPI* di interesse.

Capitolo 1

Tecniche di Analisi dei Dati

L'analisi dei dati è un processo di ispezione, pulizia, trasformazione e modellazione dei dati con il fine di supportare le decisioni strategiche aziendali. Infatti, analizzare efficacemente la sempre più crescente quantità di dati che si forma giorno dopo giorno permette di creare valore che può essere sfruttato dalle aziende per migliorare il processo decisionale e, di conseguenza, il vantaggio competitivo dell'azienda stessa sul mercato.

Ogni operazione svolta su dispositivi digitali è in grado di generare dati. Negli ultimi anni, con l'avvento dei canali digitali, è aumentato esponenzialmente il numero e la complessità di questi dati. Ma essi, se non vengono tramutati in informazioni utili, rimangono semplici statistiche. Oggigiorno ci troviamo di fronte a questa vasta quantità di informazioni che, se raccolte, selezionate, analizzate e valorizzate economicamente, possono essere per le aziende fonte di vantaggio competitivo. L'uso delle informazioni per indirizzare il processo decisionale aziendale non è di per sé una novità, ma ciò che lo differenzia dal passato è l'aumento dei dati raccolti e la diminuzione dei costi di raccolta, immagazzinamento e processing. Questa è l'era della cosiddetta *"Big Data Analytics"*, ovvero del processo di raccolta e di analisi dell'insieme di dati eterogenei (Big Data), ottenuti da diversi fonti, al fine di recuperare informazioni nascoste che, analizzate e interpretate, si possono tradurre in dati fondamentali riguardanti il comportamento dei consumatori e l'andamento generale del mercato, migliorando così in termini di efficienza ed efficacia l'attività decisionale delle imprese rispetto alla concorrenza. L'uso consapevole e mirato dei dati nelle strategie di business intelligence risulta pertanto una componente imprescindibile, ma essa non deve essere considerata una strategia univoca per tutte le imprese, in quanto per trasformare le conoscenze in vantaggio competitivo è necessario prima identificare quali sono le giuste informazioni da ricercare. Ciò che differenzia i migliori analisti Big Data è infatti proprio il saper selezionare il giusto tipo di analisi che potrebbe portare al vantaggio competitivo. Per analisi di business ci si riferisce ad un sistema avanzato di data processing che consente di ottenere un quadro esatto

della situazione attuale, di prevedere scenari futuri e di sostenere e proporre scelte efficaci e risultati concreti. In base al tipo di problema che si intende analizzare, l'analisi dei big data porta ad approfondire, più o meno dettagliatamente, le seguenti tipologie di analisi: analisi descrittiva, che consente di rispondere alla domanda *"cosa è successo?"*, analisi diagnostica, che consente di rispondere invece alla domanda *"perché è successo?"*, analisi predittiva, che permette di rispondere alla domanda *"cosa potrebbe accadere in futuro?"* ed infine analisi prescrittiva, con la quale si può affrontare il quesito *"come dovremmo rispondere a quei potenziali eventi futuri?"*.

Figura 1.1. Tipi di analisi

1.1 Analisi Descrittiva

L'analisi descrittiva si occupa della trasformazione dei dati grezzi in una forma che li renda facili da comprendere, interpretare, ordinare, riorganizzare, e modificare, al fine di creare informazione utile. L'analisi descrittiva (come indicato dal termine) "descrive" ciò che accade e, attraverso strumenti di Business Intelligence e di visualizzazione, fornisce informazioni. In base all'Analytics for the Customer-Driven Supply Chain, l'analisi descrittiva viene usata dalle aziende per capire, spesso a livello aggregato, cosa accade all'interno di esse, per descrivere vari aspetti del business, per comprendere e spiegare le performance e le dinamiche di specifiche metriche di

interesse al fine di estrarne indicazioni utili su come influenzare i risultati futuri. È una forma semplice e molto diffusa di analisi di superficie. Le analisi descrittive permettono quindi alle imprese di creare un riepilogo dei dati storici, utili a capire come migliorare le attività future. Un esempio di questo tipo di analisi è la classificazione dei consumatori sulla base di caratteristiche note: si guardano le informazioni in possesso e le si usano per acquisire una visione di dettaglio o d'insieme. Le imprese sfruttano solitamente l'analisi descrittiva per indagare:

- Tabulazioni delle metriche sociali, quali tweet, followers e like su Facebook
- Eventi del passato, quali il successo di una campagna pubblicitaria o l'aumento della clientela
- Report aziendali che riportano una revisione storica delle operazioni, dei dati finanziari, dei clienti, degli stakeholders e delle vendite
- Previsione di tendenze generali

Tra le fonti dei dati comuni usate in questa fase vi sono le osservazioni, i sondaggi e i casi di studio. In sostanza, questo tipo di analisi si pone lo scopo di visualizzare i dati nel giusto contesto, trovare le informazioni rilevanti, valutare la qualità dei dati e riconoscere i limiti e le ipotesi di ciò che viene ricavato. I passi fondamentali da compiere per svolgere questo tipo di analisi sono i seguenti: inizialmente è necessario identificare il fenomeno di interesse, bisogna stabilire quali caratteristiche sono più importanti e trovare le misure che le rappresentino al meglio, riprodurre poi le informazioni tramite formati facilmente comprensibile (quali, ad esempio, diagrammi, grafici e tabelle), infine, il risultato ottenuto dovrà essere confrontato con altre analisi passate o future. L'analisi descrittiva è importante che sia dettagliata, chiara, accurata e comprensibile a tutti. Con l'analisi dei dati si possono ricavare informazioni più dettagliate, ma già note, che descrivono situazioni passate (è il caso della vera e propria analisi descrittiva) o si possono scoprire informazioni del tutto nuove. Quest'ultimo è invece il caso dell'Exploratory Data Analysis, che si focalizza sul cercare tendenze, ipotesi, relazioni e scoperte. Frequentemente i dati utilizzati nell'analisi vengono rappresentati tramite la statistica descrittiva, la quale, attraverso l'uso di tabulazioni, distribuzioni, varianza, medie, mediane, quartili e frequenze, facilita l'interpretazione delle proprietà degli insiemi di dati e la scoperta di informazioni importanti per il fenomeno di interesse preso in esame.

1.2 Analisi Diagnostica

L'analisi diagnostica è il secondo tipo di analisi. Risponde alla domanda "perché è successo" e indaga su quali possono essere la cause che hanno condotto allo stato attuale. Per capire la condizione corrente ed usare strategie mirate ci si avvale

dell'analisi diagnostica che, tramite un processo iterativo e continuativo, consente di trovare le cause di determinati eventi o comportamenti, al fine di ripetere e ottimizzare le azioni più efficaci e migliorare quelle che non hanno portato i risultati desiderati. Per effettuare un'analisi diagnostica è bene prima identificare un cambiamento significativo inaspettato che valga la pena investigare (come un calo delle vendite o della clientela), in seguito si esegue l'analisi, semplice o complessa, per scovare le correlazioni tra le variabili. Infine è necessario selezionare le diagnosi ed esporre in maniera chiara e concisa la conclusione. L'analisi diagnostica si avvale di tecniche di data discovery o data mining, correlazioni e drill-down. La Data Discovery, ovvero la scoperta dei dati (attraverso strumenti quali tabelle pivot, grafici a barre e a torta, mappe geografiche e di calore) esamina i dati provenienti da varie fonti e cerca di estrapolare informazioni significative per rafforzare le decisioni dell'impresa e facilitare gli utenti al conseguimento dei loro scopi. A seguito dell'aumento dei big data, la data discovery viene riconosciuta come tecnica dei dati aziendali critici che agevola l'analista nella comprensione dei dati. Se quest'ultimo riscontra problemi nell'esporre alle figure decisionali aziendali quanto scoperto si può avvalere della tecnica del drill-down. Il drill-down permette, con un semplice click, di approfondire una visualizzazione più specifica dei dati presi in analisi. Un esempio di drill-down può essere un report che mostra gli incassi di vendita per stato e che permette di vedere anche gli incassi di vendita per comune o città di uno specifico stato. Così si può ottenere una visione di insieme e una visione dettagliata: si può ad esempio capire dove un determinato prodotto è maggiormente venduto e dove invece vi è un calo delle vendite di quest'ultimo, si potrebbero anche individuare le variabili che hanno influenzato le vendite del prodotto. La tecnica più usata nell'analisi diagnostica è la correlazione, la quale indaga l'esistenza di un legame per il quale al mutare di un fenomeno corrisponde un mutamento di un altro fenomeno in base ad una specifica relazione. Attraverso una funzione regressione, lineare o non lineare, si può quantificare questa relazione. Per le imprese è importante trovare le correlazioni tra i prodotti, le vendite e i clienti al fine di migliorare i risultati ottenuti. Spesso tali tecniche si intrecciano tra di loro e, per rafforzare i dati analizzati, vengono usate a supporto l'una dell'altra.

1.3 Analisi Predittiva

Una volta completata l'analisi diagnostica si procede con la creazione di un modello predittivo. I modelli predittivi permettono di gestire più facilmente le vaste quantità di dati, riassumendo le informazioni e ampliando la loro significatività basata sugli scopi del business, al fine di poterle usare per progettare efficaci strategie di mercato. L'analisi predittiva raccoglie informazioni da insiemi di dati esistenti con lo scopo di stabilire modelli e prevedere tendenze future (per esempio per creare

una funzione di forecast sull'efficacia delle future campagne di marketing dell'impresa). Si occupa di migliorare le conoscenze del business e le strategie di marketing digitale studiando cosa accadrà in futuro. I dati disponibili sulla situazione attuale e sullo storico delle performance, vengono raccolti e raggruppati in base a criteri di rilevanza e relazioni di marketing. L'analisi predittiva contribuisce a prevedere le performance dell'impresa e il comportamento degli utenti, basandosi su stime di probabilità. All'aumentare della precisione del modello usato nell'analisi predittiva, aumenta la probabilità che l'evento si verifichi. L'analisi predittiva, se ben fatta, può portare alle imprese un aumento dei ricavi o una riduzione di costi permettendo una più celere individuazione dei problemi. Le variabili usate possono essere variabili esplicative, che riguardano le cose osservate, o variabili di risposta, che riguardano le cose che si tenta di prevedere. L'analisi predittiva può essere presa in considerazione in base all'approccio usato o al risultato trovato. Per quanto riguarda il primo, vi sono principalmente tre orientamenti (Thomas W. Miller, autore del libro Modelling Techniques in Predictive Analytics) usati nella ricerca e nella costruzione dei modelli dell'analisi predittiva: l'approccio tradizionale, l'approccio adattabile ai dati e l'approccio modello-dipendente. Il primo porta alla definizione di un modello che si basa su metodi statistici, quali la regressione logistica o lineare, seguiti dall'adattamento ai dati e dalla convalida finale del modello stesso. Il secondo approccio ha come punto di partenza lo studio dei dati, i quali definiscono poi il modello che verrà infine convalidato. L'ultimo approccio modello-dipendente crea dati e previsioni, confrontate poi con dati reali, derivate dall'inquadramento di un modello. Solitamente è bene usare congiuntamente questi metodi. La qualità delle risposte dipende dai dati e dal modello predittivo scelto. Le risposte alle previsioni possono riguardare metodi di classificazione e rispondere alla domanda "quale", oppure possono riferirsi a metodi di regressione rispondendo alla domanda "quanto". Per la creazione di un modello predittivo bisogna procedere definendo innanzitutto l'obiettivo dell'analisi predittiva valutando tempi e costi, bisogna poi munirsi dei dati utili alla ricerca e definire le variabili e gli algoritmi migliori per la costruzione del modello. Questa fase è la più critica e richiede la massima concentrazione in quanto un minimo errore può portare alla completa inaffidabilità del modello. Quando il modello è stato creato, prima del suo uso, è necessario effettuare i test di convalida per accertarsi che siano state prese in considerazione tutte le variabili fondamentali all'analisi. I modelli predittivi, con il passare del tempo, divengono sempre meno attendibili in quanto aumenta sempre più la possibilità che il comportamento preso in analisi sia cambiato. L'applicazione dell'analisi predittiva al business porta numerosi benefici, quali l'identificazione delle reali necessità dei clienti che porta ad un'ottimizzazione delle vendite, un miglioramento delle strategie di acquisizione della clientela basato sullo storico dei comportamenti, un perfezionamento della pianificazione delle campagne di marketing, l'individuazione delle migliori strategie di retention e di marketing personalizzate sulle condotte degli utenti. Utilizzando

le informazioni ricavate dall'analisi predittiva i decisori aziendali vedono ottimizzati i processi decisionali. L'analisi predittiva fa da base all'analisi prescrittiva, che stabilisce indicazioni concrete su ciò che si deve fare.

1.4 Analisi Prescrittiva

L'analisi prescrittiva è la quarta ed ultima fase dell'analisi dei dati e, consiste nella sintesi di dati e nell'uso congiunto di regole di business, tecnologie, e scienze matematiche al fine di accrescere l'efficacia del processo decisionale umano o aziendale. Tale analisi, basata sulla consulenza, aiuta a "prescrivere" varie azioni per indirizzare l'attività verso soluzioni. Avvalendosi dei suggerimenti derivati dalle applicazioni delle scienze computazionali e matematiche, questa fase ha l'obiettivo di indicare azioni a vantaggio delle predizioni e di rilevarne implicazioni. A differenza dell'analisi predittiva, quella prescrittiva va oltre la semplice previsione di risultati futuri e fornisce veri e propri metodi che ne spiegano le cause. Essa infatti, oltre ad anticipare cosa accadrà, spiega anche il "perché" accadrà. L'analisi prescrittiva è efficace solo se l'impresa sa, a monte, quali domande porsi e come comportarsi di fronte alle risposte. Solo dopo essersi posta un risultato ben definito l'impresa può formulare le giuste domande per raggiungerlo. Un uso regolare di questa analisi può condurre ad affrontare in maniera chiara ed efficace ulteriori e più ampi problemi aziendali, può favorire un processo decisionale basato sui dati, può perfezionare l'analisi aziendale e dare maggiore controllo ai vari processi dell'azienda. Un ulteriore vantaggio derivante dall'utilizzo del modello di analisi prescrittiva riguarda la possibilità di analizzare anche i feedback ricevuti dall'uso delle regole, con lo scopo di controllare le azioni e le loro conseguenze. I processi dell'analisi prescrittiva si avvalgono di varie metodologie di analisi, quali la teoria dei giochi, la simulazione, i sistemi di controllo. La tecnica più usata è quella dell'ottimizzazione. Di seguito un esempio di quanto l'analisi prescrittiva possa ottimizzare il processo decisionale. Una compagnia aerea che desidera massimizzare i profitti può avvalersi di algoritmi che adeguano automaticamente il prezzo dei biglietti e la loro disponibilità basandosi su vari elementi, quali le condizioni meteorologiche, la domanda e il costo del petrolio. Se l'algoritmo si accorge ad esempio che le vendite di biglietti di determinate tratte durate il periodo estivo sono in ritardo rispetto a quelle dello scorso anno, può in automatico abbassarne i prezzi (non oltre però una determinata soglia che dipende dal prezzo del petrolio).

Attraverso l'uso di specifici programmi non è necessario pertanto avere un addetto che monitori ogni giorno la vendita dei biglietti e l'andamento del mercato. Le analisi prescrittive spesso non vengono adottate dalle aziende perché risultano complesse da gestire dal punto di vista amministrativo. Ma, se implementate nel

giusto modo, possono indurre verso il successo. Un altro limite dell'analisi prescrittiva riguarda la validità temporale dei risultati, per tale motivo questi ultimi devono dare abbastanza tempo per agire, altrimenti sarebbero poco utili ai fini esecutivi. Pertanto è bene che le regole create dal modello siano semplici e di facile interpretazione, così il tempo per agire risulterà breve, permettendo agli organi decisionali di rientrare nel periodo di validità della predizione.

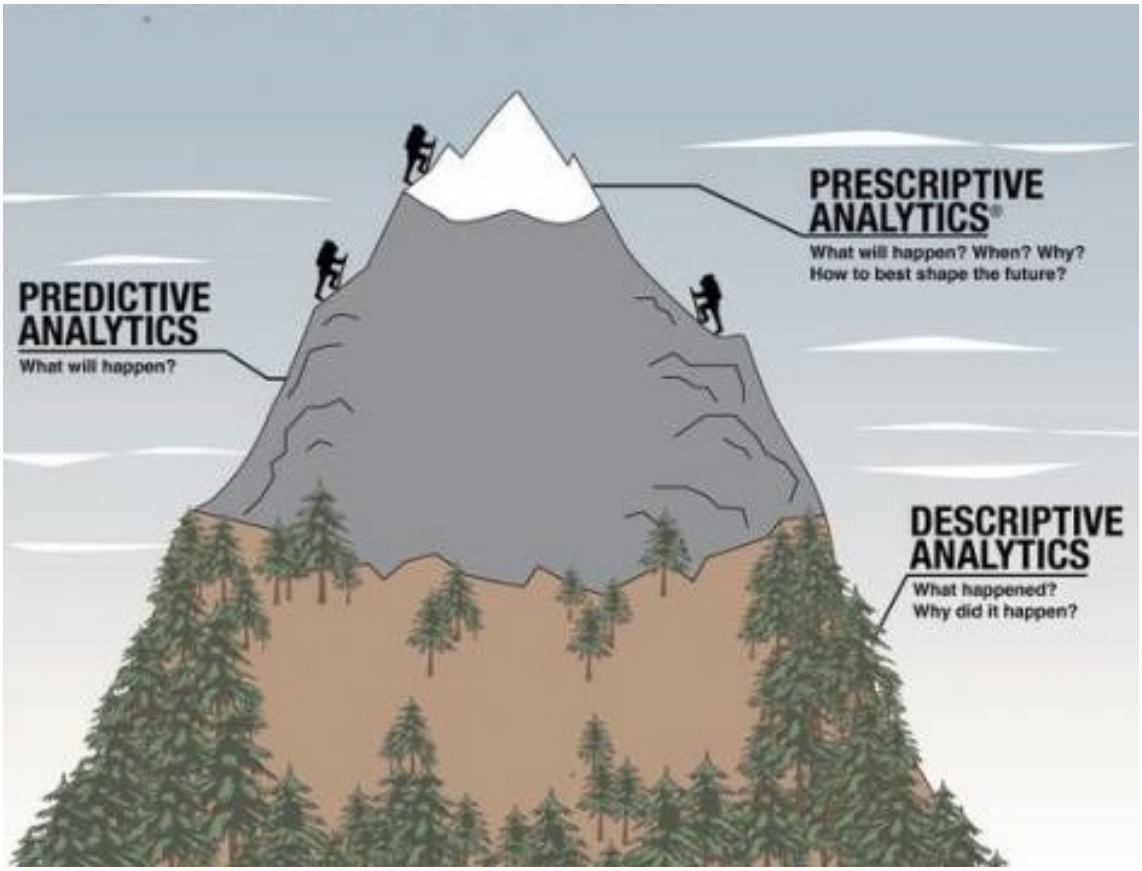

Figura 1.2. Rappresentazione delle difficoltà delle analisi

Capitolo 2

Storia e Stato dell'arte

L'uso di Analytics da parte delle imprese può essere trovato nel XIX secolo, quando Frederick Winslow Taylor iniziò la ricerca sui metodi per il miglioramento dell'efficienza nella produzione, da cui deriva il termine *Taylorismo*. Si fa riferimento a *Business Intelligence* (BI) nel lontano 1868 quando Richard Millar Devens descrisse il modo in cui un banchiere, *Sir Henry Furnese*, riuscì a comprendere la situazione economica, politica e del mercato prima dei suoi concorrenti *"attraverso l'Olanda, le Fiandre, la Francia e la Germania creò una perfetta organizzazione di business intelligence*, scrive Devens, pertanto *"le notizie...furono ricevute da lui per primo"*[1].

Importanti sviluppi avvennero nella metà del XX secolo, quando Hans Peter Luhn descrisse *"un sistema automatico...sviluppato per diffondere informazioni alle varie sezioni di un'organizzazione industriale, scientifica o di governo"*[2].

Nel 1956 IBM inventò l'hard-disk che rappresentò una rivoluzione per il salvataggio dei dati. Questo portò alla creazione dei primi Database Management Systems (DBMS).

Grazie ai DBMS le aziende iniziarono a memorizzare i dati prodotti dalle attività di business in sorgenti operazionali. Per ottenere informazioni utili dai dati si era posto necessario fare il reporting degli stessi. Le applicazioni , tuttavia, erano separate in base ai settori d'appartenenza ed avevano quindi i dati memorizzati in settori diversi, ognuno completamente diverso dall'altro. Vi era l'esigenza, però, di eseguire i report in un'unica versione dei dati, nacque perciò il problema dell'integrazione dei dati.

Nei primi anni '80 Ralph Kimball e Bill Inmon individuarono la soluzione all'integrazione nel *Data Warehouse* (DWH), struttura che memorizza i dati proveniente da sorgenti diverse. Nonostante gli approcci di Kimball e Inmon erano differenti, l'idea alla base era la medesima. Grazie al DWH le aziende furono in grado di superare l'architettura a silos favorendo una soluzione con dati integrati, non volatili, variabili nel tempo.

Nei primi anni '90 si diffuse su larga scala il termine BI. Durante questo periodo aveva due funzioni: organizzazione di dati e presentazione. La tecnologia usata, all'epoca, risultava però molto complessa. Gli utenti finali non riuscivano ad utilizzarla poichè era stata pensata per utenti esperti. Col passare degli anni sono stati sviluppati software anche per utenti meno esperti, ma il cambiamento avvenne molto lentamente.

Con l'ingresso del nuovo secolo iniziarono ad uscire sul mercato software sempre più intuitivi. I nuovi strumenti erano in grado di elaborare dati real-time consentendo di avere dati sempre aggiornati. Si iniziò a parlare di BI 2.0.

Con l'avvento dei social network quali Facebook, Twitter gli utenti avevano un nuovo modo per condividere le idee e le proprie opinioni. Questa interconnessione portò le compagnie ad avere la necessità di informazioni in tempo reale. Per rimanere competitivi la BI era diventata d'obbligo.

Gli ultimi anni hanno visto sempre più diffondersi l'idea di *Cloud*, delocalizzando quindi il software su sistemi esterni e rendendo il dato di più facile accesso rendono la *BI as-a-service*.

Nell'era moderna esiste una quantità di dati enorme, basta pensare ad esempio ad un supermercato che giornalmente produce migliaia di scontrini. La tecnica di ricavare informazioni utili dall'enorme quantità di dati presente tutt'oggi viene chiamata *Data Mining*, letteralmente dall'inglese *estrazione di dati* in riferimento alle miniere.

Il data mining perciò è diventato il passo successivo alla BI. Serve ad estrapolare informazioni nascoste e non visibile dai dati. Esempi, ormai in voga, sono la *Market Basket Analysis*, letteralemente analisi del paniere, dove vengono studiate le associazioni tra prodotti acquistati insieme; i *Recommendation System*, sistemi che sono in grado di raccomandare un prodotto che potrebbe piacere all'utente, ormai in uso su tutti i sistemi di eCommerce e di servizio. Sul fronte *Big Data* sempre più sono i Database non-relazionali, esempi sono database orientati agli oggetti come *MongoDB* oppure *Graph Database* come Neo4j.

2.1 Data Warehouse

Un DataWarehouse (DWH) è una collezione o aggregazione di dati strutturati, proveniente da dati interni ed esterni,utili all'analisi ed al reporting. E' importante che il DWH sia progettato bene per la riuscita dei processi decisionali, particolare importanza va data all'integrazione di esso con la totalità dei dati collezionati dalle diverse applicazioni, standard differenti devono essere ricodificati. Esistono due approcci per la creazioni di un DWH: il primo si basa sulla creazione di un solo archivio centralizzato, che raccoglie tutte le informazioni aziendali; il secondo invece unisce in una solo struttura diversi database chiamati *datamart*, scollegati tra loro.

L'approccio centralizzato ha il vantaggio di consentire un controllo costante sulla qualità del dato ma richiede una progettazione più attenta. I dati sono caricati sul DWH tramite un processo di estrazione, trasformazione e caricamento del dato con strumenti chiamati ETL (Extraction, Trasformation and Loading). I dati vengono poi analizzati tramite strumenti di Business Intelligence (BI).

ETL L'ETL è un processo che permette l'estrazione di dati da disparate sorgenti, tipicamente ERP (Enterprise resource planning) e/o CRM (Customer relationship management) che generalmente sono il core delle aziende, la trasformazione del dato, come la traduzione di dati codificati o semplicemente la selezione di dati utili, e il caricamento del dato finale su tabelle utili poi alle analisi. Esistono molti software di ETL, enterprise e non.

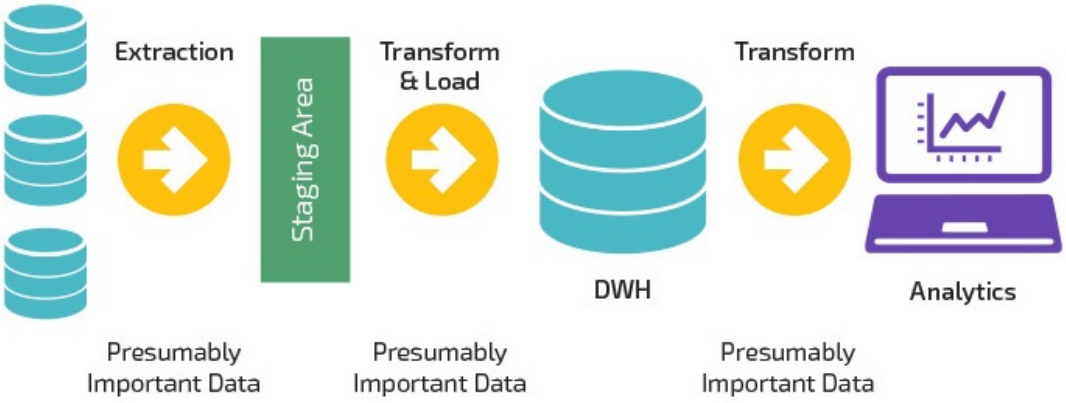

Figura 2.1. Rappresentazione schematica di un DWH

Capitolo 3

Caso di Studio

Il progetto è incentrato sulla creazione del DWH per un cliente della grande distribuzione organizzata, attiva nel Sud Italia, principalmente Campania e Basilicata. Il cliente è una holding distribuita su 4 brand riunita in unico gruppo commerciale; conta oltre 200 punti vendita ed è impegnata su business di tipo diverso.

I quattro brand differenziano per format distributivo, ovvero per differenza caratteristica.

- Un brand riunisce tutti i punti vendita di tipo iper e maxi, da 1000 mq in su. E' il supermercato completo di tutto, dove tipicamente la il consumatore si reca per effettuare una grande spesa

- Un altro brand è caratterizzato da punti vendità di prossimità, con gli store che vanno fino a 800 mq. E' il tipico supermercato "sotto casa" per la spesa di tutti i giorni

- Il terzo brand punta ai profotti formato convenienza. E' il classico discount per le persone attente alla convenienza

- L'ultimo brand è legato al settore professional con la presenza di cash&carry, tra i 2000 e 3000 mq. Si propone come negozio professionale

Il cliente, avendo ormai un DWH obsoleto, ormai troppo lento e suddiviso su datamart diversi, aveva difficoltà ad effettuare analisi del dato proveniente dai diversi datamart. La richiesta principale perciò è stata la richiesta di poter effettuare delle analisi, tramite la reportistica classica, che gli permettesse di mettere insieme i dati delle aree diverse. Dopo un approfondito studio insieme al cliente dei dati disponibili, si è proposto un modello del dato strutturato a star schema e un framework per il caricamento dei dati sul DWH. I dati a disposizione del cliente sono molteplici, per quanto riguarda i fatti abbiamo:

- Vendite
- Trasferimenti di merce da un magazzino ad un altro
- Inefficienze
- Acquisti (fatti dal gruppo al fornitore)
- Giacenze
- Tempi di preparazione
- Budget
- Listini
- Listini concorrenti
- Vendite concorrenti

Le principali dimensioni a disposizione invece sono:

- PDV (Punto di Vendita)
- Cliente Consumer (cliente finale)
- Cliente Cash (cliente con partita iva)
- Azienda
- Prodotto
- Assortimento (tipologia di assortimento: gruppi di prodotto)
- Casse
- Fornitori
- Magazzini
- Operatore
- Promozioni
- Scenario
- In più tutte le dimensioni create e pensate successivamente per la gestione del dato

La Fact principale è stata strutturata in modo da contenere più mondi aziendali riconosciuti grazie ad una dimensione pensata e creata insieme al cliente chiamata *Classe Aggregata* grazie al quale è stato possibile soddisfare la richiesta del cliente. Grazie a questo si è superato il limite importante che aveva il cliente con la vecchia struttura dati.

Il datamart in questione, chiamato *Vendite e Trasferimenti*, racchiude le informazioni di Vendita, Trasferimenti e Inefficienze. Gli altri datamart sviluppati sono:

- Acquisti: racchiude gli acquisti effettuati. Nel datamart sono presenti anche i dati dei viaggi con le informazioni delle prenotazioni, orari di arrivi, tempo impiegato per lo scarico sulla piattaforma (di magazzino)
- Giacenze: racchiude le informazioni delle giacenze sia di magazzino, sia dei singoli punti vendita
- Tempi di preparazione: è un datamart di controllo sulla qualità di preparazione dei colli/pallet della merce, indicante le informazioni su tutta la preparazione.
- Listini (factless): racchiude tutti i prezzi storicizzati nel tempo dei prodotti con in più l'informazione dei prezzi dei concorrenti di punto vendita (l'azienda acquista i dati dei prezzi dei negozi presenti ad esempio sulla stessa strada)
- IRI: contiene tutte le informazioni delle vendite dei concorrenti con l'associazione ai prodotti interni in modo da fare verifiche sul posizionamento di mercato

Il cliente chiedeva la possibilità di effettuare, su tutti i datamart disponibili, analisi del dato fino al prodotto. Tutti le Fact perciò sono al massimo dettaglio. Due delle analisi su cui il cliente puntava sono:

- il cosidetto *Bilancino di reparto*: dove il cliente calcola una KPI tramite i dati di vendita (Fatturato/Margine) e i dati delle giacenze
- la *supply-chain*: ovvero una dashboard dove poter vedere i dati dall'acquistato, dati del venduto, inefficienze, giacenza fino al cliente finale

Una volta creato e sviluppato tutto il DWH, dopo più di un anno di lavoro, e con soddisfazione del cliente, si è lavorato alla creazione per un'analisi del dato più avanzato. Questo nuovo oggetto di analisi si è soffermato sul solo datamart contenente le transazioni di vendita e dell'acquirente. La definizione della strategia di analisi è stata guidata dalla natura dei dati disponibili ed ha cercato di intercettare le necessità aziendali. Come primo approccio lavorativo infatti ci si è soffermati su un solo punto di vendita e sugli obiettivi da raggiungere: ottimizzazione delle

promozioni, valutazione dell'effetto delle promozioni sulle vendite. La prima analisi svolta è stata l'analisi delle regole di associazione tra le merci acquistate. Oltre alle misure classiche di valutazione delle regole, ovvero support, confidence e lift, si è cercato una relazione tra gli elementi nelle singole regole che incontrasse più da vicino l'esigenza aziendale. Da questo è nato l'insight *Indici di Trascinamento*, una misura di quanto, in termini di fatturato, l'antecedente influenzasse il conseguente. Allo stesso modo, si è svolta l'analisi opposta, ovvero regole di associazione negative ed allo stesso modo si è creato l'insight *Indice di Cannibalizzazione*. I due indici creati sono stati di grande supporto nella valutazione di ciascuna regola, grazie anche alla visualizzazione dei risultati in ambiente interattivo di Data Visualization che ha permesso un'analisi più veloce e agevolata dei dati. Una volta individuati *quali* prodotti posso incrementare il fatturato tramite Cross-Selling, ci si è soffermati sul *chi* tramiti analisi di categorizzazione del cliente. Infine si è svolta un'analisi combinata delle analisi precedenti. Questo tipo di analisi si vanno ad inserire in un contesto già maturo di Business Intelligence aziendale, con l'obiettivo di dare uno strumento in più all'operation aziendale.

3.1 DataWarehouse

Un Data Warehouse viene definito come una collezione di metodi, tecnologie e strumenti di ausilio aziendali per condurre analisi dei dati finalizzate all'attuazione di processi decisionali e al miglioramento del patrimonio informativo [3].

Le caratteristiche architetturali irrinunciabili per un sistema di Data Warehousing sono:

- divisione: l'elaborazione analitica e quella transazionale devono essere mantenute il più possibile separate
- scalabilità: l'architettura hardware e software deve poter essere facilmente ridimensionata a fronte della crescita nel tempo dei volumi di dati da gestire ed elaborare e del numero di utenti da soddisfare
- estendibilità: deve essere possibile accogliere nuove applicazioni e tecnologie senza riprogettare integralmente il sistema
- sicurezza: il controllo sugli accessi è essenziale a causa della natura strategica dei dati memorizzati
- fruibilità: la complessità dell'attività di amministrazione non deve risultare eccessiva

Un DWH inoltre deve essere orientato ai soggetti di interesse, integrato e consistente e rappresentativo dell'evoluzione temporale e non volatile.

Come già detto in precedenza lo scopo della creazione di un DWH è principalmente l'integrazione dei dati sulla base di un modello standard fornito dall'azienda, oltre questo deve dare accesso ai dati ad utenti con conoscenze limitate circa la struttura del dato. Deve essere sviluppato in modo di sintetizzare i dati e permettere analisi mirate e efficaci.

3.1.1 Analisi dei requisiti

La fase di analisi dei requisiti ha l'obiettivo di raccogliere le esigenze di utilizzo del datawarehouse espresse dai suoi utenti finali. È una delle fasi più importanti e delicate nel processo di creazione e progettazione ed è il fulcro per la realizzazione di tale strumento volto a migliorare il business aziendale.

L'analisi ha un'importanza strategica poiché influenza le decisioni da prendere riguardo:

- lo schema concettuale dei dati
- il progetto dell'alimentazione
- le specifiche delle applicazioni per l'analisi dei dati
- l'architettura del sistema
- il piano di avviamento e formazione
- le linee guida per la manutenzione e l'evoluzione del sistema

Per il clienti, scelte imprescindibili dettate dal business, sono state utilizzare il prodotto *Cognos Analytics* come front-end aziendale. Questo perchè l'azienda utilizzava già il seguente prodotto nelle sue versioni precedenti, perciò gli utenti erano già istruiti. Altro punto obbligato, dettata dal cliente, è stata la scelta del database che ricadeva su un prodotto microsoft, il database SQL Server. Considerando tale scelta, conoscendo i punti di forza e i punti deboli del prodotto, abbiamo suggerito al cliente una DWH a star-schema. Analizzando il numero di righe, per le sole fact da alimentare giornalmente:

SEZIONE	NUMERO DI RIGHE
Vendite/Trasferimenti/Inefficienze	500k
Acquisti	??
Giacenze	2M

- per la sezione di Vendite/Trasferimenti/Inefficienze si deduce direttamente la porzione giornaliera dalla tabella sorgente

- per gli Acquisti non si è riuscito a ricavare una porzione giornaliera, il cliente non è riuscito ad identificare i campi costituenti la chiave e perciò si è previsto un caricamento full di 3 mesi. L'ammontare delle righe da caricare giornalmente è 10M

- per le Giacenze viene calcolata la porzione di delta tramite differenza di dati scaricati in giorni diversi

Un'altra richiesta del cliente è stata di avere fino a 5 anni in linea di dati: questo a portato ad avere una quantità di dati enorme: per la sola fact delle vendite si raggiunge 1 miliardo, passando sulle giacenze dove si sfiorano i 2 miliardi di righe.

Analizzando l'analisi da loro effettuata più pesante, ovvero l'analisi della supply-chain, ipotizzando il controllo su 2 anni (anno corrente + anno precedente), la query ipotizzata leggeva sull'ordine di 10M di righe.

Considerando l'ammontare dei dati e le query pesanti richieste dal cliente si è richiesto un database:

- con un disco, possibilmente SSD per questioni di performance, di oltre 1 TB

- un disco dedicato al *tempdb*, utilizzato da SQL Server per temporizzare le query, di oltre 300 GB

- con almeno 256 GB di RAM

- con una CPU che avesse almeno 16 core

- con un collegamento verso la macchina di reporting di almeno 1Gbps

- con un collegamento verso la macchina di ETL di almeno 10Gbps

Altro punto cardine fondamentale è il caricamento dei dati sul DWH. Per la progettazione dell'ETL, per avere una continuità di prodotto con la BI, si è scelto sempre un prodotto IBM, ovvero il tool *Datastage*. Le richieste delle macchine sono state fatte al cliente seguendo le specifiche di prodotto: si è scelto la configurazione su due macchine Engine+WebServer.

Anche per quanto riguarda la macchina su cui installare il tool di reporting si è seguito la specifica di prodotto: la macchina doveva avere molta RAM, considerando i dati (oltre 256 di RAM), perchè il cliente ha scelto anche l'utilizzo dei cubi OLAP.

3.1.2 ETL

Il ruolo degli strumenti di Extraction, Transformation and Loading è quello di alimentare una sorgente dati singola o multipla, dettagliata, esauriente e di alta qualità che possa a sua volta alimentare il DWH.

Durante il processo di alimentazione si possono distinguere 4 diverse fasi:

- estrazione
- pulitura
- trasformazione
- caricamento

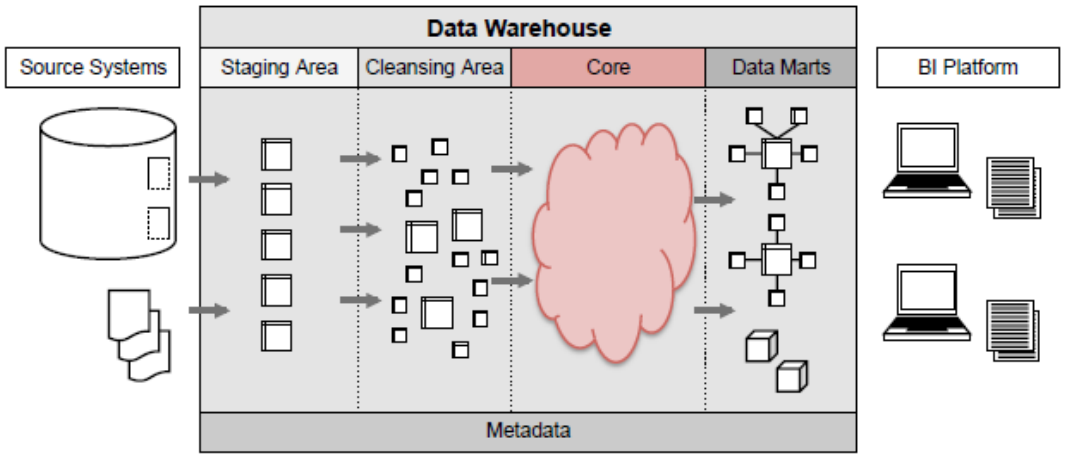

Figura 3.1. Schema Architecture

Estrazione

Durante la fase di estrazione i dati rilevanti vengono estratti dalle sorgenti. Esistono due diverse fasi di caricamento, ovvero:

- Initial Load - L'estrazione statica viene effettuata quando il DWH deve essere popolato per la prima volta e consiste concettualmente in una fotografia dei dati operazionali.
- CDC - (Change Data Capture) L'estrazione incrementale viene usata per l'aggiornamento periodico del DWH, e cattura solamente i cambiamenti avvenuti

nelle sorgenti dall'ultima estrazione. Il CDC può essere basato sul log del Database operazionale oppure basata su un campo di tipo timestamp che seleziona le righe opportune

Nel caso in cui il CDC non è attuabile, si può pensare di effettuare un delta tra le immagini periodiche di scarico.

Nel caso del cliente sono state utilizzate entrambe le soluzioni, in base alle differenti sorgenti da scaricare. Un'altra soluzione effettuata solo sul cliente è un "Initial Load" filtrato per data, esempio scaricare gli ultimi 3 mesi. Questa soluzione è stata una forzatura in quanto il cliente non era in grado di indicarci i campi chiave e perchè i dati erano in continuo movimento fino a tot giorni indietro.

Pulitura

La pulitura è necessaria in quanto i dati letti molte volte non sono congrui e presentano "rumore". Un esempio di problemi che possono verificarsi sono:

- dati duplicati
- inconsistenza tra valori logicamente associati
- dati mancanti
- uso non previsto di un campo
- valori impossibili o errati
- valori inconsistenti per la stessa entità dovuti a differenti convenzioni
- valori inconsistenti per la stessa entità dovuti a errori di battitura

Nella soluzione sul cliente, per risolvere le problematiche elencate, si è proceduto nel seguente modo:

- i dati duplicati verranno direttamente scartati grazie all'impostazione della chiave su database
- su tutte le inconsistenze chiave/valore, si è previsto la gestione di una tabella, con in chiave solo il codice, per la gestione di descrizioni diverse associate a codici uguali
- di utilizzare dei codici di default per "riempire" tutti i valori, considerati importanti per il business, che mancavano. Questo è stato fatto non appena i dati vengono scaricati dalla sorgente.
- le righe con valori impossibili o errati o con valori inconsistenti finiscono in delle tabelle di scarto e viene inviata segnalazione al cliente circa le righe scartate

Trasformazione

Converte i dati dal formato operazionale sorgente a quello del DWH. La corrispondenza con il livello sorgente è complicata dalla presenza di fonti distinte eterogenee, che richiede una complessa fase di integrazione. E' necessario perciò una fase in cui i dati vengano riconciliati tra loro: sono sottoposti quindi ad una fase di conversione e normalizzazione. Nel caso in cui tra le diversi sorgenti ci fossero codici identificativi diversi è necessaria una fase di matching, in cui viene effettuata la corrispondenza tra campi equivalenti di sorgenti diverse. Inoltre nella fase di Trasformazione avviene anche una fase di selezione, in cui tutti i record non necessari alle analisi vengono filtrati.

Sul cliente una prima fase effettuata è stato convertire tutti i campi di tipo testo: l'ERP utilizzato dal cliente infatti "riempie" tutti i campi con degli spazi fino alla totale occupazione del campo. La conversione effettuata è stata perciò convertire i campi di tipo *char* in campi di tipo *varchar*. La fase di matching tra diverse sorgenti è stata risolta inserendo una tabella di trascodifica. Tale tabella viene caricata dal cliente, con un file excel, tramite una web application sviluppata per facilitarne il lavoro. Altre trasformazioni rese necessarie sono state concatenazioni di campi, substring di alcuni campi e così via.

3.1.3 Metadati

Nella costruzione del modello di alimentazione per il DWH, per gestire i flussi di caricamento, abbiamo creato un framework che gestisce l'intero flusso di caricamento. Il framework sviluppato si basa su un modello construito su delle tabelle di 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. 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

- i metadati interni, di interesse per l'amministratore, descrivono, tra le altre cose, le sorgenti, le trasformazioni, le politiche di alimentazione, gli schemi logici e fisici, i vincoli e i profili degli utenti

- i 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 relativo 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 propri per ogni sistema. Le tabelle di metadati di processo possono esser differenti tra loro e sono fortemente legate alla tecnologia e al sistema che descrivono.

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. Molto importante è gestire le situazioni di sincronia di estrazione: ovvero il processo può partire o meno? Ci sono altre istanze dello stesso processo RUNNING o terminate in errore? Quanti processi possono esser eseguiti in una giornata? Il processo di ieri è terminato il giorno successivo. Come gestisco il lancio nel giorno successivo?

Nel caso di studio sono state previste tre tabelle di metadati:

- FLOW_AMANAGER: contiene le informazioni sul flusso, orario di partenza e fine, stato, etc.

- TABLE_MANAGER: contiene tutte le informazioni sulle taballe, orario di scarico, righe scaricate etc.

- ERROR_MANAGER: contiene le informazioni sulle righe in reject

Vedremo come sono usate queste tabelle successivamente.

Cosa descrive un metadato

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, come accennavo per motivi di fruibilità da parte dei più disparati 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 piu' 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. Nella nostro progetto di analisi prendiamo in considerazione come esempio una tabella di metadati che descrive lo STATO dei processi ovvero ogni informazione che possiamo trovare dentro la tabella descrive proprio in quel momento in che situazione è il nostro processo. Governando il carico del DWH con delle logiche di priorità possiamo gestire la sincronizzazione di tutti i processi.

Prima di ciò è necessario anticipare che il framework sviluppato è suddiviso in 3 livelli:

- Livello 0: area di staging, vanno a finire tutti i dati giornalieri scaricati
- Livello 1: area normalizzata, contiene tutti i dati storici normalizzati
- Livello 2: area di pubblicazione, è l'area da cui legge il front-end

uno seguente al successivo che verranno descritti in dettaglio successivamente.

IDENTITY	GRP_NAME	LVL	STATUS	SRC_J	JOBID	DATE_TO
ANAGRAFICHE	GRP_PRODOTTO					
ANAGRAFICHE	GRP_PDV					
ANAGRAFICHE	GRP_FORNITORI					
MOVIMENTI	VENDITE					
MOVIMENTI	ACQUISTI					

Durante la giornata fisseremo degli intervalli temporali, ad esempio: 20200517090000 con formato "YYYYMMDDHH24MISS"

Replicheremo le tuple della tabella per ogni intervallo temporale partizionando la tabella per JOBID opzionalmente, in modo da avere una situazione chiara per ogni estrazione di dove e cosa sta girando e cosa si può eventualmente richiamare. Le attività saranno forzate con delle dipendenze ferree di cui farò esempi successivamente.

La colonna DATE_TO è utilizzata per inserire la data di estrazione dei dati sul sorgente preso in considerazione per poi poter dare alla fine del carico del DWH la data di sincronia dei sistemi sorgenti caricati.

Di seguito sono esplicitati gli step e alcuni dettagli importanti su di essi per la gestione di un processo reale:

- ogni step di anagrafica può procedere indipendentemente fino all'L1 a meno che non ci siano altri processi attivi dello stesso tipo in quanto le anagrafiche sono dati di stato che servono a ogni altro processo per far lookup quando occorre caricare i dati in L1 e sono solitamente autoconsistenti, ovvero non dipendono da altri dati se non dal termine di ogni altro processo che le legga

- ogni step di L0 può essere eseguito parallelamente ad altri step dello stesso livello
- per far partire il livello L1 ogni step dovrà controllare di esser l'unico in esecuzione
- per caricare L2 tutti gli step di L1 dovranno essere terminati.
- e' possibile caricare e consolidare i dati con piu' iterazioni L0 prima di lanciare un carico cumulativo di L1 e L2 che dovremo concordare a seconda del cliente se sarà un carico di L1/L2 cumulativo o potremmo procedere con un JOBID da considerare alla volta (a seconda delle richieste di storicizzazione)
- si può invocare il lancio di più processi anche scorporati che debbano rispettare le sequenze temporali delle regole che definiremo (es. mentre carichiamo il flusso intero di anagrafiche possiamo lanciare le vendite ma fino al livello di estrazione)

Legenda Per avere dei dati più compatti sulla stato di un processo nella tabella di metadati useremo un codice numerico, di seguito la legenda che indica il significato dello stato

STATUS	VALUE	INFO
DONE	0	Job completed
RUNNING	1	Job running
ERROR	-3	Job error
TO_LOAD	2	Job ready to load

Case of use Procediamo a dare un esempio della tabella di metadati ed alcuni casi d'uso. Le varie *Identity*, che indicano l'area di lavoro (es. Anagrafica, Vendite), sono suddivise in step (o gruppi). Ogni gruppo rappresenta una porzione di lavoro consistente, ovvero può girare in modo indipendenti rispetto agli altri gruppi. Ad esempio il gruppo *GRP_PRODOTTO* scarica tutte le tabelle associate al prodotto:

- Prodotto
- Ean
- Assreteart (informazioni sugli assortimenti)
- Anartricla (Informazioni sulla riclassificazione dei prodotti)
- Promo iniz (Informazioni sulle promozioni dei prodotti)

- ...

Lo step va a buon fine, se e solo se, tutti i job contenuti nello step vanno a buon fine; se uno solo va in errore, anche tutto lo step va in errore.

In questa casistica l'estrazione parallela degli step di anagrafica, livello 0, è in esecuzione:

IDENTITY	GRP_NAME	LVL	STATUS	SRC_J	JOBID	DATE_TO
ANAG	GRP_PRODOTTO	0	1	19000101000000	20200524120000	20200523155600
ANAG	GRP_PDV	0	1	19000101000000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	0	1	19000101000000	20200524120000	20200523155600

Appena termina il livello 0 delle anagrafiche, è possibile far partire il livello 1, che caricherà tutti i livelli 0 con STATUS = 2:

IDENTITY	GRP_NAME	LVL	STATUS	SRC_J	JOBID	DATE_TO
ANAG	GRP_PRODOTTO	1	1	20200524120000	20200524120000	20200523155600
ANAG	GRP_PDV	1	1	20200524120000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	1	1	20200524120000	20200524120000	20200523155600
ANAG	GRP_PRODOTTO	0	2	19000101000000	20200524120000	20200523155600
ANAG	GRP_PDV	0	2	19000101000000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	0	2	19000101000000	20200524120000	20200523155600

Qui si vede che il livello 1, che è in fase di caricamento, ha in SRC_J il JOBID caricato precedentemente dal livello 0; infatti qui il livello 1 sta caricando i dati che sono stati scaricati dal livello 0. Quando le anagrafiche saranno tutte caricate, si avrà una situazione in cui tutti i livelli sono in stato DONE:

IDENTITY	GRP_NAME	LVL	STATUS	SRC_J	JOBID	DATE_TO
ANAG	GRP_PRODOTTO	2	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_PDV	2	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	2	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_PRODOTTO	1	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_PDV	1	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	1	0	20200524120000	20200524120000	20200523155600
ANAG	GRP_PRODOTTO	0	0	19000101000000	20200524120000	20200523155600
ANAG	GRP_PDV	0	0	19000101000000	20200524120000	20200523155600
ANAG	GRP_FORNITORI	0	0	19000101000000	20200524120000	20200523155600

La stessa logica di caricamento viene applicata anche al caricamento dei MOVIMENTI.

3.2 L0 - Staging Area

Procediamo adesso a descrivere nel dettaglio i livelli precedentemente introdotti.

Il livello L0 rappresenta la fase iniziale o di staging del DWH dove avviene lo scarico delle informazioni dai sistemi sorgenti. I sistemi sorgente possono esser di diverso genere, i più comuni sono i sistemi operazionali su database o i file prodotti dal fornitore. Si prevedono 2 tipologia di lettura:

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

Per facilitare il riconoscimento dei file e l'archiviazione per eventuali necessità di ricarico utilizzeremo una nomenclatura ferrea che renderà più semplice il riconoscimento del singolo file. Tali file dovranno poi risiedere in un'alberatura che consente la navigazione storica e concettuale all'interno delle cartelle. Durante il processo i file verranno spostati nella cartella da caricare e una volta letti spostati nella cartella con funzione di archivio. Un esempio di nomenclatura file potrebbe essere: <nome_sistema>_<data_estrazioneYYYYMMDDHH24MISSS>.<estensione_file>.

Una possibile alberatura delle cartelle potrebbe essere:

- folder IN: file da caricare
- fodler RUN: file in corso di lettura
- folder OUT: file già letti che vengono archiviati (si può pensare a dividerli in sottocartelle distinte per mese o anno)
- fodler LOG: file di log del processo di loading
- folder BAD: file scartati per errori di lettura o generali

3.2.1 Modalità di accesso ai dati

I processi di elaborazione per il caricamento della Staging Area si occupano di verificare la disponibilità dei dati dai sistemi alimentanti. I dati da scaricare possono essere suddivisi in entità differenti:

- per sistema alimentante (i dati vengono caricati quando tutti i flussi provenienti da un sistema sono allineati)
- per entità funzionale (i dati vengono caricati quando tutti i flussi relativi ad una entità funzionale sono allineati)
- per flusso dati (i dati vengono caricati appena disponibili, per singolo flusso)

Per gestire i processi di scarico verrà utilizzata una tabella di metadati, chiamata FLOW_MANAGER, ubicata su un database possibilmente raggiungibile da tutti i sistemi.

Esistono due modalità di scarico del dato:

- MODALITA' SINCRONA

  - si registra il JOB_ID di scarico dei dati dai sistemi sorgenti in modo da avere la contestualizzazione temporale della replica dei dati
  - si differenzia un JOB_ID per ogni sistema se non sono allineati
  - se i sistemi sono allineati abbiamo un JOB_ID unico di scarico

- MODALITA' ASINCRONA
  - si attende la creazione di un file tappo o di un aggiornamento di un metadato per far partire la replica (tipica modalità di replica utilizzata quando i sistemi alimentanti sono su file)
  - start dei processi condizionale
  - considerare se far partire la replica giornaliera di tutti i sistemi o solo di alcuni (verificare la fattibilità)

Inoltre vengono seguite le seguenti regole:

- lettura dei dati senza trasformazione dei data type per preservare l'informazione senza alterarne il contenuto
- inserimento dei dati in tabelle DLT (delta) che hanno esattamente la stessa struttura dei sorgenti con una colonna in più contenente la data di estrazione a cui fanno riferimenti i dati
- i dati devono esser scaricati senza esser messi in join con niente e senza applicare nessun tipo di trasformazione
- le tabelle DLT essendo temporizzate devono esser accedibili in modo ottimizzato per la lettura della fetta temporale da considerare nel carico

Nel caso in cui non sia possibile effettuare il delta direttamente dal sistema sorgente, le tabelle verranno scaricate in modalità FULL, ovvero tutti i giorni tutti i dati, e verranno inseriti in delle tabelle STG. Da queste STG, tramite il software di ETL verranno effettuati dei delta a partire dalle due partizioni temporali. Questi delta calcolati andranno inseriti nelle medesime tabelle DLT descritte sopra.

Nel nostro ETL, tutte le anagrafiche seguono il primo approccio, ovvero scarico tutti i giorni in modalità full. Per il controllo di questa modalità si è creata la necessità di avere una tabella di metadati, chiamata *TABLE_MANAGER*, dove si indica per JOBID e nome tabella quante righe sono state caricate. In questa tabella, ho anche la profondità storica presenta sulla tabella; nel nostro caso teniamo sempre una storia di 7 giorni (vedere sezione 3.2.1). Vediamo un esempio di dato per la tabella Prodotto. Avrò una situazione simile:

JOBID	TABLE_NAME	CNT_ROWS
20200624023004	STG_PRODOTTO	209119
20200623023002	STG_PRODOTTO	209075
20200622023002	STG_PRODOTTO	209049
20200621023003	STG_PRODOTTO	209049
20200620023002	STG_PRODOTTO	209049
20200619023002	STG_PRODOTTO	209020
20200618023003	STG_PRODOTTO	209013

Nella fig. 3.2 vediamo un esempio di calcolo effettuato su Datastage.

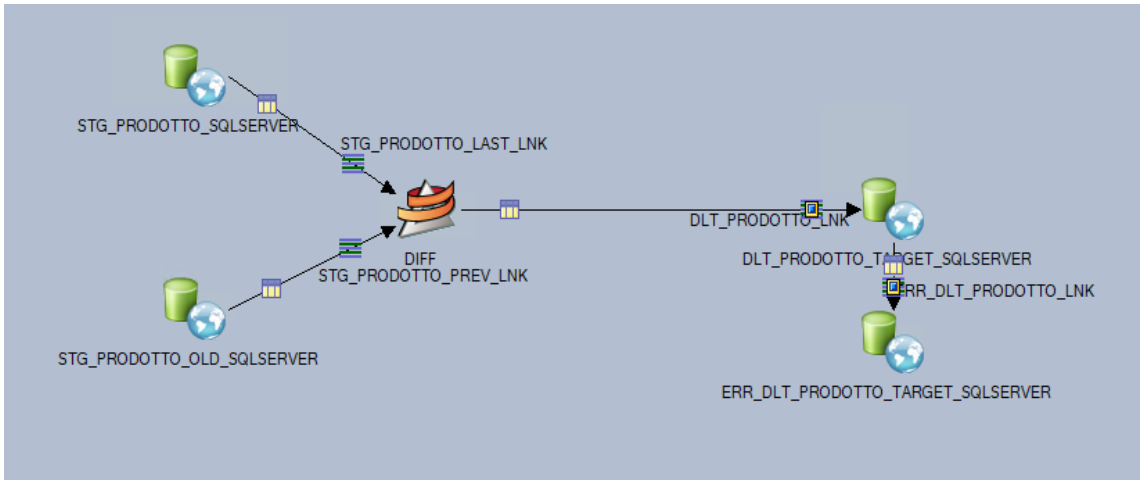

Figura 3.2. Job DLT

Da STG_PRODOTTO prendiamo l'immagine scaricata attualmente, per poterlo fare eseguiamo la query:

```
1  SELECT JOBID_LO
2       , PROD_CODE
3       , PROD_VAR_CODE
4       , PROD_DESC
5       , PROD_INS_DATE
6       , PROD_STATUS
7       , CEDI_PREV_CODE
8       , INTERCETT_CONTROPARTITA_DETT
9       , CLASSE_MERCE_SETTORE_CODE
10      , CLASSE_MERCE_SETTORE_DESC
11      , CLASSE_MERCE_REPARTO_CODE
12      , CLASSE_MERCE_REPARTO_DESC
13      , CLASSE_MERCE_CATEGORIA_CODE
14      , CLASSE_MERCE_CATEGORIA_DESC
15      , CLASSE_MERCE_SUBCATEGORIA_CODE
16      , CLASSE_MERCE_SUBCATEGORIA_DESC
17      , CLASSE_MERCE_SEGMENTO_CODE
18      , CLASSE_MERCE_SEGMENTO_DESC
19      , MARCA_CODE
20      , MARCA_DESC
21      , SOTTOMARCA_CODE
22      , SOTTOMARCA_DESC
23      , PRODUTTORE_CODE
```

```
      , PRODUTTORE_DESC
      , CATEGORY_CODE
      , CATEGORY_DESC
      , PROD_MERCATO_CODE
      , PROD_MERCATO_DESC
      , FUNZIONE_PREZZO_CODE
      , FUNZIONE_PREZZO_DESC
      , UM_CONFEZIONE_CODE
      , UM_CONFEZIONE_DESC
      , UM_PREZZO_CLIENTI_CODE
      , FUNZIONE_COMM_ARTICOLI_CODE
      , FUNZIONE_COMM_ARTICOLI_DESC
      , STAGIONE_CODE
      , STAGIONE_DESC
      , STIME_PREVISIONI_CODE
      , STIME_PREVISIONI_DESC
      , SUPPORTO_CONTENIMENTO_CODE
      , SUPPORTO_CONTENIMENTO_DESC
      , FORMATO_UNITA_VENDITA_CODE
      , FORMATO_UNITA_VENDITA_DESC
      , PESO_VARIABILE_FLG
      , PESO_NETTO_UNIT_KG
      , PESO_LORDO_UNIT_KG
      , DURATA_MINISTERIALE_GG
      , ENTRATA_MERCI_MIN_DURATA_GG
      , USCITA_MERCI_MIN_DURATA_GG
      , ALLARME1_DURATA_GG
      , ALLARME2_DURATA_GG
      , ALLARME3_DURATA_GG
      , FORN_PREV_CODE
      , FORN_PREV_VAR_CODE
      , UM_FISCALE_CODE
      , CONTENUTO_FISCALE_NETTO_QTA
      , CONTENUTO_FISCALE_LORDO_QTA
      , TRACCIABILITA_CODE
      , UM_MINISTERIALE_CODE
      , INTRASTAT_PESO_KG
      , INTRASTAT_COEFF_CONV_UM_MINISTERO
      , PROD_H_CM
      , PROD_L_CM
      , PROD_P_CM
      , TERRITORIO_CODE
      , TERRITORIO_DESC
      , MONDO_APPARTENENZA_CODE
      , MONDO_APPARTENENZA_DESC
      , PLU_CODE
      , IMPONIBILE_IVA
      , IVA_MULTIPLA_FLG
      , IMPONIBILE1_IVA
      , IMPONIBILE1_IVA_PERC
      , IMPONIBILE2_IVA
      , IMPONIBILE2_IVA_PERC
      , IMPONIBILE3_IVA
      , IMPONIBILE3_IVA_PERC
      , IMPONIBILE4_IVA
      , IMPONIBILE4_IVA_PERC
      , IMPONIBILE5_IVA
      , IMPONIBILE5_IVA_PERC
      , REPARTO_VENDITA_POS_CODE
      , REPARTO_VENDITA_POS_DESC
      , CALO_PESO_PERC
      , COEFF_CONV_NUMERATORE
      , COEFF_CONV_DENOMINATORE
      , PEZZI_X_COLLO
      , STRATI_X_U_MOVIMENTAZIONE
      , COLLI_X_STRATO
      , COLLI_X_U_MOVIMENTAZIONE
      , AZIENDA_CRIS_CODE
      , UM_VENDITA_CODE
      , KG_PESO_LORDO
      , MOD_ESPOSIZIONE
      , MOD_ESPOSIZIONE_DESC
      , ARTICOLI_DESC_BREVE
      , ORD_LARG
      , ORD_ALT
      , ORD_LUNG
      , IMBALLI_COLLO
      , COD_CLASSE_MERC
      , SETTORE_MERC
      , SETTORE_MERC_DESC
      , REPARTO_MERC
      , REPARTO_MERC_DESC
      , CATEGORIA_MERC
```

```
      , CATEGORIA_MERC_DESC
      , SUBCATEGORIA_MERC
      , SUBCATEGORIA_MERC_DESC
      , SEGMENTO_MERC
      , SEGMENTO_MERC_DESC
      , DATA_INIZ_MERC
      , DATA_FINE_MERC
      , PROGR_MERC
      , COD_CLASSE_MERC1
      , SETTORE_MERC1
      , SETTORE_MERC_DESC1
      , REPARTO_MERC1
      , REPARTO_MERC_DESC1
      , CATEGORIA_MERC1
      , CATEGORIA_MERC_DESC1
      , SUBCATEGORIA_MERC1
      , SUBCATEGORIA_MERC_DESC1
      , SEGMENTO_MERC1
      , SEGMENTO_MERC_DESC1
      , DATA_INIZ_MERC1
      , DATA_FINE_MERC1
      , PROGR_MERC1
  FROM LO.STG_PRODOTTO
  JOIN DWH.TABLE_MANAGER
    ON SCHEMA_NAME = 'LO'
   AND TABLE_NAME = 'STG_PRODOTTO'
 WHERE JOBID_LO = #P_JOBID_LO#
   AND JOBID_LO = JOBID
   AND ROW_COUNT <> 0
 ORDER BY PROD_CODE
      , PROD_VAR_CODE
```

Il valore *#P_JOBID_L0#* è una variabile e sarà sostituito da Dataastage impostando il JOBID del running in esecuzione. Si vede inoltre che verrà scartato tutto se nella tabella di TABLE_MANAGER con il JOBID e il nome della tabella si avranno 0 righe; questo perchè probabilmente ci sarà stato un errore lato ERP che non dovrà intaccare i dati (in caso contrario tutte le righe sulla dimensione Prodotto verrebbero segnate come cancellate).

Mentre da STG_PRODOTTO_OLD scarichiamo l'ultima immagine precedente a queste RUN. Di seguito la query utilizzata:

```
SELECT JOBID_LO
      , PROD_CODE
      , PROD_VAR_CODE
      , PROD_DESC
      , PROD_INS_DATE
      , PROD_STATUS
      , CEDI_PREV_CODE
      , INTERCETT_CONTROPARTITA_DETT
      , CLASSE_MERCE_SETTORE_CODE
      , CLASSE_MERCE_SETTORE_DESC
      , CLASSE_MERCE_REPARTO_CODE
      , CLASSE_MERCE_REPARTO_DESC
      , CLASSE_MERCE_CATEGORIA_CODE
      , CLASSE_MERCE_CATEGORIA_DESC
      , CLASSE_MERCE_SUBCATEGORIA_CODE
      , CLASSE_MERCE_SUBCATEGORIA_DESC
      , CLASSE_MERCE_SEGMENTO_CODE
      , CLASSE_MERCE_SEGMENTO_DESC
      , MARCA_CODE
      , MARCA_DESC
      , SOTTOMARCA_CODE
      , SOTTOMARCA_DESC
      , PRODUTTORE_CODE
      , PRODUTTORE_DESC
      , CATEGORY_CODE
      , CATEGORY_DESC
      , PROD_MERCATO_CODE
      , PROD_MERCATO_DESC
      , FUNZIONE_PREZZO_CODE
      , FUNZIONE_PREZZO_DESC
```

```
      , UM_CONFEZIONE_CODE
      , UM_CONFEZIONE_DESC
      , UM_PREZZO_CLIENTI_CODE
      , FUNZIONE_COMM_ARTICOLI_CODE
      , FUNZIONE_COMM_ARTICOLI_DESC
      , STAGIONE_CODE
      , STAGIONE_DESC
      , STIME_PREVISIONI_CODE
      , STIME_PREVISIONI_DESC
      , SUPPORTO_CONTENIMENTO_CODE
      , SUPPORTO_CONTENIMENTO_DESC
      , FORMATO_UNITA_VENDITA_CODE
      , FORMATO_UNITA_VENDITA_DESC
      , PESO_VARIABILE_FLG
      , PESO_NETTO_UNIT_KG
      , PESO_LORDO_UNIT_KG
      , DURATA_MINISTERIALE_GG
      , ENTRATA_MERCI_MIN_DURATA_GG
      , USCITA_MERCI_MIN_DURATA_GG
      , ALLARME1_DURATA_GG
      , ALLARME2_DURATA_GG
      , ALLARME3_DURATA_GG
      , FORN_PREV_CODE
      , FORN_PREV_VAR_CODE
      , UM_FISCALE_CODE
      , CONTENUTO_FISCALE_NETTO_QTA
      , CONTENUTO_FISCALE_LORDO_QTA
      , TRACCIABILITA_CODE
      , UM_MINISTERIALE_CODE
      , INTRASTAT_PESO_KG
      , INTRASTAT_COEFF_CONV_UM_MINISTERO
      , PROD_H_CM
      , PROD_L_CM
      , PROD_P_CM
      , TERRITORIO_CODE
      , TERRITORIO_DESC
      , MONDO_APPARTENENZA_CODE
      , MONDO_APPARTENENZA_DESC
      , PLU_CODE
      , IMPONIBILE_IVA
      , IVA_MULTIPLA_FLG
      , IMPONIBILE1_IVA
      , IMPONIBILE1_IVA_PERC
      , IMPONIBILE2_IVA
      , IMPONIBILE2_IVA_PERC
      , IMPONIBILE3_IVA
      , IMPONIBILE3_IVA_PERC
      , IMPONIBILE4_IVA
      , IMPONIBILE4_IVA_PERC
      , IMPONIBILE5_IVA
      , IMPONIBILE5_IVA_PERC
      , REPARTO_VENDITA_POS_CODE
      , REPARTO_VENDITA_POS_DESC
      , CALO_PESO_PERC
      , COEFF_CONV_NUMERATORE
      , COEFF_CONV_DENOMINATORE
      , PEZZI_X_COLLO
      , STRATI_X_U_MOVIMENTAZIONE
      , COLLI_X_STRATO
      , COLLI_X_U_MOVIMENTAZIONE
      , AZIENDA_CRIS_CODE
      , UM_VENDITA_CODE
      , KG_PESO_LORDO
      , MOD_ESPOSIZIONE
      , MOD_ESPOSIZIONE_DESC
      , ARTICOLI_DESC_BREVE
      , ORD_LARG
      , ORD_ALT
      , ORD_LUNG
      , IMBALLI_COLLO
      , COD_CLASSE_MERC
      , SETTORE_MERC
      , SETTORE_MERC_DESC
      , REPARTO_MERC
      , REPARTO_MERC_DESC
      , CATEGORIA_MERC
      , CATEGORIA_MERC_DESC
      , SUBCATEGORIA_MERC
      , SUBCATEGORIA_MERC_DESC
      , SEGMENTO_MERC
      , SEGMENTO_MERC_DESC
      , DATA_INIZ_MERC
      , DATA_FINE_MERC
```

```
114       , PROGR_MERC
115       , COD_CLASSE_MERC1
116       , SETTORE_MERC1
117       , SETTORE_MERC_DESC1
118       , REPARTO_MERC1
119       , REPARTO_MERC_DESC1
120       , CATEGORIA_MERC1
121       , CATEGORIA_MERC_DESC1
122       , SUBCATEGORIA_MERC1
123       , SUBCATEGORIA_MERC_DESC1
124       , SEGMENTO_MERC1
125       , SEGMENTO_MERC_DESC1
126       , DATA_INIZ_MERC1
127       , DATA_FINE_MERC1
128       , PROGR_MERC1
129    FROM L0.STG_PRODOTTO
130    JOIN DWH.TABLE_MANAGER
131      ON SCHEMA_NAME = 'L0'
132     AND TABLE_NAME = 'STG_PRODOTTO'
133   WHERE JOBID = #P_JOBID_L0#
134     AND ROW_COUNT <> 0
135     AND JOBID_L0 =
136       ( SELECT MAX(TRG_JOBID) TRG_JOBID
137           FROM DWH.FLOW_MANAGER
138          WHERE [IDENTITY] = '#PS_DWH_PROCESS_API.IDENTITY#'
139            AND [LEVEL] = #PS_DWH_PROCESS_API.LEVEL#
140            AND [GRP_NAME] = '#PS_DWH_PROCESS_API.GRP_NAME#'
141            AND [STATUS] = #PS_DWH_PROCESS_CODE.P_DONE#
142       )
143   ORDER BY PROD_CODE
144       , PROD_VAR_CODE
```

Come si nota dalla query sopra, il JOBID_L0 in questo caso è preso direttamente dalla tabella di metadati; viene preso il max jobid per gruppo che si sta eseguendo, prendendo ovviamente i soli jobid che hanno terminato correttamente.

In ogni livello/gruppo ci sono tanti job che sono eseguiti in parallelo. Un esempio di esecuzione parallela fatta in datastage è data nella fig. 3.3 dove sono presenti tutti i job del livello 0 per il gruppo PRODOTTO. Nell'esempio sono presenti i job STG che scaricano l'immagine così come è presente sull'ERP e i job di count che servono per popolare la tabella di metadati TABLE_MANAGER.

Storicizzazione

Le tabelle DLT possono avere una storicizzazione necessaria al ricalcolo del DWH causa perdita di dati o carico di dati parziali. Per aver una storicizzazione del dato abbiamo due scelte:

- creare tabelle ombra delle DLT partizionate per JOB_ID di estrazione chiamate HIS
  - DLT in truncate/insert che contengano solo i dati del JOB_ID attuale e non partizionate
  - HIS in insert con il partizionamento per JOB_ID per velocizzare le estrazioni
- avere le storico direttamente sulle DLT

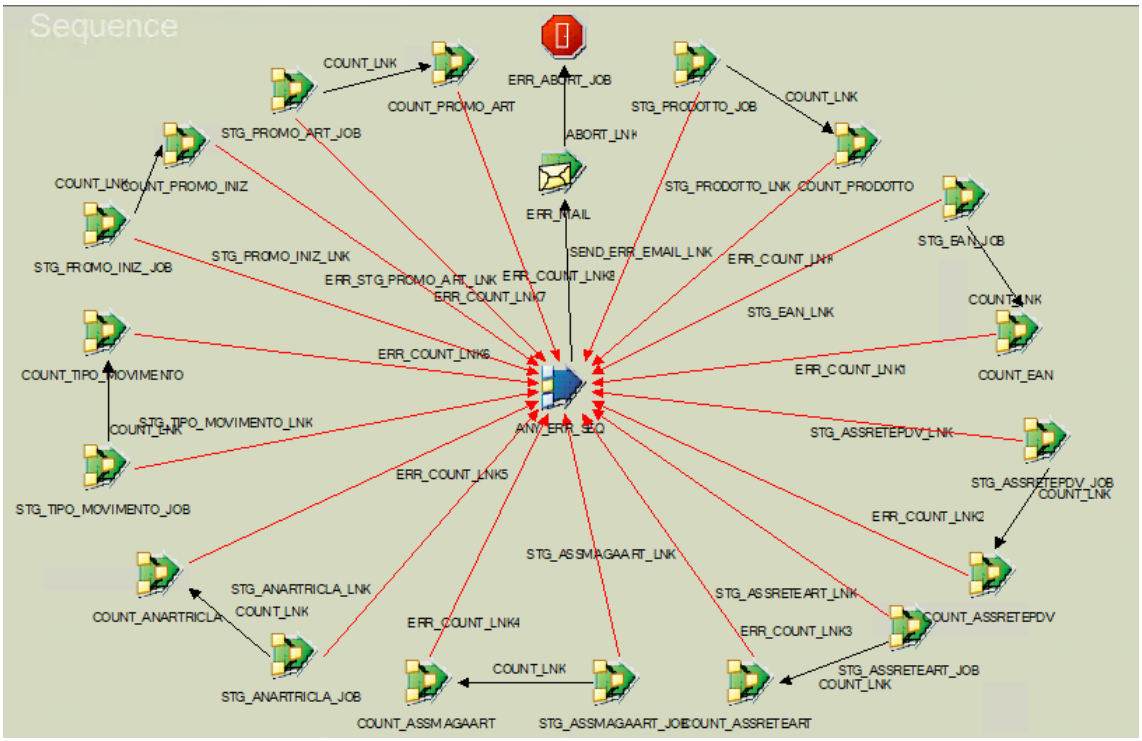

Figura 3.3. Esempio Sequence Parallel

– DLT in insert con il partizionamento per JOB_ID per velocizzare le estrazioni

Eventualmente è possibile agire sulla compressione più elevata delle partizioni più vecchie e meno accedute oppure spostare tali porzioni su hardware con costi meno elevati e prestazioni minori. E' possibile utilizzare una sola delle due modalità in modo da uniformare l'architettura delle tabelle a un modello unico. A seconda della modalità utilizzata sarà necessario differenziare il codice di un eventuale ricalcolo.

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 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). Considerare sempre il driving site dove effettuare la minus a seconda di dove conviene spostare la minima mole di dati per effettuare la differenza tra le slide di dati. Considerare anche i vari statement disponibili (MINUS, OUTER JOIN ecc.)

- FULL replica per intero giornaliera dei dati della tabella date le dimensioni contenute e la variabilità bassa del contenuto

La modalità MINUS prevede il passaggio dalle taballe STG prima descritte, mentre le altre modalità scrivono i record direttamente sulle tabelle DLT.

3.3 Replica da L0 a L1

Livello intermedio di verifica contenuto formalmente nel livello di estrazione L0 dove si effettuano controlli di integrità dei dati e si fa data quality sui dati estratti. Le tabelle sorgenti in questa fase sono le tabelle DLT (tabelle replica dei sistemi sorgenti che caratterizzano il livello di accesso L0) che inseriscono i dati ripuliti in tabelle con suffisso OK (repliche delle tabelle DLT con i datatype coerenti con i livelli successivi) che conterranno solo record necessari all'elaborazione successiva. In questo livello verranno utilizzate le tabelle di metadati di sistema che descriveranno per ogni dettaglio o addirittura ogni tabella i parametri da utilizzare per le pulizie dei dati (percentuale di errori tollerati, quali controlli effettuare a seconda del tipo di tabella).

Nel carico verso le tabelle OK verranno effettuati dei check sui dati:

- 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

- validazione record: i dati devono subire controlli per scartare record che non soddisfano i requisiti stabiliti

- constraints Nullable: nel caso in cui un campo NOT NULLABLE sia NULL, verrà assegnato un valore di default letto dalle tabelle dei metadati

- CONDIZIONI SIMPLE/COMPLEX (es.: data compresa in intervalli o campi testo di lunghezza definita ecc.)

3.3.1 Gestione degli scarti

Il ciclo del software deve prevedere o meno la gestione del reinserimento dei record scartati solitamente per integrità referenziale. Creazione di strutture ombra delle tabelle OK chiamate con suffisso ERR con l'aggiunta di una colonna reason per specificare la motivazione dello scarto che può prevedere un messaggio fisso per tipologia o contenere la reference esistente per cui si è scartato il record. Importante sarà concordare col cliente come e dove rendere fruibili i dati degli scarti in quanto si può proporre una reportistica che sottolinei le tipologie di errori più comuni e guidi la correzione dei sorgenti dati, oppure gestisca la metodologia di comunicazione di errori ai vari sistemi con log inviati tramite mail.

Importante è anche definire un periodo e una modalità di eliminazione degli errori dopo aver cercato di inserirli per un certo numero di giorni o archiviarli su storage a più basso costo o con più elevata compressione. Si parla di "Retetion Period". Il numero di giorni deve esser un parametro della tabella di metadati in modo da esser modificabile e gestibile indipendentemente dal processo invocato e senza necessità di ricompilare codice applicativo. Tale meccanismo serve per cercare di non scartare record che intercettati risolvano la reference con i loro padri solo in istanti di tempo successivi. I ricicli devono esser gestiti in modo da reinserire i dati scartati nelle tabelle prima che vengano bonificati nel calcolo delle tabelle OK per il numero di giorni che la retention prevede. Distinguiamo due modalità di gestione dei ricicli degli scarti a seconda della modalità di storicizzazione delle tabelle nel livello 1:

- Entità storicizzazione nel livello 1: se gli scarti prevedono una storicizzazione dell'informazione scartata dovremmo prevedere un ciclo che cerca di inserire le varie immagini del record in modalità successiva e temporale in modo da preservarne l'ordine di modifica

- Entità non storicizzate nel livello 1: se gli scarti non prevedono una storicizzazione dell'informazione scartata nel riciclo tenteremo sempre di reinserire l'ultima immagine del record presente

3.4 L1 - Relation Data Store

Il livello L1 contiene i dati al massimo dettaglio disponibile normalizzati e controllati, memorizzati in tecnologia relazionale. In questo livello si passa dalla replica delle strutture contenute nei gestionali all'integrazione e rivoluzione delle informazioni per tramutarle in tabelle volte all'analisi che si dovrà fare nel livello successivo. Su quest'area insistono processi di trasformazione deputati a

- normalizzazione

- integrazione
- generazione e risoluzione delle chiavi interne

Si identificano le entità e si accostano le informazioni correlate, il modello dati che sottende a quest'area è tipicamente normalizzato. Dal punto di vista funzionale le informazioni contenute in quest'area sono modellate secondo le esigenze di tutta l'azienda. Nell'ambito di quest'area sono inoltre situati:

- i dati delle tabelle di lookup
- i dati necessari per risolvere le transcodifiche e decodifiche per integrare dati provenienti da fonti differenti

I processi di trasformazione tra l'area di staging ed il relational data-store si occupano della normalizzazione ed integrazione. L'attività di integrazione risolve anche la consistenza temporale dei dati ove questi provengano da fonti differenti e non sincronizzate. Per l'attività di trascodifica il processo di alimentazione si affida a strutture apposite che risolvono la transcodifica delle chiavi. Il processo di alimentazione può essere eventualmente segmentato in sotto-processi. I processi sono l'implementazione delle regole che traducono le rappresentazioni operazionali delle entità nella rappresentazione analitica ed unificata delle stesse, necessaria per l'analisi dei dati ed esistono in ragione di uno per entità nel Relational Data Store.

Le tabelle contenute in questo livello generalmente sono accedute e alimentate da ogni genere di operazione DML possibile perciò non devono avere un livello di compressione elevate in quanto vengono accedute e aggiornate molto spesso. La modalità di alimentazione di tali tabelle è tipicamente uno statement di MERGE; le chiavi primarie su tali tabelle non sono necessarie ma devono esser almeno dichiarate in fase di modellazione per permettere ai tool di integrazione e ETL di avere una chiave di accesso alle tabelle. In questo livello normalmente si prevede di selezionare ciò che si vuole storicizzare ovvero decidere cosa, essendo importante a livello di analisi, debba esser salvato in quanto modificato nel tempo.

3.5 L2 - Dimensional Data Store

Il modello più utilizzato per il design logico di un datawarehouse è probabilmente il cosiddetto schema a stella (star schema) o lo schema a fiocco di neve (snowflake schema). Il datawarehouse rappresenta un modello dimensionale, composto da una "fact table" centrale e una serie di "dimension table" correlate, che possono poi essere a loro volta scomposte in tabelle sottodimensionali. In termini di database relazionali, in uno Star Schema, la fact table contiene generalmente una o più misure (di cui almeno una contenente un'informazione temporale) e tutte le chiavi esterne,

corrispondenti alle chiavi primarie delle varie dimension table. La fact table e' quindi l'unica ad avere join multiple con altre tabelle, mentre l'unica relazione della chiave primaria di ciascuna dimension table, avviene appunto con la fact table stessa. A loro volta, schemi a stella di questo tipo, possono diventare schemi a "snowflake" (ovvero a fiocco di neve), con l'apporto di gerarchie sugli attributi, permettendo alle dimension table di essere normalizzate a loro volta e quindi di avere tabelle sottodimensionali. Quest'area e' destinata a contenere i dati pubblicati secondo aree tematiche, tipicamente coincidenti con specifici processi aziendali.

Tutti i dati per il caricamento di quest'area provengono dal Relational Data Store L1. La tecnologia di memorizzazione dei dati in quest'area può essere relazionale (ROLAP), multidimensionale (OLAP) o mista. Dimensioni e fatti pubblicati nelle varie aree sono conformi.

Nel nostro studio i dati sono stati pubblicati in un'area relazione. Il cliente, ad ogni modo, ha anche delle fact in forma OLAP che vengono comunque alimentate dall'area in forma relazionale.

Nella fig. 3.4 vediamo lo star schema pubblicato per la Fact Vendite Trasferimenti.

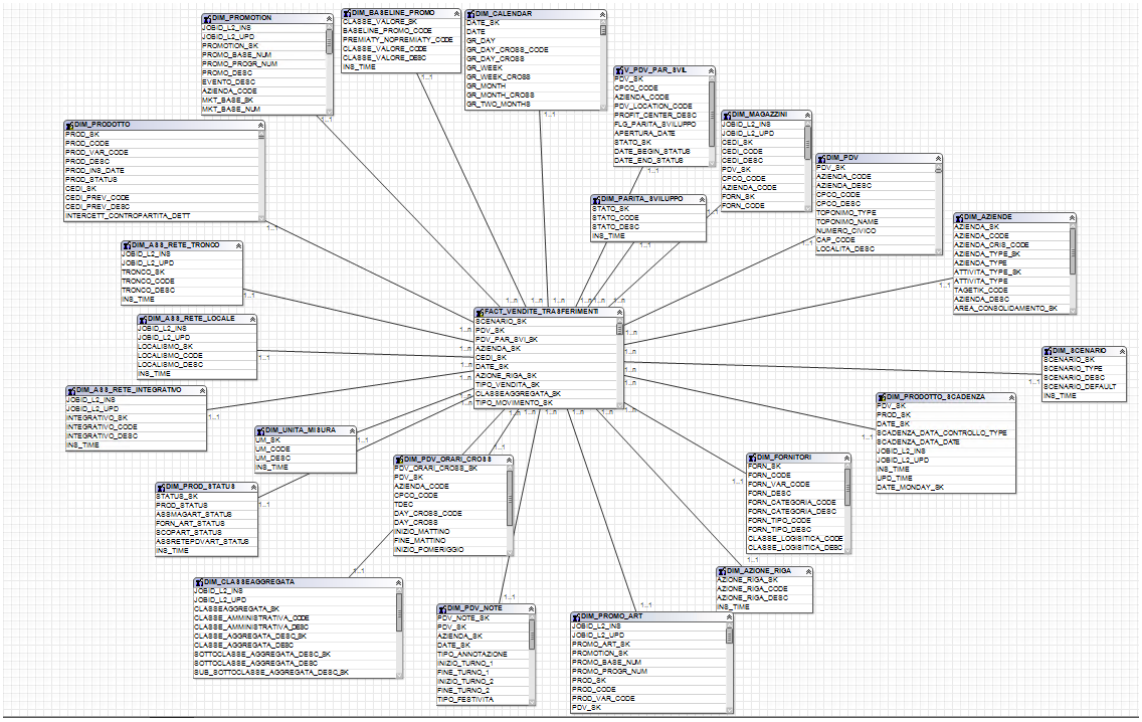

Figura 3.4. Star Schema Fact Vendite

3.5.1 Fact Table

Questa tabella contiene tutte le misure, ovvero gli elementi d'indagine che variano in continuazione nelle entità che stiamo considerando. Contiene praticamente una colonna per ogni dimensione del cubo di analisi, oltre che una o più colonne "value" con il valore della misura e una o più referenze temporali. La chiave primaria di una fact table è composta da tutte le chiavi esterne che la legano alle dimension table, oltre che dall'elemento temporale; se ne deduce che ogni record della fact table è individuato dai record delle dimension table: uno per ogni tabella dimensionale. Pensando allo star schema come ad un grafico multidimensionale, ciascun record della fact table si trova in un punto, le cui coordinate finite e discrete sono determinate da un elemento preso da ciascuna dimensione. E' bene far si che queste chiavi esterne siano tutte chiavi "surrogate", ovvero una chiave artificiale, a sostituzione di eventuali chiavi naturali, usate in produzione. L'utilizzo di chiavi surrogate (di tipo numerico) comporta generalmente un notevole risparmio di spazio, rispetto all'utilizzo, ad esempio di chiavi naturali, di tipo testuale.

La fact table risulta quindi fortemente denormalizzata, perchè contiene solamente la chiave primaria e i pochi attributi che variano nel tempo, che costituiscono le misure delle indagini; la cosa non è sempre vera invece nelle dimension table. Di solito una fact table viene aggiornata giornalmente: per questa ragione, se l'arco di tempo che contiene è di qualche anno, il numero dei suoi record può raggiungere e superare qualche milione. Bisogna allora porre particolare attenzione nella progettazione di questa tabella, soprattutto nella scelta del tipo di campi che vogliamo salvare e degli indici che permetteranno l'accesso selezionato. Si prediligono solitamente i campi di dimensione ridotta (numerici)

Esistoni diversi tipi di Fact table:

- Transaction: (store sales) è la tipologia di fact table base con granularità che associa a ogni riga un dettaglio di transazione. E' la più dettagliata di tutti i tipi di fact table
- Periodic Snapshots: (inventory, banking balance) misure semiadditive di un periodo considerato come finestra della fact
- Accumulating Snapshot: descrive l'attività di un processo di business definito in un intervallo di tempo. Questo tipo di tabella ha più colonne temporali per rappresentare le fasi del processo
- Factless: sono letteralmente da intendersi come fact table che non "immagazzinano" alcuna misura. Un tipo di tabelle di questo tipo è quello preposto alla registrazione di eventi; un esempio puo' essere fornito da un sistema che tiene conto delle partecipazioni dei ricercatori nei progetti; non dev'essere inserita una misura, come ad esempio, le ore di lavoro eseguito o previsto, ma solo un

record che attesti una partecipazione; ecco come in questo caso sia possibile avere una tabella contenente chiavi esterne di dimension table (progetti, ricercatori) ed eventualmente elementi temporali (anno in cui e' partito il progetto, ad esempio), ma non vi e' la neccessita' di registrare alcuna misura

Le fact table si presenteranno altamente denormalizzate e prive di campi testuali (essendo quelle maggiormente popolate, ciò assicura un cospicuo risparmio di spazio), mentre tutte le descrizioni per gli elementi dimensionali troveranno spazio nelle dimension table, composte per lo più da campi testuali appunto e talvolta denormalizzate. Le fact table devono essere alimentate con statement di inserimento diretto (APPEND) per velocizzare l'inserimento della grossa mole di dati nelle tabelle e possibilmente con multi processo in scrittura; inoltre devono avere una struttura che ci permetta di inserire e interrogare le slice temporali di dati in maniera molto veloce. Infatti molto spesso le fact table vengono partizionate per la misura temporale necessaria per le query del front-end.

Un esempio di fact table è dato dalla *FACT_VENDITE_TRASFERIMENTO*:

DIM/MISURA	NOME_CAMPO
DIMENSIONE	SCENARIO_SK
DIMENSIONE	PDV_SK
DIMENSIONE	PDV_PAR_SVI_SK
DIMENSIONE	AZIENDA_SK
DIMENSIONE	CEDI_SK
DIMENSIONE	DATE_SK
DIMENSIONE	ORA_SK
DIMENSIONE	CASSE_SK
DIMENSIONE	CLIENTI_CONSUMER_SK
DIMENSIONE	ETA_SK
DIMENSIONE	CLIENTI_CASH_SK
DIMENSIONE	AZIONE_RIGA_SK
DIMENSIONE	TIPO_VENDITA_SK
DIMENSIONE	CLASSEAGGREGATA_SK
DIMENSIONE	TIPO_MOVIMENTO_SK
DIMENSIONE	CLASSE_VALORE_SK
DIMENSIONE	PROMOTION_SK
DIMENSIONE	EAN_SK
DIMENSIONE	PROD_SK
DIMENSIONE	PROD_STATUS_SK
DIMENSIONE	TRONCO_SK
DIMENSIONE	LOCALISMO_SK

DIMENSIONE	INTEGRATIVO_SK
DIMENSIONE	UM_SK
DIMENSIONE	TESTATA_SK
DIMENSIONE	DOCUMENTO_ACCOM_SK
DIMENSIONE	FASCIA_SCONTRINO_SK
DIMENSIONE	DOCUMENTO_FATTURA_SK
DIMENSIONE	COMPETENCE_HOUR
DIMENSIONE	RIGA_NUM
METRICA	PUNTI_NUM
METRICA	PREZZO_VENDITA_UM_QTA
METRICA	PEZZI_QTA
METRICA	KG_QTA
METRICA	COLLI_QTA
METRICA	PALLET_QTA
METRICA	VALORE_LORDO_IVA
METRICA	ALIQUOTA_IVA
METRICA	VALORE_NETTO_IVA
METRICA	COSTO_VENDUTO_NETTO_IVA
METRICA	VALORE_PUBBLICO_CESSIONE
METRICA	VALORE_PUBBLICO_IVA_TOT
METRICA	MARGINE
METRICA	MARGINE_DENOM
METRICA	PUBBLICO_TEORICO
METRICA	RICARICO_IMMESSO
METRICA	RICARICO_IMMESSO_DENOM
METRICA	FATTURATO_FINALE
METRICA	MARGINE_FINALE
METRICA	MARGINE_FINALE_DENOM
METRICA	CARRELLI_ZERO_NUM
METRICA	VALORE_ASSOLUTO
METRICA	NUM_CARRELLO_RETT
METRICA	PREZZO_VENDITA_UM_QTA_FINALE
METRICA	PEZZI_QTA_FINALE
METRICA	KG_QTA_FINALE
METRICA	COLLI_QTA_FINALE
METRICA	PALLET_QTA_FINALE
INFO	JOBID_L2
INFO	SOURCE_TABLE

INFO	RIGA_NEGATA_FLG

3.5.2 Dimension Table

Le dimension table contengono le descrizioni delle cosiddette “dimensioni”. In una dimension table possiamo trovare la completa definizione di un prodotto con tutti i suoi attributi. I migliori attributi sono quelli testuali che possono essere usati come sorgente di restrizioni nelle query degli utenti o come intestazioni degli insiemi di risposta agli end-user. La chiave primaria di una dimension table è composta da un solo attributo (a differenza di quella di una fact table che è composita) che si ripete come chiave esterna (surrogate key) nella fact table. Non è possibile mettere direttamente in relazione tra loro due dimension table e spesso non ha neanche senso cercare di connettere due dimensioni, perché riguardano soggetti logicamente diversi; la loro unione acquista significato solo attraverso il verificarsi di un fatto. Sempre per questioni di prestazioni le dimension table sono quasi sempre denormalizzate, perchè sono quelle che subiscono più accessi in lettura e, quindi, sebbene i dati siano ridondanti, risultano più rapidi i tempi di risposta. Inoltre il risparmio di spazio che si ottiene dalla loro normalizzazione è solitamente insignificante, data la differenza di dimensione tra la fact table e le dimension table.

Nella soluzione proposta, per tutti i datamart, esistono più di 50 dimensioni. Ne elenchiamo le dimensioni create per la fact vendite trasferimenti:

- Dim Calendar
- Dim Parita Sviluppo
- Dim Magazzini
- Dim Pdv
- Dim Aziende
- Dim Scenario
- Dim Prodotto Scadenza
- Dim Fornitori
- Dim Azione Riga
- Dim Promo Art

- Dim Pdv Note
- Dim Classeaggregata
- Dim Unità Misura
- Dim Prod Status
- Dim Ass Rete Tronco
- Dim Ass Rete Locale
- Dim Ass Rete Integrativo
- Dim Prodotto
- Dim Promotion
- Dim Baseline Promo

Ogni dimensione è formata dalla Chiave Surrogata, usata per il legame con la Fact, e da campi descrittivi. Prendiamo come esempio la tabella *Dim Prodotto* con alcuni suoi campi:

Nome campo	Esempio di valorizzazione
PROD_SK	4
PROD_CODE	1004
PROD_VAR_CODE	1
PROD_DESC	POMODORINI G.400
PROD_INS_DATE	2006-03-15
PROD_STATUS	ATT
CLASSE_MERCE_SETTORE_CODE	101
CLASSE_MERCE_SETTORE_DESC	GROCERY
CLASSE_MERCE_REPARTO_CODE	110
CLASSE_MERCE_REPARTO_DESC	ALIMENTARE SALATO
CLASSE_MERCE_CATEGORIA_CODE	124
CLASSE_MERCE_CATEGORIA_CODE	POMODORI

SCD - Slowly Changing Dimension

Le dimension table devono esser alimentate previa considerazione della modalità di storicizzazione dei dati. Tale comportamento di alimentazione (in SCD) serve

prettamente per tenere traccia e storia delle variazioni delle dimensioni per alcuni dei loro attributi. Il metodo migliore per gestire tali eventualità deriva direttamente dall'utilizzo delle chiavi surrogate citate in precedenza e consiste nell'inserimento di un nuovo record in una dimension table, ogni qualvolta un attributo del soggetto interessato cambi. E' sicuramente l'approccio migliore, perché permette di navigare avanti ed indietro nella storia dell'elemento e come unico svantaggio comporta una maggior neccessità di spazio su disco. Considerando comunque che la cardinalità delle dimension table è generalmente di diversi ordini di grandezza inferiore rispetto alla fact table, è spesso possibile trascurare tale particolare.

Un esempio in cui si è utilizzato l'SCD è stato sulla dimensione prodotto per catturare lo stato del prodotto, quando ad esempio passa dallo stato NUOVO a stato ATT, oppure ad ESAURITO e così via

3.6 Indici e Parallelismi

Per migliorare le performance di caricamento esistono diverse metodologie. Una in particolare è l'utilizzo di parallelismi, sia lato database sia lato software.

Prendiamo in considerazione il software datastage, utilizzato dal cliente e prendiamo in considerazione la replica di dati da un db sorgente ad un db target, con uno stage intermedio di trasformazione. Invece di attendere che l'operazione di lettura termini per poi passare i dati ad uno stage intermedio, il parallel job divide la lettura in multi processi e passa i dati direttamente allo stage successivo tramite pipeline man mano che vengono letti. A sua volta lo stage intermedio crea una pipeline che passa i dati allo stage di scrittura mano mano che li ha pronti senza attendere che i processi terminino. I tre processi operano simultaneamente.

Quando sono coinvolti grandi volumi di dati, è possibile utilizzare la potenza di elaborazione in parallelo partizionando i dati e gestendo il carico in ogni partizione tramite sessioni indipendenti. Il Partition parallelism viene gestito a runtime e non manualmente come nei sistemi tradizionali. In fase di configurazione si deve solo specificare l'algoritmo di partizionamento dei dati, non il grado di parallelismo o dove il lavoro verrà eseguito. Nella fig. 3.5 vi è un esempio di come indicare il tipo di partizionamento per l'elemento indicato.

Per avere successo, un data warehouse deve permettere un accesso ai dati intuitivo, facile e soprattutto immediato; uno strumento molto utile per velocizzare le interrogazioni sono gli indici, strutture opzionali associati alle tabelle, permettendo di recuperare le informazioni in modo rapido.

Nel disegno scelto è importante che tutte le dimension table abbiano un indice sulla chiave surrogata, o semplicemente che la chiave surrogata sia la primary key (questo genere automaticamente un indice sulla chiave). Inoltre dopo uno studio sulle query maggiormente usate dagli utenti si può procedere alla creazione di indici

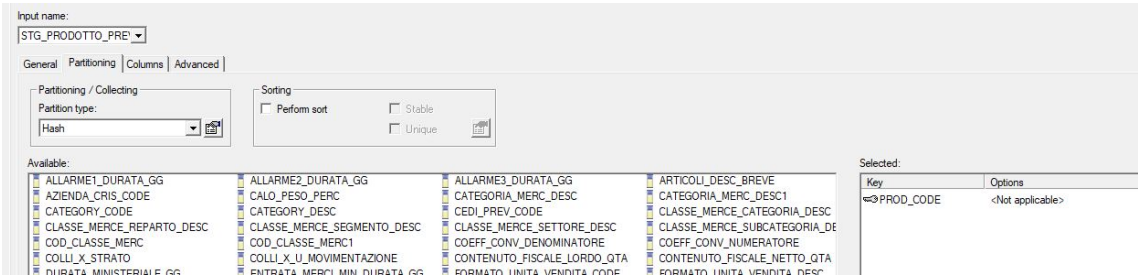

Figura 3.5. Esempio Partitioning

sugli attributi delle dimension table più utilizzati per il filtering di dati in modo da rendere molto più perfomrmante il reporting all'utente finale.

Di seguito riportiamo un estratto di una query proveniente da un report:

```
SELECT "DIM_PDV"."BRAND_DESC" AS "Des_Brand"
, ((((ltrim(rtrim(CAST("DIM_PDV"."AZIENDA_CODE" AS VARCHAR(10))))+'-')+CAST("DIM_PDV"."CPCO_CODE" AS VARCHAR
    (10)))+'-')+"DIM_PDV"."CPCO_DESC_BREVE") AS "Cod___Des_CPCO_Breve"
, ((CAST("DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_CODE" AS VARCHAR(10))+'-')+"DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_
    DESC") AS "Cod___Des_Settore"
, ((CAST("DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_CODE" AS VARCHAR(10))+'-')+"DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_
    DESC") AS "Cod___Des_Reparto"
, "DIM_CALENDAR__Alias_Current_Week_0"."ISO_WEEK" AS "ISO_WEEK"
, "DIM_PDV"."BRAND_CODE" AS "BRAND_CODE"
, "DIM_CALENDAR"."ISO_YEAR" AS "ISO_YEAR"
, "DIM_CALENDAR__Alias_Current_Year_2"."ISO_YEAR" AS "ISO_YEAR1"
, "DIM_CLASSEAGGREGATA"."CLASSE_AGGREGATA_DESC" AS "CLASSE_AGGREGATA_DESC"
, "DIM_CLASSEAGGREGATA"."SOTTOCLASSE_AGGREGATA_DESC" AS "SOTTOCLASSE_AGGREGATA_DESC"
, "FACT_VENDITE_TRASFERIMENTI"."VALORE_ASSOLUTO" AS "VALORE_ASSOLUTO"
, "FACT_VENDITE_TRASFERIMENTI"."VALORE_NETTO_IVA" AS "VALORE_NETTO_IVA"
, "FACT_VENDITE_TRASFERIMENTI"."MARGINE_FINALE_DENOM" AS "MARGINE_FINALE_DENOM"
, CASE WHEN ROW_NUMBER() OVER ( PARTITION BY "DIM_PDV"."BRAND_DESC", (
(((ltrim(rtrim(CAST("DIM_PDV"."AZIENDA_CODE" AS VARCHAR(10))))+'-')+CAST("DIM_PDV"."CPCO_CODE" AS VARCHAR(10)))
    +'-')+
"DIM_PDV"."CPCO_DESC_BREVE"), ((CAST("DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_CODE" AS VARCHAR(10))+'-')+
"DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_DESC"), ((CAST("DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_CODE" AS VARCHAR(10))+
    '-')+
"DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_DESC"), "DIM_CALENDAR__Alias_Current_Week_0"."DATE_SK"
ORDER BY "DIM_PDV"."BRAND_DESC" ASC,
((((ltrim(rtrim(CAST("DIM_PDV"."AZIENDA_CODE" AS VARCHAR(10))))+'-')+CAST("DIM_PDV"."CPCO_CODE" AS VARCHAR(10))
    )+'-')+
"DIM_PDV"."CPCO_DESC_BREVE") ASC,
((CAST("DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_CODE" AS VARCHAR(10))+'-')+"DIM_PRODOTTO"."CLASSE_MERCE_SETTORE_
    DESC") ASC,
((CAST("DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_CODE" AS VARCHAR(10))+'-')+"DIM_PRODOTTO"."CLASSE_MERCE_REPARTO_
    DESC") ASC,
"DIM_CALENDAR__Alias_Current_Week_0"."DATE_SK" ASC )=1
  THEN "DIM_CALENDAR__Alias_Current_Week_0"."ISO_WEEK"
  ELSE NULL END AS "ISO_WEEK_u"
FROM "COGNOSDWH"."L2"."DIM_AZIENDE" "DIM_AZIENDE" INNER JOIN
"COGNOSDWH"."L2"."AGGR_VENDITE_TRASFERIMENTI" "FACT_VENDITE_TRASFERIMENTI"
ON "DIM_AZIENDE"."AZIENDA_SK"="FACT_VENDITE_TRASFERIMENTI"."AZIENDA_SK"
INNER JOIN "COGNOSDWH"."L2"."DIM_CALENDAR" "DIM_CALENDAR"
ON "DIM_CALENDAR"."DATE_SK"="FACT_VENDITE_TRASFERIMENTI"."DATE_SK"
INNER JOIN "COGNOSDWH"."L2"."DIM_CLASSEAGGREGATA" "DIM_CLASSEAGGREGATA"
ON "DIM_CLASSEAGGREGATA"."CLASSEAGGREGATA_SK"="FACT_VENDITE_TRASFERIMENTI"."CLASSEAGGREGATA_SK"
INNER JOIN "COGNOSDWH"."L2"."DIM_PDV" "DIM_PDV"
ON "DIM_PDV"."PDV_SK"="FACT_VENDITE_TRASFERIMENTI"."PDV_SK"
INNER JOIN "COGNOSDWH"."L2"."DIM_PRODOTTO" "DIM_PRODOTTO"
ON "DIM_PRODOTTO"."PROD_SK"="FACT_VENDITE_TRASFERIMENTI"."PROD_SK"
INNER JOIN "COGNOSDWH"."L2"."DIM_SCENARIO" "DIM_SCENARIO"
ON "DIM_SCENARIO"."SCENARIO_SK"="FACT_VENDITE_TRASFERIMENTI"."SCENARIO_SK"
INNER JOIN "DIM_CALENDAR__Alias_Current_Week_0"
ON "DIM_CALENDAR__Alias_Current_Week_0"."ISO_WEEK"="DIM_CALENDAR"."ISO_WEEK"
INNER JOIN "DIM_CALENDAR__Alias_Current_Year_2"
ON "DIM_CALENDAR__Alias_Current_Year_2"."ISO_YEAR"="DIM_CALENDAR"."ISO_YEAR"
WHERE "DIM_CALENDAR"."ISO_YEAR"=2020
AND "DIM_PDV"."BRAND_CODE" IN ('100', '180', '200', '400') AND "DIM_SCENARIO"."SCENARIO_DESC"='CONSUNTIVO' AND
```

```
46 "DIM_PDV"."GRUPPO_DESC"='Diretto'
```

In questa query l'utente vuole visualizzare per i soli dati del 2020, i dati del *consuntivo* (ovvero le vendite), per i soli punti vendita di tipo diretto e per i brand che hanno codice 100, 180, 200 e 400. Infatti dalla query si può notare la presenza dei filtri:

- "DIM_CALENDAR"."ISO_YEAR"=2020
- "DIM_PDV"."BRAND_CODE" IN ('100', '180', '200', '400')
- "DIM_SCENARIO"."SCENARIO_DESC"='CONSUNTIVO'
- "DIM_PDV"."GRUPPO_DESC"='Diretto'

Questi filtri sono proprio filtri che si ripetono molto spesso in tanti report, è buona regola perciò creare degli indici su tali colonne e tabelle specifiche per rendere i report più performanti.

Un tipo di indice che è stato utilizzato per le fact table, è l'indice *columnstore*. Quest'indice è stato introdotto da Microsoft non molti anni fa ed ha la capacità di migliorare fino a 10 volte la compressione dei dati rispetto alla dimensione dei dati non compressi ed a migliorare fino a 10 volte le prestazioni delle query [4].

3.7 Risultati

Tutto il caricamento del DWH è gestito tramite una sequence chiamata *Main*. La sequence in questione può essere richiamata sia da schedulazione, quindi nei normali caricamenti notturni, sia apposito run da web application per la gestione dei file excel caricati dal cliente e/o per dei ricarichi giornalieri. In fig. 3.6 è raffigurato parte della sequence; la gestione è molto semplice e si ripete per tutte le parti presenti nel DWH.

Alla partenza la sequence stacca un *jobid* che è l'identificativo che userà per riconoscere il run nella tabella di metadati. Successivamente fa dei controlli, nell'esempio riportato abbiamo il controllo *Framework Setup* e il controllo *Anagrafiche*, se il flag relativo al controllo è attivo allora esegue quell'attività richiamando un'altra sequence e passa al controllo successivo oppure passa semplicemente al controllo successivo. I parametri da passare sono scritti in dei file di configurazione oppure devono essere specificati al momento del lancio.

Tutto il progetto è trasportabile su qualsiasi macchina e può appoggiarsi su qualsiasi Database, purché sia un SQL Server. Infatti il primo controllo, relativo al *Framework Setup*, indica se deve essere lanciato la creazione del Framework: la creazione di tutte le strutture su database per la gestione di tutto il progetto.

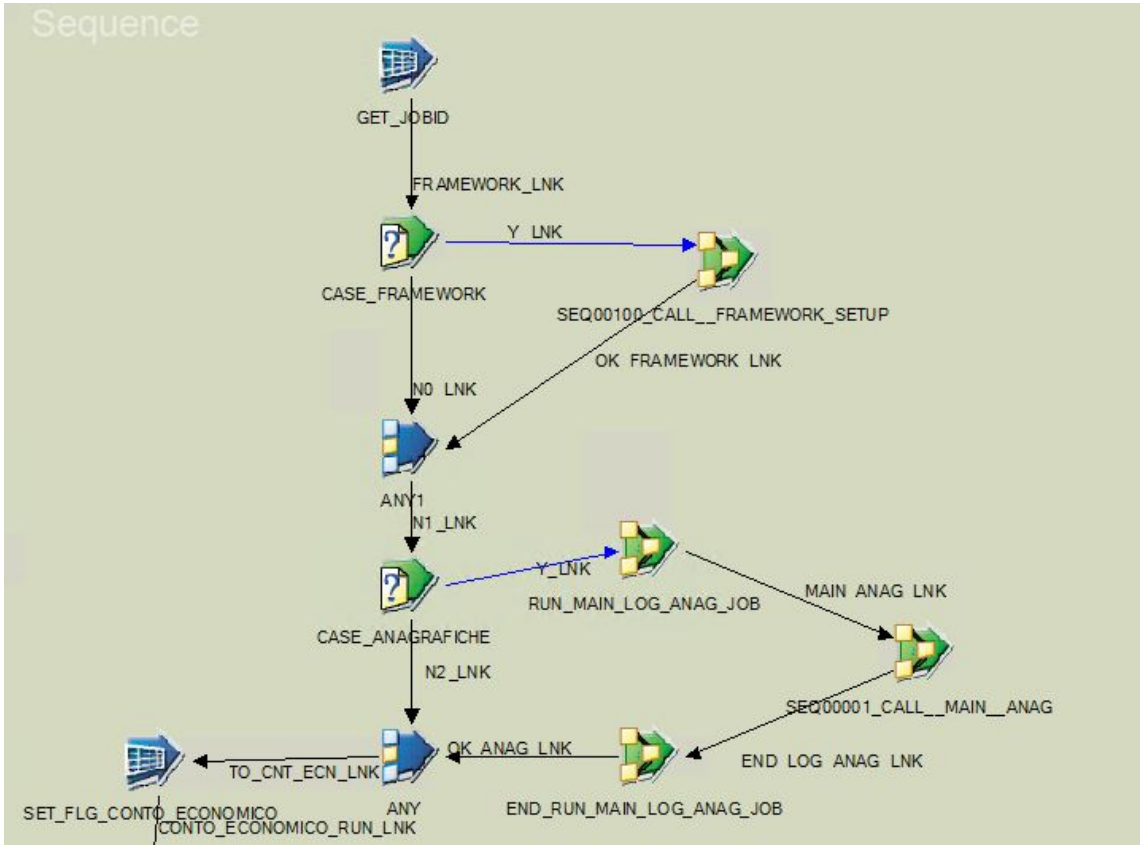

Figura 3.6. Esempio Main

Nell'immagine 3.7 è possibile visionare i tempi di tutto il DWH suddiviso per sezioni. La schedulazione del DWH è unica (coincide con la partenza delle Anagrafiche, 2:30), le partenze delle varie sezioni sono gestite direttamente dalla sequence: legge in modo parametrico quando , ad esempio, le *Vendite* sono pronte per essere caricate, nel nostro caso alle 5:30 ed aspetta fino all'orario di partenza.

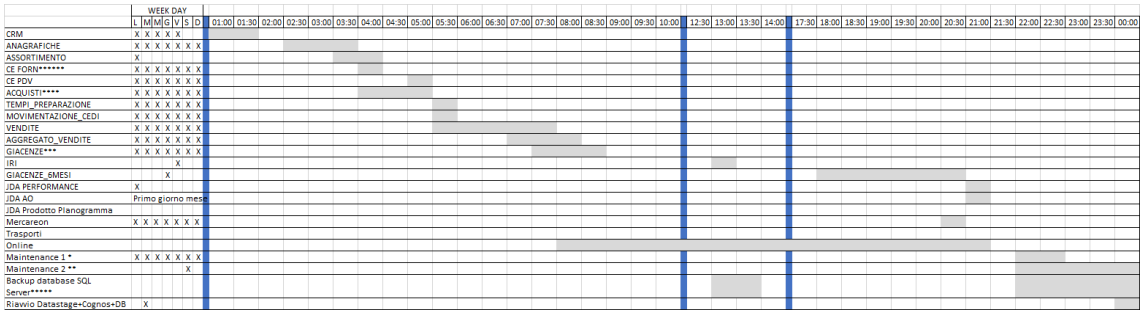

Figura 3.7. Schedulazione Main

Una sezione importante, parallela al DWH, è la sezione delle Online. Il DWH con il giro notturno carica i dati contabilizzati, quindi relativi al giorno prima. Il cliente durante il giorno ha la possibilità di leggere i dati direttamente dalle casse automatiche; perciò è stato creato un progetto parallelo near-realtime dove tramite Datastage vengono letti i dati giornalieri a cicli di 5 minuti.

Esistono anche sezioni che sono di contorno, cioè non caricano direttamente dati sul DWH ma servono per la gestione. Le sezioni di *Maintenance1* e *Maintenance2* sono delle schedulazioni che fanno pulizia di dati, ricalcolano gli indici delle tabelle, fanno lo shrink dei datafile e così via.

Gli obiettivi preposti sono stati raggiunti, perciò il cliente si ritiene molto soddisfatto del lavoro fatto.

Capitolo 4

Market Basket Analysis

Oggi ci troviamo in un mondo sempre più data-oriented e ormai tutti i principali negozi al dettaglio dispongono di un enorme quantità di dati relativi alle vendite al cliente. Osservando i dati sorge spontaneo chiedersi "Quali prodotti vengono solitamente acquistati insieme?". Entrano qui in gioco le tecniche di Market Basket Analysis che consentono di identificare le relazioni tra un infinità di prodotti acquistati da differenti consumatori, in base alla teoria secondo cui l'acquisto di un prodotto trascina l'acquisto di un altro prodotto. I risultati del market basket analysis non sono altro che delle regole che mettono in associazione due o più articoli. I clienti possono utilizzare tali regole per numerose strategie di marketing:

- Modifica del layout del negozio (posizione dei prodotti sugli scaffali)
- Analisi del comportamento del cliente
- Cross marketing

Consideriamo il famoso esempio Pannolini, Birra. Dato l'insieme di transazioni in 4.1, ogni transazione mostra gli articoli acquistati in quella transizione (basket). In questo esempio si può vedere che Birra e Pannolini sono acquistati insieme in tre transizioni. Allo stesso modo, il pane viene acquistato con il latte in tre transizioni rendendo entrambi gli articoli frequenti.

Le regole di associazione vengono scritte in questo modo [5]:

$$A => B[Support, Confidence] \tag{4.1}$$

La parte prima del => viene indicato come (Antecedente) e la parte successiva del => viene indicato come (Conseguente).

Dove A e B sono insiemi di articoli nei dati di transazione e A e B sono insiemi disgiunti.

- Itemset: collezione di uno o più items. K-item-set è un set di k items

ID	Items
1	{Pane,Latte}
2	{Pane,Pannolini,Birra,Uova}
3	{Latte,Pannolini,Birra,CocaCola}
4	{Pane,Latte,Pannolini,Birra}
5	{Pane,Latte,Pannolini,CocaCola}
...	...

Figura 4.1. Esempio: Transizioni

- Support Count: Numero di occorrenze id un item-set (frequenza)
- Support(s): Frazione delle transazioni che contengono l'item-set 'X'

$$Support(X) = \frac{frequency(X)}{N} \tag{4.2}$$

- Confidence(c): Frazione tra la frequenza dell'item-set e la frequenza dell'antecedente

$$Confidence(A => B) = \frac{frequency(A,B)}{frequency(A)} \tag{4.3}$$

- Lift(l): Frazione tra il supporto della regola e il prodotto dei supporti singoli

$$Lift(A => B) = \frac{support(A,B)}{support(A)support(B)} \tag{4.4}$$

Per una regola A => B, il supporto è dato da:

$$Support(A => B) = \frac{frequency(A,B)}{N} \tag{4.5}$$

La Confidence(c) può essere calcolata anche:

$$Confidence(A => B) = \frac{P(A \cap B)}{P(A)} \tag{4.6}$$

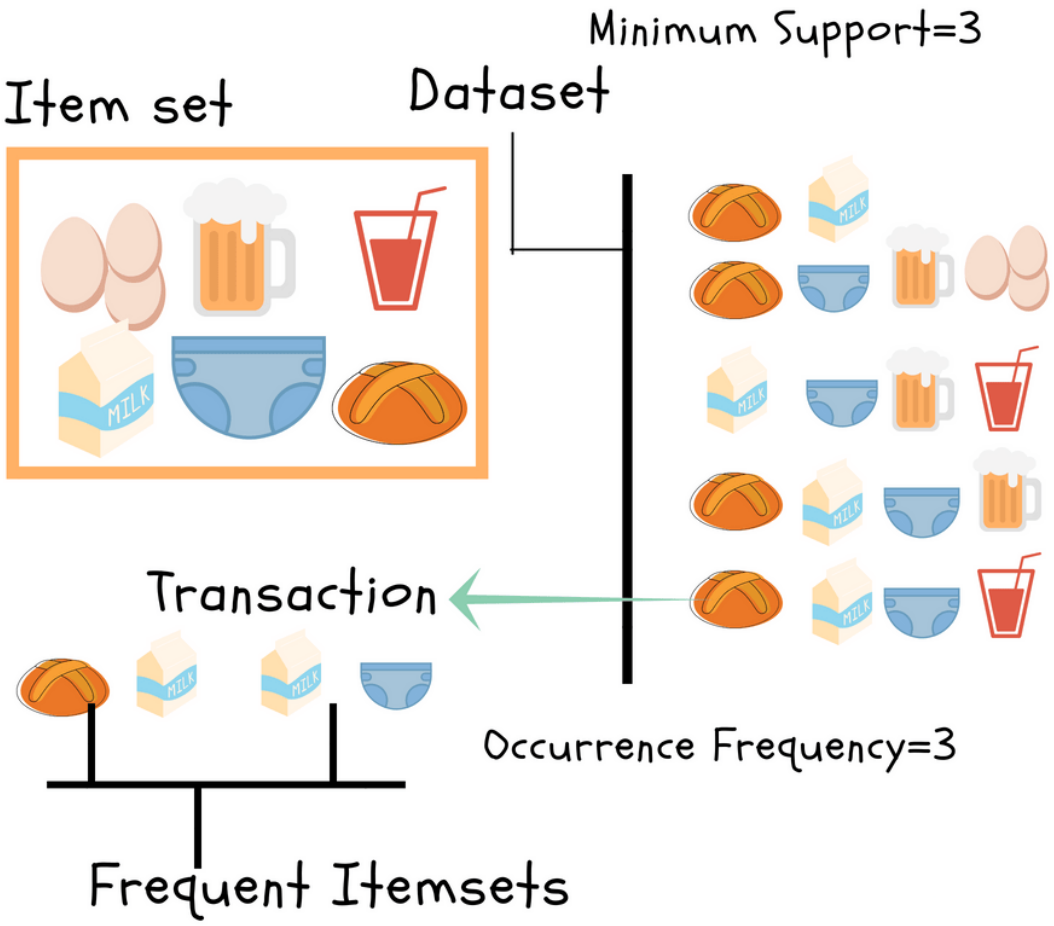

Figura 4.2. Esempio: Transizioni

dove P(x) è la probabilità dell'evento x.

Il supporto e la confidenza misurano quanto sia interessante la regola. Quando si effettua uno studio di Market Basket Analysis vengono specificati un sopporto minimo ed una confidenza minima. Questi parametri vengono scelti dal cliente in accordo con le regole aziendali. Da questi parametri derivano i

- frequent itemsets: sono tutti gli itemset che hanno un supporto maggiore o uguale al minimo supporto scelto
- strong rules: se una regola soddisfa entrambi i parametri (siano maggiori o uguali) allora si parla di regola forte

Prendendo in esame l'esempio sottoposto, avremo

$$Confidence(Pane => Latte) = \frac{3}{4} = 0.75 = 75\% \tag{4.7}$$

$$Support(Pane) = \frac{4}{5} = 0.8 \tag{4.8}$$

$$Support(Latte) = \frac{4}{5} = 0.8 \tag{4.9}$$

$$Lift(Pane => Latte) = \frac{0.6}{0.8 * 0.8} = 0.90 \tag{4.10}$$

Se la regola ha un lift uguale ad 1, allora A e B sono indipendenti e non può essere ricavata nessuna regola. Se il lift è maggiore di 1, A e B sono dipendenti uno dall'altro, e il grado di dipendenza è dato dal suo valore; mentre se il lift è minore di 1 la presenza di A ha un effetto negativo su B.

4.1 Algoritmo Apriori

L'algoritmo APRIORI è il metodo più utilizzo per la ricerca delle regole di associazione. La ricerca delle regole tipicamente avviene in due passaggi:

- generazione di set di articoli frequenti: vengono trovati tutti i set di articoli frequenti con supporto >= al minimo supporto scelto
- generazione di regole: vengono elencate tutte le regole di associazione dal set di articoli frequenti, vengono calcolati supporto e confidenza , successivamente vengono eliminate tutte le regole che non superano le soglie scelte

La generazione dei set di articoli frequenti è il passaggio più costoso dal punto di vista computazionale perché richiede una scansione completa dei dati.

Nell'esempio di cui sopra abbiamo visto un caso con sole 5 transazioni, ma nei dati reali le transazioni per la vendita al dettaglio possono superare i GB ed arrivare anche a TB di dati. Questa montagna di dati porta alla necessità di avere un algoritmo ottimizzato per eliminare gli insiemi di articoli non necessari.

L'algoritmo APRIORI afferma che: qualsiasi sottoinsieme di un insieme di elementi frequente deve essere frequente. In altre parole, non è necessario generare o testare nessun superset di un set di articoli non frequente.

Nella figura 4.3 si può vedere che come ultimo elemento in fondo ci sono tutti gli elementi presenti nei dati delle transazioni. L'algoritmo inizia a spostarsi verso l'alto creando sottoinsiemi fino al set nullo. Questa rappresentazione, chiamata *Lattice*, mostra tutte le combinazioni possibili dati d elementi; inoltre si sa che dati d elementi il numero di nodi sarà 2^d. Ciò dimostra quanto sarà difficile generare un set di oggetti frequenti trovando il supporto per ogni combinazione. L'algoritmo, infatti per ridurre il numeri di nodi da controllare, elimina tutti i superset di un set

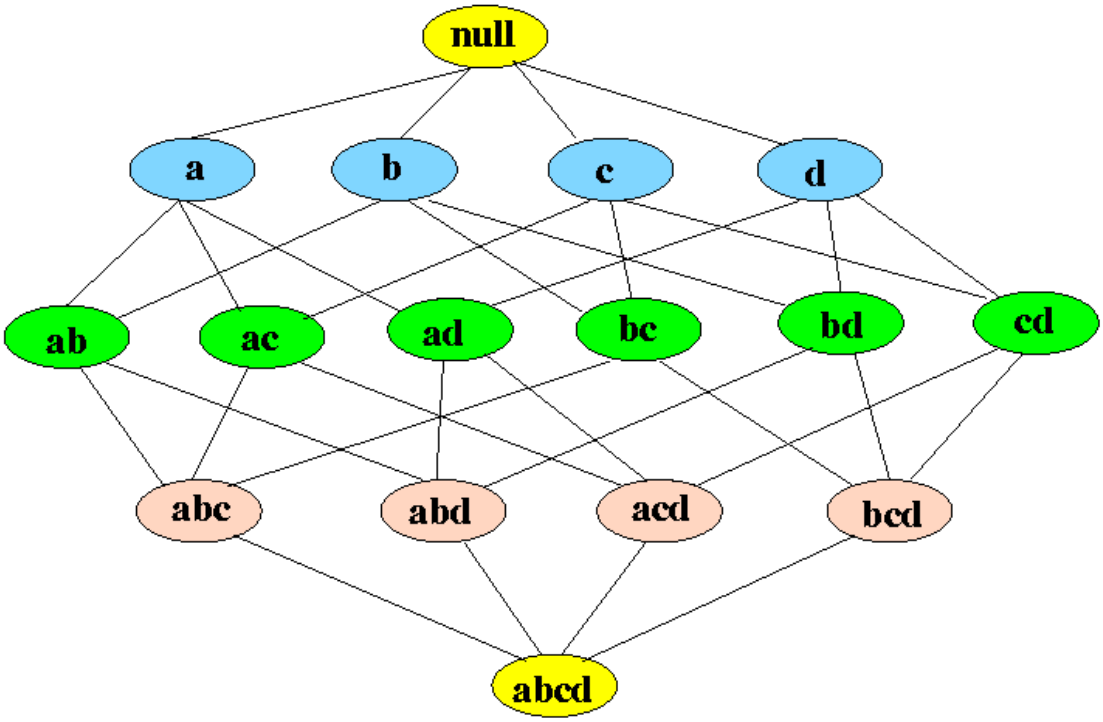

Figura 4.3. Rappresentazione itemset

non frequente. Nell'esempio considerato, consideriamo ad esempio l'elemento *ab* non frequente. Vediamo, come nella figura 4.4, che non è necessario considerare tutti i suoi super-set.

4.2 Regole Associazioni Positive

Per la ricerca delle regole di associazione si è scelto l'utilizzo del linguaggio **R**, tramite l'utilizzo delle librerie *odbc* per effettuare la lettura del dato direttamente dal DWH e *arules* per la ricerca vera e propria delle associazioni.

La ricerca delle regole è suddivisa in tre macroprocessi:

- Pre-processing: è la fase in cui il dato viene filtrato e preparato
- Processing: è l'utilizzo dell'algoritmo che genera le regole
- Post-Processing: le regole raccolte vengono interpretate e ne viene dato un significato aziendale

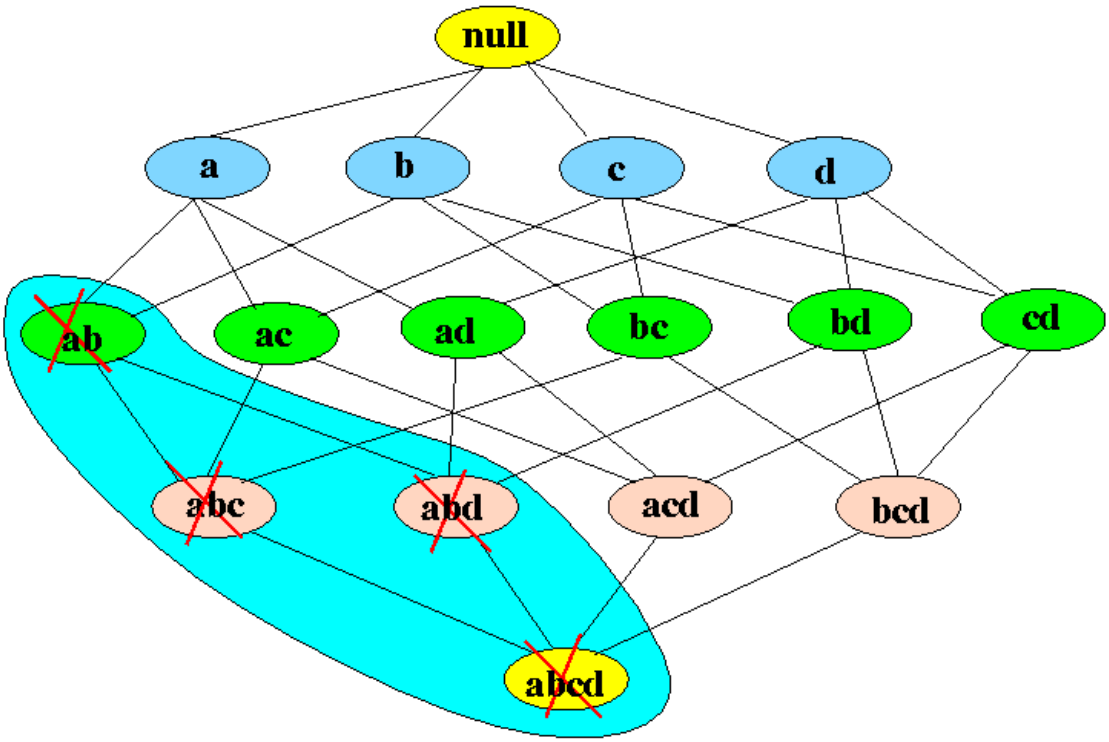

Figura 4.4. Eliminazione itemset

La parte più complessa è la fase di pre-processing. Questa prevede la conoscenza profonda dei dati aziendali in quanto non solo bisogna filtrare i dati di vendita ma anche i prodotti non necessari, ad esempio prodotti di imbustamento.

4.2.1 Pre-Processing

La fase di Pre-Processing è necessaria per la preparazione del dato da poter utilizzare con l'algoritmo APRIORI. Nell'ambito di questa tesi si vedrà l'utilizzo dell'algoritmo tramite il linguaggio di scripting **R**.

L'algoritmo Apriori prevede di ricevere il dato nel seguente modo:

- Ogni riga è una transazione: ovvero uno scontrino
- La riga contiene tutti i prodotti in un array

Per dare in pasto i dati all'algoritmo perciò abbiamo bisogno di preparare il dato nel formato necessario. Prima su tutto il dato va filtrato per i soli dati di interesse.

Nella figura 4.5 possiamo vedere una piccola parte dello Star Schema realizzato.

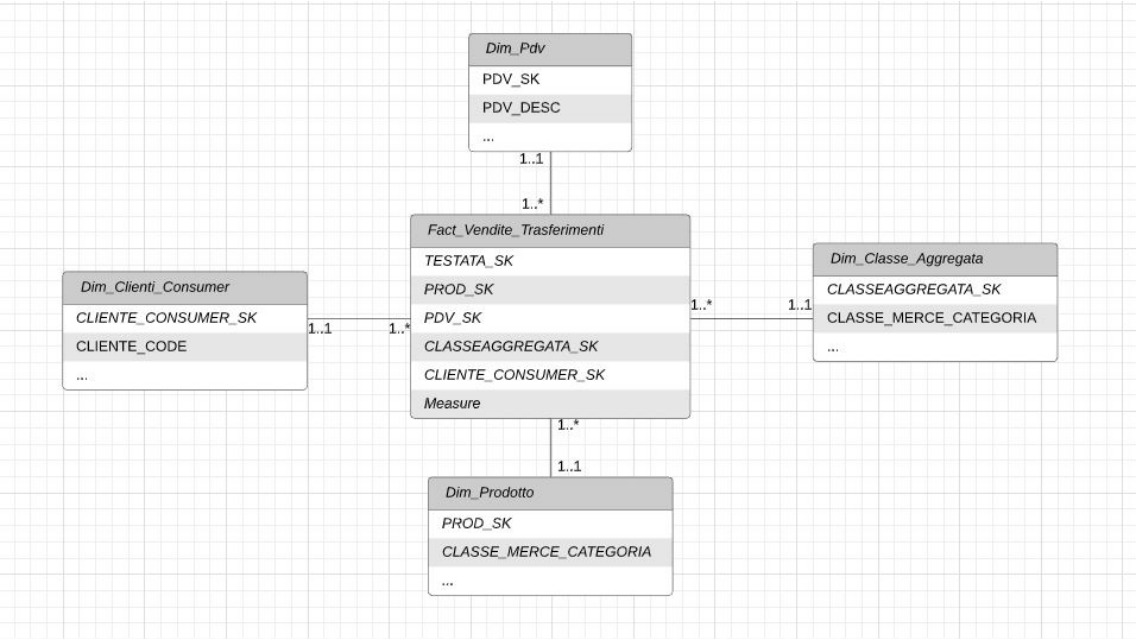

Figura 4.5. Star Schema

Come già detto nel capitolo 3.1, la tabella Fact contiene tutti i legami verso le tabelle dimensioni tramite la chiave surrogata creata dal framework; la chiave surrogata viene riconosciuta perchè ha nel finale del nome il termine *SK*. Tutte le restanti colonne sono le misure utili alle analisi.

Nel dettaglio mostrato nell'immagine, abbiamo le dimensioni

- Dim Prodotto: contiene tutte le informazioni sui prodotti
- Dim Pdv: che sta per "Punto di Vendita", contiene tutte le informazioni sui negozi
- Dim Cliente Consumer: contiene tutte le informazione sui clienti. Bisogna specificare che i clienti, nel DWH sviluppato, sono classificati in due modi: Consumer e Cash. Il cliente consumer è il consumatore finale, ci soffermeremo su questo cliente nelle nostre analisi, mentre il cliente cash sono i clienti possessori di partita iva
- Dim Classe Aggregata: questa dimensione identifica il movimento, che può essere un trasferimento da un magazzino ad un altro, un inefficienza di prodotto, una vendita, etc

Le colonne prese in considerazione per il calcolo delle regole di associazione saranno

- TESTATA_SK che rappresenta l'identificativo dello scontrino
- PROD_SK che indica l'identificativo del prodotto

Considerando che nella Fact esistono diversi mondi insiemi, bisogna perciò effettuare dei filtri. Applicheremo i seguenti filtri:

- Verranno escluse dall'analisi le categorie merceologiche non utili, ad esempio tutto i prodotti facenti parte di *Materiale di confezionamento* o *Materiale di consumo ufficio*
- Verrà selezionata la sola quota parte delle vendite al consumatore finale
- Verranno esclusi tutti i prodotti non gestiti direttamente dalla società, come i prodotti venduti da franchising e i prodotti da Bar
- Verranno esclusi tutti i prodotti omaggi
- Filtreremo i risultati per il solo punto vendita oggetto di analisi
- Filtreremo i risultati per il solo anno solare 2019

Una volta ottenuti i soli dati di interesse, di seguito un esempio, bisogna preparare il dato da dare in pasto all'algoritmo.

TESTATA_SK	PROD_SK
1	100
1	101
1	152
2	101
2	232
2	423

Lo scontrino con Testata 1 ha tante righe quante sono i prodotti acquistati, e così anche lo scontrino con Testata 2; bisogna ottenere invece una riga per ogni Testata con l'elenco dei prodotti. Per ottenere tale risultato utilizzeremo una funzione di aggregazione del Database chiamata *LISTAGG* che ci permetterà di aggregare il dato per Testata e concatenare la colonna PROD_SK in un unico campo con un separatore scelto da noi.

4.2.2 Processing

Una volta costruito il dato tramite la fase di *pre-processing*, si passa al vero e proprio *processing* tramite script R.

```
1  library(odbc)
2  library(arules)
3  con <- dbConnect(odbc::odbc(), DRIVER = "SQL Server", SERVER= "...", DATABASE = "...", UID="...", PWD="...")
4  query_AR <- dbSendQuery( con, "Select ...")
5  AR_df <- dbFetch(query_AR)
6  items_AR <- strsplit(as.character(AR_df$LST_NEG), ",")
7  trans_AR <- as(items_AR, "transactions")
8  rules_AR <- apriori(trans_AR, parameter = list(support = 0.0002, confidence = 0.4, minlen=2, maxlen=3))
9  rules_AR_sort <- sort(rules_AR, by= 'lift')
10 rulesDataFrame <- as(rules_AR_sort, "data.frame")
11 dbWriteTable(con, "TMP_RULES_NEG_R", rulesDataFrame, overwrite=TRUE)
```

Listing 4.1. Script R Positive Rules

Nello script 4.1 vediamo come viene eseguita la ricerca delle regole a partire dalla query delle transazioni. Nelle prime righe vengono importate le librerie necessarie; nella riga 3 viene definita la connessione e nella riga 4 viene inviata la query da far eseguire al db: è qui che va inserita la query costruita nella fase di pre-processing. Una volta ottenuti i dati, prendiamo la colonna contenente i prodotti dello scontrino e lo suddividiamo per il carattere con cui erano stati concatenati a livello database. Nella riga 8 vanno specificati il min-support e la min-confidence, inoltre abbiamo specificato che ci interessano le regole con minimo 2 e massimo 3 prodotti coinvolti. La min-support e la min-confidence sono stati scelti dopo diversi tentativi in modo da ottenere un numero di regole accettabile per il cliente. Le prove effettuate sono state:

MIN_SUPPORT	MIN_CONFIDENCE	NUMBER_RULES
0.0004	0.5	960
0.0004	0.4	2254
0.0003	0.5	1565
0.0003	0.4	3756
0.0002	0.5	3678
0.0002	0.4	8693

Una volta che le regole sono state trovate dall'algoritmo, le scriviamo sul database in modo da aver modo di consultarle più facilmente.

4.2.3 Post-Processing

Una volta ottenuti i risultati si passa allo studio delle prime regole di associazione, si nota subito che ci sono due evidenti limiti. Il primo è un problema di tipo temporale: la relazione ottenuta infatti non ha nessun significato di business, la regola è stata calcolata su un periodo temporale impostato in fase di analisi ma non si sa se tale regola è ancora valida, oppure se era anche valida nel passato. La regola ottenuta infatti è un punto di partenza per le analisi successive, dove abbiamo introdotto anche il fattore temporale. Il secondo limite è la non interpretabilità dei valori di support, confidence e lift da parte del business. Infatti le regole non permettono di comprendere i fattori scatenanti e non suggeriscono eventuali previsioni sul futuro.

Il termine di questa analisi si è concretizzato nell'introduzione di una KPI chiamata **Indici di Trascinamento**.

Indice di Trascinamento L'indice di trascinamento è un indice calcolato per arricchire di significato la regola positiva, quantificando quanto la regola sia forte in un determinato periodo di tempo. E' una misura che mette in relazione i prodotti della regola calcolata dall'analisi del market basket analysis e la realtà economica in termini di fatturato. Si costruisce come:

$$\text{Data una regola tale che } \{A\} \rightarrow \{B\}, \text{ allora} \tag{4.11}$$

$$TR_{t0} = \frac{FATT(B)_{t0}}{FATT(A)_{t0}} \tag{4.12}$$

dove:

- TR_{t0}: indice di trascinamento al tempo t0
- $FATT(B)_{t0}$: Fatturato di B al tempo t0
- $FATT(A)_{t0}$: Fatturato di A al tempo t0

Nella fig. 4.6 si può vedere un'analisi effettuata per una regola del tipo $A,B => C$, dove filtrando per regola estratta si può vedere l'andamento del fatturato e se è stata utilizzata una promo sul prodotto durante il periodo di tempo.

4.3 Regole Associazioni Negative

Oltre al classico rapporto di causalità tra i prodotti acquistati dai clienti, ci si è chiesto se si potessero ottenere delle informazioni circa i prodotti non acquistati dai clienti. Anche se Apriori non è la scelta giusta come algoritmo, risulta essere valido per un'analisi esplorativa. Infatti per poter implementare le regole di associazione negative tramite l'algoritmo basta predeterminare il vettore dei prodotti non venduto nello scontrino marcandolo ad esempio con un **!** per indicare la negazione dell'elemento. Nel caso in esame però il cliente ha circa 200 mila prodotti in anagrafica, utilizzare l'algoritmo con delle transazioni da 200 mila prodotti ogni riga significherebbe avere una matrice troppo grande, in più i risultati sarebbero falsati, ad esempio dai prodotti in anagrafica ma mai venduti. Nella fase di pre-processing perciò si sceglieranno quali prodotti utilizzare.

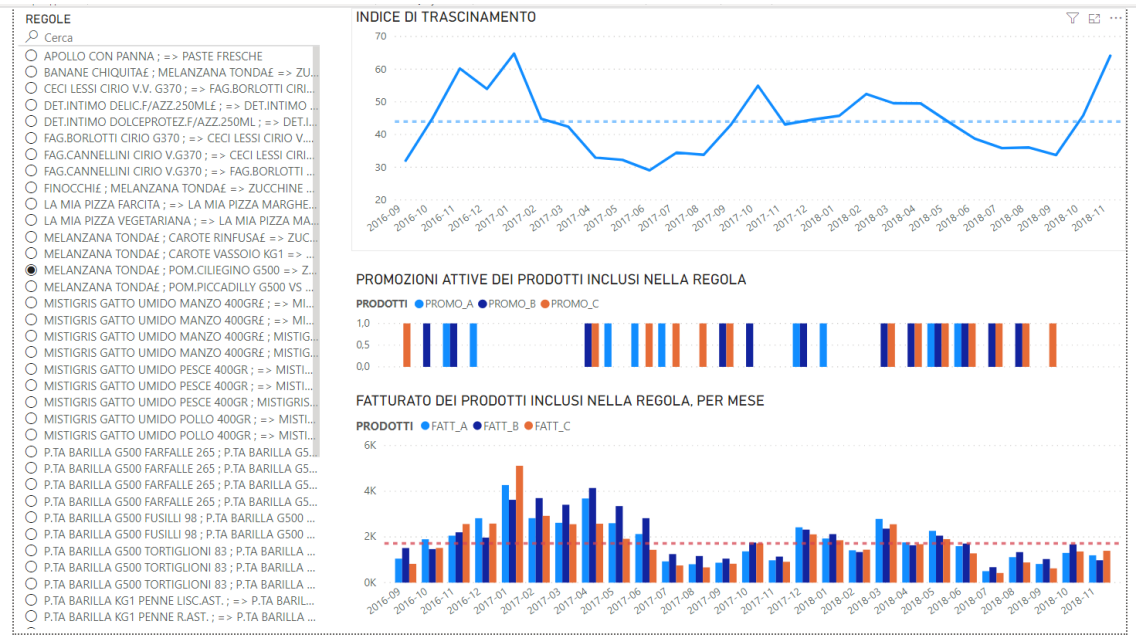

Figura 4.6. Positive Rules

4.3.1 Pre-Processing

L'output di Apriori per le regole negative potrebbero non avere significato di business, in fase di post-processing infatti le regole verranno filtrate per le sole regole che hanno come antecedente un prodotto non negato e come conseguente un prodotto negato, per avere solo le regole di interesse. Infatti, è l'acquisto dell'elemento antecedente che implica il non acquisto del prodotto conseguente.

Per ridurre il numero dei prodotti negati ad ogni singola transazione, si è svolto un lavoro di analisi insieme al cliente per capire quali prodotti potrebbero essere interessanti. Si è giunti alla conclusione di spezzare l'analisi in due:

- Una prima analisi con i prodotti negati per le stesse categorie merceologiche vendute sulla singola transazione
- Una seconda analisi con i prodotti negati per le sole categorie merceologiche che non comparivano nei prodotti della singola transazione

L'analisi divisa in questa modo genera comunque tanti prodotti per ogni singola transazione, perciò si è pensato di generare un "basket" di prodotti da quale attingere per la generazione dei negati. Questo basket è formato dai primi x elementi in termini di fatturato nella sua categoria per il periodo scelto. Il numero x e il *periodo* sono delle variabili che si possono scegliere, per la nostra analisi si è scelto un periodo fisso, l'anno solare 2019 mentre per il numero di prodotto si è scelto i primi 10

prodotti per categoria per l'analisi dei prodotti intra-categoria, mentre i primi 3 prodotti della categoria per l'analisi extra-categoria.

Questo tipo di analisi effettuate, sono fasi di pre-processing aggiuntive rispetto a quanto spiegato sopra.

4.3.2 Processing

Il processing per il calcolo delle regole negative è esattamente il medesimo calcolato per le regole positive; la differenza tra le due è:

- la fase di preprocessing: differenza di preparazione del dato
- scelta del min support e min confidence

Per le regole extra-categoria ovviamente cerchiamo di ottenere molte più regole, in quanto i prodotti negati sono stati introdotti in modo fittizio. I risultati ottenuti sono stati:

MIN_SUPPORT	MIN_CONFIDENCE	NUMBER_RULES
0.0004	0.5	2'919'343
0.0004	0.8	2'771'019
0.0005	0.8	2'374'902
0.0008	0.8	1'693'571
0.002	0.8	877'567
0.005	0.8	508'574

Si è scelto perciò l'ultimo set di cui sopra.

Mentre per le regole extra-categoria otteniamo le seguenti numerosità:

MIN_SUPPORT	MIN_CONFIDENCE	NUMBER_RULES
0.0002	0.4	103'684'601
0.002	0.8	8'029'257
0.005	0.8	3'392'146
0.01	0.8	1'321'445
0.05	0.8	18'791

In questo caso si è voluto utilizzare il min support 0.01 e la min confidence 0.8, analizzando 1 milione di regole.

4.3.3 Post-Processing

Una volta ottenuti i risultati e filtrato le sole regole di interesse, allo stesso modo come per le regole positive, aggiungeremo l'analisi temporale. La *KPI* che andremo a creare per le regole negative sarà chiamata **Indice di Cannibalizzazione**. Questo perchè l'indice indicherà quanto un prodotto sostituisce prodotti più deboli.

Indice di Cannibalizzazione L'indice di cannibalizzazione è un po' il duale dell'inidice di trascinamento, mentre il primo misura quanto un prodotto viene trascinato da un altro, questo indice calcola quanto un prodotto ne cannibalizza un altro. Si costruisce come:

$$\text{Data una regola tale che } \{A\} \rightarrow \{!B\}, \text{ allora} \tag{4.13}$$

$$CN_{t0} = ((\frac{FATT(A)_{t0}}{FATT(A)_{t0} + FATT(B)_{t0}}) - 0.5) * 2 \tag{4.14}$$

dove:

- CN_{t0}: indice di cannibalizzazione al tempo t0
- $FATT(A)_{t0}$: Fatturato di A al tempo t0
- $FATT(B)_{t0}$: Fatturato di B al tempo t0

La sottrazione di 0.5 è il prodotto per 2 è un modo per normalizzare l'indice tra -1 e 1. Un indice negativo indica che il fatturato del prodotto negato è più alto del prodotto che cannibalizza. Un indice positivo indica invece che il fatturato del prodotto che cannibalizza ha un fatturato maggiore, più il valore è vicino in valore ssoluto all'1 più il prodotto è cannibalizzato.

In figura 4.7 vediamo un esempio di cannibalizzazione di prodotto intra-categoria.

Il valore dell'indice di cannibalizzazione è mostrato in valore assoluto. In questo esempio si vede che il prodotto è fortemente cannibalizzato.

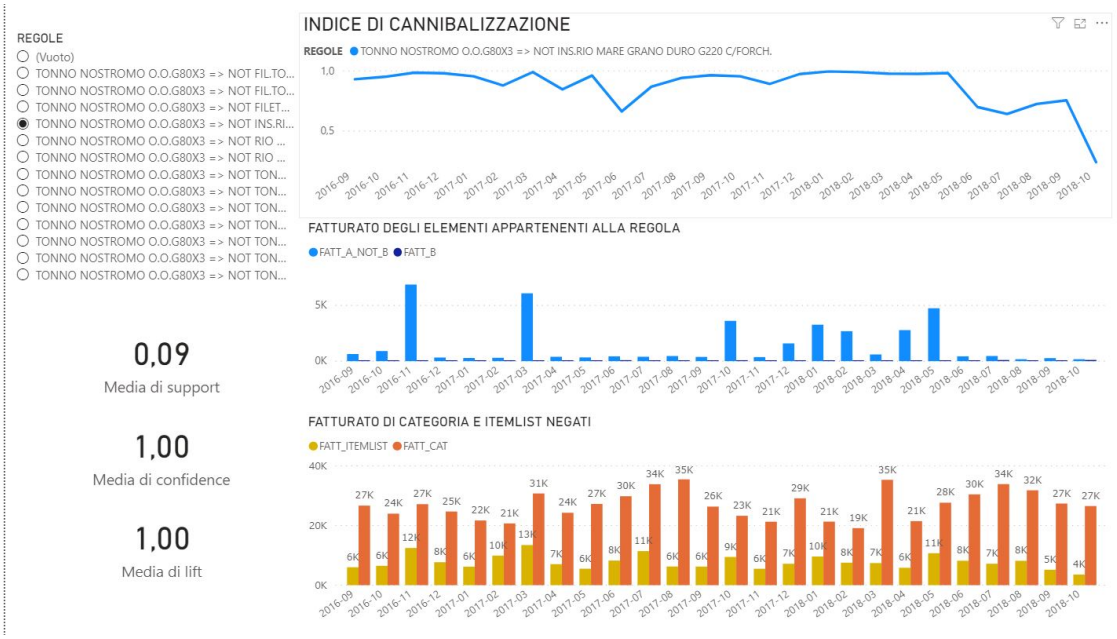

Figura 4.7. Negative Rules

Capitolo 5

Modelli di classificazione

Una volta ottenute le informazioni dal market basket analysis e ricavato le informazioni positive e negative sui prodotti acquistati, la domanda che sorge spontanea è *che tipo di cliente* sia ad effettuare tale acquisto. Per dare una risposta a questo quesito si è affrontato uno studio sul comportamento di acquisto degli utenti e si è scelto di utilizzare un analisi di classificazione, assegnando perciò ai clienti determinate *etichette*.

Le etichette scelte sono:

- Consumatore di carne
- Consumatore di alcool
- Goloso
- Amante degli animali
- Con figli
- Consumatore di piatti pronti
- Attento alla linea
- Sportivo
- Utente normale

5.1 Pre-processing

Per il calcolo delle etichette si è seguito due metodi, una prevede l'utilizzo di una rete neurale sviluppata grazie alla libreria keras; mentre nell'altro metodo si è calcolato

un albero decisionale grazie alla libreria sklearn. Entrambi i metodi sono stati sviluppati in linguaggio *Python*.

Gli algoritmi scelti sono algoritmi di machine learnig, come tali perciò hanno bisogno di "imparare" dai dati. Per la creazione della funzione desiderata perciò c'è bisogno di un esempio di dati di training e un esempio di dati di test. L'algoritmo impara da dati e training e confronta i risultati con i dati di test per misurare quanto sia efficace la funzione calcolata. L'utente che utilizza tale algoritmo deve avere conoscenza, almeno per una parte di dati (training set), quale sia il risultato atteso dato un certo set di input.

Una volta studiati, insieme al cliente, le etichette che si volevano assegnare ai clienti si è preceduto ad una prima analisi di come una "persona" avrebbe etichettato un determinato cliente. Per far ciò si è creata una matrice di acquisto dei clienti, ovvero una matrice in cui:

- sulle righe erano presenti i clienti
- sulle colonne le categorie dei prodotti
- ogni riga in corrispondenza della categoria la percentuale del fatturato totale spesa dal cliente per quella categoria

Tramite questa matrice si può già intuire come etichettare un cliente. Infatti se un cliente spende ad esempio oltre il 50% di quello compra in alcool, questo assumerà l'etichetta *Consumatore di alcool*. In prima analisi si è voluto dare un etichetta a tutti i clienti, oltre 500 mila. Considerando che non è umano analizzare così tanti clienti, si è utilizzato un report per creare delle condizioni, utilizzabili tramite *case when* per una prima etichettatura automatica. Infatti le etichette sono state scelte in base a questo report dal cliente con la promessa che le etichette sarebbero state poi modificabili in maniera semplice per future aggiunte.

Nella figura 5.1 si può vedere il numero di clienti che spendono una percentuale in quella categoria. Il numero di clienti è calcolato come sommarizzazione dei precedenti più i nuovi clienti, per gli esperti del settore è una *sum over partition con rows unbounded preceding*.

Questa figura da un'indicazione di quanto la categoria in questione sia "forte". Ad esempio la categoria più in basso è la *"BIO"*, si nota che è la categoria più bassa; questo non sta a significare che è poco acquistata ma che nel carrello di un utente i prodotti che possono cadere in quella categoria sono tipicamente pochi; perciò il cliente che spende tanto, esempio il 20% del suo fatturato in quella categoria è una persona attenta ai prodotti che acquista è quindi cadrà nell'etichetta pensata per i clienti "attenti". Questo mi porta alla creazione della prima etichettatura automatica: si pone che un utente può appartenere ad una sola etichetta; il controllo sull'appartenenza è effettuata a partire dalle categorie meno forti. In questo è possibile creare un etichetta molto velocemente per tutti gli utenti. Ma come scegliere la

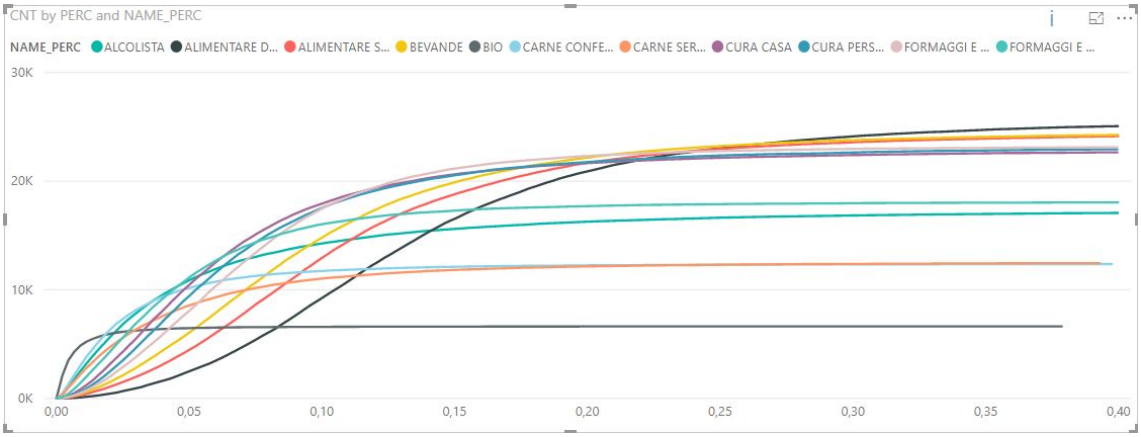

Figura 5.1. Fatturato Percentuale per Categoria

soglia tale per cui se un utente spende quella percentuale su quella categoria allora appartiene a quella etichetta?

Per capire quale sia la percentuale tale per cui un utente è classificato per quella categoria merceologica, dopo un po' di tentativi a "mano", siamo giunti insieme al cliente ad un punto preciso a partire dalla figura 5.1. Il punto in cui la tangente della funzione rappresentata dalla curva forma un angolo α con l'asse delle ascisse; l'angolo α lo si è lasciato parametrico. Con il cliente si è scelto un angolo pari a 45 gradi. Una volta individuato i punti, perciò la percentuale della categoria, tale per cui se un cliente spende quella percentuale rientra in una determinata "etichetta", bisogna decidere quale etichetta dare dato un cliente che soddisfa più di una condizione. Come già detto si è scelto di scegliere sempre l'etichetta mena "forte", ovvero la funzione il cui limite ad infinito è inferiore, questo sostanzialmente indica l'ordine di controllo. Questo lavoro è stato fatto perchè molto oneroso da far fare al cliente e serviva una prima classificazione per testare gli algoritmi.

5.2 Rete Neurale

Le reti neurali nascono dal bisogno di sviluppare un modello da precisi processi alla base del problema che spesso risultano difficili, se non impossibili, da trovare. Hanno un elevata velocità di elaborazione ed hanno la capacità di imparare la soluzione da un determinato set di esempi. Tuttavia le reti neurali vanno in sofferenza se

- non si è scelto un set di esempi adeguato al problema
- quando deve rispondere ad un problema sostanzialmente diverso al set di esempi che si è scelto

Durante la progettazione e la configurazione di un modello per l'apprendimento di una rete neurale ci sono molte decisioni da prendere. Molte di queste possono essere risolte utilizzando la struttura di una rete già configurata e usare l'euristica. Infatti la tecnica migliore è quella di progettare piccoli esperimenti e valutare empiricamente le opzioni usando i dati reali. Le decisioni da prendere sono:

- numero, dimensione e tipo di livelli nella rete
- funzione di perdita
- funzione di attivazione
- procedura di ottimizzazione
- numero di epoche

La grande quantità di dati e la complessità dei modelli richiedono tempi di addestramento molto lunghi. In generale è un buon metodo utilizzare una semplice separazione dei dati in set di dati di training e test o set di dati di training e validazione.

5.2.1 Processing

La libreria di Keras utilizzata offre due modi convenienti per valutare gli algoritmi di apprendimento:

- Utilizzare un set di dati di verifica automatica
- Utilizzare un set di dati di verifica manuale

Per il caso di studio si è utilizzato l'approccio di verifica automatica; per utilizzarla bisogna impostare l' argomento validation_split sulla funzione di fit() su una percentuale della dimensione del tuo set di dati di allenamento. Ad esempio, un valore ragionevole potrebbe essere 0,2 o 0,33 per il 20% o il 33% dei dati di allenamento trattenuti per la convalida.

```
# split into input (X) and output (y) variables
trainingSet = dataset[dataset['ETICHETTA']>=0].to_numpy()
X = trainingSet[:,1:49]
y = trainingSet[:,50]

# define the keras model
model = Sequential()
model.add(Dense(12, input_dim=48, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(7, activation='softmax'))

# compile the keras model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# fit the keras model on the dataset
model.fit(X, y, validation_split=0.33, epochs=150, batch_size=10)
```

Listing 5.1. Script Neural Network

Nello script 5.1 vediamo la definizione del modello. Il dataset è formato dalle 48 colonne di input che indicano la percentuale sul fatturato totale delle categorie, più l'ultima colonna che indica l'etichetta assegnata.

Nel run 5.2 vediamo che l'algoritmo si comporta bene tra il 70% e il 75% dei casi.

```
...
Epoch 145/150
514/514 [==============================] - 0s - loss: 0.5252 - acc: 0.7335 - val_loss: 0.5489 - val_acc: 0.7244
Epoch 146/150
514/514 [==============================] - 0s - loss: 0.5198 - acc: 0.7296 - val_loss: 0.5918 - val_acc: 0.7244
Epoch 147/150
514/514 [==============================] - 0s - loss: 0.5175 - acc: 0.7335 - val_loss: 0.5365 - val_acc: 0.7441
Epoch 148/150
514/514 [==============================] - 0s - loss: 0.5219 - acc: 0.7354 - val_loss: 0.5414 - val_acc: 0.7520
Epoch 149/150
514/514 [==============================] - 0s - loss: 0.5089 - acc: 0.7432 - val_loss: 0.5417 - val_acc: 0.7520
Epoch 150/150
514/514 [==============================] - 0s - loss: 0.5148 - acc: 0.7490 - val_loss: 0.5549 - val_acc: 0.7520
```

Listing 5.2. Output Neural Network

Questo run è stato effettuato su tutto il set di client tramite etichettatura automatica spiegata sopra. Se calcolato su un set di 100 utenti, con etichetta scelta da noi, l'algoritmo si comporta meglio portando l'accuratezza dall'80% all'85%.

5.3 Albero decisionale

L'algoritmo dell'albero decisionale, mostrato nella fig. 5.2, segue i seguenti passi [6]:

- selezione dell'attributo migliore per dividere i record
- trasformare l'attributo selezionato in un nodo e suddividere il set di dati in sottoinsiemi più piccoli
- costruisce l'albero ripetendo in modo ricorsivo il punto precedente fino a che:
  - tutte le righe appartengono alla stessa foglia
  - non ci sono più attributi rimanenti
  - non ci sono più casi

La selezione dell'attributo migliore è un'euristica per selezionare il criterio di suddivisione che suddivide i dati nel miglior modo possibile. L'algoritmo crea una classifica per ogni caratteristica (o attributo). L'attributo con il miglior punteggio verrà selezionato come attributo di divisione. Le misure di selezione più popolari sono *Information Gain*, *Gain Ratio* e *Gini Index*.

Vediamo ad esempio come avviene il calcolo per l'*Information Gain*:

$$Info(D) = \sum_{i=1}^{m} pi * log_2 * pi \tag{5.1}$$

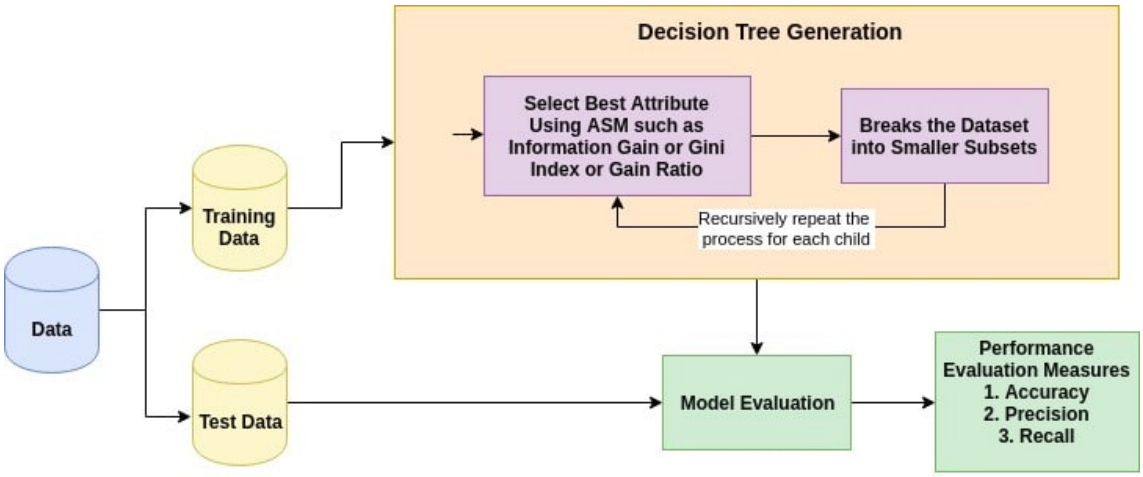

Figura 5.2. Algoritmo Decision Tree

Dove Pi è la probabilità che una tupla arbitraria in D appartenga alla classe Ci.

$$Info_A(D) = \sum_{i=1}^{V} \frac{|Dj|}{|D|} \tag{5.2}$$

$$GAIN(A) = Info(D) - Info_A(D) \tag{5.3}$$

Dove

- $Info(D)$ è la quantità media di informazioni necessarie per identificare l'etichetta di classe di una tupla in D
- $\frac{|Dj|}{|D|}$ agisce come il peso della jth partizione
- $Info_A(D)$ è l'informazione prevista richiesta per classificare una tupla da D in base al partizionamento di A

E il Gini Index, che utilizzeremo noi nel calcolo dell'albero:

$$Gini(D) = 1 - \sum_{i=1}^{m} pi^2 \tag{5.4}$$

L'indice Gini considera una divisione binaria per ciascun attributo. E' possibile calcolare una somma ponderata dell'impurità di ciascuna partizione. Se una divisione binaria sull'attributo A partiziona i dati D in D1 e D2, l'indice Gini di D è:

$$Gini_A(D) = \frac{|D1|}{|D|}Gini(D1) + \frac{|D2|}{|D|}Gini(D2) \tag{5.5}$$

Per l'*Information Gain* l'attributo A con il più alto guadagno di informazioni, Guadagno (A), viene scelto come attributo di divisione nel nodo N; mentre per il *Gini index* viene scelto l'attributo con l'indice Gini minimo.

5.3.1 Processing

Anche con l'algoritmo di Decision Tree è possibile dividere i dati in training e test data.

```
# split into input (X) and output (y) variables
X = dataset[feature_cols] # Features
y = dataset.ETICHETTA     # Target variable

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test

# Create Decision Tree classifer object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Listing 5.3. Script Decision Tree

Nello script 5.3 si nota che si è scelto una dimensione di test del 30%. L'algoritmo riesce a dare una metrica di accuratezza indicante quanto sia preciso nell'assegnazione dell'etichetta.

```
Accuracy: 0.775568741321539
```

Listing 5.4. Script Decision Tree

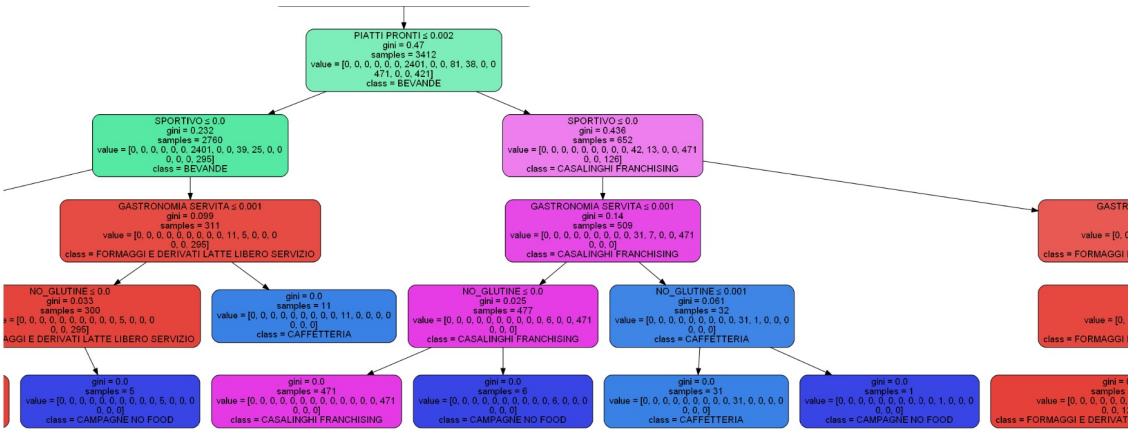

Figura 5.3. Output Decision Tree

La percentuale ottenuta dal Decision Tree con i medesimi dati della Neural Network, sul set completo di utenti con etichetta automatica, si attesta sul 77%, come

mostrato nell'output dell'algoritmo in 5.4. Nella fig. 5.3 vediamo un'estrazione dell'albero ottenuto dall'algoritmo. Utilizzando il set controllato con 100 utenti, l'algoritmo migliora portando l'accuratezza all'89%.

Capitolo 6

Analisi Combinata

Nel capitolo precedente dedicato alla classificazione abbiamo visto come classificare i clienti. I due algoritmi si comportando similmente. Il cliente si ritiene soddisfatto per i risultati ottenuti sul set dei 100 clienti. Controllando i dati ottenuti, considerando il set completo degli utenti, si hanno gli utenti così suddivisi:

ETICHETTA	COUNT
CONSUMATORE DI CARNE	3802
CONSUMATORE DI ALCOOL	1673
GOLOSO	1487
AMANTE DEGLI ANIMALI	796
CON FIGLI	714
CONSUMATORE DI PIATTI PRONTI	636
ATTENTO ALLA LINEA	175
SPORTIVO	108

Non viene preso in considerazione l'etichetta dell'Utente normale. Queste *etichette* possono essere considerate dei cluster creati a partire dal comportamento d'acquisto. I cluster hanno numerosità differenti, in fase di analisi perciò si è scelto di normalizzare i dati del fatturato sulla base della numerosità dei clienti.

Si è previsto la creazione di due dashboard per lo studio combinato tra classificazione del cliente e regole: una dashboard per le regole positive, l'altra per le regole negative.

6.1 Analisi con Regole Positive

Nella dashboard è visibile come cambia nel tempo l'*indice di trascinamento* combinata alle promozioni applicate per i prodotti coinvolti, inoltre è visibile un indicatore

di quanto un cliente spende per i prodotti considerati. L'indicatore per cliente è calcolato come Fatturato Medio per cliente nel periodo di tempo in esame.

Nella fig. 6.1 si può vedere un esempio di studio combinato.

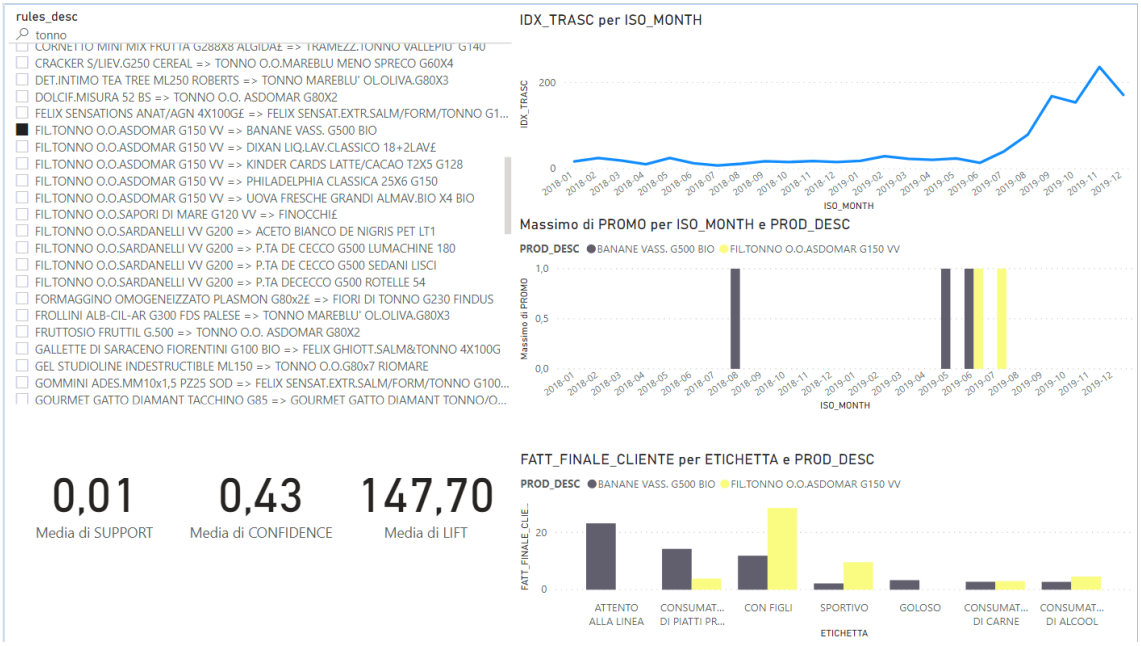

Figura 6.1. Dashboard Regole positive

In questo esempio notiamo che nel momento in cui è presente una promozione sul prodotto trascinante, il nostro indice diventa più forte, ovvero il fatturato del trascinato aumenta. La promozione sul trascinante ha fatto aumentare le vendite sul prodotto trascinato. Sempre dall'esempio notiamo che i clienti *Sportivi* e *Con Figli* sono soliti comprare il *tonno*, cioè il prodotto trascinante; si può pensare quindi di effettuare una promozione sul prodotto trascinato direttamente ai clienti che sono soliti comprare il tonno.

6.2 Analisi con Regole Negative

Nella dashboard creata per l'analisi combinata con le regole negative, allo stesso modo per come avviene con le regole positive, è visibile come cambia nel tempo l'*indice di cannibalizzazione* insieme al grafico di come sono acquistati i prodotti dai clienti.

Nella fig. 6.2 vediamo un esempio di dashboard; nell'esempio è visibile la forte cannibalizzazione tra Birra e Torta farcita. Dall'analisi dei clienti si nota che questa cannibalizzazione è molto forte per i clienti *Sportivi* e *Attenti alla linea*.

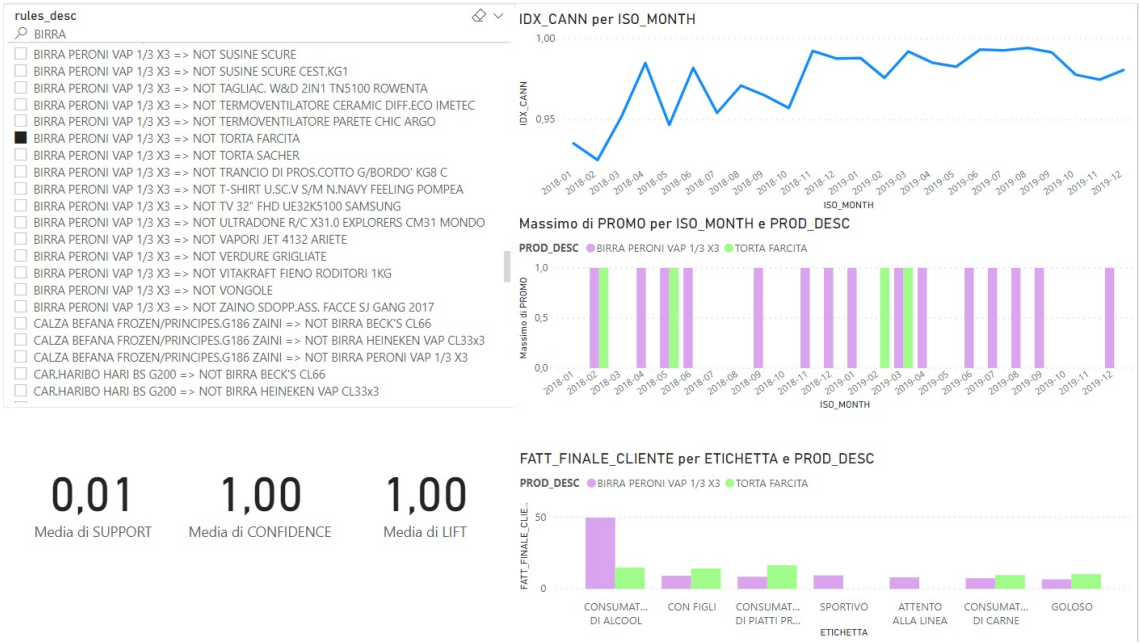

Figura 6.2. Dashboard Regole negative

6.3 Future implementazioni

Il cliente è stato molto soddisfatto della presentazione dei dati e della navigabilità. La dashboard inoltre si presta molto facilmente a future modifiche che potrebbero portare ad analisi più dettagliate. Possibili dati che porterebbero informazioni utili potrebbero essere:

- Analisi delle giacenze dei prodotti nel tempo
- Analisi del planogramma dei prodotti (posizione dei prodotti sugli scaffali)
- Analisi sulla marginalità dei prodotti
- Analisi sui listini concorrenti

Tutti questi dati sono presenti già nel DWH realizzato perciò sono facilmente portabili anche nel tipo di analisi avanzate.

Ringraziamenti

Grazie ad anni di lavoro che mi hanno insegnato tantissimo a livello personale e lavorativo, sono giunto a scrivere questa tesi.

Rivolgo i miei ringraziamenti a tutte le persone che mi hanno sostenuto e aiutato durante questo periodo.

Anzitutto, vorrei ringraziare la mia ragazza, Stefania, per il supporto datomi, senza la quale probabilmente non sarei giunto fino a qui.

Zaino in spalla e PC mi hanno accompagnato insieme a fantastici colleghi che ho incontrato durante il mio lavoro per la loro splendida collaborazione. Mi avete sostenuto e siete sempre stati pronti ad aiutarmi.

In particolare, mi rivolgo al mio supervisore presso Mediamente Consulting, il dott. Morassutto, vorrei ringraziarla per l'incredibile disponibilità e per tutte le opportunità che mi sono state date nel condurre la mia ricerca per la tesi di laurea.

E una menzione speciale, per l'aiuto che mi hanno dato nella stesura della tesi, va anche ai colleghi Antonio, Leonardo e Filippo.

Forte dei consigli ricevuti dal mio relatore, il professor P. Garza, sono arrivato alla conclusione dell'elaborato. Mi ha fornito tutti gli strumenti di cui avevo bisogno per intraprendere la strada giusta e portare a compimento la mia tesi. Un grazie speciale!

Infine vorrei ringraziare i miei genitori e tutta la mia famiglia per i loro saggi consigli e la loro capacità di ascoltarmi. Siete sempre stati al mio fianco.

Con tutto il mio affetto e la mia stima ringrazio il presidente D'Anghela e l'A.D. Scinicariello per le opportunità datemi e la fiducia dimostratami durante tutto il mio percorso.

Amici! Ultimi, ma non meno importanti, sono contento di poter condividere con voi questo mio traguardo.

GRAZIE A TUTTI!

Bibliografia

[1] Richard Miller Devens. *Cyclopaedia of Commercial and Business Anecdotes*. D. Appleton, 1868.

[2] H. P. Luhn. A business intelligence system. *IBM J. Res. Dev.*, 2(4):314–319, October 1958.

[3] Ralph Kinball. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley, 2013.

[4] Microsoft. *Indici columnstore: Panoramica*, 2020.

[5] Datacamp. *Market Basket Analysis*, 2020.

[6] Datacamp. *Decision Tree algorithm*, 2020.

[7] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.