UNIVERSITÀ DI PISA
Dipartimento di Informatica
Corso di Laurea Magistrale in Informatica per l'economia e per l'azienda
(Business Informatics)

TESI DI LAUREA

REALIZZAZIONE DI UNA PIATTAFORMA BIG DATA PER UN'AZIENDA LEADER NEL SETTORE MANUFACTORING

RELATORE
Prof. Roberto TRASARTI

CANDIDATO
Chiara Sabina PESTILLO

ANNO ACCADEMICO 2015-16

Sommario

Il presente documento descrive la realizzazione e la progettazione di una piattaforma tecnologica basata su strumenti Big Data quali Hadoop e le sue componenti principali per l'analisi e la gestione dei dati di un'azienda leader nel settore manufactoring.
Tale piattaforma ha l'obiettivo di creare uno strumento di reportistica utile a supportare le decisioni aziendali ed in particolare le decisioni riguardanti i prezzi da applicare e le strategie competitive da utilizzare per aumentare il profitto e la quota di mercato attuale.
Il lavoro é introdotto con una panoramica sui Big Data e sulle loro caratteristiche principali. Successivamente vengono descritti gli strumenti tecnologici utilizzati per lo svolgimento del progetto e le attivitá di progettazione ed implementazione del *data lake*, soffermandosi sul processo di elaborazione dei dati in ingresso e sulla rappresentazione grafica delle informazioni richieste dagli utenti finali.

Indice

Elenco delle figure

Capitolo 1

INTRODUZIONE

Questo lavoro di tesi, svolto presso la societá di consulenza Target Reply, si basa sulla progettazione di una piattaforma tecnologica basata su strumenti Big Data per l'analisi e la gestione dei dati di un'azienda leader nel settore manufactoring con l'obiettivo di creare uno strumento di reportistica utile a supportare le decisioni aziendali.
Nel Capitolo 2 saranno esposti i concetti fondamenti riguardanti il mondo Big Data e verranno descritte le caratteristiche principali e i relativi campi di applicazione, analizzando le motivazioni che spingono sempre piú imprese ad adottare sistemi Big Data avanzati per fornire un valido supporto ai loro manager.
Nel Capitolo 3 si entrerá maggiormente nel merito degli strumenti utilizzati in azienda per lo svolgimento del progetto, quali Hadoop e le sue componenti principali e verrá data una breve descrizione di ogni componente utilizzato.
Nel Capitolo 4 sará presentato il frutto del lavoro svolto in azienda. In particolare, partendo dai requisiti e dalle richieste espresse dal cliente, verranno descritte tutte le fasi che vanno dall'importazione dei dati alla creazione dei report finali, sia a livello architetturale che a livello analitico, mostrando nel dettaglio le trasformazioni effettuate sui dati sorgente. In particolare, verrá mostrato un algortimo attraverso cui sono stati scoperti Motif (ovvero pattern) ricorrenti all'interno di serie temporali.

Il Capitolo 5 é dedicato alla fase finale di reportistica ottenuta utilizzando il software di Business Intelligence a Analytics *Tableau* in cui verranno mostrati e analizzati i cruscotti principali.
Infine, nel Capitolo 6, saranno analizzati gli obiettivi raggiunti e i possibili sviluppi futuri applicabili al progetto.
Per quanto riguarda la rassegna della letteratura, per la descrizione degli strumenti utilizzati si é fatto riferimento a manuali tecnici, in particolare [Rezzani A. 2013] e [Turkington et al., 2014]. Invece, per la descrizione dell'algortimo utilizzato nel Capitolo 3 nell'ambito della *motif discovery* si é fatto riferimento a numerosi paper elencati in bibliografia, quali [1],[2],[3],[4],[5],[6],[7].

Capitolo 2

I BIG DATA

Il termine Big Data indica grandi aggregazioni di dati che non possono essere processati o analizzati con i tradizionali processi e strumenti di analisi. I Big Data aprono le porte verso nuovi modi di percepire il mondo e di prendere decisioni. Essi, infatti, possono essere definiti come *"il nuovo microscopio che rende misurabile la societá"*, poiché spingono verso una nuova scienza di dati in grado di diffondere opinioni, distribuire risorse economiche o energetiche, prevedere crisi economiche e soddisfare bisogni di mobilitá.
Da un lato, alcuni fattori, quali ad esempio la crescita dei dati scientifici, hanno fortemente contribuito all'accelerazione della produzione di questi dati. Dall'altro, questa crescita esponenziale é il risultato di alcuni mutamenti sociali ed economici estremamente positivi avvenuti nella nostra societá.
Si consideri, ad esempio, la rapida diffusione dei dispositivi mobili, ricchi di contenuti multimediali e compatibili con i sistemi GPS, e dei social network, che hanno consentito a miliardi di persone in tutto il mondo di tenersi in contatto in modo digitale. Una visione interessante di cosa sono i Big Data é stata esposta da Alexander Jaimes, ricercatore presso Yahoo Research, che ha affermato "i dati siamo noi".
Anche se il termine Big Data non si riferisce ad alcuna quantitá in particolare, solitamente si inizia a parlare di Big Data a partire da Terabyte di dati, cioé quando i dati non possono essere piú memorizzati o processati da una singola

macchina.

I Big Data hanno numerose caratteristiche che li differenziano dalle tradizionali collezioni di dati. Queste caratteristiche sono note anche come le Tre V: **Volume**, **Velocitá** e **Varietá**.

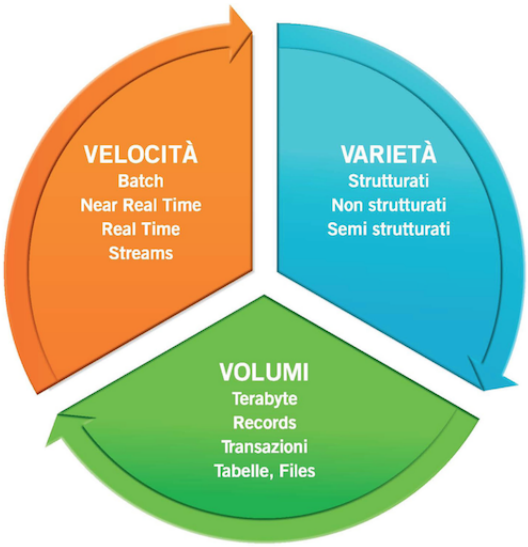


Figura 2.1: Le 3 V dei Big Data

Un equivoco di uso comune é la convinzione che il concetto di Big Data riguardi esclusivamente le dimensioni dei dati. Per quanto le dimensioni rappresentino sicuramente un elemento fondamentale esistono anche altri aspetti che non sono necessariamente associati ad esse.

Si prendano in considerazione, ad esempio, la velocitá di generazione dei Big Data da un lato e il numero e la varietá di origini da cui i dati vengono generati contemporaneamente dall'altro. Per stabilire quando effettivamente i vari tipi di dati rientrano nella categoria dei Big Data é necessario analizzare gli elementi sopra citati.

Per quanto riguarda il **Volume**, esso fa riferimento alla capacitá di acquisire, memorizzare ed accedere a grandi volumi di dati, non gestibili con i database tradizionali; la **Velocitá** non é riferita alla crescita o al volume dei dati ma alla necessitá di comprire i tempi di gestione e analisi. In breve tempo, infatti, il dato puó diventare obsoleto.

Inizialmente le aziende hanno analizzato i dati usando processi batch, ma con le nuove fonti dei dati, quali applicazioni sociali e mobili, il processo batch cede il posto a un nuovo modo di gestire ed elaborare i dati: i dati ora arrivano in streaming al server, in tempo reale ed in modo continuo.
Infine, l'aspetto **Varietá** rappresenta un cambiamento nel modo in cui i dati vengono analizzati e memorizzati, essendo di diversa natura e non strutturati. L'era dei Big Data é caratterizata dalla necessitá di esplorare anche dati non strutturati oltre e insieme alle informazioni tradizionali. Se pensiamo ad un post su Facebook, un tweet o un blog, essi possono essere in un formato strutturato (JSON), ma il vero valore si trova nella parte dei dati non strutturati.
Quelle finora descritte sono le 3 classiche caratteristiche dei Big Data, ma ve ne possono essere aggiunte altre: la **Veridicitá**, la **Variabilitá** e la **Complessitá**.
Per quanto riguarda la **Veridicitá**, si può asserire che tutti i dati raccolti costituiscono un valore per l'azienda. Infatti, é dall'analisi dei dati che si colgono le opportunitá e si trae supporto per i processi decisionali in modo tale che questi possano avere un grande impatto sulle attivitá. Piú dati si hanno a disposizione, piú informazioni e valore si riescono ad estrarre. Tuttavia il solo volume dei dati non garantisce sufficientemente la qualitá dei dati.
Si deve essere sicuri che i dati siano affidabili e usabili a supporto dei processi decisionali. La veridicitá e la qualitá dei dati diventa, pertanto, un requisito fondamentale affinché i dati possano davvero "alimentare" nuove intuizioni e idee e costituire valore per l'azienda.
La variabilitá si riferisce all'inconsistenza presente nei dati, che ostacola il processo di manipolazione e gestione efficace dei dati.
Infine, la **Complessitá** indica il fatto che i dati provengono da fonti diverse e devono essere collegati tra loro per ricavare informazioni utili.
Un altro aspetto molto importante che va considerato quando si parla di Big Data riguarda le diverse fonti da cui traggono origine. Esse sono molteplici, ma possono essere ricondotte a due distinte categorie.

La prima é relativa alle tipologie tradizionali di dati; la seconda é relativa ai dati generati dalla rete. Dunque un'azienda che avesse intenzione di sfruttare l'immenso patrimonio informativo all'interno della rete si troverebbe ad affrontare, oltre al problema della quantitá dei dati, anche l'aspetto della varietá e dell'eterogeneitá.
Le principali fonti di dati sono le seguenti:

- dati strutturati in tabelle (relazionali): sono i dati sui quali si basa la tradizionale Business Intelligence;
- dati semistrutturati (XML): le applicazioni transazionali e non, forniscono nativamente output di dati in formato XML. Si tratta perlopiú di dati business-to-business organizzabili gerarchicamente.
- dati di eventi e macchinari (messaggi, batch o real time,sensori, RFID e periferiche): sono i tipici dati definibili Big Data che sino a pochi anni fa venivano memorizzati con capacitá temporali molto brevi per problemi di *storage*.
- dati non strutturati (linguaggio umano, audio, video): sono enormi quantita di metadati, memorizzati sul web, dai quali é possibilie estrarre informazione attraverso tecniche di analisi semantica.
- dati non strutturati provenienti dai social media (social network, blog, tweet): sono l'ultima frontiera delle fonti dati non strutturate. I loro volumi aumentano esponenzialmente nel tempo e il loro utilizzo puó aprire nuovi paradigmi di analisi.
- dati dalla navigazione web (*clickstream*): sono enormi quantitá di dati che portano informazioni sui consumi e le propensioni di milioni di utenti. Anche per questo tipo di dati i volumi aumentano esponenzialmente nel tempo.

Altro aspetto che bisogna considerare quando si parla di Big Data riguarda l'acquisizione dei dati.

L'acquisizione dei Big Data puó avvenire in modi diversi, a seconda della fonte dati. I mezzi per l'acquisizione dei dati possono essere suddivisi in quattro categorie:

- *Application Programming Interface*: le API sono protocolli usati come interfaccia di comunicazione tra componenti software. Alcuni esempi di API sono la Twitter API, la Graph Facebook API e le API offerte da alcuni motori di ricerca come Google e Yahoo. Esse permettono, ad esempio, di ottenere i tweet relativi a determinati argomenti o di esaminare i contenuti pubblcitari che rispondono a determinati criteri di ricerca.

- *Web Scraping* attraverso cui si possono prendere dati semplicemente analizzando il Web, cioé la rete di pagine collegate tra loro tramite *hyperlinks*.

- Strumenti ETL: gli strumenti ETL (Extract, Trasform and Load) sono quegli strumenti utilizzati nel Data Warehousing per convertire i database da un formato o tipo ad un altro. L'applicazione ETL di punta per Hadoop é Apache Sqoop, che permette di caricare i dati strutturati presenti in un RDBMS in HDFS, Hive o HBASE, cosí come permette di fare l'operazione inversa.

- Stream di dati: sono disponibili tecnologie per la cattura e il trasferimento di dati in tempo reale.

Infine, l'ultimo aspetto da considerare riguarda la gestione e memorizzazione dei Big Data.
Il bisogno di un'elevata scalabilitá e di una memorizzazione di dati che potrebbero essere non strutturati fa sí che i tradizionali DBMS relazionali non siano adatti alla memorizzazione dei Big Data.
Per questo motivo sono stati creati nuovi sistemi che permettono di memorizzare tipi di dati non relazionale offrendo scalabilitá orizzontale, cioé le

prestazioni aumentano in maniera lineare rispetto al numero di nodi/macchine presenti. Ció si contrappone all'aumento di prestazioni di una singola macchina, operazione complessa e costosa.

Tra le tecnologie presenti per la memorizzazione dei Big Data, la piú diffusa é Hadoop, un software open-source affidabile e scalabile per il calcolo distribuito. I software di calcolo distribuito sono stati progettati per sfruttare la potenza di calcolo e la memoria di un insieme di macchine, suddividendo il lavoro da svolgere tra le stesse.

Maggiori dettagli sull'architettura e sul funzionamento di Hadoop saranno dati nel capitolo successivo.

Capitolo 3

LA PIATTAFORMA HADOOP

Apache Hadoop é un software open-source per l'archiviazione e l'analisi di quantitá elevatissime di dati strutturati e non.
É possibile gestire terabyte o quantitá superiori di dati di qualsiasi tipo, dalla posta elettronica a letture di sensori, log di server, segnali GPS e altro ancora.
Hadoop é stato ispirato dalla MapReduce di Google e dal Google File System.
Il nome del progetto é stato scelto dal suo creatore Doug Cutting, il quale ha scelto Hadoop, il nome dell'elefante di pezza del figlio.
In origine fu sviluppato per supportare la distribuzione per il progetto del motore di ricerca Nutch. L'architettura, realizzata in Java, permette di poter scalare da pochi server fino a migliaia di sistemi: ogni server contribuisce con le proprie risorse di calcolo e la propria capacità di memorizzare i dati,e quindi aggiungendo server, chiamati anche "nodi", é possibile far crescere un sistema Hadoop in modo pressocché lineare.
Una delle ragioni della popolaritá di Hadoop é anche la convenienza economica. In precedenza, per l'elaborazione di set di Big Data erano necessari super-computer e altro hardware costoso e specializzato. Hadoop rende possibile l'elaborazione affidabile, scalabile e distribuita in server standard di settore, permettendo di gestire grandi quantitá di dati con budget ridotti.
In base ad alcune stime, circa l'80 percento di dati gestiti attualmente dalle organizzazioni non é strutturato in colonne e righe ben definite. Si tratta

piuttosto di una valanga confusa di messagi di posta elettronica, feed di social media, immagini da satellite, segnali GPS, log di server e altri file non strutturali e non relazionali.
Un altro grande vantaggio offerto da Hadoop é la possibilitá di gestire quasi tutti i tipi di file o formati, permettendo alle aziende di ottenere risposte in precedenza impossibili da ottenere.
L'alta affidabilitá, e dunque la protezione dei dati, viene realizzata non basandosi sulle caratteristiche hardware dei server, ma bensí a livello software: sono le librerie di Hadoop che identificano se e quali componenti presentano un malfunzionamento ed intervengono per ripristinare le operazioni.
Rispetto ai file system *general purpose* quali NFS o CIFS, HDFS é costruito attorno all'idea di un utilizzo dei dati in modalitá *write-once, read many*. Questo ovviamente non significa che i dati non possano essere modificati, ma indica un approccio differente all'utilizzo dei dati stessi.
Come é tipico del mondo Big Data, non ci si aspetta di accedere in modo puntuale ad un singolo file o ad una porzione di questo, ma piuttosto é previsto che le analisi operino su una gran parte dei dati memorizzati (se non tutti). La possibilitá di poter accedere in modo efficiente a tutti i dati e non solo ad una porzione é in effetti una delle chiavi di volta dei Big Data.
É possibile confrontare le pecurialitá di storage e data management offerte da Hadoop con quelle offerte da un "classico" RDBMS.

RDBMS	Hadoop
Schema *on Write*: lo schema dei dati deve essere creato prima che i dati stessi vengano caricati	Schema *on Read*: i dati sono semplicemente copiati nel file system, nessuna trasformazione è richiesta
Ogni dato da caricare deve essere trasformato nella struttura interna del database	I dati delle colonne sono estratte durante la fase di lettura
Nuove colonne devono essere aggiunte esplicitamente prima che i nuovi dati per tali colonne siano caricate nel database	I nuovi dati possono essere aggiunti ed estratti in qualsiasi momento

Figura 3.1: Differenze RDBMS vs HADOOP

Le componenti che costituiscono il nucleo centrale della piattaforma Hadoop sono le seguenti:

- **Hadoop Common**, che fornisce l'accesso al file system supportato da Hadoop. L'Hadoop Common package contiene i file jar e gli script necessari per avviare Hadoop. Il package fornisce inoltre il codice sorgente, la documentazione e una sezione di contributi che include i progetti della comunitá Hadoop.

- **HDFS** (Hadoop Distributed File System), é il file system distribuito che fornisce un'efficace modalitá di accesso ai dati e garantisce che i dati siano ridondanti nel cluster rendendo le operazioni sui dati stessi immuni dall'eventuale guasto di un nodo. HDFS accetta dati in qualsiasi formato, strutturati e non.

- **MAP REDUCE**, che permette di realizzare sistemi di computazione parallela e distribuita di grandi quantitá di dati lavorando secondo il principio"*divide-et-impera*".

- **YARN**, framework che consente di creare applicazioni o infrastrutture per il calcolo distribituito (sulla base di MapReduce). Esso si occupa della gestione delle risorse del cluster (memoria/CPU/storage).

Inoltre, si aggiungono al nucleo centrale una serie di altri sistemi software che completano quello che viene denominato "L'ecosistema Hadoop".

3.1 Hadoop Distributed File System

HDFS é un file system distribuito ideato per soddisfare requisiti quali l'affidabilitá e la scalabilitá e per gestire un numero molto elevato di file, anche di dimensioni ragguardevoli (dell'ordine dei gigabyte o terabyte), attraverso la realizzazione di cluster che possono contenere migliaia di nodi.
In HDFS i file sono organizzati in una struttura gerarchica di cartelle.

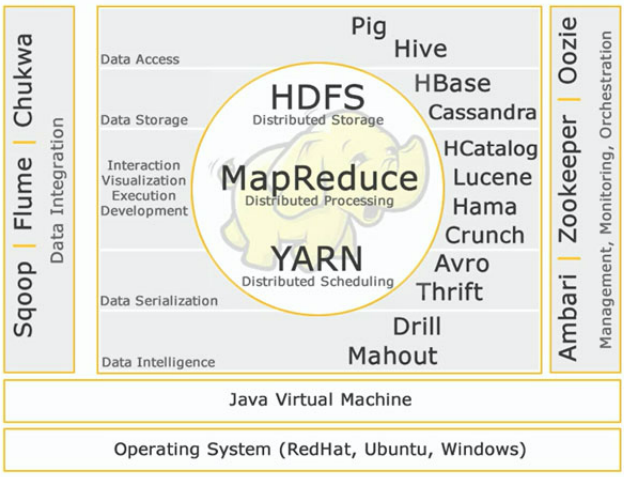


Figura 3.2: Ecosistema Hadoop

Dal punto di vista dell'architettura, un cluster é costituito dai seguenti tipi di nodi:

- **NameNode**: é l'applicazione che gira sul server principale. Gestisce il file system ed in particolare il *namespace*, cioé l'elenco dei nomi dei file e dei blocchi (solitamente i file infatti vengono divisi in blocchi da 64/128 MB) e controlla l'accesso ai file, eseguendo le operazioni di apertura, chiusura e modifica dei nomi dei file. Inoltre, determina come i blocchi dati siamo distribuiti sui nodi del cluster e la strategia di replica che garantisce l'affidabilitá del sistema.
  Il NameNode monitora anche che i singoli nodi siano in esecuzione senza problemi e in caso contrario decide come riallocare i blocchi. Il NameNode distribuisce le informazioni contenute nel namespace su due file: il primo prende il nome *fsimage* e rappresenta l'ultima immagine del namespace; il secondo é un log dei cambiamenti avvenuti al namespace a partire dall'ultimo aggiornamento del file *fsimage*.

- **DataNode**: é una applicazione che gira su altri nodi del cluster, generalmente ve ne é una per nodo, e gestisce fisicamente lo storage di ciascuno di essi. Queste applicazioni eseguono, logicamente, le operazio-

ni di lettura e scrittura richieste dai *client* e gestiscono fisicamente la creazione, la cancellazione e la replica dei blocchi dati.

- **SecondaryNameNode**: é un servizio che aiuta il NameNode ad essere piú efficiente.
- **BackupNode**: é il nodo di *failover* e consente di avere un nodo simile al SecondaryNameNode sempre sincronizzato con il NameNode.

Come detto in precedenza, i file sono organizzati in blocchi da 64 o 128 MB e sono ridondanti su piú nodi. Sia la dimensione dei blocchi, sia il numero di repliche possono essere configurate da ciascun utente per ogni file.
Le repliche sono utilizzate per garantire l'accesso a tutti i dati (anche in presenza di problemi a uno o piú nodi) e per rendere piú efficiente il recupero dei dati.
In HDFS le richieste di lettura dati seguono una politica relativamente semplice: esse avvengono scegliendo i nodi piú vicini al client che effettua la lettura e , ovviamente, in presenza di dati ridondanti risulta piú semplice soddisfare questo requisito.

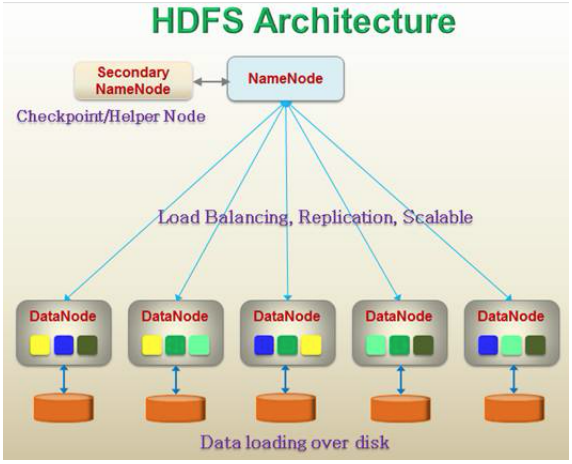


Figura 3.3: Architettura HDFS

Quanto detto fino ad ora, ci permette di asserire che quando vengono trattati file di grandi dimensioni HDFS é molto efficiente. Ma quando vengono

trattati file di piccole dimensioni, dove per piccole dimensioni si intendono dimensioni inferiori al blocco, HDFS risulta molto inefficiente perché i file utilizzano spazio all'interno del namespace, cioé l'elenco dei file mantenuti dal NameNode, che ha un limite dato dalla memoria del server.
Questo problema viene risolto compattando molti file piccoli in file piú grandi ai quali é possibile accedere dal sistema in parallelo e senza necessitá di espanderli.

3.2 MapReduce

MapReduce é un framework per la creazione di applicazioni in grado di elaborare grandi quantiá di dati in parallelo basandosi sul concetto di *functional programming*.
A differenza della programmazione *multithreading*, in cui i thread condividono i dati oggetto delle elaborazioni presentando una certa complessitá nel coordinare l'accesso alle risorse condivise, nel *functional programming*, invece, la condivisione dei dati é eliminata e i dati sono passati tra le funzioni come paramentri o valori di ritorno.
MapReduce lavora secondo il principio del *divide-et-impera*. Ció significa che suddivide l'operazione di calcolo in diverse parti, ognuna prepocessata in modo autonomo. Una volta che ogni parte del problema é stata calcolata, i vari risultati parziali sono ricomposti in un unico risultato finale.
É MapReduce stesso che si occupa dell'esecuzione dei vari task di calcolo, del loro monitoraggio e della ripetizione dell'esecuzione in caso di problemi o errori.
Il framework lavora attraverso i *compute node*, cioé i nodi di calcolo che si trovano assieme ai DataNode di HDFS. Infatti, lavorando congiuntamente con HDFS, MapReduce puó eseguire i task di calcolo sui nodi dove i dati sono giá presenti, aumentando cosí l'efficienza di calcolo. Una descrizione di un generico job MapReduce é mostrato nella figura 3.4.

Un job MapReduce é costituito da 4 componenti principali:

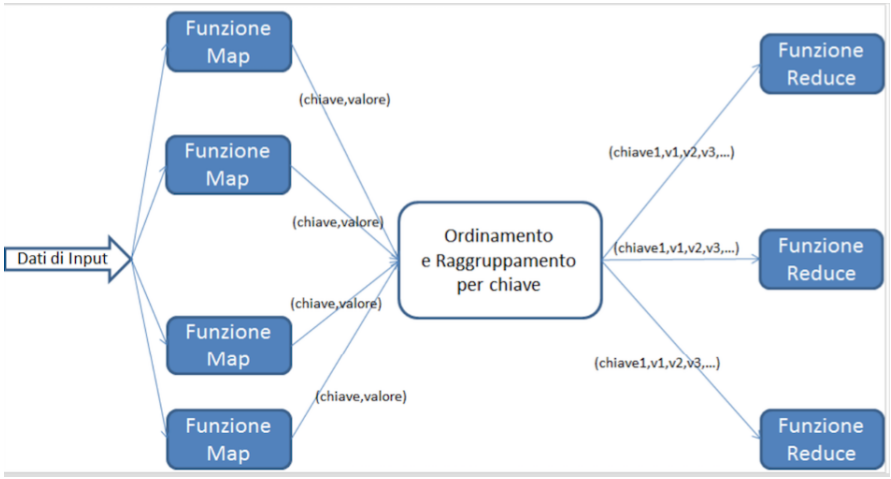


Figura 3.4: Fasi Job MapReduce

- i dati di input, memorizzati su HDFS;
- una funzione *map*, che trasforma i dati di input in una serie di coppie chiave/valore;
- una funzione *reduce* che, per ogni chiave, elabora i valori ad essa associati e crea, come output, una o piú coppie chiave/valore.
  L'esecuzione della funzione reduce é preceduta da una fase di raccolta delle coppie chiave/valore prodotte dalla funzione *map*. Le coppie sono ordinate per chiave e i valori con la stessa chiave vengono raggruppati;
- l'output, scritto su un file su HDFS.

3.3 YARN

Yarn (Yet Another Resource Negotiator) é una caratteristica principale della seconda versione di Hadoop. Prima di Yarn, uno stesso nodo del cluster, su cui stava in esecuzione il JobTracker, si occupava sia della gestione delle risorse del cluster sia della schedulazione delle attivitá delle applicazioni MapReduce.
Con l'avvento di YARN, i due compiti sono stati separati e sono svolti

rispettivamente dal ResourceManager e dall'ApplicationMaster. Inoltre, i TaskTracker presenti nei nodi del cluster per svolgere le operazioni di Map Reduce sono stati sostituiti dai NodeManager che si occupano di lanciare e monitorare i *container*, ovvero quei componenti che svolgono lavori specifici e a cui sono allocati una certa quantitá di risorse del nodo (RAM, CPU).
YARN permette di eseguire applicazioni diverse da MapReduce, tra cui Spark o Impala. In questo modo é possibile fare *stream processing* ed eseguire query interattive. Inoltre, YARN permette a piú utenti di connettersi al cluster e lanciare applicazioni diverse in maniera concorrente.
Il ResourceManager di YARN é l'entitá che governa il cluster decidendo l'allocazione delle risorse alle applicazioni concorrenti che sono in esecuzione. Le risorse vengono richieste dall'ApplicationMaster, il primo *container* allocato per un'applicazione.
Esso comunica con i NodeManager per inizializzare i *container* e monitorare la loro esecuzione.
Il NodeManager si occupa di creare e distruggere *container* e monitorare le risorse utilizzate in un nodo.
Il ciclo di vita di un'applicazione é il seguente:

- arriva una richiesta da un client per l'esecuzione di un'applicazione;
- il ResourceManager crea un container per l'ApplicationMaster;
- l'ApplicationMaster negozia con il ResourceManager le risorse per il container;
- viene eseguita l'applicazione, mentre l'ApplicationMaster ne monitora la corretta esecuzione;
- quando l'applicazione termina la sua esecuzione, l'ApplicationMaster libera le risorse comunicandolo al ResourceManager.

É da notare che é l'ApplicationMaster a garantire la *fault-tollerance* di un'applicazione, monitorando lo stato dei *container* e richiederne nuovi al

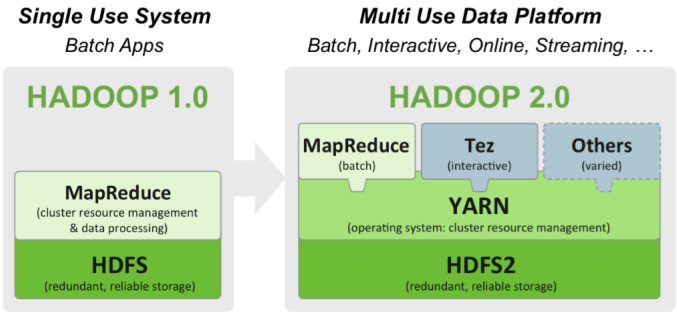


Figura 3.5: Versioni Hadoop

ResourceManager in caso di necessitá. Ció fa sí che il nodo contenente il ResourceManager non sia sovraccaricato permettendo, quindi, di avere una scalabilitá superiore.

3.4 Altri componenti dell'ecosistema Hadoop

Oltre le componenti principali descritte nelle sezioni precedenti, ne sono presenti altrettanto importanti. Quelli utilizzati per lo svolgimento di questo progetto di tesi sono i seguenti:

- **Hive**: é un sistema di data warehouse per Hadoop che consente di eseguire query e analisi di volumi di dati tramite HiveQL, un linguaggio di interrogazione basato su SQL.
  Puó essere usato per esplorare i dati in modo interattivo o per creare processi di elaborazione di batch riusabili. Hive prende le istruzione HiveQL e traduce in maniera automatica ed istantanea le query in uno o piú job MapReduce.
  Hive utilizza sempre un componente chiamato *metastore* in cui conserva tutti i metadati riguardanti le tabelle definite nel sistema. Puó essere anche esteso tramite funzioni definite dall'utente (denominate UDF), che consentono di implementare funzionalitá o logiche non facilmente modellabili in HiveQL.
  Le funzionalitá di Hive possono essere riassunte nei seguenti punti:

  - ETL (Extract, Trasform, Load): in Hive sono presenti strumenti per il caricamento e la manipolazione dei dati.
  - Con Hive é possibile dare ai dati una struttura (simile alle tabelle dei database relazionali)
  - Accesso ai dati presenti in HDFS o in altri sistemi quali HBase.
  - Utilizzo del framework MapReduce per l'esecuzione delle query.
  - Indicizzazione, per ottenere migliori performance di esecuzione.
  - Estensibilitá, attraverso funzioni create dall'utente.
  - Linguaggio simile a SQL, chiamato HiveQL.

- **Sqoop**: é uno strumento progettato per il trasferimento di dati tra Hadoop e i database relazionali. Puó essere usato per importare dati in HDFS da un sistema di gestione di database relazionali (RDBMS), ad esempio MySQL oppure Oracle. Sqoop dispone di un'interfaccia da riga di comando attraverso la quale si eseguono le istruzioni per la movimentazione dei dati.
  Inoltre, supporta la lettura incrementale di una tabella di un database relazionale (o di una query SQL) e la scrittura su file in HDFS. Nel processo di import genera una classe Java che rappresenta la struttura del record.Il collegamento con i database relazionali é realizzato attraverso driver JDBC.

- **Oozie**: é un sistema scalabile, affidabile ed estensibile per la schedulazione di workflow e serve per gestire i vari processi di Hadoop.

- **Spark**: é un framework open-source per l'analisi di grandi quantitá di dati, nato per essere veloce e flessibile come alternativa a MapReduce. É caratterizzato dalla capacitá di memorizzare risultati (solitamente parziali) in memoria centrale, a differenza di MapReduce, il quale memorizza obbligatoriamente i risultati delle computazioni su disco. L'utilizzo ottimale della memoria permette a Spark di essere ordini di

grandezza piú veloce, rispetto a MapReduce, nell'esecuzione di algoritmi che svolgono iterativamente le stesse istruzioni sui dati fino a che non é verificata una certa condizione.

Capitolo 4

IL CASO DI STUDIO

4.1 Parte I: L'architettura

La parte architetturale su cui si basa l'intero progetto viene mostrato nella figura seguente:

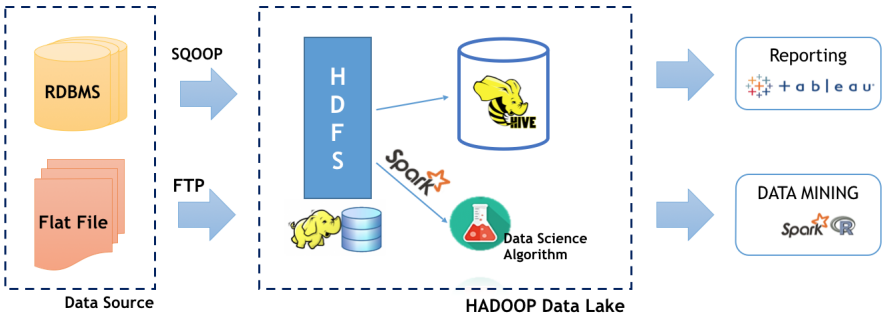


Figura 4.1: Architettura della piattaforma realizzata

Le sorgenti dati principali sono tabelle provenienti dai database relazionali aziendali e *flat file* provenienti da fonti esterne quali ad esempio siti web che raccolgono informazioni sia anagrafiche sia di tipo quantitativo sui prodotti venduti dall'azienda e dai suoi principali competitors sul mercato o file ".csv" caricati manualmente su HDFS.
Dopo che tutte le tabelle necessarie sono state caricati sul file system di Hadoop, tramite Hive sono state applicate le trasformazioni necessarie alla

creazione delle tabelle finali su cui sono stati costruiti i report finali.
Le tabelle presenti nel datawarehouse aziendale vengono importate all'interno di HDFS utilizzando Sqoop. Sqoop, come giá descritto nel Capitolo 4, é uno strumento di ETL che si collega ai database relazionali attraverso driver JDBC, generando una classe JAVA che rappresenta la struttura di ogni record.
Un esempio di template di un comando Sqoop é riportato nel seguente pseudo-codice:

```
1 - sqoop import
2 - connect dbc:Oracle://indirizzo_db/nome_db
3 - username nome_utente
4 - password password
5 - table tab1
6 - target-dir /folder_hdfs_1/folder_hdfs_2
```

La riga di comando n2 serve per la connessione tramite driver jdbc al database di tipo Oracle e richiede come parametri l'indirizzo del database ed il nome del database. Nella riga 3 deve essere inserito il nome utente. Nella riga 4 é necessario inserire la password associata al nome utente definito sopra. Nella riga 5 bisogna inserire il nome della tabella da importare su HDFS. Ed infine nell'ultima riga viene indicato il percorso di destinazione su HDFS in cui va inserita la tabella da importare.
Per alcune tabelle di dimensioni molto elevate e delle quali non é necessario un import massivo si é deciso di effettuare un import incrementale.
Un template di import incrementale tramite Sqoop é il seguente:

```
1 - sqoop import
2 - connect dbc:Oracle://indirizzodb/nomedb
3 -  username nomeutente
4 - password password
5 - table tab1
6 - target-dir /folderhdfs1/foldehdfs2
7 - incremental append
```

```
8 - check-column col1
9 - last-value 1000
```

Questo pseudocodice é simile al precedente con la differenza che é stato specificato un import incrementale con l'opzione append. Nel caso specificio saranno importate le righe che presentano un valore della colonna "col1" maggiore di 1000.
Le tabelle vengono importate in formato Avro. La struttura di ciascuna tabella viene catturata in un record Avro, che contiene le informazioni dell'header (un nome e un namespace facoltativo per qualificare il nome) e un array di campi.
Ogni campo é specificato con il suo nome e il tipo, oltre che con una stringa di documentazione (facoltativa). Per alcuni campi, il tipo non é un valore singolo, ma una coppia di valori, uno dei quali é null. Questo é il modo caratteristico di gestire le colonne che potrebbero contenere valori null.
Ad ogni tabella, inoltre, é associato lo schema Avro opportuno. Un esempio di schema Avro viene mostrato in figura 4.2.

Dopo che tutte le tabelle sono state importate tramite Sqoop, vengono create delle *external table* che puntano al file Avro contenente i dati e allo schema Avro che serve per leggere il file dati importato.
Per quanto riguarda i *flat file* essi vengono caricati direttamente su HDFS tramite l'interfaccia WEB e successivamente vengono create le *external table* in cui vanno indicati i nomi dei campi del file e il percorso di destinazione su HDFS. Ogni campo avrá il tipo impostato a Stringa di default.

Un esempio di creazione di tabella esterna viene mostrato in figura 4.4.
Dopo la creazione di tutte le tabelle esterne, vengono create le *parsing view* per ognuna di esse. Le *parsing view* non sono altro che delle viste in cui verranno effettuati i cast necessari.

Un esempio di *parsing view* é mostrato in figura 4.5.
Dopo la creazione di tutte le viste, si passa alla creazione del flusso ETL che punterá non alle tabelle esterne ma alle *parsing view* che contengono i dati e i vari campi con i tipi corretti.

```
{
  "type" : "record",
  "name" : "sqoop_import_DW_DIM_MATERIAL_PRICE",
  "doc" : "Sqoop import of DW.DIM_MATERIAL_PRICE",
  "fields" : [ {
    "name" : "MAT_ID_MATERIAL",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_ID_MATERIAL",
    "sqlType" : "2"
  }, {
    "name" : "MAT_ID_PPP_NA",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_ID_PPP_NA",
    "sqlType" : "2"
  }, {
    "name" : "MAT_ID_DEEP_NA",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_ID_DEEP_NA",
    "sqlType" : "2"
  }, {
    "name" : "MAT_SUGG_EU_PR_LIST",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_SUGG_EU_PR_LIST",
    "sqlType" : "2"
  }, {
    "name" : "MAT_EU_PPP_PR_LIST",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_EU_PPP_PR_LIST",
    "sqlType" : "2"
  }, {
    "name" : "MAT_DEEP_EU_PR_LIST",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_DEEP_EU_PR_LIST",
    "sqlType" : "2"
  }, {
    "name" : "MAT_DW_AUDIT_UPD",
    "type" : [ "string", "null" ],
    "columnName" : "MAT_DW_AUDIT_UPD",
    "sqlType" : "2"
  } ],
  "tableName" : "DW.DIM_MATERIAL_PRICE"
}
```


Figura 4.2: Esempio schema Avro

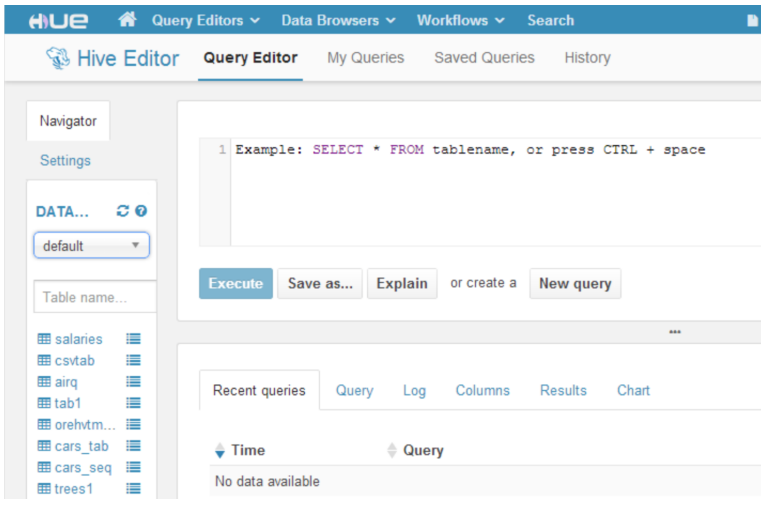


Figura 4.3: Interfaccia Web HUE

```
use ${hive_raw_db};

DROP TABLE IF EXISTS ${hive_raw_db}.price_data_discovery_gfk__ext;
CREATE EXTERNAL TABLE ${hive_raw_db}.price_data_discovery_gfk__ext(
 brand_comma_pattern string,
 type string,
 seasonality string,
 dimension string,
 width int,
 series int,
 inches string,
 li int,
 si_vmax string,
 run_flat string,
 extra_load string,
 oe_marking string,
 mount_year string,
 country string,
 sales_units double,
 sales_value_eur double,
 price_eur double
)
ROW FORMAT
DELIMITED FIELDS TERMINATED BY '\073'
STORED AS TEXTFILE
LOCATION '${sw_raw_file_path_standard}'
;
```

Figura 4.4: Esempio tabella esterna

```
use s__patriot;

drop view if exists s__patriot.dw__dim_material_price__pv;
create view s__patriot.dw__dim_material_price__pv
as
select cast(MAT_ID_MATERIAL as int) MAT_ID_MATERIAL,
 cast(MAT_ID_SUGG_EU as int) MAT_ID_SUGG_EU,
 cast(MAT_ID_PPP_EU as int) MAT_ID_PPP_EU,
 cast(MAT_ID_PPP_NA as int) MAT_ID_PPP_NA,
 cast(MAT_ID_DEEP_EU as int) MAT_ID_DEEP_EU,
 cast(MAT_ID_DEEP_NA as int) MAT_ID_DEEP_NA,
 cast(MAT_ID_DEEP_EXP as int) MAT_ID_DEEP_EXP,
 cast(MAT_SUGG_EU_PR_LIST as float) MAT_SUGG_EU_PR_LIST,
 cast(MAT_EU_PPP_PR_LIST as float) MAT_EU_PPP_PR_LIST,
 cast(MAT_NA_PPP_PR_LIST as float) MAT_NA_PPP_PR_LIST,
 cast(MAT_DEEP_EU_PR_LIST as float) MAT_DEEP_EU_PR_LIST,
 cast(MAT_DEEP_NA_PR_LIST as float) MAT_DEEP_NA_PR_LIST,
 cast(MAT_DEEP_EXP_PR_LIST as float) MAT_DEEP_EXP_PR_LIST,
 cast(MAT_DW_AUDIT_INS as int) MAT_DW_AUDIT_INS,
 cast(MAT_DW_AUDIT_UPD as int) MAT_DW_AUDIT_UPD
from s__patriot.dw__dim_material_price__ext;
```

Figura 4.5: Esempio Parsing View

Il flusso ETL é stato creato utilizzando HIVE. Sono stati creati vari script in liguaggio HIVE-QL ognuno dei quali effettua varie trasformazioni sui dati iniziali per ottenere le tabelle finali utili per un'analisi efficace e completa dell'intero datalake costruito.
Data la complessitá e la numerositá dei dati e delle tabelle presenti su HDFS si é reso necessario l'utilizzo di uno strumento per l'esecuzione sequenziale degli script. Questo strumento prende il nome di Oozie e consente la creazione di un *workflow* unico in cui vengono inseriti i vari script Hive in sequenza.

I workflow Oozie sono scritti in liguaggio XML.

Un esempio di workflow Oozie é il seguente:

```
<workflow-app name="r__price_data_discovery__f_distributor" xmlns="uri:oozie:workflow:0.4">
    <start to="GET_START_TIMESTAMP"/>
    <action name="GET_START_TIMESTAMP">
        <shell xmlns="uri:oozie:shell-action:0.1">
            <job-tracker>${jobTracker}</job-tracker>
            <name-node>${nameNode}</name-node>
            <exec>0050_get_start_timestamp.sh</exec>
            <file>0050_get_start_timestamp.sh</file>
            <capture-output/>
        </shell>
        <ok to="WDR__F_MATERIAL_BRAND"/>
        <error to="kill"/>
    </action>
    <action name="WDR__F_MATERIAL_BRAND">
        <hive2 xmlns="uri:oozie:hive2-action:0.1">
            <job-tracker>${jobTracker}</job-tracker>
            <name-node>${nameNode}</name-node>
            <jdbc-url>${jdbcUrl}</jdbc-url>
            <script>0200_wdr__f_material_brand_temp.hql</script>
            <param>reservoir_db=${reservoir_db}</param>
            <param>working_db=${working_db}</param>
            <param>swamp_patriot_db=${swamp_patriot_db}</param>
            <param>raw_file_db=${raw_file_db}</param>
            <file>0200_wdr__f_material_brand_temp.hql</file>
        </hive2>
        <ok to="IMPALA_METADATA"/>
        <error to="kill"/>
    </action>
    <action name='IMPALA_METADATA'>
        <shell xmlns="uri:oozie:shell-action:0.1">
            <job-tracker>${jobTracker}</job-tracker>
            <name-node>${nameNode}</name-node>
            <exec>sh</exec>
            <argument>0800_update_impala_metadata.sh</argument>
            <argument>${reservoir_db}</argument>
            <file>0800_update_impala_metadata.sh</file>
            <file>0801_update_impala_metadata_template.iql</file>
        </shell>
        <ok to="end" />
        <error to="kill" />
    </action>
    <kill name="kill">
        <message>Action failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>
    </kill>
      <end name="end"/>
</workflow-app>
```

Figura 4.6: Esempio workflow Oozie

All'interno di ogni *workflow* possono essere inserite delle azioni di tipo differente. Ad esempio, alcune azioni possibili sono le azioni di tipo shell o di tipo hive. Nel caso di azioni di tipo hive, ogni script viene salvato con estensione .hql e successivamente viene passato come parametro all'interno della sua specifica azione.

Ad ogni workflow é associato un file di configurazione, in formato XML, in cui vengono passati i nomi dei database necessari all'esecuzione dei vari script e altri parametri quali ad esempio il tipo di driver jdbc e l'indirizzo IP del server Hive.

```
<configuration>
    <property>
        <name>oozie.use.system.libpath</name>
        <value>true</value>
    </property>
    <property>
        <name>hiveServer</name>
        <value>10.131.203.175</value>
    </property>
    <property>
        <name>working_db</name>
        <value>w__price_data_discovery</value>
    </property>
    <property>
        <name>reservoir_db</name>
        <value>r__price_data_discovery</value>
    </property>
    <property>
        <name>raw_file_db</name>
        <value>s__raw_file</value>
    </property>
    <property>
        <name>jdbcUrl</name>
        <value>jdbc:hive2://${hiveServer}:10000/${reservoir_db}</value>
    </property>
</configuration>
```

Figura 4.7: Esempio file configurazione workflow

Un esempio di struttura del file di configurazione viene mostrato in figura 4.7. Ogni workflow, quindi, ha lo scopo di creare le tabelle principali a cui punterá il tool di reportistica "Tableau" nell'ultima fase del progetto.
Poiché il numero di workflow creati é elevato si é reso necessario l'utilizzo di uno strumento per coordinare la loro esecuzione. Lo strumento utilizzato é Talend. Talend é uno strumento di ETL, di tipo "drag and drop", i cui componenti principali e utilizzati per lo scopo da noi prefissato, ovvero il coordinamento dei vari workflow Oozie, sono i seguenti:

- tLibraryLoad: permette di caricare nell'ambiente di lavoro le librerie e i jar necessari;
- tJava: permette l'esecuzione di codice Java custom;
- tParallelize: separa l'esecuzione del job corrente in piú thread e viene utilizzato per eseguire workflow in parallelo;
- tLogCatcher: utilizzato per gestire le eccezioni;
- tFileOutputDelimited: utilizzato per scrivere su un file i messaggi di log.

Ogni job Talend viene quindi eseguito periodicamente, utilizzando il comando "crontab" di Windows. Questo comando permette di lanciare i vari job ad una determinata ora del giorno, nel caso di aggiornamenti giornalieri, o una volta alla settimana, nel caso di aggiornamenti settimanali e cosí via.

4.2 Parte II: Analisi

4.2.1 Analisi dei requisiti e contesto dati

I requisiti principali su cui si basa la costruzione della piattaforma tecnologica oggetto dello studio sono i seguenti:

- Costruire una piattaforma unica per tutti i mercati, customizzata in accordo con le singole richieste locali;
- Analizzare in maniera veloce e rappresentativa il trend di prezzo e di volume per capire il posizionamente dell'azienda e dei suoi competitors nel mercato europeo ed extra-europeo;
- Facilitare l'integrazione di eventuali nuovi sorgenti dati;
- Simulare l'impatto che potrebbe avere una modifica sulle condizioni di vendita;
- Utilizzare le analisi effettuate sul trend di prezzo per consentire la ricostruzione dell'intera *Value Chain*.

I dati utilizzati per la creazione dei report finali utili al raggiungimento degli obiettivi indicati nei requisiti sopra citati provengono da diverse fonti. Le principali fonti dati sono le seguenti:

- Database di proprietá aziendale
- Row File
- Dati provenienti da siti web specializzati nella vendita online dei prodotti

Nella sezione successiva si entrerá maggiormente nel dettaglio delle tabelle di input utilizzate e delle trasformazioni applicate ad esse.

4.2.2 Data Undertanding e data Manipulation

La parte di analisi del progetto puó essere sintetizzata in figura 4.8.

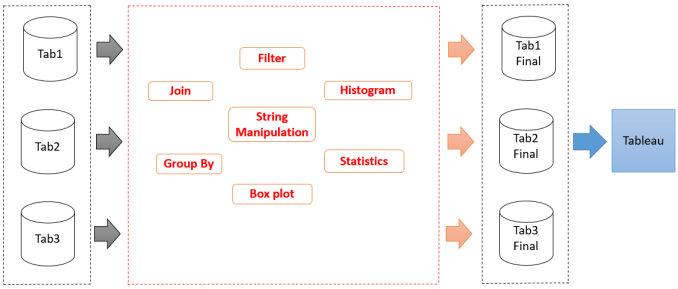


Figura 4.8: Flusso sintetico fase analitica

L'obiettivo della fase analitica é trasformare e aggregare i dati provenienti da piú tabelle per ottenere le tabelle finali contenenti le informazioni necessarie all'utente per le sue analisi di mercato.
Le tabella finali create per la costruzione del sistema di reportistica utile al supporto delle desicisioni aziendali sono quattro:

1. Anagrafica dei prodotti
2. Distributori
3. Tabella dei volumi gestionali di piano
4. Tabella dei prezzi

La prima tabella contiene i dati dell'anagrafica dei prodotti. Essa é stata ottenuta mettendo in *join* le informazioni provenienti da due tabelle principali:

- Anagrafica 1 che contiene l'anagrafica dei prodotti venduti esclusivamente dall'azienda committente;

- Angrafica Tot che contiene l'anagrafica di tutti i prodotti venduti sul mercato dalle principali aziende. I dati contenuti in questa tabella provengono da una delle più grandi piattaforme B2B per il commercio di tali prodotti.

Ogni record della tabella finale descrive le caratteristiche tecniche di ogni prodotto ed é ottenuta estraendo i campi di interesse dalle due tabelle di imput, privilegiando l'informazione proveniente dalla tabella Anagrafica 1 poiché é ritenuta piú affidabile in quanto le informazioni in essa contenute sono state validate dal reparto Marketing e Pricing dell'azienda ed integrando, nel caso di campi con valori mancanti, i dati provenienti dalla tabella Anagrafica Tot.

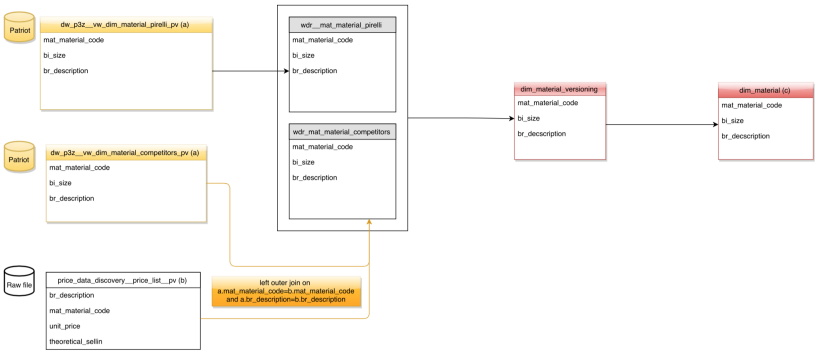


Figura 4.9: Schema logico della tabella anagrafica dei prodotti

In figura 4.9 viene mostrato lo schema logico che ha portato alla costruzione della tabella di anagrafica.
Nello schema logico in figura 4.9 il colore giallo é associato alle tabelle provenienti dal datawarehouse aziendale, il colore bianco ai file csv, il colore grigio alla tabelle intermedie create per l'ottenimento della tabella finale ed infine il colore rosso é associato alla tabella finale.
Dalla tabella Anagrafica 1, che contiene i dati anagrafici dei soli prodotti aziendali, sono stati trasformati alcuni campi per ottenere una maggiore chiarezza sul loro significato:

- Il campo *Seasonality*, che in origine era un campo di tipo Integer, é stato trasformato in Stringa secondo la seguente regola:
  - se il campo assume valore uguale 1 allora esso assume valore di tipo stringa = SUMMER;
  - se il campo é uguale a 2 allora assume valore di tipo stringa = "WINTER";
  - se il campo é uguale a 10 allora assume valore di tipo stringa = "ALL SEASONS";
  - in tutti gli altri casi assumerá valore = "UNKNOWN".
- Il campo Stato gestionale é stato trasformato secondo la regola seguente:
  - se il campo é uguale a 2 allora esso assume valore = GAMMA;
  - altrimenti assume valore uguale a "PHASE OUT".

  Questa trasformazione consente le suddivisione dei prodotti in due categorie: prodotti ancora in commercio, ovvero in Gamma, e prodotti non piú commercializzati.

Dalla tabella Anagrafica Tot, invece, sono stati selezionati solamente i record delle prime 6 aziende leader di mercato e i dati dei prodotti appartenenti alle categorie "CAR", "VAN" e "SUV".
Anche in questo caso, sono state applicate le stesse trasformazioni riguardanti la stagionalitá e la classificazione dei prodotti descritte sopra.
Uno dei problemi riscontrati durante la creazione di questa prima tabella riguarda il campo "descrizione prodotto". Si é notato, infatti, che alcuni prodotti presentavano degli errori sintattici nelle due tabelle di input.
Lo stesso prodotto, pur avendo la stessa descrizione, presentava dei caratteri a volte minuscoli e a volte maiuscoli. Ció avrebbe portato ad una errata costruzione della tabella poiché si sarebbero create righe duplicate per lo stesso prodotto.
La soluzione adottata per risolvere questo inconveniente é stata trasformare

tutte le stringhe del campo "descrizione prodotto" in caratteri minuscoli.
La seconda tabella é stata denominata "Distributori" e contiene i dati delle rilevazioni di prezzo settimanali per ogni distributore, la cui lista é stata estrapolata dai dati provenienti dalla piattaforma B2B citata sopra.
Le tabelle di input necessarie alla creazione di questa tabella sono:

- Daily Price che contiene le rilevazioni di prezzo giornaliere per ciascun distributore;
- Anagrafica D che contiene l'anagrafica dei distributori;
- Anagrafica dei prodotti che contiene l'anagrafica unificata dei prodotti creata precedentemente;

La tabella "Distributori" é stata costruita in due fasi:

- Nella prima fase viene generata una tabella temporanea in cui viene calcolato il prezzo medio settimanale di ciascun distributore.
  Questo prezzo é ottenuto calcolando la media pesata sullo stock venduto da ogni rivendito e filtrando i record aventi uno stock minore di 8 poiché sono considerati non rilevanti ai fini delle analisi di mercato.
- Nella seconda fase viene calcolato, invece, il rank di ogni distributore per ogni settimana e per ogni prodotto, dove rank=1 indica che quel determinato distributore ha applicato il prezzo piú basso in quella settimana per quel determinato prodotto. Questa metrica é stata utilizzata per calcolare il posizionamento di un distributore rispetto ai rimanenti presenti sul mercato.

In figura 4.10 viene mostrato lo schema logico con tutti i vari passaggi che hanno portato alla creazione della tabella finale dei distributori.
Le trasformazioni applicate per l'ottenimento della tabella finale sono le seguenti:

- Sono stati filtrati i record aventi il campo denominato "canale distribuzione" diverso da 115 e il campo "codice mercato" diverso da 711;

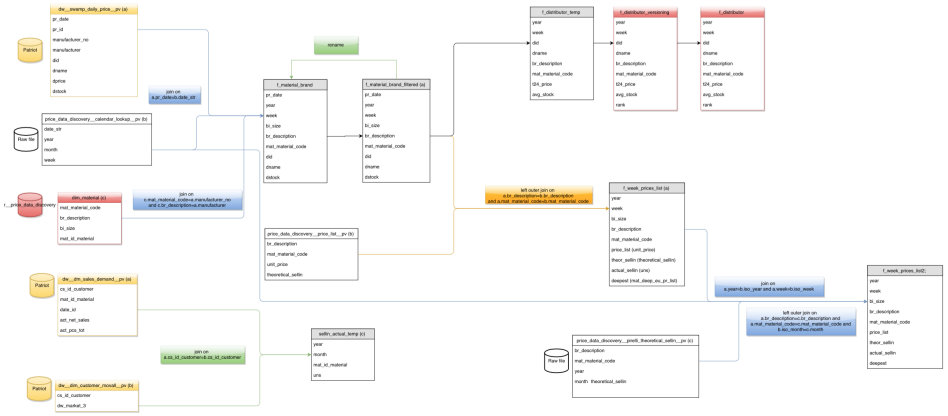


Figura 4.10: Schema logico della tabella dei distributori

- il campo "data", essendo un timestamp é stato modificato per ottenere il campo year come data/1000 e il campo mese applicando la funzione substring.

- é stato creato un campo denominato "UNS" ottenuto come rapporto tra i pezzi venduti ed il totale dei pezzi disponibili. Se uno di questi due valori é uguale a zero allora il campo viene settato a NULL.

Un'altra tabella di fondamentale importanza per le analisi finali é la tabella dei volumi gestionali di piano che contiene le informazioni riguardanti i volumi di vendita pianificati dall'azienda per ogni anno e per ogni prodotto. Questa tabella é stata ottenuta a partire da tre tabelle di input:

- Tabella Volumi Piano che contiene le informazioni dei volumi di vendita calcolate dal reparto Marketing e Prodotto dell'azienda in base a stime e previsioni derivanti dai valori a consuntivo degli anni precedenti.

- Anagrafica Clienti

- Anagrafica Prodotti

La costruzione della tabella finale é stata creata aggregando i volumi di vendita di ciascun prodotto a livello annuale e filtrando i record che non soddisfavano i seguenti requisiti:

- Codice regione geografica = "EU"
- Codice canale di distribuzione = 115
- Codice Prodotto con ultime due cifre uguali a zero
- Codice Unitá Business = "CAR"

Infine, la tabella dei prezzi contiene le informazioni, per ogni settimana e per ogni prodotto, di tutti i prezzi, (quali ad esempio i prezzi di listino, i prezzi di vendita tra azienda e rivenditori, i prezzi di vendita tra rivenditori e clienti finali, i prezzi di vendita tra azienda e cliente finale senza intermediari), dei volumi di vendita e dei primi 3 rivenditori per ogni prodotto (questa informazione viene estrapolata dalla tabella dei distributori).
Le tabelle di input utilizzate per la creazione di questa tabella finale sono molteplici:

- Tabella con i prezzi di listino dell'azienda;
- Tabella daily price con i prezzi giornalieri di tutti i rivenditori e delle principali aziende;
- Tabella prezzi sell-out con i prezzi di vendita tra rivenditore e cliente finale;
- Tabella con i dati di vendita dell'azienda;
- Tabella con i prezzi di listino delle aziende concorrenti;
- Tabella dei distributori con i prezzi settimanali di ciascun rivenditore;
- Anagrafica totale dei prodotti
- Tabella con i volumi di vendita pianificati dall'azienda.

In figura 4.11 viene mostrato lo schema logico per la creazione della tabella dei prezzi.
Durante la creazione di questa tabella é stata implementata una regola

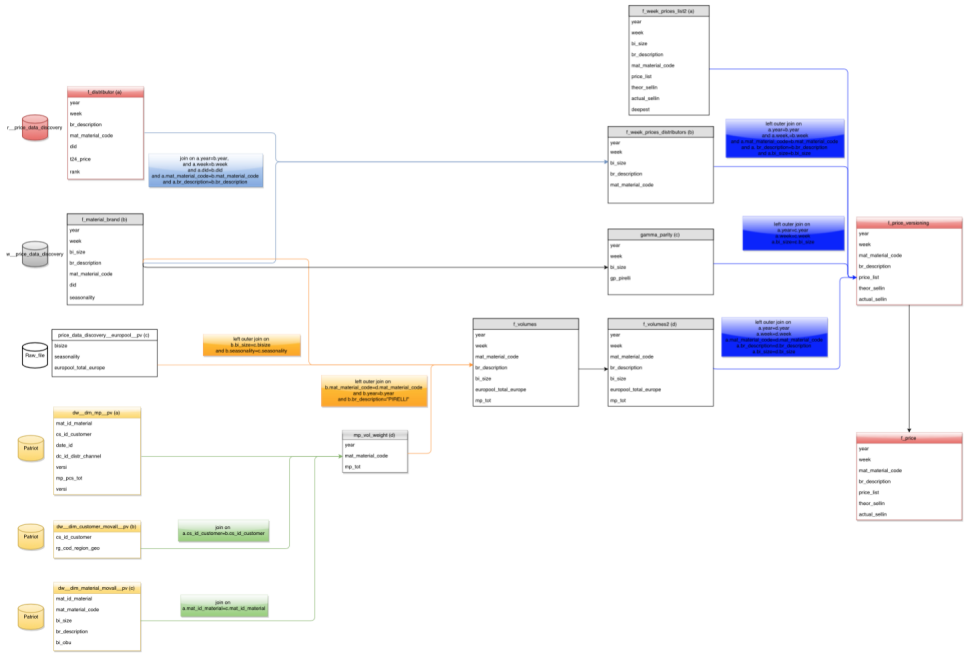


Figura 4.11: Schema logico della tabella dei prezzi

riguardante la lunghezza dei codici prodotto. Questa regola é stata introdotta poiché i dati provenienti dalle diverse sorgenti presentavano discrepanze sui codici di alcuni prodotti.
Si é notato, infatti, che un unico prodotto risultava avere codici diversi a seconda della fonte dati prescelta.
Le regole implementate sono distinte per Brand:

- Per ogni prodotto venduto dal Brand 1
  - se la lunghezza del codice del prodotto é uguale a 5 vengono aggiunti 2 zero alla fine;
  - se la lunghezza del codice é uguale a 6 viene aggiunto uno zero all'inizio del codice;
  - in tutti gli altri casi viene mantenuto il codice originale.
- Per ogni prodotto venduto dal Brand 2

  - se la lunghezza del codice é pari a 5 viene aggiunto uno zero all'inizio della stringa;
  - in tutti gli altri casi il codice non viene modificato;

- Per ogni prodotto venduto dal Brand 3 se la lunghezza del codice é maggiore o uguale a 9:
  - se il secondo carattere a partire da sinistra é uguale a 1 vengono presi i primi 7 caratteri a partire dal secondo carattere;
  - se il secondo carattere é invece diverso da 1 vengono presi i primi 6 caratteri a partire dal secondo carattere;
  - in tutti gli altri casi il codice del prodotto rimane inalterato
- Per ogni prodotto commercializzato dal Brand 4 vengono eliminati tutti gli zero iniziali;
- Per i prodotti venduti dagli altri Brand principali il codice del prodotto non viene modificato

Un altro aspetto molto importante che é stato considerato riguarda la qualitá dei dati, che rappresenta un punto cruciale al fine di ottenere tabelle i cui dati siano significativi e utili al top management.
Per ognuna delle tabelle utilizzate come sorgente sono stati analizzati il numero di outlier presenti, ovvero valori anomali rispetto alla maggior parte dei dati, ed il numero di valori mancanti o errati.
Per quanto riguarda gli outlier, attraverso l'utilizzo dei box plot, si é scoperto che alcuni prodotti presentavano dei prezzi oggettivamente errati poiché molto superiori rispetto al prezzo medio dello stesso prodotto.
Per eliminare gli eventuali outlier si é deciso di applicare la seguente regola: filtrare tutti i record il cui prezzo é superiore a 3 volte la media del prezzo calcolato in ogni settimana.
Per quanto riguarda l'analisi dei valori mancanti si é deciso di applicare i seguenti criteri risolutivi:

- Per i campi di tipo numerico che presentano un valore NULL derivante dalle trasformazioni applicate precedentemente si é deciso di lasciare il campo senza un valore specifico.

- Per i campi di tipo stringa, quali il nome prodotto o la descrizione dello stesso, nel caso di prodotto proprio dell'azienda si é deciso di creare una lista con tutti i valori mancanti per ogni prodotto che é stata successivamente modificata dal reparto Marketing dell'azienda. Viceversa, nel caso di prodotti di altri competitors, si é deciso di non applicare nessuna modifica poiché non si era in grado di conoscere con esattezza il valore di quei campi mancanti.

Negli altri attributi delle tabelle non si sono rilevati ulteriori anomalie o errori. Una ulteriore analisi é stata effettuata considerando l'andamento dei volumi di vendita negli ultimi due anni disponibili (2014 e 2015). In particolare si é calcolato l'andamento della distribuzione mediana delle 6 serie temporali e del conseguente delta, positivo o negativo, tra ogni distribuzione dei rispettivi brand e la mediana.
Questa analisi consente lo studio dell'andamento delle vendite di ciascun brand rispetto all'andamento mediano del mercato di cui fa parte.
Si suppone, infatti, che un andamento con poche oscillazioni significhi che il volume delle vendite segua in maniera pressoché costante l'andamento delle vendite dell'intero mercato. Tuttavia, il caso piú interessante si manifesta quando si hanno dei picchi, sia positivi che negativi, rispetto all'andamento medio del mercato.
Un picco negativo significa che quel determinato Brand in un determinato mese o range di tempo ha avuto un crollo nelle vendite che non si é manifestato nelle aziende concorrenti. E quindi sarebbero necessarie ulteriori analisi più approfondite per capire la causa e l'origine di questa discrepanza.
Viceversa, un picco positivo significa che il Brand considerato ha venduto in un determinato periodo di tempo molti più prodotti rispetto al resto del mercato. Anche in questo caso, quest'analisi é molto utile perché rappresenta un campanello d'allarme per il top management. Infatti un elevato volume

delle vendite rispetto all'intero mercato circostante potrebbe dipendere sia da un abbassamento dei prezzi o aumento dei prezzi delle aziende concorrenti, sia da una maggiore pubblicitá dei prodotti.

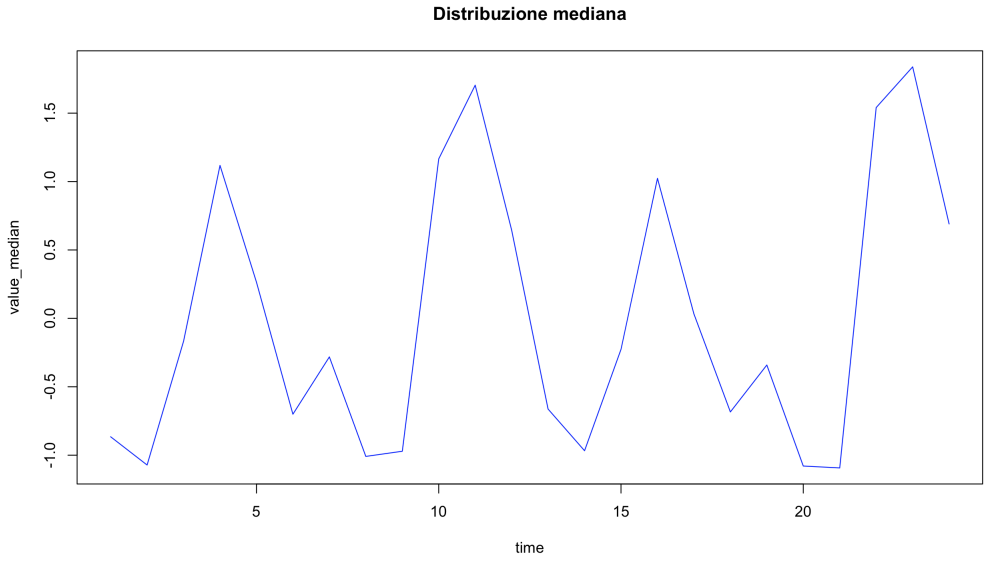


Figura 4.12: Rappresentazione Mediana

Come é possibile notare nelle figure successive:

- Il Brand 1 presenta 5 picchi positivi in corrispondenza dei mesi di Febbraio 2014, Settembre 2014, Novembre 2014 e Maggio 2015 e Novembre 2015 e 4 picchi negativi in corrispondenza di Aprile 2014, Ottobre 2014, Marzo 2015 ed Ottobre 2015;

- Il Brand 2 presenta due picchi positivi in corrispondenza di Aprile 2014 e Dicembre 2014 e tre picchi negativi nei mesi di Agosto2014, Ottobre 2014 e Ottobre 2015;

- Il Brand 3 presenta 4 picchi positivi in corrispondenza dei mesi Aprile 2014, Ottobre 2014, Aprile 2015 e Ottobre 2015 e 3 picchi negativi in corrispondenza di Luglio 2014, Dicembre 2014 e Novembre 2015;

- Il Brand 4 presenta 5 picchi positivi in corrispondenza dei mesi Gennaio, Marzo e Luglio 2014 e Luglio 2015 e 4 picchi negativi in Maggio 2014, Ottobre 2014, Febbraio 2015 e Maggio 2015;
- Il Brand 5 presenta 4 picchi positivi nei mesi Marzo 2014, Ottobre 2014, Febbraio 2015 ed Aprile 2015 e 3 picchi negativi in corrispondenza di Luglio 2014, Luglio 2015 e Novembre 2015;
- Il Brand 6 presenta5 picchi positivi nei mesi Luglio 2014, Ottobre 2014, Gennaio 2015, Luglio 2015 e Novembre 2015 e 2 picchi negativi nei mesi di Maggio 2014 e Aprile 2015.

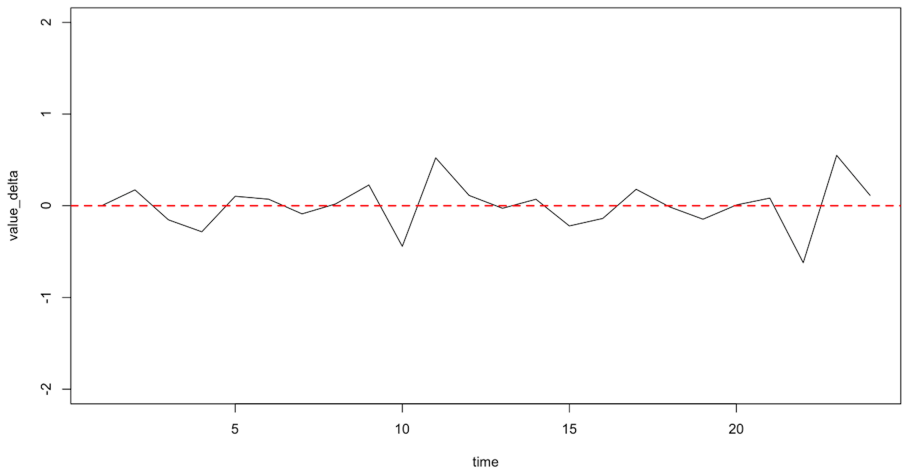


Figura 4.13: Rappresentazione delta Brand 1

Dalle analisi viste sopra emerge che, ad esempio, nei mesi di Ottobre 2014 e 2015 i due Brand 1 e 2 presentano dei picchi negativi rispetto all'andamento mediano del mercato. Nello stesso mese, peró, si nota che gli altri Brand 3, 4 e 6 presentano dei picchi positivi, ovvero hanno un incremento delle vendite rispetto agli altri competitor. Invece il Brand 5 ha un picco positivo nell'anno 2014 ed un picco negativo del 2015.
Questo andamento non conforme tra tutte le aziende del mercato puó derivare da un aumento dei prezzi da parte dei Brand 1 e 2 (che rappresentano coloro che detengono una maggior quota di mercato) con consequente diminuzione

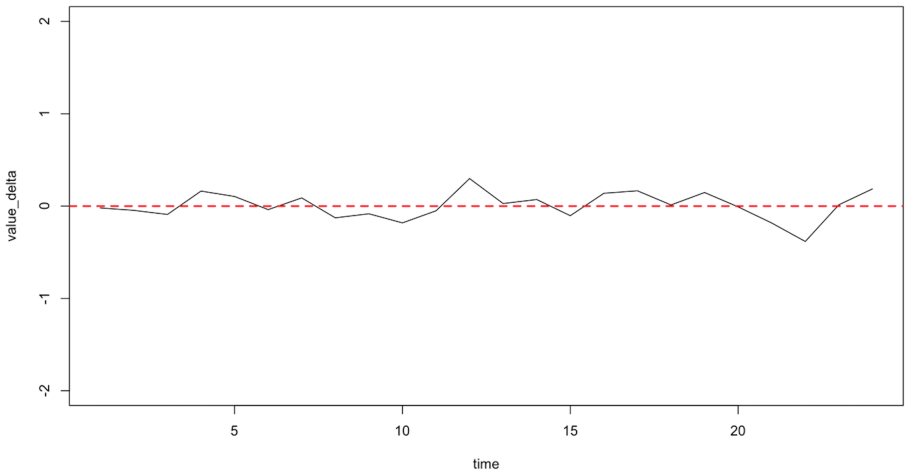


Figura 4.14: Rappresentazione delta Brand 2

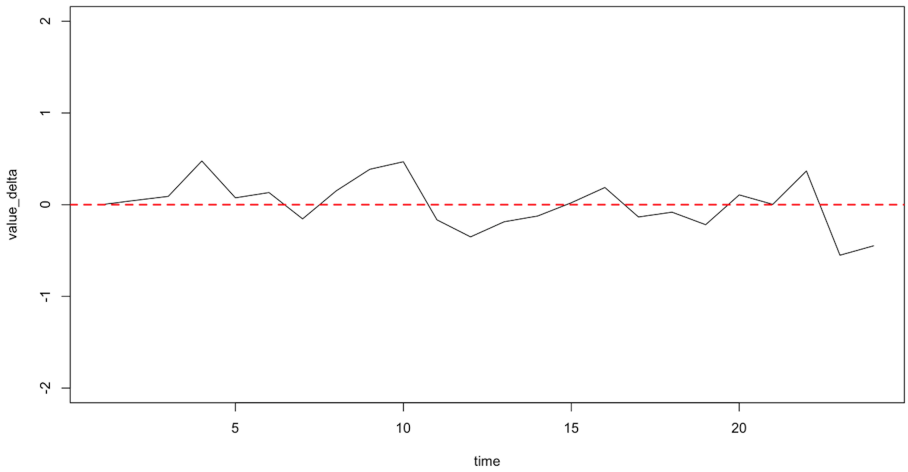


Figura 4.15: Rappresentazione delta Brand 3

delle vendite a favore degli altri Brand che tendenzialmente offrono prodotti con prezzi inferiori e che hanno beneficiato di questo aumento dei prezzi per incrementare le loro vendite.

Un'altra considerazione riguarda i due Brand 3 e 4, i quali presentano una diminuzione delle vendite rispetto alla media del mercato nei mesi di Luglio 2014 e Novembre 2015, viceversa i Brand 1 e 6 negli stessi mesi hanno un incremento delle vendite.

Anche in questo caso, questo comportamento potrebbe derivare da un cam-

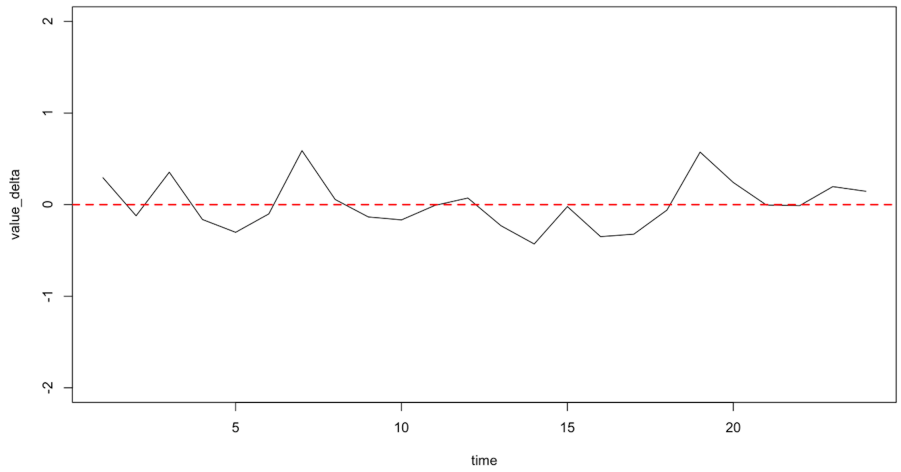


Figura 4.16: Rappresentazione delta Brand 4

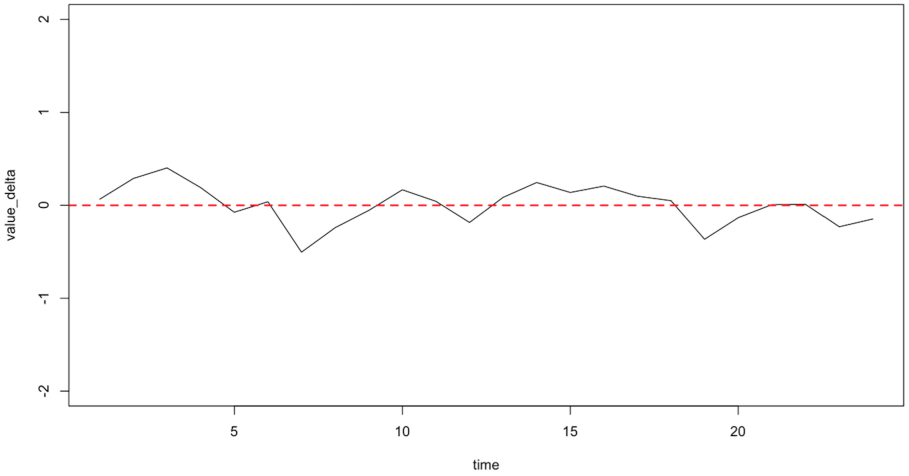


Figura 4.17: Rappresentazione delta Brand 5

biamento di prezzo da parte di alcuni dei Brand principali del mercato, cambiamento che influenza anche le scelte di prezzo dei competitors, con conseguente aumento o diminuzione della quantitá venduta.
Molto probabilmente, dato che questo andamento si verifica nei mesi di Luglio e Novembre, che possono essere considerati come i mesi finali di una campagna di vendita Summer e Winter rispettivamente, si puó supporre che in quel periodo i due Brand che presentano dei picchi positivi hanno diminuito i loro prezzi per aumentare la quantitá venduta e diminuire il numero di scorte

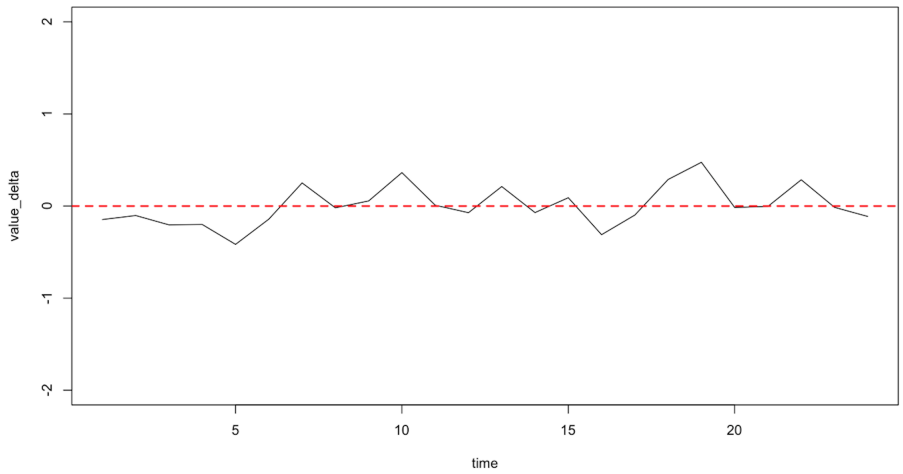


Figura 4.18: Rappresentazione delta Brand 6

presenti in magazzino.
Un'altra analisi che emerge dagli andamenti dei Brand nel mercato riguarda il Brand 1 che presenta un picco negativo nel mese di Aprile 2014, a differenza dei Brand 2 e 3 in cui si ha invece un aumento delle vendite. Gli altri Brand si attestano su valori mediani.
Una possibile giustificazione di questo andamento potrebbe essere quella di una diminuzione di prezzo su determinati tipi di prodotti con caratteristiche simili ai prodotti venduti dal Brand 1, con conseguente aumento delle loro vendite.
Questo tipo di azione puó essere giustificata dal fatto che questi 3 Brand sono quelli che occupano la porzione piú grande del mercato e che, quindi, hanno l'obiettivo di aumentare ancora di piú la loro quota di mercato a discapito dei competitors.
Nei restanti mesi si nota un andamento abbastanza lineare rispetto all'andamento mediano dell'intero mercato.

4.2.3 Data mining: Motif discovery

La *Motif Discovery* consente di scoprire pattern ricorrenti, sconosciuti in precedenza, all'interno di serie temporali. Questi pattern prendono il nome di

motif. Al fine di trarre conoscenza e vantaggio competitivo dai dati disponibili, é stato applicato alle serie temporali dei principali 6 Brand oggetto di questo studio un algoritmo il cui obiettivo é trovare tali *motif* all'interno delle serie storiche. Nello specifico, si é deciso di applicare l'algoritmo alle serie temporali riguardanti l'andamento dei volumi di vendita in un arco temporale di due anni.
L'approccio proposto in questo lavoro si basa su una rappresentazione simbolica delle serie storiche utilizzando l'algortimo denominato SAX (Symbolic Aggregate Approximation) ([5],[6]).
Un grande vantaggio di questa rappresentazione é la possibilitá di ridurre estremamente la quantitá di dati senza che questi perdano le proprie caratteristiche e le informazioni necessarie per una corretta analisi.
Infatti, mediante SAX, é possibile ridurre una serie temporale a valori reali di lunghezza n in una stringa di simboli di lunghezza w (tipicamente $w \ll n$). Ciascun carattere puó essere scelto da un alfabeto Σ di grandezza α, specificabile dall'utente. La scelta di w e di α dipende dai requisiti applicativi.
Durante il primo step, ogni serie temporale originale viene normalizzata. Infatti, confrontare serie temporali con *offset* ed ampiezza differenti acquista significato solo se esse subiscono prima un processo di normalizzazione, atto a renderle di uguale valore medio e uguale deviazione standard.
Esistono varie forma di normalizzazione dei dati; in questo lavoro di tesi si é deciso di utilizzare la *z-normalization*:

$$Z = \frac{X - \mu}{\sigma}$$

Essa richiede in input un vettore x il quale viene trasformato in un array x' i cui valori hanno media uguale zero mentre la deviazione standard (e quindi anche la varianza) varrá uno.
Questa trasformazione richiede due operazioni:

- la media μ della serie temporale viene sottratta da ciascuno degli elementi dell'array di input

- ogni valore ottenuto al punto precedente viene diviso per la deviazione standard σ della serie temporale iniziale.

Dopo che ogni serie temporale é stata normalizzata, ognuna di esse subisce un processo di approssimazione con conseguente trasformazione nella rappresentazione PAA (*Piecewise Aggregate Approximation*).
Tale trasformazione converte una serie temporale $T = t_1, t_2, ..., t_n$ in una serie temporale T' di lunghezza w, dove solitamente $w \ll n$.
T' é calcolata dividendo T in w segmenti della medesima dimensione (chiamati *frame*), ciascuno dei quali viene mappato in un elemento della nuova sequenza.
Vengono, quindi,calcolati i valori medi dei dati appartenenti a ciascun *frame* ed un vettore contenente tali medie diventa la rappresentazione ridotta dei dati di partenza.
Nell'ultima fase dell'approccio SAX, la rappresentazione PAA viene sostituita da una stringa di simboli, ciascuno dei quali é selezionato da un alfabeto Σ.
Per riflettere al meglio le caratteristiche della serie temporale, una discretizzazione ideale dovrebbe produrre simboli equiprobabili.
Tale risultato é ottenibile grazie al fatto che quasi tutti i segnali normalizzati possiedono una distribuzione Gaussiana.
Al fine di utilizzare tale risultato per la discretizzazione mediante SAX, l'area sottostante la distribuzione normale (di media 0 e varianza 1) viene suddivisa in α regioni delimitate dai *breakpoint* $B = \beta_0, \beta_1, ..., \beta_\alpha$.
Tutti i coefficienti della rappresentazione PAA il cui valore é inferiore a β_1 vengono mappati sul simbolo $\alpha_1 = a$, tutti quelli piú grandi o uguali a β_1 e minori di β_2 sono rappresentati da $\alpha_2 = b$, etc.
La concatenazione diei w simboli risultante viene chiamata **parola**.
I *breakpoint* non dipendono dai dati di input ma solo dalla dimensione dell'alfabeto. É, quindi, possibile costruire una *lookup table* di *breakpoint* per ciascun valore assegnabile ad α.

La tabella 4.19 mostra i *breakpoint* per un alfabeto avente una dimensione che varia tra 2 e 10 simboli.

	2	3	4	5	6	7	8	9	10
β_1	0	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	-	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3	-	-	0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4	-	-	-	0.84	0.43	0.18	0	-0.14	-0.25
β_5	-	-	-	-	0.97	0.57	0.32	0.14	0
β_6	-	-	-	-	-	1.07	0.67	0.43	0.25
β_7	-	-	-	-	-	-	1.15	0.76	0.52
β_8	-	-	-	-	-	-	-	1.22	0.84
β_9	-	-	-	-	-	-	-	-	1.28

Figura 4.19: Lookup table di breakpoint

Utilizzando tale tabella é possibile convertire tutti i valori reali di ciascuna rappresentazione PAA nei corrispondenti simboli di Σ.
Nelle figure sottostanti é possibile vedere le rappresentazioni PAA delle serie temporali dei principali Brand; l'arco temporale considerato è 24 mesi ed ogni serie temporale rappresenta l'andamento mensile dei volumi all'interno dell'arco temporale prescelto.

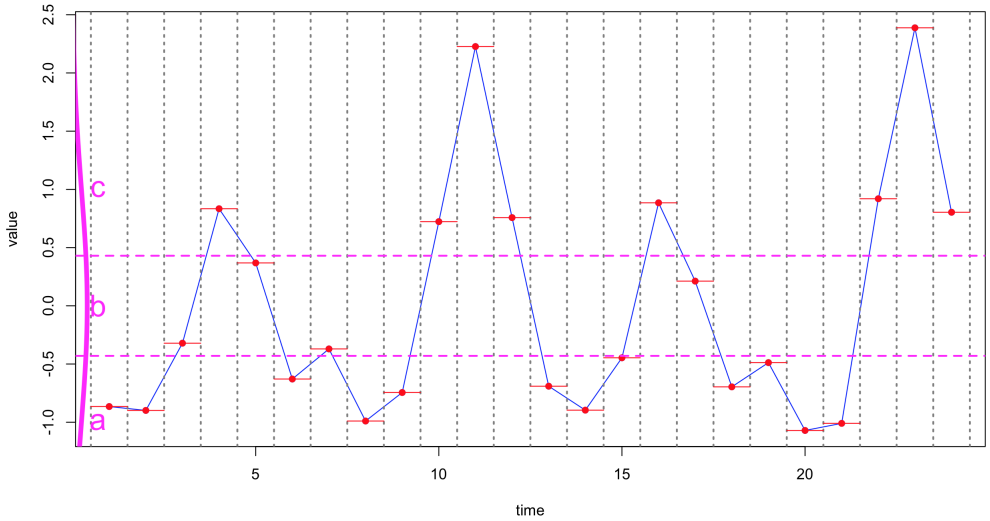


Figura 4.20: Discretizzazione SAX della serie temporale del Brand 1. Nell'esempio, con $\alpha = 3$ e $w = 24$, la parola risultante é **aabcbabaacccaaacbaaaaccc**

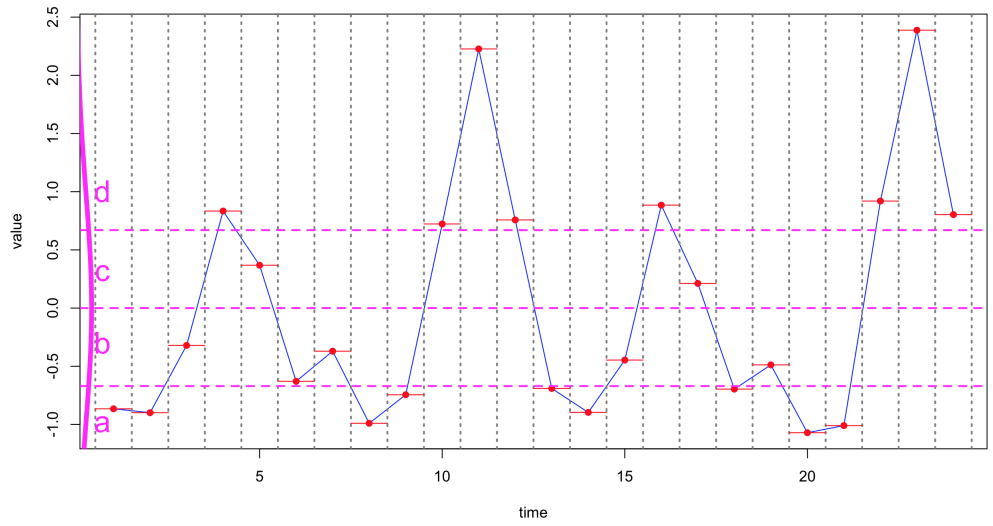


Figura 4.21: Discretizzazione SAX della serie temporale del Brand 1. Nell'esempio, con $\alpha = 4$ e $w = 24$, la parola risultante é **aabdcbbaadddaabdcabaaddd**

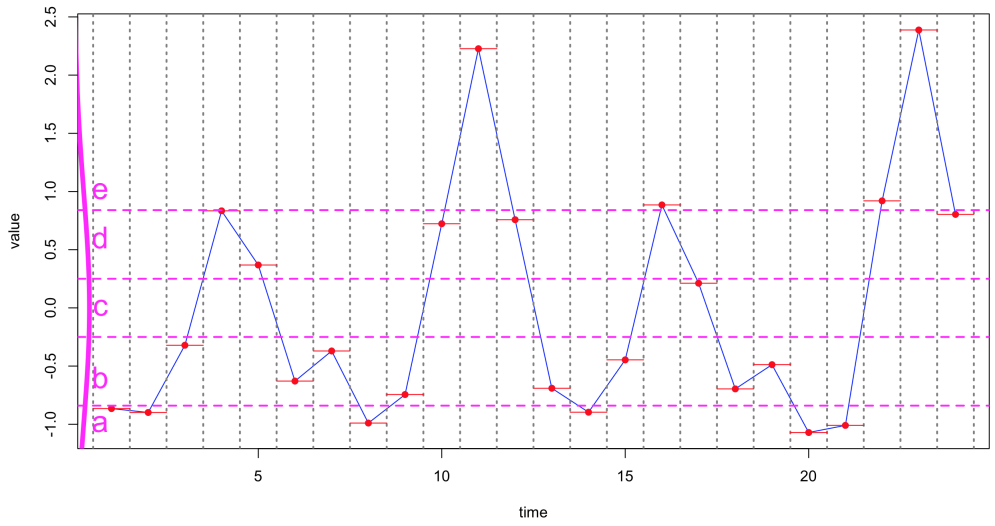


Figura 4.22: Discretizzazione SAX della serie temporale del Brand 1. Nell'esempio, con $\alpha = 5$ e $w = 24$, la parola risultante é **aabddbbabdedbabecbbaaeed**

Allo scopo di confrontare due o piú serie temporali differenti, é necessario determinare un modo per calcolare la distanza tra esse; piú bassa sará la distanza e piú le serie temporali risulteranno simili.

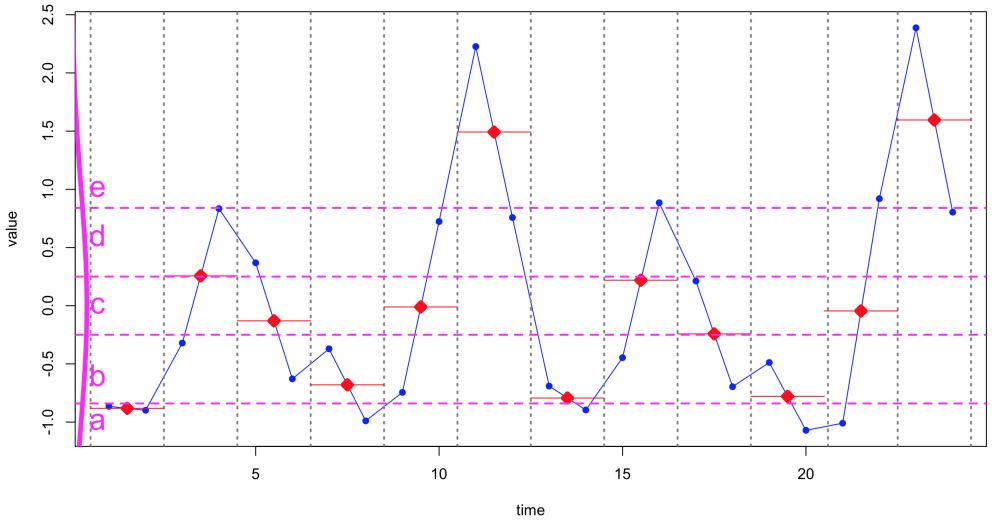


Figura 4.23: Discretizzazione SAX della serie temporale del Brand 1. Nell'esempio, con $\alpha = 5$ e $w = 12$, la parola risultante é **adcbcebccbce**

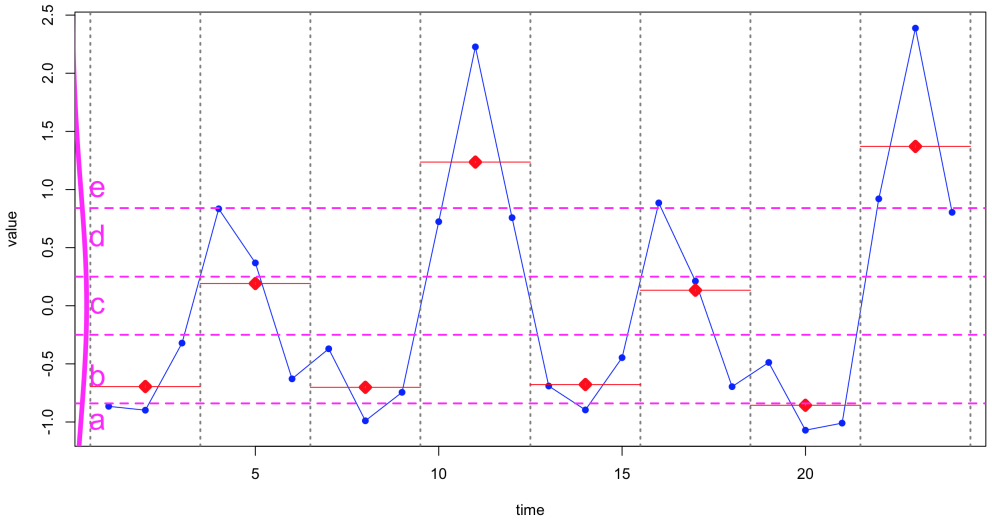


Figura 4.24: Discretizzazione SAX della serie temporale del Brand 1. Nell'esempio, con $\alpha = 5$ e $w = 8$, la parola risultante é **bcbebcae**

SAX introduce una nuova metrica per misurare la similaritiá tra stringhe estendendo la distanza Euclidea e quella basata su PAA. Questa nuova metrica possiede la proprieá di *lower bounding* della distanza Euclidea.
In sintesi, la distanza calcolata sui dati approssimati é sempre inferiore a

quella calcolabile sulle serie originali. Questa caratteristica é indispensabile, poiché permette di applicare molti algoritmi di *data mining* esistenti sulle rappresentazioni SAX producendo gli stessi risultai che si otterrebbero sui segnali di partenza.
Tramite l'utilizzo di SAX é stato possibile scoprire dei *motif* ricorrenti all'interno delle serie temporali. Uno dei parametri configurabili riguarda i segmenti in cui viene suddivisa ogni serie storica utilizzata, il cui numero rappresenterá la lunghezza della **parola** di ciascuna serie temporale.
La scelta del parametro é stata fatta considerando tre possibili casi:

1. segmentare ogni serie storica in 24 segmenti, quanti sono i mesi dell'intero arco temporale;

2. segmentare ciascuna serie in 12 segmenti, prendendo in considerazione, quindi, intervalli di 2 mesi;

3. segmentare le differenti serie temporali in 8 mesi, in modo da considerare sotto periodi di 3 mesi.

Questa scelta é scaturita per trovare *motif* di varia lunghezza e che potevano non essere visibili se si fossero considerati solamente periodi di minimo 3 mesi.
Così facendo, é possibile verificare se esistono dei pattern frequenti in comune tra diverse serie temporali, anche riferiti ad intervalli di tempo brevi.
Un altro parametro che é possibile scegliere nell'esecuzione dell'algoritmo riguarda il numero di simboli dell'alfabeto da utilizzare per convertire ogni serie temporale in stringa. La scelta di questa parametro é stata effettuata considerando tre opzioni possibili:

1. trasformare ogni serie temporale in stringa utilizzando un alfabeto di dimensione 3, ottenendo così **parole** contenenti solamente i simboli **a,b,c**;

2. trasformare ogni serie temporale in stringa utilizzando un alfabeto di dimensione 4, ottenendo così **parole** contenenti solamente i simboli **a,b,c,d**;

3. trasformare ogni serie temporale in stringa utilizzando un alfabeto di dimensione 5, ottenendo così **parole** contenenti solamente i simboli **a,b,c,d,e**.

Questa scelta deriva dal fatto che utilizzando un numero piú elevato di simboli é possibile ottenere *motif* piú accurati rispetto all'utilizzo di un numero inferiore di simboli in cui ogni area contiene un numero maggiore di punti.

	numero_simboli		
num_mesi	3	4	5
1	5	6	7
2	5	6	6
3	2	5	5

Figura 4.25: Numero Motif trovati

Nella figura 4.25, per ogni possibile combinazione dei parametri impostabili nell'algoritmo, si mostra il numero di *motif* trovati.
Per trovare i *motif* frequenti all'interno delle 6 serie temporali analizzate, si é deciso di utilizzare sotto-sequenze costituite da 2 simboli.
Avremo, quindi, nel caso di parole aventi lunghezza 24, in cui ogni simbolo corrisponde ad un mese solare, sotto-sequenze e di conseguenza *motif* corrispondenti a due mesi.
Invece, nel caso di parole aventi lunghezza 12, in cui ogni simbolo corrisponde a due mesi, sotto-sequenze e di conseguenza *motif* di lunghezza 4 mesi.
Infine, nel caso di parole aventi lunghezza 8, in cui ogni simbolo corrisponde a tre mesi , sotto-sequenze e di conseguenza *motif* corrispondenti a 6 mesi.
Nelle seguenti tabelle si mostra per ogni motif trovato il relativo supporto, a seconda dei parametri scelti utilizzati nell'applicazione dell'algoritmo.

	numero_simboli		
motif	3	4	5
AA	17	10	6
AC	13		
AD		10	
AE			10
BA	18	15	9
BC	11		
BD		11	
BE			5
CB		4	8
CC	10		
DB			3
DD		6	
ED			9

Figura 4.26: Elenco motif con relativo supporto e numero mesi = 1

	numero_simboli		
motif	3	4	5
AA	2		
AB	7		
AC	6	9	
AD		2	6
BA	8	5	
BB		7	4
BC	11		
BD		5	4
CB			6
CD		7	
CE			7
DE			3

Figura 4.27: Elenco motif con relativo supporto e numero mesi = 2

Un ulteriore appunto deve essere fatto sulle informazioni contenute nelle tabelle mostrate nelle figure 4.26, 4.27 e 4.28.
Il fatto che la stessa occorrenza di simboli compaia in piú tabelle contemporaneamente non deve portare a conclusioni errate.

motif	numero_simboli		
	3	4	5
AB	11	2	
AC	13	4	2
AD		8	
AE			5
BC		6	7
BD		4	3
BE			7

Figura 4.28: Elenco motif con relativo supporto e numero mesi = 3

Infatti, le occorrenze uguali che ricorrono nelle diverse tabelle sono riferiti a *motif* diversi.
Ad esempio, il *motif* "AA" della tabella 4.26 avrá lunghezza pari a due mesi a differenza del *motif* della tabella 4.27 che avrá lunghezza pari a 4 mesi.
I *motif* totali che sono stati trovati sono 47. Nelle figure seguenti ne vengono mostrati solamente alcuni, poiché considerati di maggior rilevanza e aventi un supporto maggiore rispetto ad altri.

- Motif 1: questo *motif* presenta un supporto pari a 18. É stato trovato impostando i seguenti parametri:

  - dimensione dell'alfabeto=3, ovvero numero di simboli utilizzati per la conversione in stringa=3;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=24 (ogni serie temporale é stata segmentata in 24 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 2 mesi).

  Questo *motif* é riferito al primo bimestre del 2014 e del 2015 ed è ricorrente in tutte le 6 serie storiche analizzate. Inoltre, limitatamente alla serie temporale dei Brand1, Brand3 e Brand5 questo *motif* é riferito anche al periodo di tempo compreso tra Luglio e Agosto 2015.

- Motif 2: Questo *motif* presenta un supporto pari a 11. É stato trovato impostando i seguenti parametri:
  - dimensione dell'alfabeto=4, ovvero numero di simboli utilizzati per la conversione in stringa=4;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=24 (ogni serie temporale é stata segmentata in 24 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 2 mesi).

  Questo *motif* é riferito al periodo compreso tra Marzo e Aprile di entrambi gli anni, ad eccezione del Brand 4 e del Brand 5 in cui ricorre solamente nell'anno 2015.

- Motif 3: questo *motif* presenta un supporto pari a 8. É stato trovato impostando i seguenti parametri:
  - dimensione dell'alfabeto=3, ovvero numero di simboli utilizzati per la conversione in stringa=3;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=12 (ogni serie temporale é stata segmentata in 12 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 4 mesi).

  Questo *motif* é riferito al secondo quadrimestre (Maggio-Agosto) di entrambi gli anni analizzati. Fanno eccezione il Brand 3 che presenta questo *motif* solamente nel secondo quadrimestre del 2014 e il Brand 6 che presenta questo *motif* nel secondo quadrimestre del 2015.

- Motif 4: questo *motif* presenta un supporto pari a 9. É stato trovato impostando i seguenti parametri:
  - dimensione dell'alfabeto=4, ovvero numero di simboli utilizzati per la conversione in stringa=4;

  - numero segmenti in cui é stata suddivisa ogni serie temporale=12 (ogni serie temporale é stata segmentata in 12 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 4 mesi).

  Questo *motif* é riferito al primo quadrimestre di ciascun anno considerato, ad eccezione del Brand 3 in cui questo motif è ricorrente solamente nell'anno 2015 ed il Brand 5 che non presenta questo motif.

- Motif 5: questo *motif* presenta un supporto pari a 7. É stato trovato impostando i seguenti parametri:

  - dimensione dell'alfabeto=5, ovvero numero di simboli utilizzati per la conversione in stringa=5;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=12 (ogni serie temporale é stata segmentata in 12 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 4 mesi).

  Questo *motif* é riferito al periodo compreso tra Settembre e Dicembre di entrambi gli anni, ad eccezione del Brand 3 e del Brand 6 in cui non compare.

- Motif 6: questo *motif* presenta un supporto pari a 13. É stato trovato impostando i seguenti parametri:

  - dimensione dell'alfabeto=3, ovvero numero di simboli utilizzati per la conversione in stringa=3;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=8 (ogni serie temporale é stata segmentata in 8 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 6 mesi).

Questo *motif* é riferito al periodo compreso tra Luglio e Dicembre di entrambi gli anni, ad eccezione del Brand 3 in cui questo motif si ripete ogni sei mesi per tutto l'arco temporale considerato.

- Motif 7: questo *motif* presenta un supporto pari a 8. É stato trovato impostando i seguenti parametri:
  - dimensione dell'alfabeto=4, ovvero numero di simboli utilizzati per la conversione in stringa=4;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=8 (ogni serie temporale é stata segmentata in 8 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 6 mesi).

  Questo *motif* é riferito al secondo semestre di ciascun anno per ogni Brand considerato, ad eccezione del Brand 3 e del Brand 6 in cui questo motif ricorre solamente nell'anno 2015 e il Brand 4 in cui non compare.

- Motif 8: questo *motif* presenta un supporto pari a 7. É stato trovato impostando i seguenti parametri:
  - dimensione dell'alfabeto=5, ovvero numero di simboli utilizzati per la conversione in stringa=5;
  - numero segmenti in cui é stata suddivisa ogni serie temporale=8 (ogni serie temporale é stata segmentata in 8 *frames* di uguale dimensione);
  - lunghezza di ciascuna sottosequenza=2 (ovvero 6 mesi).

  Questo Motif é riferito al primo semestre di ogni anno. Fanno eccezione i Brand 2,3 e 5 che presentano questo motif solamente nel primo semestre 2015.

In conclusione, l'applicazione di questo algoritmo ha fatto emergere un andamento dei volumi di vendita pressocché identico nei due anni considerati,

per quanto riguarda il Brand1. Infatti, considerando i vari esperimenti effettuati facendo variare i parametri descritti sopra, si é notato che nei due anni consecutivi i volumi di vendita non presentano andamenti anomali.
Per quanto riguarda la comparazioni con gli altri Brand, é possibile evidenziare, ad esempio, che determinati *motif* compaiono all'interno di tutte le serie temporali analizzate ad eccezione di qualche periodo o in certi casi non compaiono completamente per un determinato Brand.
Ad esempio, nel *motif* n.8, la maggior parte dei Brand presenta lo stesso andamento lungo il primo semestre di entrambi gli anni. Fanno eccezione i Brand 2, 3 e 5 in cui compare solamente nel 2015.
Questo comportamento potrebbe derivare dal fatto che queste aziende concorrenti, analizzando a loro volta gli andamenti dei volumi di vendita delle altre aziende, hanno applicato delle strategie di prezzo o di vendita tali che l'anno successivo il loro volume di vendita seguisse l'andamento dell'interno mercato.
Stesso ragionamento puó essere fatto per il *motif* n.7, con la differenza che in questo caso il Brand 4 nello specifico non presenta questo *motif* nel periodo considerato.

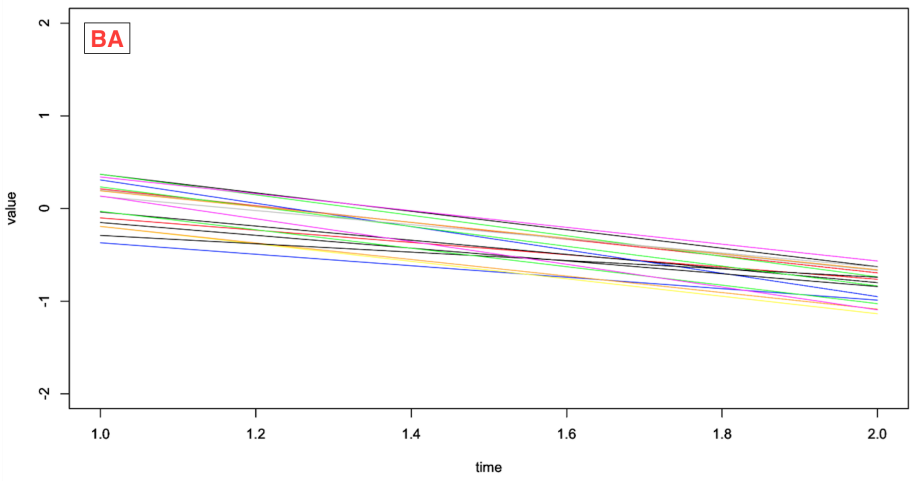


Figura 4.29: Motif n.1 avente lunghezza pari a 2 mesi e supporto pari a 18

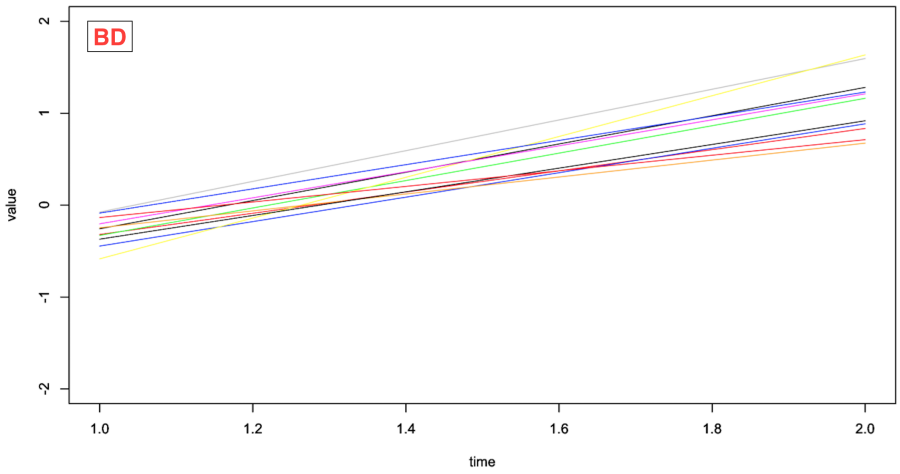


Figura 4.30: Motif n.2 avente lunghezza pari a 2 mesi e supporto pari a 11

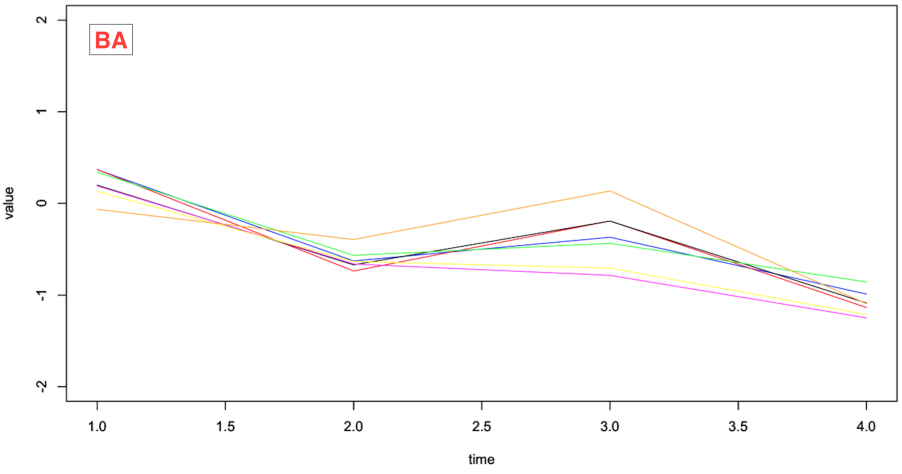


Figura 4.31: Motif n.3 avente lunghezza pari a 4 mesi e supporto pari a 8

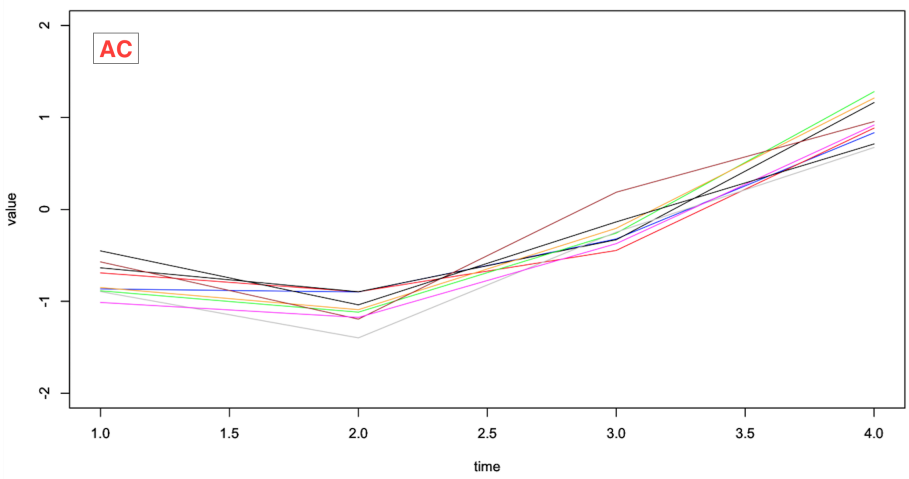


Figura 4.32: Motif n.4 avente lunghezza pari a 4 mesi e supporto pari a 9

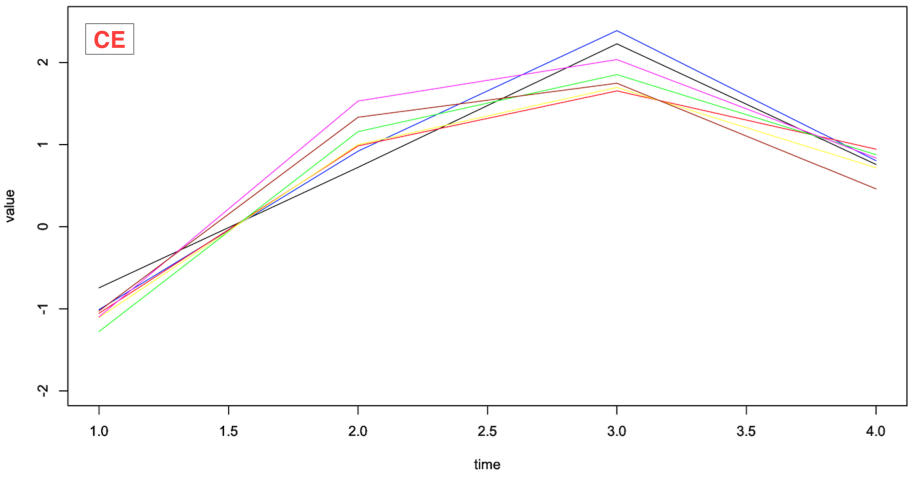


Figura 4.33: Motif n.5 avente lunghezza pari a 4 mesi e supporto pari a 7

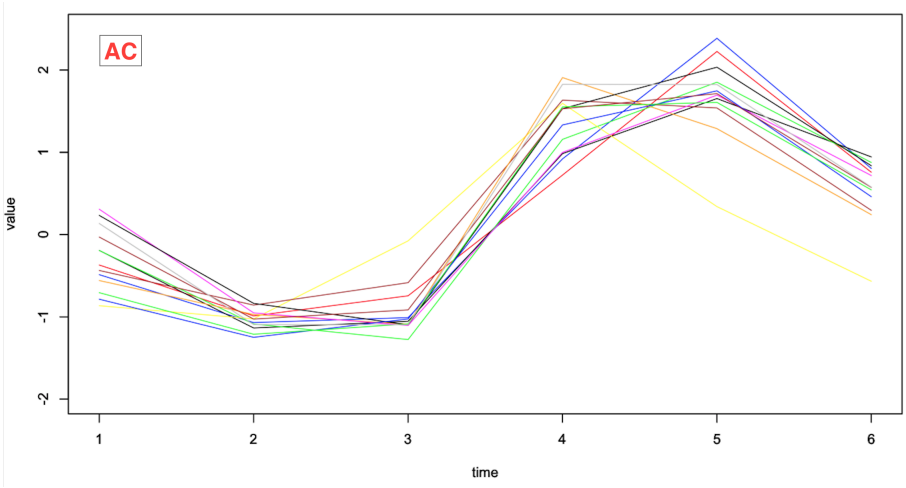


Figura 4.34: Motif n.6 avente lunghezza pari a 6 mesi e supporto pari a 13

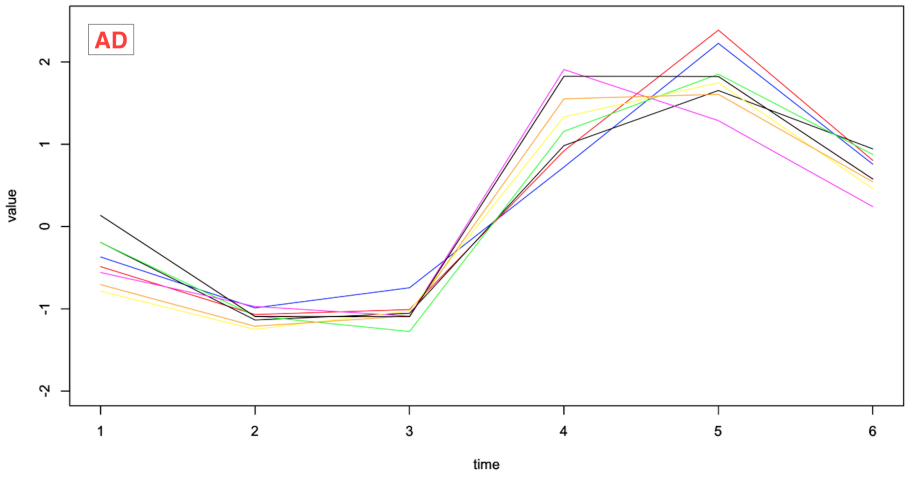


Figura 4.35: Motif n.7 avente lunghezza pari a 6 mesi e supporto pari a 8

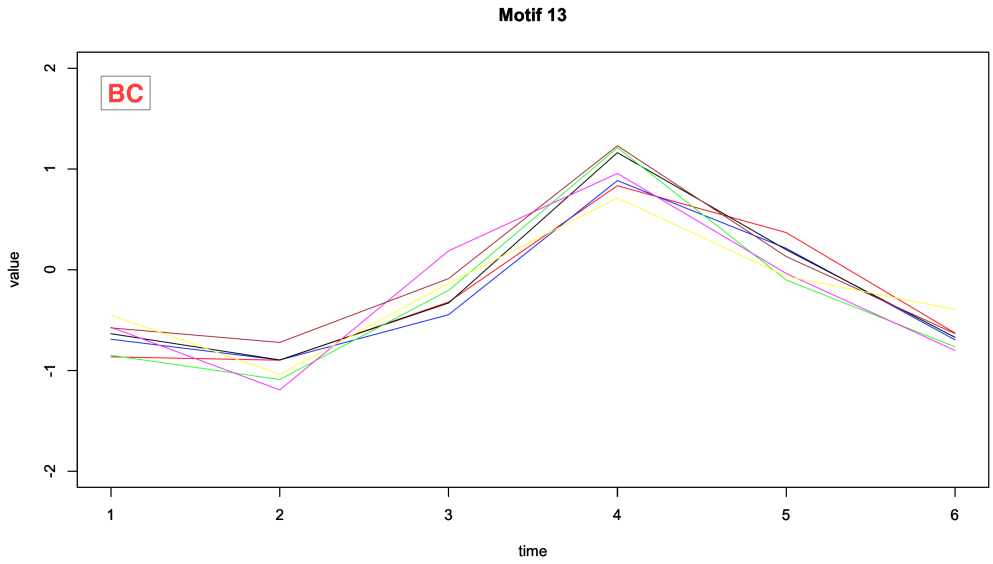


Figura 4.36: Motif n.8 avente lunghezza pari a 6 mesi e supporto pari a 7

Capitolo 5

REPORTING

L'ultima fase del progetto riguarda la costruzione di un sistema di reportistica utile al supporto delle decisioni aziendali. Per la costruzioni di tali report é stato utilizzato lo strumento di Data Visualization "Tableau".
Questo strumento consente di rappresentare in forma tabellare e grafica i dati secondo modalitá preconfezionate di navigazione e interattive.
Si parla di report interattivi nel senso che, una volta costruiti, possono essere calcolati, quando desiderato, in funzione dei nuovi dati disponibili. Inoltre, un report puó essere personalizzato e distribuito (per esempio per via elettronica o cartacea) a seconda delle esigenze della persona a cui é destinato.
Disporre di un tale strumento consente a manager o persone di alti livelli gerarchici aziendali di capire l'andamento della propria azienda in maniera molto semplice e veloce.
Questi strumenti sono, infatti, diventati indispensabili per garantire il corretto svolgimento delle attivitá aziendali e per intraprendere azioni correttive in casi di scostamenti da valori medi o superamento di valori di soglia.
Tableau é uno strumento costituito da vari componenti, I principali sono:

- la componente "desktop" utilizzata per la costruzione effettiva dei cruscotti ;
- la componente "server" in cui vengono caricati i vari cruscotti creati precedentemente a cui posso accedere gli utenti autorizzati utilizzando

delle credenziali apposite.

Per ogni cruscotto creato e caricato sul server sono stati impostati dei permessi che consentono di limitare la visione di alcune dashaboard o alcuni dati a singoli o gruppi di utenti.
Nel seguito verrano mostrati i cruscotti piú rappresentativi che sono stati costruiti durante quest'ultima fase progettuale.
La prima dashboard viene mostrata in figura 5.1 e consente l'analisi del trend di prezzo del Brand principale e dei Brand competitors considerando solamente le categorie di prodotti che rientrano nella cosiddetta paritá di gamma.
Si ha paritá di gamma quando un Brand offre la stessa categoria di prodotti offerti dal Brand principale in un determinato mese.
Il prezzo di ogni Brand in ciascun mese é calcolato come media pesata sui volumi di vendita e in maniera dinamica: ogni qualvolta che viene applicato o rimosso un filtro la paritá di gamma viene ricalcolata.
Questa analisi é stata effettuata seguendo tre differenti approcci:

1. la prima analisi ha l'obiettivo di mostrare in che modo le aziende concorrenti si posizionano sul mercato rispetto al Brand principale. Per ottenere questa analisi, l'indice di prezzo del Brand 1 é stato settato a 100 per tutto l'arco temporale considerato; invece, gli indici di prezzo delle aziende concorrenti sono stati calcolati per ogni mese, rapportando il loro prezzo a quello del Brand 1 sulla base dei prodotti che entrano in paritá di gamma.

2. la seconda analisi ha l'obiettivo di mostrare la variazione di prezzo di ogni Brand rispetto ad un mese di riferimento nel quale ogni Brand assume valore = 100. Per analizzare il trend é stato costruito un mese "fittizio" che comprende tutti i prodotti venduti sul mercato nel periodo compreso tra Gennaio ed Ottobre dell'anno precedente.
   Quindi, per ogni mese considerato, l'indice di ogni Brand é calcolato rapportando i prezzi dei prodotti del mese considerato ai prezzi dei

prodotti venduti nel mese "fittizio".
A differenza della prima analisi, gli indici di prezzo vengono calcolati considerando solamente i prodotti in comune tra il mese "fittizio" e il mese che si sta analizzando, all'interno dello stesso Brand, senza considerare i prodotti venduti anche dal Brand principale.

3. il terzo grafico é stato costruito come unione dei primi due ed ha l'obiettivo di analizzare la variazione di prezzo rispetto ad un mese di riferimento. É stato costruito riproporzionando i trend visibili nel primo grafico al trend del Brand 1 del secondo grafico.

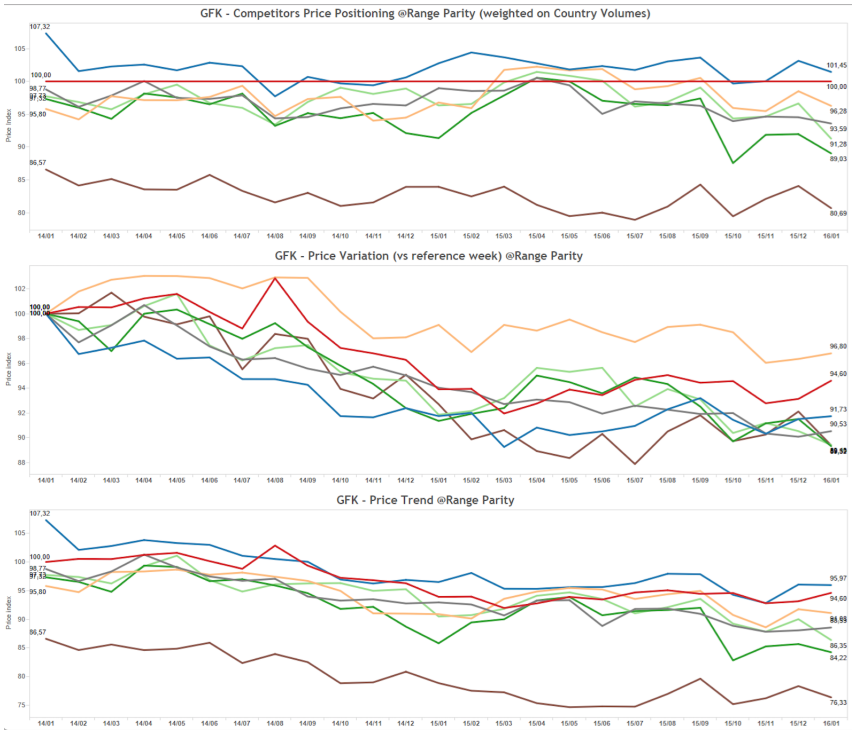


Figura 5.1: Posizionamento dei Brand sull'intero mercato

La dashboard mostrata in figura 5.3 riassume in un unico cruscotto l'andamento del *markup* e la percentuale di stock venduto, suddiviso per stagionalitá.
Nella parte alta della dashboard sono stati creati dei pulsanti, uno per

ciascun Brand. Cliccando su uno di essi, i grafici sottostanti si aggiornano automaticamente mostrando solamente le informazioni relativi al Brand prescelto.
Questo cruscotto presenta 3 livelli distinti di analisi. Nel primo livello, si é mostrato il trend del Markup, in valore percentuale, che ogni Brand possiede in ciascuna settimana. Il Markup é dato dalla differenza tra il prezzo di un prodotto e il suo costo.
Nel secondo livello, tramite un grafico a barre, vengono mostrati i cosiddetti "markup cluster" che indicano la percentuale della somma totale dei volumi di vendita per ogni cluster definito precedentemente.

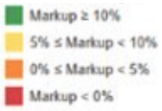


Figura 5.2: Markup Range

Infine, nel terzo ed ultimo livello, sono stati creati due grafici, distinti per stagionalitá, il cui obiettivo é mostrare la quantiá di stock venduto in ogni settimana da ciascun Brand.
Sono stati individuati, in base ai valori ottenuti negli anni precedenti, quattro cluster principali del markup, mostrati in figura 5.2.

In figura 5.4 si mostrano due grafici a bolle utili per analizzare il numero di prodotti offetti dai vari distributori ed il markup ad essi associato.

Il numero di *bubble* corrisponde al numero di distributori presenti nella rispettiva tabella dei distributori.
Ogni bubble corrisponde ad una micro-categoria, la dimensione della *bubble* é data dal volume associato ad ogni categoria ed il colore rappresenta i differenti Brand.
Questo grafico é di tipo interattivo: cliccando su una determinata *bubble* vengono evidenziate le altre *bubble* riferite alla stessa micro-categoria, ma vendute dagli altri Brand, e viene mostrata una tabella di dettaglio contenente le varie specifiche di prodotto.

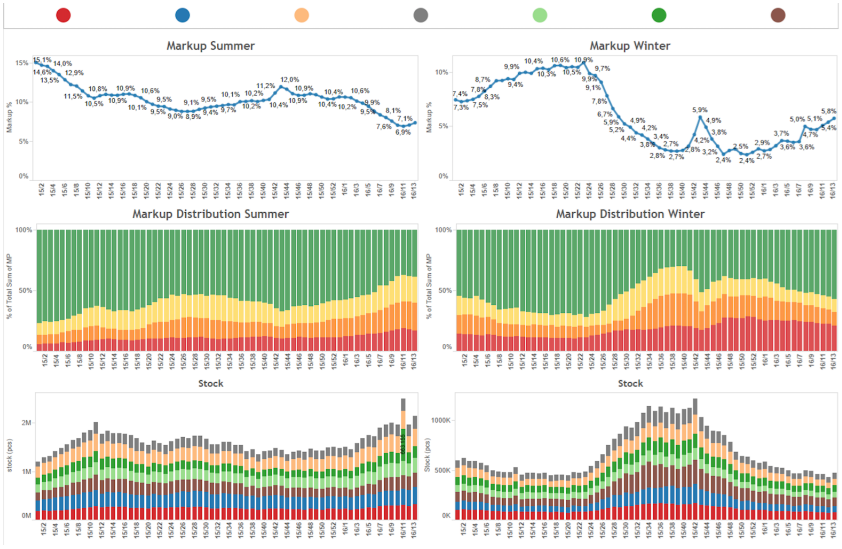


Figura 5.3: Trend quota di mercato e Stock venduto

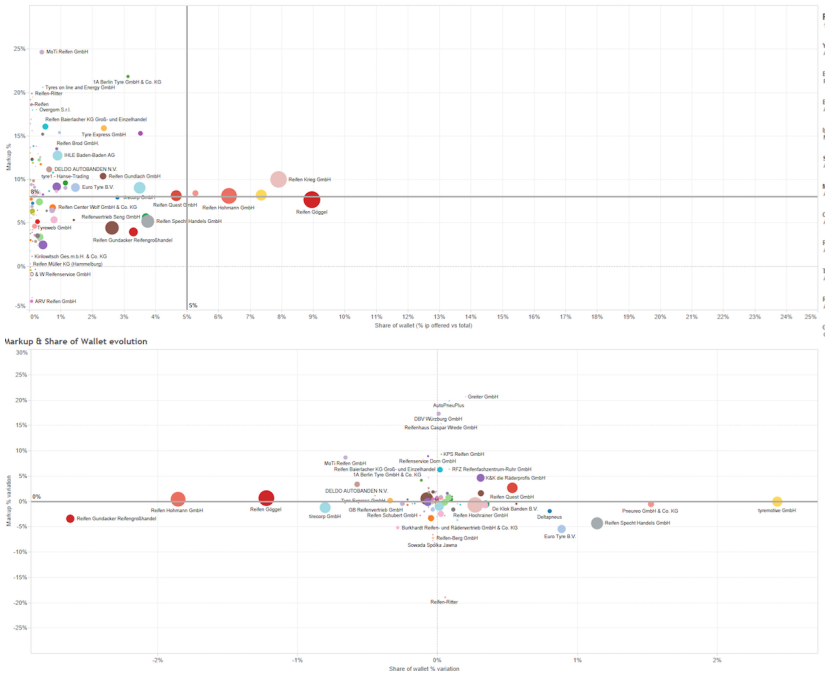


Figura 5.4: Rappresentazioni bubble

In figura 5.5 viene mostrato un tipo particolare di grafico, chiamato *Sankey*. É un diagramma di flusso in cui l'ampiezza delle varie "frecce" é

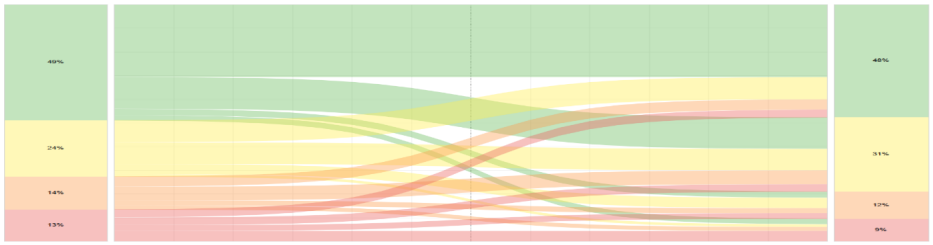


Figura 5.5: Sankey Markup

proporzionale alla quantitá del flusso della variabile utilizzata. Il *Sankey* mostra l'evoluzione del Markup settimana dopo settimana. Le settimane sono selezionabil dall'utente. Ogni colore é riferito ad un cluster del markup (per la definizione di cluster vedere la figura 5.2).
Anche in questo caso il grafico é di tipo interattivo. Infatti, cliccando su un cluster, viene evidenziato il flusso specifico e viene mostrata una tabella di dettaglio sui produtti venduti appartenenti a quel determinato cluster.

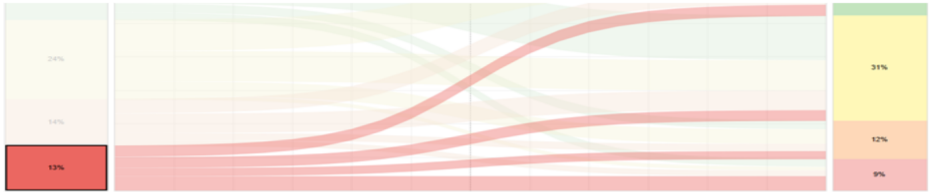


Figura 5.6: Sankey Markup

In figura 5.6 viene mostrato un esempio di interattivitá possibile.
Questo tipo di grafico é stato creato per capire l'andamento dei prodotti in un arco temporale delimitato dalle due settimane prescelte. Come si vede in figura 5.6, alcuni prodotti che in un certa settimana presentava un valore percentuale di markup negativo hanno ottenuto un valore di markup $> 10\%$ in corrispondenza di un'altra settimana.
Un'analisi effettuata considerando in particolare i distributori presenti sul mercato viene mostrata in figura 5.7.
La prima tabella della dashbard contiene le informazioni relative al prezzo, aggregato settimanalmente, di tutti i prodotti offerti da ogni rivenditore, lo

stock medio, il numero di prodotti offerti in ogni settimana, lil markup in valore assoluto e in percentuale.
Tutte queste informazioni sono relative ai primi 3 distributori che offrono il prezzo piú basso di ogni prodotto in ogni settimana.

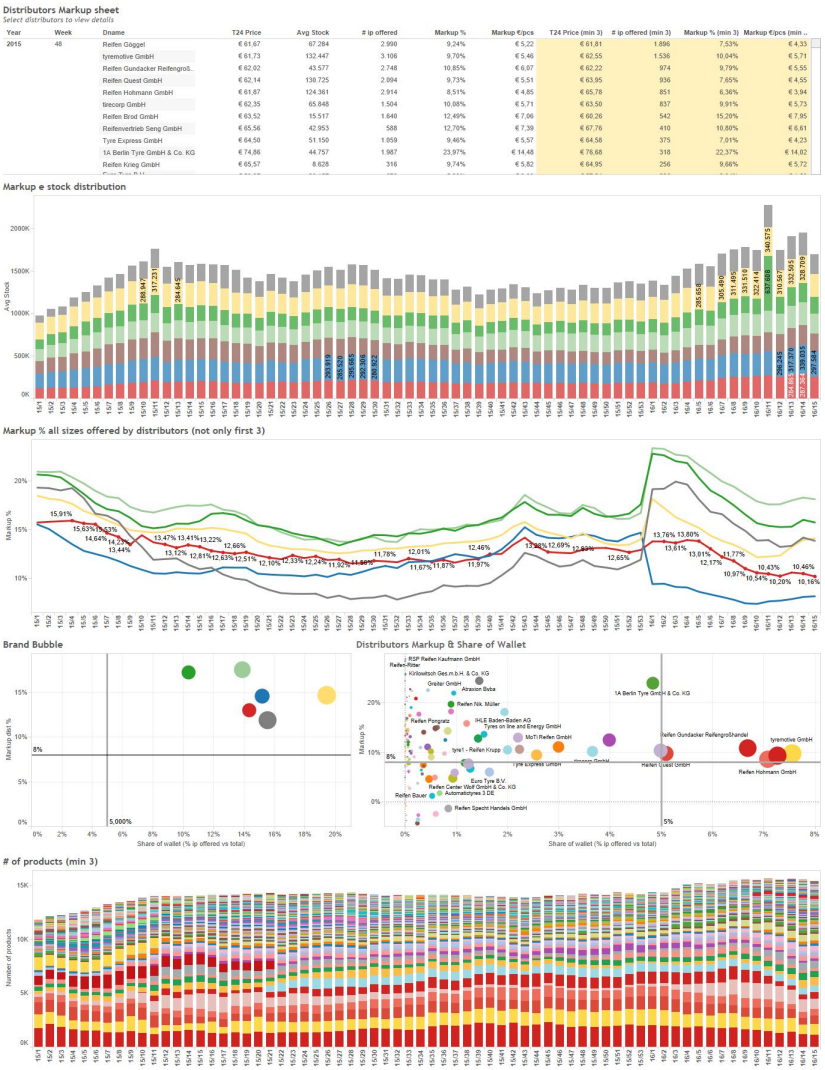


Figura 5.7: Analisi distributori

Sotto la tabella é stato creato un grafico a barre che mostra lo stock medio di ogni Brand in ogni settimana. In questo modo é possibile analizzare la distribuzione dello stock totale venduto, evidenziando eventuali picchi positivi o negativi nella distribuzione.

Il grafico a linee mostra l'andamento del markup offerta da tutti i distributori, non solamente dei primi 3, distinto per Brand.
A seguire troviamo due grafici a bolle. Il primo mostra in che modo i Brand sono posizionati rispetto alla percentuale dei prodotti offerti sul totale dei prodotti del mercato ed al markup medio che ogni Brand garantisce al distributore.
Nel secondo grafico, invece, ogni *bubble* rappresenta uno specifico distributore ed il suo posizionamento rispetto al markup medio e alla percentuale dei prodotti offerti sul totale del mercato.
La dimensione delle bubble é data dal volume totale associato ad ogni categoria di prodotto.
Infine, é stato creato un grafico a barre che mostra per ogni settimana il numero di prodotti venduto da ogni distributore (filtrando solamente i primi 3 con prezzo piú basso).
Il colore á associato ad ogni distributore e sono stati ordinati in maniera decrescente sul totale della quantitá venduta.

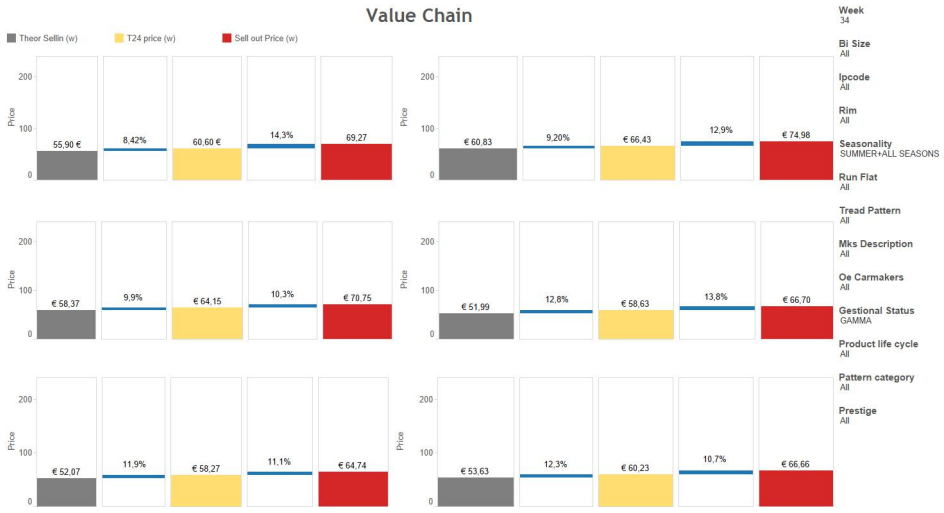


Figura 5.8: Waterfall

In figura 5.8 viene mostrato un grafico chiamato "Waterfall". Questo cruscotto consente di analizzare per ogni Brand la rispettiva catena del valore in una determinata settimana ed il markup ottenuto a diversi livelli della

catena.
É un grafico a cascata che consente di vedere facilmente i valori positivi e negativi che influiscono sul valore totale. Inoltre, permette di mettere in evidenza gli step che portano una certa grandezza da uno stato iniziale ad uno stato finale.
Nella figura 5.8 i vari step rappresentano i diversi prezzi che vengono applicati a seconda che il prodotto venga venduto direttamente dall'azienda al rivenditore o dal rivenditore al cliente finale.
Come si puó immaginare, il prezzo aumenta man mano che ci si avvicina allo step finale, ovvero la vendita tra distributori e cliente. La differenza tra i prezzi applicati nei vari step rappresenta il cosiddetto *markup* mostrato anche nelle analisi precedenti.
Infine, nella parte laterale del cruscotto viene mostrata una lista di filtri applicabili alla dashboard ed in base ai quali il cruscotto si aggiorna in maniera dinamica.
Grazie a questi filtri, é possibile da parte dell'utente effettuare delle analisi piú dettagliate focalizzandosi solamente su una porzione dei dati piuttosto che sull'intero dataset disponibile.
La dashboard mostrata in figura 5.10 riassume le informazioni sulla quota di mercato, il trend dei volumi di vendita, il trend di prezzo del Brand principale e dei sui concorrenti.
Nella parte alta della dashboard si mostra una mappa geografica rappresentante gli Stati in cui sono stati venduti. Per ogni Stato la percentuale associata rappresenta la quota di mercato che ogni Brand possiede in quella nazione.
La quota di mercato é stata calcolata come somma dei volumi di vendita di un Brand sul totale dei volumi di vendita di tutti i Brand. Accanto alla mappa geografica é stat costruita una griglia in cui viene mostrata la quota di mercato ed il prezzo pesato sui volumi di vendita considerando diversi livelli di dettaglio, quali la stagionalitá, il tipo di prodotto e la categoria.
Questa prima parte consente di avere una panoramica su tutti i prodotti venduti dai *Brand* principali nelle differenti nazioni.

É possibile selezionare un solo Stato o considerare l'intero mercato. La stessa selezione é possibile effettuarla sui Brand.
Il colore associato ad ogni bubble corrisponde al valore della quota di mercato e al cluster a cui appartiene. Invece, la dimensione si basa sul prezzo associato ad ogni possibile combinazione.
Sotto questa prima panoramica vengono mostrati altri grafici che si concentrano su particolari misure o che analizzano determinati trend.
Nel primo grafico a barre si mostra la percentuale della quota di mercato che i vari Brand posseggono in ogni mese e in uno o piú stati selezionabili dall'utente.
A seguire troviamo un altro grafico a barre che mostra il trend dei volumi di vendita totali, senza distinzione di Brand, in uno o piú stati e per ogni mese.
Un ulteriore grafico che é stato inserito in questa dashboard riassuntiva riguarda il concetto esposto precedentemente di partiá di gamma. L'obiettivo é analizzare il posizionamento dei concorrenti rispetto al Brand principale.
Nella parte finale, infine, vengono mostrati dei grafici atttraverso i quali é possibile comparare la quota di mercato dell'anno corrente rispetto alla quota di mercato dell'anno precedente e la quota di mercato di un mese dell'anno corrente rispetto alla quota di mercato del medesimo mese dell'anno precedente.
Per ognuna di queste due possibiliá é stato creato un grafico a barre orizzontali in cui viene mostrato il δ positivo o negativo tra la quota di mercato dell'anno in corso e la quota di mercato dell'anno prima.
Tutti questi grafici si aggiornano automaticamente in base alla nazione applicata o al Brand prescelto.
In Figura 5.9 viene mostrata una dashboard che consente di analizzare ogni singola Country separatamente e avere in questo modo una panoramica sull'intero mercato europeo.
Ogni riguadro, infatti, é riferito ad una specifica nazione ed é stato costruito in modo tale da dare piú informazioni di tipo differente all'utente che lo utilizzerá successivamente.

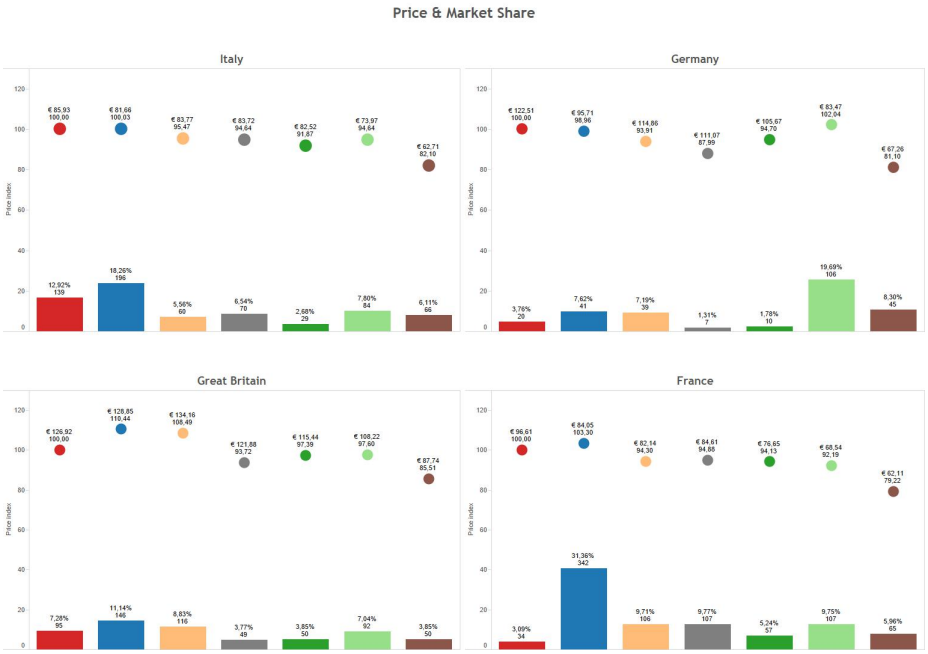


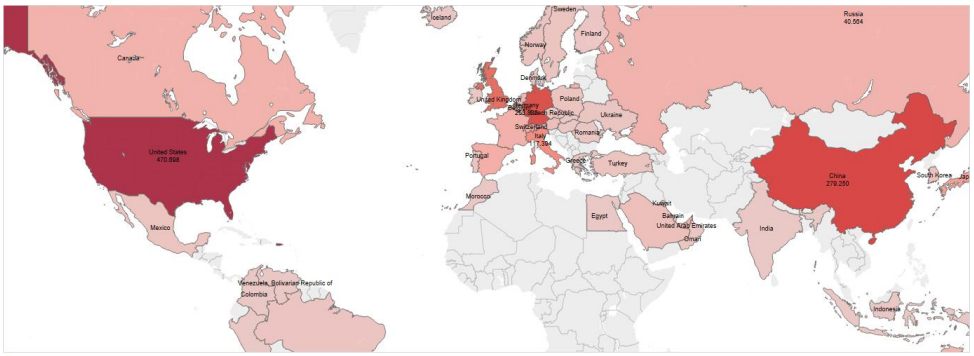
Figura 5.9: Analisi mercato europeo

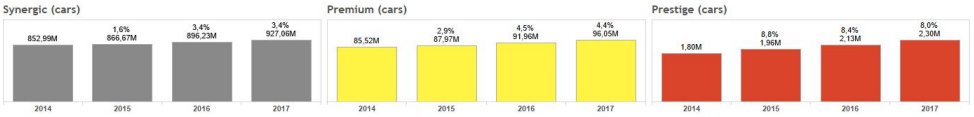
Le informazioni presenti in questa dashboard sono molteplici. In alto troviamo le *bubble*, una per Brand, che indicano il posizionamento di un determinato Brand rispetto ai suoi competitors nella Country selezionata.
Ad ogni bubble é associata un'ulteriore informazione, ovvero il prezzo calcolato come media pesata sui volumi della singola country.
Nella parte sottostante, é stato utilizzato invece un grafico a barre che mostra la distribuzione, distinta per Brand, della quota di mercato, espressa in percentuale. Ad ogni bin é associata un'altra informazione che riguarda il numero di unitá vendute dai differenti Brand.

Infine, in figura 5.11, viene mostrata una dashboard riassuntiva creata per analizzare il numero di prodotti, distinti per categoria, venduti non solo nel mercato europeo ma estendendo l'analisi anche al di fuori di questo perimetro. Nella parte alta del cruscotto viene mostrata una mappa geografica in cui per ogni nazione é stato calcolato il numero di prodotti venduti. Il colore é associato a questa quantitá ed in particolare un colore piú intenso assume il significato di un elevato numero di prodotti venduti, per una determinata

categoria che é possibile scegliere tramite filtri. Questa dashboard é utile per analizzare il potenziale di vendita di ogni Stato analizzato.
Sono state create tre macro-categorie: prodotti di tipo standard, luxury e industrial. Per ognuna delle categorie citate sopra, sono stati creati tre grafici a barre in cui ogni bin é riferito ad un anno specifico.
In questo cruscotto sono stati inseriti anche dati di *forecast*, ovvero dati ottenuti facendo delle previsioni sulle vendite dell'anno successivo. Infatti, i bin ricoprono un arco temporale che va dal 2014 al 2017.
Per dare una visione dei dati meno aggregata e piú dettagliata viene mostrata una tabella sul potenziale di mercato, distinto per anno.
A seguire é stato inserito un *sankey diagram* che mostra il flusso che va da ogni macro-categoria ad una micro-categoria riferita ad una specifica di prodotto che a sua volta si muove verso lo step finale, suddiviso in ulteriori sotto-categorie create appositamente per la creazione di questo cruscotto.
Sempre per dare un maggior dettaglio sui singoli prodotti venduti, é stata costruita una tabella contenente il dettaglio della quota di mercato espressa in percentuale, distinta per tipo di mercato selezionato.
A seguire si mostra una matrice in cui ogni bubble é ottenuta come punto di intersezione tra i valori della quota di mercato dei due principali mercati di vendita. La dimensione é associata, invece, al numero di prodotti venduti.
Infine, é stata creata una tabella di dettaglio limitatamente agli ultimi due anni effettivi contenenti le informazioni sul numero di prodotti venduti in cui viene specificato il tipo di modello dei vari prodotti.

Figura 5.10: Rappresentazione prezzi e volumi nelle country principali



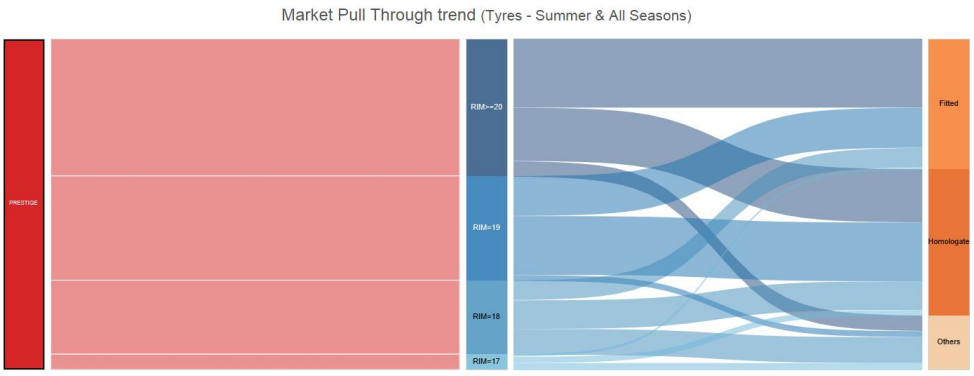


Market potential (Tyres Summer & all season)

	2014	2015	2016	2017
Fitted	6.082.046	11.086.960	13.544.631	15.837.588
Homologated	18.204.863	20.354.561	23.171.375	24.655.154
Others	722.131.753	716.536.377	758.460.174	784.617.529
Tot market potential	746.418.662	747.977.899	795.176.181	825.110.271

Market trend YoY (Tyres Summer & all season)

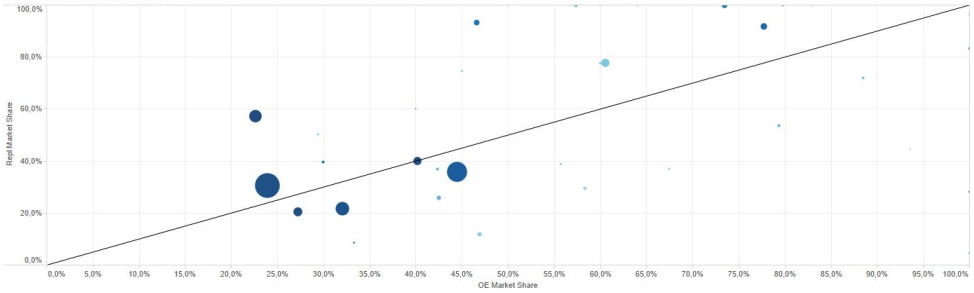
	2014	2015	2016	2017
Fitted		82,29%	22,17%	16,93%
Homologated		11,81%	13,84%	6,40%
Others		-0,77%	5,85%	3,45%
Market trend		0,21%	6,31%	3,76%



Car model detail

Car model	Market potential (tyres)	of which Homologated (tyres)	of which Fitted (tyres)	Potential growth Ebit (€)	Scenario 2015 OE Market Share	Scenario 2015 Repl Market Share
PORSCHE/ CAYENNE	474.235	323.264	113.599	€0K	24,0%	30,5%
PORSCHE/ 911	310.954	274.246	138.444	€867K	44,5%	35,8%
PORSCHE/ BOXSTER	140.127	110.546	44.933	€383K	32,1%	21,5%
PORSCHE/ PANAMERA	117.908	117.172	26.718	€0K	22,7%	57,1%
PORSCHE/ CAYMAN	60.600	37.609	16.517	€113K	27,3%	20,3%
PORSCHE/ MACAN	50.182	50.182	20.163	€6K	40,2%	39,8%
BENTLEY/ GT	44.325	26.855	26.855	€0K	60,6%	77,6%
MASERATI/ QUATTROPORTE	31.928	31.229	24.817	€0K	77,7%	91,7%
MASERATI/ GRANTURISMO	23.725	16.514	11.065	€0K	46,6%	93,1%
MASERATI/ GHIBLI	21.498	20.891	15.798	€0K	73,5%	100,0%
PORSCHE/ PORSCHE OTHER	21.063	0	0	€0K		
LOTUS/ ELISE	15.573	0	0	€0K		
ASTON MARTIN/ VANTAGE	14.470	6.786	6.786	€376K	46,9%	11,8%
FERRARI/ F458	13.320	11.323	5.662	€149K	42,5%	25,9%
FERRARI/ FERRARI OTHER	13.143	0	0	€0K		

Car model market share matrix



Growing cars

Car model	Market potential (tyres) 2015	Market potential (tyres) 2016	2016 vs 2015
MCLAREN/ SPORTS SERIES	244	1.374	463,50%
FERRARI/ F488	665	2.642	297,28%
ASTON MARTIN/ LAGONDA	28	102	262,93%
LAMBORGHINI/ HURACAN	784	1.820	138,24%
PORSCHE/ MACAN	50.182	112.396	123,98%
MCLAREN/ P1	120	264	118,87%
MASERATI/ GHIBLI	21.498	45.788	112,99%
ROLLS-ROYCE/ WRAITH	1.845	3.370	104,91%
FERRARI/ LAFERRARI	163	313	92,06%
PORSCHE/ 918 SPYDER	254	448	76,03%

Figura 5.11: Dashboard riassuntiva

Capitolo 6

CONCLUSIONI

In questo elaborato di tesi, si é visto come i Big Data assumono un'importanza vitale nella vita di tutti i giorni, e soprattutto come il corretto utilizzo di ingenti quantitá di dati diversi tra loro per volume, varietá e struttura, consenta alle aziende di ottenere vantaggi competitivi.
La realizzazione del progetto é stata particolarmente complessa, soprattutto a causa della notevole mole di dati da gestire e dell'eterogeneitá dei sistemi dai quali estrarli, ma alla fine sono state portate a termine tutte le richieste espresse dal cliente, il quale si é ritenuto soddisfato della soluzione implementata.
In particolare sono stati raggiunti gli obiettivi che erano stati prefissati nella primissina fase progettuale:

- é stata realizzata un'unica piattaforma contenente dati provenienti da fonti diverse;
- sono stati realizzati molteplici cruscotti utili al top management per le analisi di mercato;
- sono state effettuate tutte le trasformazioni necessarie per poter integrare in un'unica piattaforma dati provenienti da sorgenti differenti, aventi in alcuni casi discrepanze ad esempio nei codici dei prodotti;

- sono state implementate numerose regole e create varie strutture per poter integrare ulteriori dati, provenienti anche da altre fonti;
- sono stati presentati i risultati ottenuti nell'ambito della *motif discovery*.

Per quanto riguarda gli sviluppi futuri saranno introdotte nuove fonti dati, riguardanti anche i dati sui prezzi dei prodotti venduti sui maggiori siti web. Verranno integrati ulteriori dati sui volumi e sui prezzi di vendita anche di altri stati sia europei che extra-europei in modo da ottenere un quadro di analisi piú ampio e dettagliato.
In conclusione possiamo affermare che l'esperienza lavoriativa é stata molto interessante e formativa, in quanto ha offerto la possibilitá di mettere in pratica all'interno di una realtá aziendale complessa quanto appreso durante il corso di studi.
Il progetto é stato realizzato da un gruppo di lavoro composto da 8 consulenti, ognuno con particolari compiti e competenze.
Io ho collaborato alla realizzazione delle procedure ETL e alla costruzione dei report finali richiesti dal cliente. Ho avuto inoltre la possibilitá di approfondire alcune tematiche studiate, come ad esempio le procedure ETL e l'applicazione di algoritmi di machine learning per scoprire pattern frequenti, e di imparare l'utilizzo di nuovi strumenti, in particolare Hadoop e il suo ecosistema e lo strumento di Data Visualization Tableau.

Bibliografia

[1] Patel P., Keogh E., Lin J., Lonardi S., *Mining Motifs in Massive Time Series Databases*, 2003.

[2] Chiu B., Keogh E., Lonardi S., *Probabilistic Discovery of Time Series Motifs*, 2003.

[3] Mueen A., Keogh E., *Online Discovery and Maintenance of Time Series Motifs*, 2010.

[4] Lin J., Keogh E., Lonardi S., Patel P., *Finding Motifs in Time Series*, 2002.

[5] Lin J., Keogh E., Lonardi S., Chiu B., *A Symbolic Representation of Time Series, with Implications for Streaming Algorithms*, 2003.

[6] Lin J., Keogh E., Wei L., Lonardi S., *Experiencing SAX: a novel symbolic representation of time series*, 2003.

[7] Mueen A., Keogh E., Zhu Q., Cash S., Westover B., *Exact Discovery of Time Series Motifs*, 2009.

[8] Rezzani A., *Big Data: Architettura, tecnoloige e metodi per l'utilizzo di grandi basi di dati*, 2013.

[9] Turkington G., Modena G., *Big Data con Hadoop*, 2014.