ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Scienze
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Un prototipo per analizzare dati multidimensionali sulla base delle intenzioni dell'utente

Elaborato in
BUSINESS INTELLIGENCE

Relatore
Prof. STEFANO RIZZI

Presentato da
LORENZO PACINI

Terza Sessione di Laurea

Anno Accademico 2019 – 2020

INDICE

INTRODUZIONE

Il lavoro di tesi si basa sulla definizione dei nuovi sistemi IAM (Intentional Analytics Model) come evoluzione ed integrazione dei sistemi OLAP ed ha portato alla realizzazione di un prototipo per analizzare dati multidimensionali sulla base delle intenzioni dell'utente.
Il prototipo è stato realizzato in tutte le sue componenti, dalla formulazione dell'intenzione alla generazione delle query per il recupero dei dati, fino alla definizione degli algoritmi di data mining da utilizzare per l'analisi multidimensionale.
Il lavoro si è concentrato sull'attività di parsing intesa come interpretazione ed elaborazione dell'intenzione utente, sulla definizione e costruzione degli algoritmi di data mining da associare ai diversi operatori intenzionali.
Nel prototipo è stato anche realizzato un supporto per l'utente guidandolo alla formulazione dell'intenzione attraverso modelli e strutture dei metadati. In questo ambito è stata prevista anche un'interfaccia vocale per poter formulare la richiesta intenzionale direttamente in modalità parlata. Sono stati implementati due operatori intenzionali, Describe e Explain, insieme a una serie di algoritmi di analisi dei dati previsti per tali operatori, quali algoritmi di clustering, analisi delle varianze, trend detection, K-means, outlier, top-k, …
Lo schema generale dell'applicazione si presenta secondo il modello MVC che realizza il supporto per l'inserimento da parte dell'utente delle richieste intenzionali e la rappresentazione dei risultati in forma grafica e tabellare.

Tutto il lavoro viene presentato nei seguenti capitoli:
1. Sistemi OLAP. Breve introduzione ai sistemi OLAP con lo stato di evoluzione ed uso attuale e la presentazione dei loro principali operatori.
2. Intentional Analytics Model. Sono presentate le caratteristiche principali dei sistemi IAM, gli operatori con le loro caratteristiche e i principali algoritmi associati.
3. Prototipo del modello. Sono definite tutte le componenti strutturali del sistema a partire dalla formulazione dell'intenzione, parsing dell'intenzione, database dei metadati, generazione delle query, struttura del prototipo e algoritmi di data mining con interpretazione dei risultati.
4. Realizzazione del prototipo. Sono descritte in dettaglio tutte le strutture organizzate secondo un modello MVC.

1 SISTEMI OLAP

1.1 Descrizione generale

OLAP (On-Line Analytical Processing) è un insieme di tecniche software per l'analisi interattiva di grandi quantità di dati, che possono essere esaminati con modalità complesse. La tecnologia OLAP consente di organizzare i database aziendali di grandi dimensioni e supporta l'esecuzione di analisi complesse.

Questa tecnologia rappresenta un potente strumento di rilevamento dei dati progettato per consentire agli utenti di eseguire analisi multidimensionali dei dati. Viene usata per eseguire interrogazioni (query) per analisi dei dati senza influire negativamente sui sistemi transazionali. Costituisce una componente dei data warehouse (DW) e la sua applicazione tipica è all'interno di strumenti di business intelligence (BI) o a supporto di processi decisionali basati sui dati.

OLAP fa parte della più ampia categoria di strumenti di business intelligence che comprende anche database relazionali per la generazione di report. In sintesi un sistema OLAP permette di:

- esaminare grandi quantità di dati
- organizzare i dati in strutture multidimensionali
- analizzare i dati da prospettive diverse
- supportare i processi decisionali.

Si noti che il termine OLAP si contrappone al termine OLTP (Online Transactional Processing), utilizzato per indicare le elaborazioni basate su transazioni dei database tradizionali.

Le interrogazioni di un sistema OLAP richiedono la scansione di un’enorme quantità di record per calcolare i dati di sintesi richiesti per le analisi. Inoltre dovendo fornire risposte in tempi rapidi, l’efficienza e l’interattività diventano una caratteristica irrinunciabile delle sessioni di analisi.

Per fornire adeguati tempi di risposta, nei sistemi OLAP vengono create apposite strutture dedicate e vengono separate le query dell'utente rispetto ai processi di caricamento, aggiornamento e modifica dei dati propri dei sistemi transazionali. Un database OLAP è quindi una rappresentazione delle informazioni del database di origine, fissata in un determinato momento, con la trasformare dei dati da formato relazionale a multidimensionale.

I sistemi OLAP utilizzano il modello multidimensionale per organizzare i dati ed esprimere le relazioni tra i dati stessi, tale modello permette di realizzare query analitiche anche complesse e mirate con ridotti tempi di esecuzione.

La struttura multidimensionale, spesso rappresentata tramite la metafora di un cubo, è utilizzata per organizzare i dati ed esprimere le relazioni tra essi. I cubi memorizzano i dati e forniscono la struttura multidimensionale per l'accesso alle informazioni contenute all'interno del cubo. Ogni cella del cubo contiene dati aggregati relativi agli elementi lungo ciascuna delle sue dimensioni. [4]

1.2 Uso della tecnologia OLAP

La tecnologia OLAP è uno strumento sempre più diffuso e basilare in ambito BI, in grado di rispondere a molteplici esigenze, quali:

- realizzare aggregazioni multidimensionali dei dati per ottenere risultati rapidi e coerenti;
- eseguire query specifiche ed analisi complesse sui dati senza interferire con i sistemi OLTP;
- eseguire calcoli ed aggregazione di grandi quantità di dati;
- generare report dei dati in modo estremamente semplice.

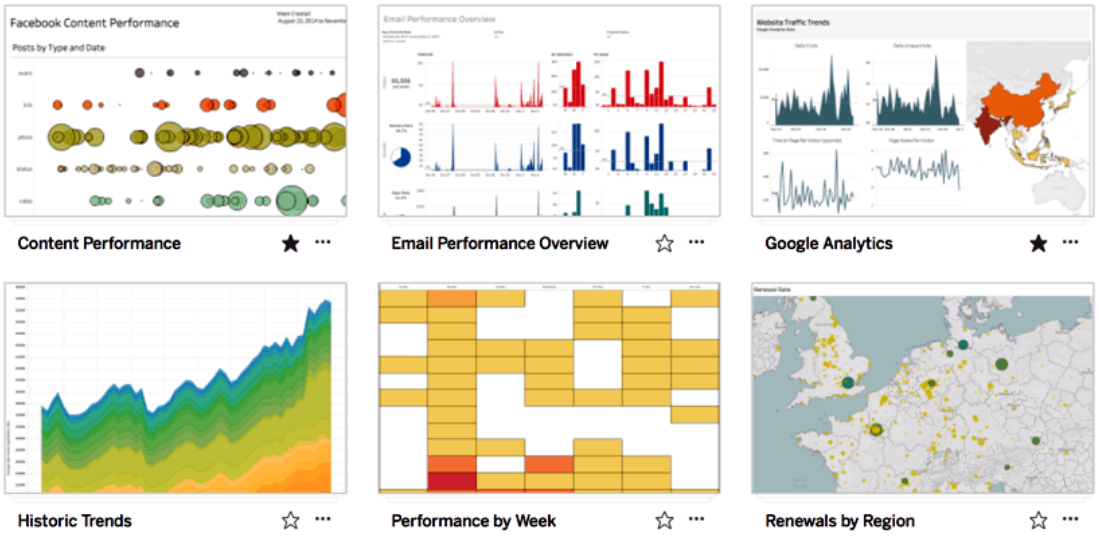
I sistemi OLAP consentono di segmentare i dati multidimensionali visualizzabili in strutture a due dimensioni (ad esempio in una tabella pivot), o di filtrare i dati in base a specifici valori. I processi di segmentazione e selezione dei dati possono essere eseguiti anche su dati provenienti da fonti diverse. Questo consente agli utenti di esplorare i dati, individuare le tendenze,

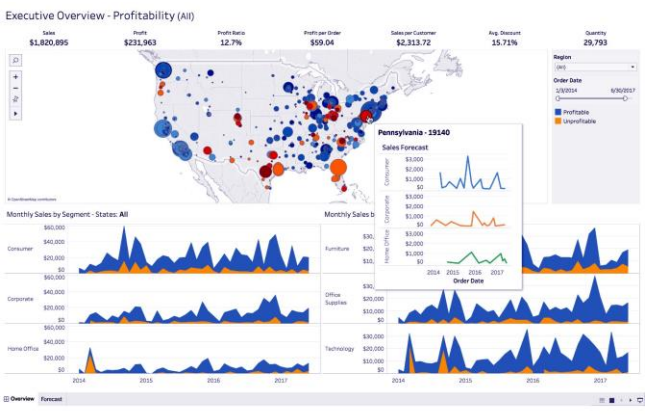
identificare schemi ricorrenti ed altro, senza dover conoscere tutti i dettagli richiesti nell'analisi dei dati tradizionali.

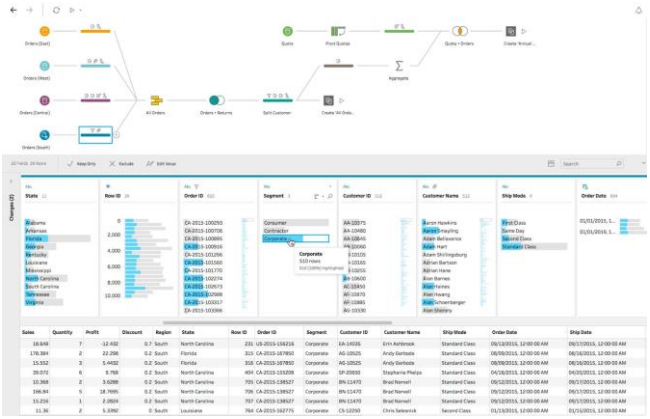
I sistemi OLAP sono ottimizzati prevalentemente per gli scenari che richiedono intensi carichi di lavoro in lettura, quali ad esempio le applicazioni di analisi e business intelligence. Sono in grado di offrire prestazioni molto elevate facilitando l'estrazione e l'analisi delle informazioni.

Uno degli aspetti di forza dei sistema OLAP è sicuramente il livello di sviluppo che hanno avuto in tutti questi anni. Questo ha portando alla realizzazione di ambienti estremamente sofisticati e fortemente interattivi, basati su rappresentazioni grafiche molto efficaci ed accattivanti per l'utente, trasformando il processo di analisi in una sorta di game molto coinvolgente.

Nelle immagini di fig. 1.1 alcuni esempi di output di sistemi OLAP commerciali.







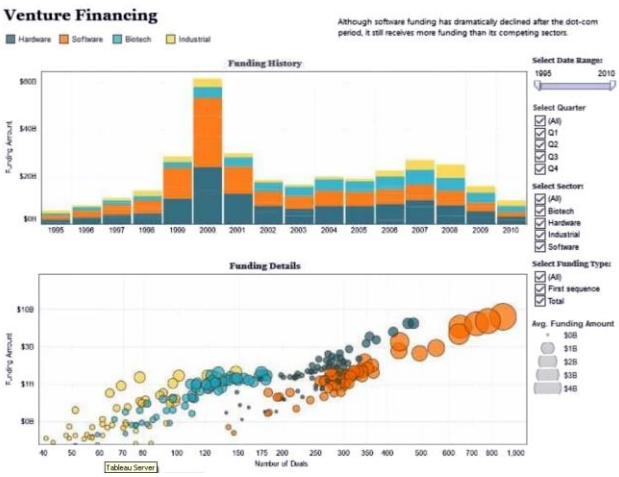


Fig. 1.1 Esempi di sistemi OLAP

1.3 Rappresentazione a cubo

Un cubo OLAP è la struttura per la memorizzazione di dati che permette di eseguire analisi in tempi rapidi, superando uno dei limiti dei database relazionali. In generale un cubo OLAP può avere più di tre dimensioni, per questo si parla di ipercubo.
Il cubo OLAP (cubo multidimensionale o ipercubo) contiene i fatti numerici (**misure**), che sono classificati in base ai diversi aspetti (**dimensioni**).

Le **misure** rappresentano proprietà misurabili come ricavi, costi, tempi, ecc.. e forniscono informazioni sulle quantità che interessa esaminare e valutare. Le **dimensioni** rappresentano elenchi o categorie di elementi correlati che definiscono un determinato aspetto di un'azienda. Possono essere viste come le righe o le colonne di una tabella pivot.

Ad esempio, per un'azienda automobilistica, le dimensioni possono essere i modelli di veicoli, i punti vendita, i reparti di produzione. Nella figura 1.1 possiamo vedere un cubo OLAP che rappresenta i volumi di vendite di una azienda. Il cubo è formato dalle tre **dimensioni** città, tempo e articoli e dalla **misura** vendita, quantificata dal valore all'interno di ogni cella.

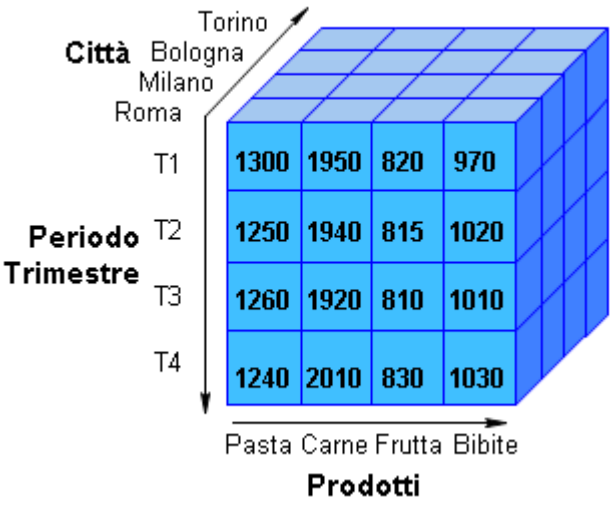


Figura 1.2 Rappresentazione a Cubo

Un **cubo multidimensionale** viene normalmente implementato su database relazionali con uno schema a stella, con al centro la **tabella dei fatti** a cui sono collegate le tabelle che rappresentano le diverse dimensioni. La tabella dei fatti elenca i principali elementi su cui sarà costruita l'interrogazione, mentre le tabelle delle dimensioni specificano le aggregazioni dei dati.

Lo schema **snowflake** si ottiene a partire dallo schema a stella normalizzando le tabelle delle dimensioni. Ogni misura ha un insieme di etichette o metadati ad essa associati ed i valori delle misure hanno significato all'interno delle coordinate del cubo. I metadati del cubo si riferiscono ai nomi delle tabelle, ai nomi delle misure ed ai nomi delle dimensioni nel modello relazionale. I valori delle misure sono memorizzati nei record nella tabella dei fatti ed i valori delle coordinate del cubo sono memorizzati come record nelle tabelle delle dimensioni.

1.4 Sessioni ed operatori OLAP

L'analisi OLAP è una modalità avanzata rispetto alle query SQL per accedere alle informazioni di un data warehouse e permette di esplorare in modo interattivo i dati con sessioni complesse. Una sessione OLAP è una sequenza di interrogazioni del data warehouse attraverso il modello multidimensionale. Ad ogni passo della sessione sono applicati operatori specifici (operatori

OLAP) per trasformare il risultato del passo precedente in una nuova rappresentazione (nuova interrogazione). Gli operatori OLAP più diffusi sono: roll-up, drill-down, slice-and-dice, pivoting, drill-across, drill-through. Di seguito una breve descrizione dei diversi operatori.

- Roll-up

Il **roll-up** (o consolidamento) è l'operazione per aggregare i dati omogenei, accumulandoli o calcolandoli in una o più dimensioni. Produce un cubo con un ridotto numero di piani, eliminando alcune informazioni e riducendo il livello di granularità.

L'operatore di aggregazione è particolarmente utile per analisi su grandi quantità di dati, ad esempio, tutti i punti vendite vengono aggregati in base all'area geografica.

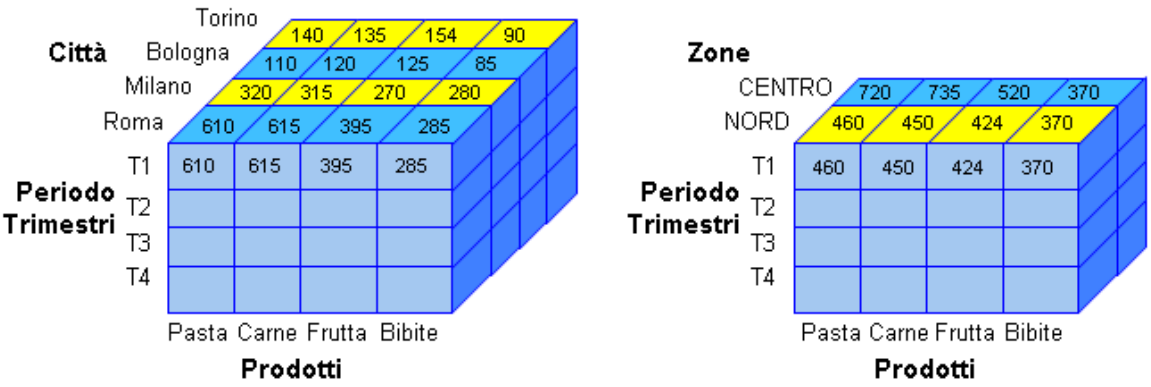


Figura 1.3 Roll-up da Città a Zone

- Drill-down

Il **drill-down** è l'operazione di estensione o esplorazione del dato nelle sue componenti di base, con l'aggiunta di informazioni di dettaglio. Rappresenta l'operazione inversa del roll-up, riduce l'aggregazione dei dati aumentando il livello di granularità.

Il drill-down permette di eseguire un'analisi di dettaglio, esplorando i dati che compongono una determinata misura e suddividendo l'insieme dei dati iniziali in parti più piccole. L'operazione riveste un ruolo chiave ai fini analitici per la comprensione delle costituenti di un dato, ad esempio, per visualizzare le vendite dei singoli prodotti che costituiscono le vendite di una regione.

L'operazione di drill-down può essere eseguita con due diversi percorsi:

a) scendendo la gerarchia costruita sulla dimensione di analisi (ad esempio con il passaggio dai trimestri ai singoli mesi o da una famiglia di prodotti all'insieme dei prodotti che ne fanno parte).
b) applicando relazioni matematiche che legano i dati per ricavare da un dato calcolato le sue componenti originarie (esempio: determinazione del ricavo e costo a partire dal margine generato).

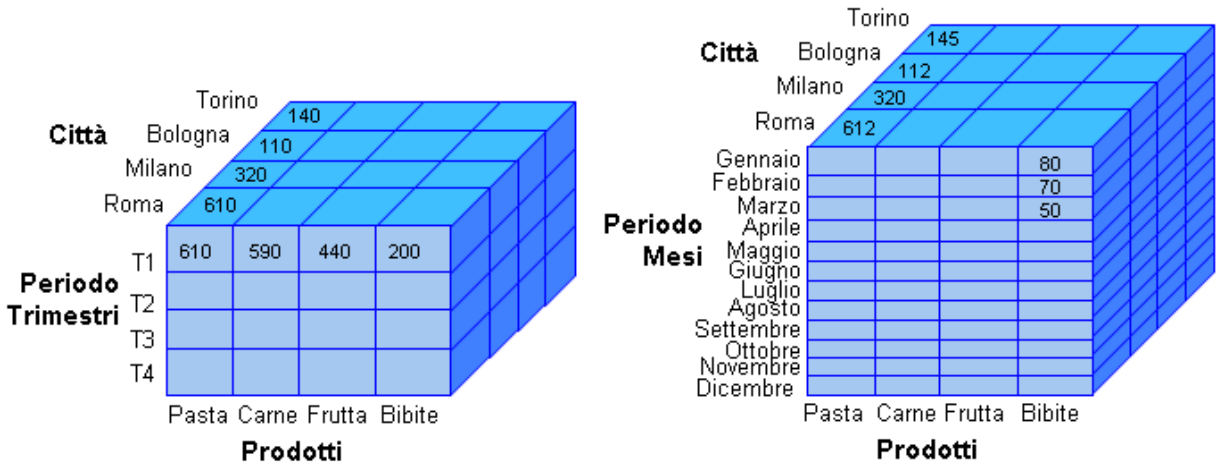


Figura 1.4 Drill-down da Trimestri a Mesi

- Slicing

Lo slicing (affettare) permette di estrarre un piano dal cubo OLAP e corrisponde all'operatore di selezione riferita ad una dimensione; con questa operazione ci si focalizza su specifici sottoinsiemi di informazioni.

L'operazione viene eseguita fissando uno valore per una delle dimensioni del cubo ed estraendo la "fetta" corrispondente. Il risultato è un nuovo cubo con una dimensione in meno rispetto a quello di partenza. Pertanto lo slicing corrisponde ad un taglio su una dimensione.

Esempio: permette di osservare le stesse vendite per venditore, o per data, o per cliente, o per prodotto, o per regione, ecc..

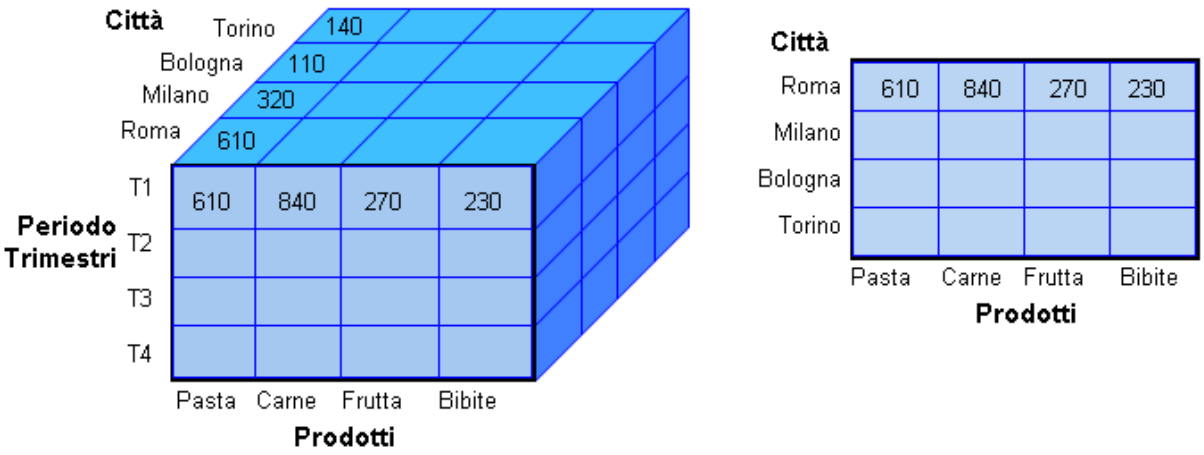


Figura 1.5 Slice per Periodo = 'T1'

- Dicing

Il **dicing** (tagliare a cubetti) corrisponde all'operatore di proiezione su più dimensioni e produce sottocubi dal cubo di partenza. L'operazione permette di estrarre parti dal cubo OLAP producendo un nuovo cubo con le stesse dimensioni lungo gli assi.

L'operazione di dicing viene eseguita per focalizzare l'analisi su un sottoinsieme del cubo di particolare interesse, oltre che per filtrare le informazioni di maggiore interesse, può essere usata anche per estrarre i contenuti dal cubo per visualizzarli da diversi punti di vista.

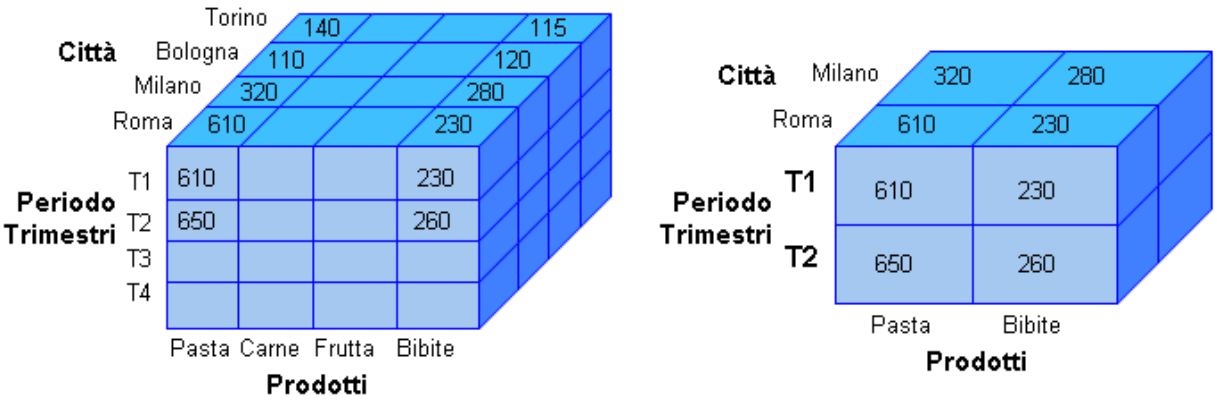


Figura 1.6 Dicing per Città = 'Roma' or 'Milano' and Periodo = 'T1' or 'T2'

- Pivoting

Il pivoting è l'operazione di rotazione delle dimensioni di analisi e permette di analizzare totali ottenuti in base a dimensioni diverse o analizzare aggregazioni differenti. Il pivoting permette di modificare il layout di una rappresentazione multidimensionale e quindi la vista del dato da visualizzare dal cubo OLAP.

Immagine da modificare e fondere: **Pivot**

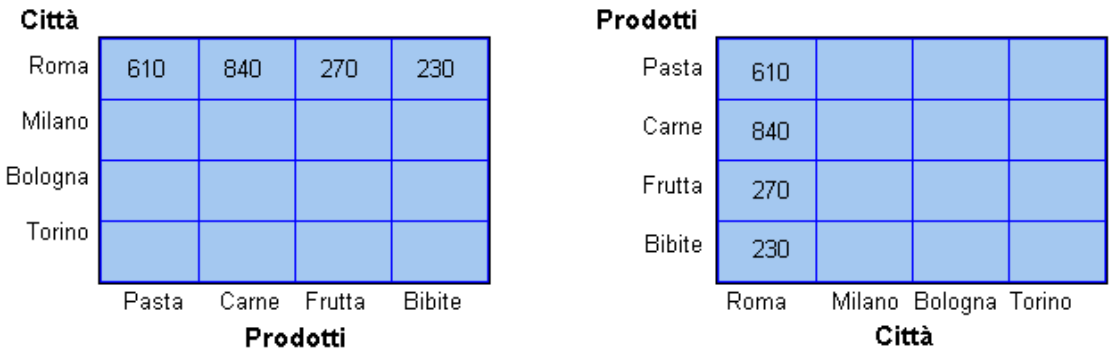


Figura 1.7 Pivot da Città a Prodotti

- Drill-through

Il **Drill-through** è concettualmente simile al drill-down e permette di passare da un livello aggregato al livello di dettaglio appartenente però alla base dati sorgente. Attraverso questo comando è possibile passare dai dati di partenza del data warehouse ai dati operazionali presenti nelle sorgenti riconciliate.

- Drill-across

L'operatore di **Drill-across** permette di navigare attraverso uno stesso livello all'interno di una stessa gerarchia, stabilendo un collegamento tra i dati di cubi diversi per poter confrontare i rispettivi valori. Il collegamento può anche essere realizzato mediante l'applicazione di una funzione alle diverse misure.

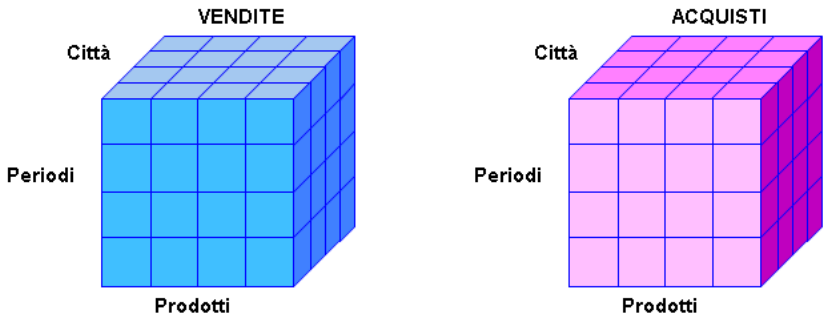


Fig. 1.8 Drill-Across da Vendite ad Acquisti

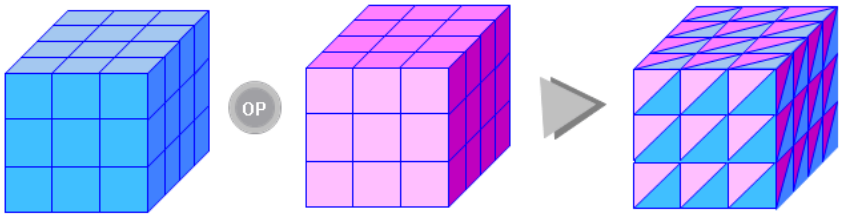


Fig. 1.9 Drill-Across

1.5 Modelli semantici

Per evidenziare il significato dei dati e rendere il processo di analisi il più possibile indipendente dalla strutturazione dei dati stessi vengono adottati modelli semantici.

Un modello di dati semantico è un modello concettuale per descrive il significato dei dati. L'uso dei modelli semantici permette agli utenti di identificare le relazioni anche complesse tra i dati ed eseguirne le analisi più rapidamente.

La modellazione semantica garantisce un livello di astrazione dello schema del database, in modo che non sia necessario che gli utenti conoscano le strutture di dati sottostanti. Questo semplifica la formulazione delle query da parte degli utenti finali in quanto non richiede aggregazioni e join nello schema dei dati. Inoltre le denominazioni (tabelle, colonne, attributi, ..) possono essere rinominate usando alias o nomi più descrittivi, in modo da avere un significato immediato e renderli più adatti al contesto.

Questo tipo di modellazione è prevalentemente usata per gli scenari con intensa attività di lettura dei dati, ad esempio business intelligence e l'analisi OLAP, piuttosto che per l'elaborazione dei dati transazionali con più intensa attività di scrittura (OLTP).

1.6 Problematiche e limiti dei sistemi OLAP

A fronte di una estrema versatilità e dei molteplici vantaggi offerti, i sistemi OLAP presentano alcune problematiche che ne limitano l'impiego e le potenzialità.
Le principali sono:

- Difficoltà di uso e di individuare pattern
- Struttura denormalizzata
- Difficoltà di aggiornamento
- Altri punti deboli (assenza di standardizzazione e modellazione, inaccessibilità ai dati atomici, proliferazione di codice SQL)

- Difficoltà di uso e di individuare pattern

Nonostante l'uso dei modelli semantici, rimane una difficoltà di fondo nell'uso dei sistemi OLAP legata al dover conoscere la strutturazione dei dati e la loro semantica per poter eseguire analisi approfondite.
Una delle limitazioni di una sessione OLAP è la difficoltà a ricavare tutte le informazioni utili e individuare dei pattern per i dati che possano rispondere alle esigenze dell'utente. Da qui anche la necessità di sviluppare modelli avanzati che possano rispondere più agevolmente alle reali intenzioni dell'utente, come verrà descritto nel capitolo successivo .

- Struttura denormalizzata

I dati multidimensionali dei sistemi OLAP non sono normalizzati a differenza delle tabelle relazionali presenti nei sistemi tradizionali. Per funzionare in maniera efficiente OLAP richiede una struttura denormalizzata perché, generando grandi quantità di dati, per ottimizzare i tempi di risposta vengono memorizzate chiavi ridondanti e valori ripetuti.
L'assenza di normalizzazione rende difficile o impossibile il mapping diretto con modelli di tipo entità-relazione od orientati agli oggetti, dove ogni attributo deve essere mappato in una singola colonna.

- Difficoltà di aggiornamento

I sistemi OLAP risultano più adatti a supportare le decisioni aziendali strategiche piuttosto che a fornire indicazioni in tempo reale sui cambiamenti in atto. Questo perché i database OLAP sono in genere aggiornati ad intervalli meno frequenti rispetto ai sistemi transazionali (OLTP). Inoltre, per mantenere aggiornati i dati OLAP, è necessario prevedere un adeguato livello di controllo sulla correttezza e congruenza dei dati. Queste operazioni possono rendere le fasi di aggiornamento lunghe e complesse.

- Altri punti deboli

Altri punti deboli degli strumenti OLAP sono i seguenti:

Assenza di standardizzazione e modellazione. Gli strumenti OLAP sono basati prevalentemente sui modelli dei database tradizionali, ma non hanno un proprio paradigma concettuale di riferimento (come la teoria dei database relazionali), la loro realizzazione è soggetta alle interpretazioni dei diversi produttori software.

Inaccessibilità ai dati atomici. Difficoltà ad accedere al livello atomico del dato. Gli strumenti OLAP funzionano molto bene su dati di sintesi, ma presentano limiti nell'uso su dati analitici di base.

Proliferazione di codice SQL. Nel caso il database su cui vengono effettuate le analisi OLAP non sia multidimensionale (MOLAP), ma sia relazionale (ROLAP), le operazioni di base (slicing, dicing, drilling,..) sono realizzate con complesse query SQL e richiedono molte risorse di elaborazione.

2 INTENTIONAL ANALYTICS MODEL

Il modello intenzionale analitico (IAM) è un nuovo paradigma di interazione complementare ad OLAP [1]. Esso costituisce una nuova proposta di evoluzione del modello OLAP per renderlo più adatto a rispondere alle esigenze dell'utente. Le esigenze sono espresse e formulate a partire dai reali obiettivi dell'utente, con modalità più vicine al linguaggio naturale, piuttosto che legate alla strutturazione dei dati.

La risposta ad una query non è più solo un insieme di tuple, ma viene completata con modelli in grado di rappresentare (ed evidenziare sinteticamente) la correlazione fra i dati e la loro strutturazione evidenziando la loro significatività. In pratica i risultati della query vengono elaborati con algoritmi di data mining per offrire un'analisi più approfondita dei dati estratti e fornire i punti salienti dei dati. L'obiettivo è di evidenziare raggruppamenti di dati che concentrano nuove informazioni utili a soddisfare le intenzioni dell'utente.

Il modello permette all'utente di fare interrogazioni sul database ed eseguire complesse operazioni di estrazione dal cubo senza usare operatori OLAP ma usando espressioni più vicine al linguaggio umano. Le query intenzionali offrono un valido supporto al processo decisionale degli utenti in ambito BI (Business Intelligence), permettendo di esplorare i dati in modo semplificato ed ottenendo informazioni altamente interessanti ai fini delle analisi.

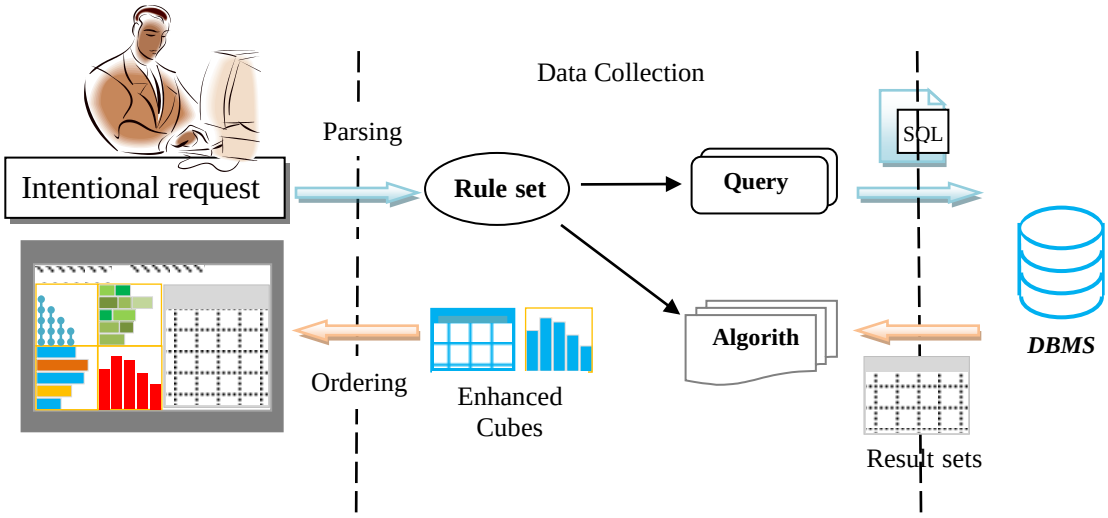
2.1 Descrizione del modello e concetti di base

Il modello prevede che l'utente possa interrogare il database con istruzioni intenzionali, quindi non con istruzioni SQL o tramite operatori OLAP. I risultati delle sessioni di analisi che nell'operatore OLAP erano rappresentate da cubi, ora sono rappresentate mediante DashBoard composto da una serie di grafici e tabelle pivot (vedi figura 2.1).

La realizzazione della dashboard è basata sulla raccolta dei dati e sull'applicazione di algoritmi ai dati stessi. I dati sono raccolti attraverso query al DBMS e successivamete sui dati vengono applicati gli algoritmi pertinenti per estrarre informazioni approfondite. Questo processo viene

definiti sulla base di un insieme di regole (Rule Set) che definiscono ed interpretano la conoscenza sul dominio di analisi.

I dati delle query ed i risultati ottenuti dalla loro elaborazione con gli algoritmi associati vengono rappresentati con un Enhanced Cube (EC). Dopo che tutti gli EC sono stati costruiti vengono ordinati (Ordering) per aumentarne la leggibilità e la comprensione. Infine i risultati sono presentati all'utente come descrizione (Describe) o spiegazione (Explain) o più in generale come risposta alla sua query intenzionale.



Schema fig. 2.1 Processo di valutazione di una query intenzionale

2.2 Espressione di una query intenzionale

Idealmente un utente potrebbe formulare direttamente una sua esigenza sotto forma di una domanda in linguaggio naturale con un sistema in grado di interpretarla e di tradurla in query intenzionale. Viene proposto invece un semplice linguaggio di query intenzionale basato su una serie di operatori specifici costruito con semplici parole chiave del linguaggio naturale.

L’utente esprime la sua intenzione riferendo il cubo dei dati ed esprimendo il suo obiettivo con una serie di operatori intenzionali (EXPLAIN, DESCRIBE, ...) con delle misure di riferimento. L’intenzione è completata con specifiche del modello di dati, misure ed attributi da utilizzare.

Una query intenzionale è sintatticamente definibile come:

WITH <cube> <IamOperator> <measure> [FOR <subcube>]
{USING <explanation model> (<attribute list>)} BY <group> SIZE <integer>

Dove:

WITH è la clausola che specifica il cubo su cui viene eseguita la query.

<IamOperator> è uno degli operatori intenzionali previsti: DESCRIBE, EXPLAIN, ASSESS, PREDICT, SUGGEST. che specifica la misura che l’utente desidera comprendere, spiegare, valutare, ... Ad esempio EXPLAIN specifica la misura che l’utente desidera comprendere.

FOR è una clausola facoltativa che specifica la selezione effettuata sul cubo (per esempio Time.year = 1997, Geo.country = France). In mancanza della clausola FOR non viene applicata nessuna selezione.

USING è la clausola che consente all’utente di specificare il tipo di algoritmi che desidera applicare; può essere usata sia nell’operatore DESCRIBE che nel caso di operatore EXPLAIN.

BY definisce la modalità di raggruppamento dei dati

SIZE specifica la cardinalità desiderata per il risultato.

Un esempio di query intenzionale è il seguente:

WITH foodmart EXPLAIN store_sales FOR the_year=1997 USING correlation (unit_sale)

che estrae i dati dal cubo foodmart relativi alle misure store_sales e unit_sale relativi all'anno 1997 correlandoli fra di loro.

2.3 Data collection ed Enhanced Cube

Tutte le informazioni associate ad una query intenzionale vengono prodotte attraverso un processo di raccolta dei dati e di applicazione degli algoritmi.

In modo formale questo processo di raccolta dati (data collection) può essere definito con una coppia di insiemi (Q, A) dove:

Q è l'insieme delle query sul cubo o sui dati

A è l'insieme dei modelli o algoritmi che possono essere applicati ai risultati delle query.

L'insieme di tutti i dati e dei risultati prodotti degli algoritmi per una specifica query intenzionale costituisce l'insieme di tutte le informazioni disponibili per l'utente a livello di analisi. Questo livello viene costruito mediate una sequenza di azioni costituite da query per selezionare i dati ed algoritmi da applicare sui dati selezionati.

Un'azione può quindi essere definita come una coppia (*q*, *a*) dove *q* è una query del cubo e *a* è un algoritmo scelto fra quelli previsti per lo specifico operatore intenzionale. L'esecuzione di un'azione produce un enhanced cube.

Un cubo avanzato (Enhanced Cube) è una vista multidimensionale dei dati restituiti da una query su un cubo e di una serie di valori e proprietà calcolati con uno specifico algoritmo. Può essere rappresentato da una tabella con le celle significative evidenziate e può estendersi con delle dimensioni particolari dei dati (quali ad esempio tempo e spazio).

Il primo EC corrisponde all'estrazione (visualizzazione) dei dati di partenza specificati nella query intenzionale da parte dell'utente. I successivi EC saranno ottenuti applicando i diversi algoritmi proposti dai modelli della query intenzionale.

Un EC può quindi essere definito come tripla: IC = (C, M, H) dove C rappresenta il cubo di base, M un insieme di modelli ed H una serie di dati salienti (highlights).

2.4 Regole di costruzione (Rule set)

La raccolta dei dati (Data Collector) dipendono dalla query intenzionale dell'utente e costituiscono la base per tutto il processo di analisi. Per questo è necessario definire delle regole per la costruzione di questi dati che possano rispondere al meglio alle intenzioni dell'utente. Tali regole devono anche limitando l'insieme dei dati da generare, ma assicurare la selezione degli insiemi di dati più adatti alle elaborazioni successive.

In genere, le regole per raccogliere i dati devono essere il più possibile di tipo generale per essere applicate a tutti gli scenari di analisi. Allo stesso tempo devono poter essere sufficientemente dettagliate per esprimere al meglio la conoscenza sul dominio informativo.

Inoltre per ridurre la dimensione dei dati di ricerca è necessario prevedere solo regole che abbiano la massima significatività escludendo inutili o ridondanti query sui dati.

2.5 Algoritmi

Gli algoritmi che possono essere utilizzati sui dati raccolti sono molteplici e sono sia di tipo generale che dipendenti dalla tipologia dei dati da analizzare. Algoritmi di tipo generale che possono essere applicati a tutte le diverse dei dati sono elencati di seguito.

- **Detail** determina i valori di dettaglio di una misura, elencando tutti i dati che producono il valore aggregato, rappresenta l'operazione equivalente al drill-down dei sistemi OLAP.

- **Clustering** comprende gli algoritmi da eseguire per raggruppare dati simili del cubo. Per il clustering possono essere utilizzati i classici algoritmi della letteratura quali K-means, DB-Scan ed altri).

- **Correlation** prevede algoritmi per rilevare una correlazione significativa tra una misura ed i predicati di selezioni dello stesso livello. Possono essere realizzati in vario modo, con il test ANOVA, con semplici misure di correlazione, con l'indice di correlazione di Pearson o con Mutual Information per misurare la reciproca dipendenza fra due misure.

- **Trend detection**, comprende gli algoritmi per il rilevamento delle tendenze da applicare alle serie temporali. Possono essere utilizzati algoritmi più o meno sofisticati di trend detection o anche semplici regressioni lineari.

- **Outlier**, algoritmi per il rilevamento dei valori anomali per ricercare dati con una particolare significatività che portano ad evidenziare dei dati del cubo da utilizzare anche come possibili componenti di nuove analisi. Questi algoritmi possono essere basati sulla misura di un punteggio quali z-score o in base alla misura dello scostamento rispetto alla deviazione standard (esempio scostamenti $> 2\sigma$).

2.6 Operatori intenzionali

Il modello intende superare i tradizionali operatori OLAP rivolti alla manipolazione di dati multidimensionali proponendo per l'utente operatori in grado di esprimere direttamente le sue intenzioni sull'analisi dei dati.
Per questo vengono definiti i seguenti operatori da utilizzare per l'interrogazione intenzionale sui dati: Describe, Explain, Assess, Predict e Suggest.
Di seguito sono descritti brevemente tali operatori con un maggior dettaglio per i primi due che sono stati sviluppati nel prototipo di questa tesi.

2.6.1 Operatore Describe

L'operatore Describe permette di arricchire la dashboard dell'utente con ulteriori dati ed informazioni. Questo operatore viene impiegato per ottenere una descrizione di dettaglio della relazione tra livelli e misure espresse nell'intenzione.
Le misure sono possibilmente incentrate su uno o più membri della dimensione (specificando una o più clausole di FOR) o con una determinata granularità (specificando il livello di GROUP BY), utilizzando un determinato numero di cluster (SIZE).
Il modello dell'operatore Describe è il seguente:

WITH cube DESCRIBE measure {,measure}* [FOR subcube]
[BY ({level}) | SIZE (integer)]

Il modello può essere descritto mediante le seguenti due forme.
Prima forma di Describe:

With cube DESCRIBE m FOR v BY level (in questo caso level è un livello)

Seconda forma di Describe:

With cube DESCRIBE m SIZE k (dove k in questo caso è un numero intero)

La prima è usata quando si vogliono raggruppare i dati riferendendoli ad una dimensione che funge da livello. La seconda forma è usata nel caso in cui si voglia ottenere un certo numero di cluster dall'algoritmo K-means.

2.6.2 Operatore Explain

L'operatore Explain fornisce un modello per spiegare osservazioni rilevanti (su una o più misure) sulla base di relazioni nascoste fra i livelli e le misure. In pratica, l'operatore fornisce un modello di causa - effetto dedotto dalle relazioni nascoste fra i dati per risponde alla domanda del perché di una certa osservazione. Attraverso questo operatore si vogliono fornire le spiegazioni dei risultati osservati dall'utente per aumentare le sue capacità di analisi.
L'operatore di Explain viene realizzato mediante la rivelazione automatizzata delle correlazioni nascoste, e fornisce informazioni non direttamente osservabili dai dati.

In genere la spiegazione può essere più complessa rispetto ad una correlazione, perché può richiedere di individuare elementi di causalità con processi lunghi e complessi rispetto ad un semplice algoritmo che determina la correlazione in modo automatico.

Il modello dell'operatore Explain è il seguente:

WITH cube EXPLAIN measure [FOR subcube]
USING explanation model (attribute list) {, explanation model (attribute list)}*

L'operatore di Explain può essere visto come un operatore Describe che invece di classificare i dati li confronta fra di loro utilizzando algoritmi di correlazione.

Per la funzionalità di Explain si possono utilizzare i diversi modelli:

Test statistici tra diverse misure a differenti livelli di granularità dello stesso spazio multidimensionale (con operazioni di roll-up o drill-down) per stabilire un'analogia tra le loro misure statistiche utilizzando specifici test quali ad esempio, t-test ([1]) o F-test ([2]) .

Correlazione di una misura con una o più misure elencate in una lista di altre misure e/o (attributi di) livelli di dimensione.

Formula di regressione che mette in relazione la misura con un elenco di attributi di altre misure e/o (attributi di) livelli e confronto fra i risultati ottenuti dalla formula stessa.

Albero decisionale che classifica l'indicatore rispetto a un elenco di misure raggruppate con livelli di dimensione.

La sequenza di esecuzione dell'operatore Explain è la seguente:

1. **Acquisizione dati**, si ottengono i dati riferiti alle misure specificate dall'intenzione e si costruisce il sotto cubo relativo.
2. **Applicazione del modello**, si calcolano tutti i modelli di spiegazione previsti per l'elenco di attributi specificati. In base ai modelli, l'output comprende le diverse misure calcolate (componenti del modello) che caratterizzano i dati del sottocubo.

---

[1] T-test: test parametrico per verificare se il valore medio di una distribuzione si discosta significativamente da un valore di riferimento. Viene usato quando non si conosce la varianza.

[2] F-test: test statistico per il confronto fra le varianze di due insiemi che hanno la stessa distribuzione normale.

3. **Evidenziazione**, si selezionano i dati (righe) corrispondenti ai valori delle misure più significativi stabiliti dal modello, in modo da ottenere l'insieme delle celle del cubo che costituiscono i punti salienti.

2.6.3 Operatore Assess

L'obiettivo dell'operatore di valutazione è giudicare una o più misure riferendole ad una valore di base ed utilizzando alcuni KPI (Key Performance Indicator) per il confronto. La valutazione, deve fornire una risposta all'utente che chiede un giudizio qualitativo (e/o quantitativo) su una determinata attività in base alle attese o a precedenti risultati.
Il modello caratterizza ogni fatto, o l'intero cubo, confrontandolo con un riferimento di base. Ad esempio, per valutare se le vendite complessive di un prodotto in un periodo dell'anno corrente risultino deludenti o soddisfacenti rispetto alla media degli ultimi anni.
Le misure, riferite ad uno o più membri della dimensione, vengono comparate, per esempio, ai valori passati della stessa misura, o ai valori di altre misure, o a valori di benchmark. Sulla base del confronto viene determinato il grado di soddisfazione della o delle misure da analizzare.

2.6.4 Operatore Predict

Il modello ha lo scopo di prevedere un'evoluzione dei fatti del cubo per fornire una risposta all'utente che chiede una previsione sulla sua attività. Per esempio, una previsione sulle vendite nel prossimo periodo potrebbe portare a predire vendite superiori del 10% rispetto all'anno precedente.
La previsione viene realizzata elaborando i dati derivati dai cubi originali con analisi predittive quali ad esempio analisi di serie temporali o regressioni.

2.6.5 Operatore Suggest

Il modello fornisce un suggerimento di ricerca all'utente per continuare l'analisi. Lo scopo dell'operatore è essenzialmente quello di guidare l'utente verso i dati di maggiore interesse. Ad

esempio, gli utenti che hanno analizzato le vendite di un prodotto hanno poi visto le vendite dello stesso prodotto nei paesi vicini.
L'operazione di suggerire richiede un modello per individuare il tipo di suggerimento ed una strategia per calcolarlo. L'obiettivo è mostrare all'utente dati analoghi a quelli a cui sono stati interessati utenti precedenti o ricercati in analisi simili.
Un sistema di suggest può essere visto come un sistema di previsione in grado di classificare le query dell'utente, calcolare un punteggio di interesse per poi suggerire quelle con il punteggio maggiore.

2.7 Valutazione delle query intenzionali

Sulla base degli operatori e delle componenti precedentemente descritte, la struttura di valutazione di una query intenzionale può essere rappresentata con uno schema su quattro livelli: intentional level, collector level, action level e physical level.
La struttura rappresenta anche uno schema da utilizzare per la realizzazione del modello. Una implementazione basata su tali livelli garantisce una buona separazione delle strutture e bene si adatta anche a differenti modelli implementativi (MVC, …).
- **Intentional level**, rappresenta il livello di interazione con l'utente, comprende la formulazione della query intenzionale e le risposte ottenute che vanno a costruire il dashboard. Può anche comprendere la definizione di regole per la selezione dei tipi di IC.
- **Collector level**, è il livello di raccolta di tutte le informazioni, contiene le regole generali di generazione degli IC e tutte le istanze degli infoCollector selezionati dalle regole e gli algoritmi associati agli IC
- **Action level**, rappresenta il livello delle azioni necessarie a costruire il piano (logical plane) del Collector level. Del livello fanno parte tutti i dati delle query e i risultati degli algoritmi di elaborazione (calcolo min, max, correlazioni, ..)
- **Physical level**, contiene tutte le strutture per l'accesso e l'organizzazione dei dati. A tale livello è anche demandato il ruolo di separazioni della struttura dati di esame in modo da rendere i livelli superiori indipendenti dalla specifica applicazione su cui opera il sistema IAM.

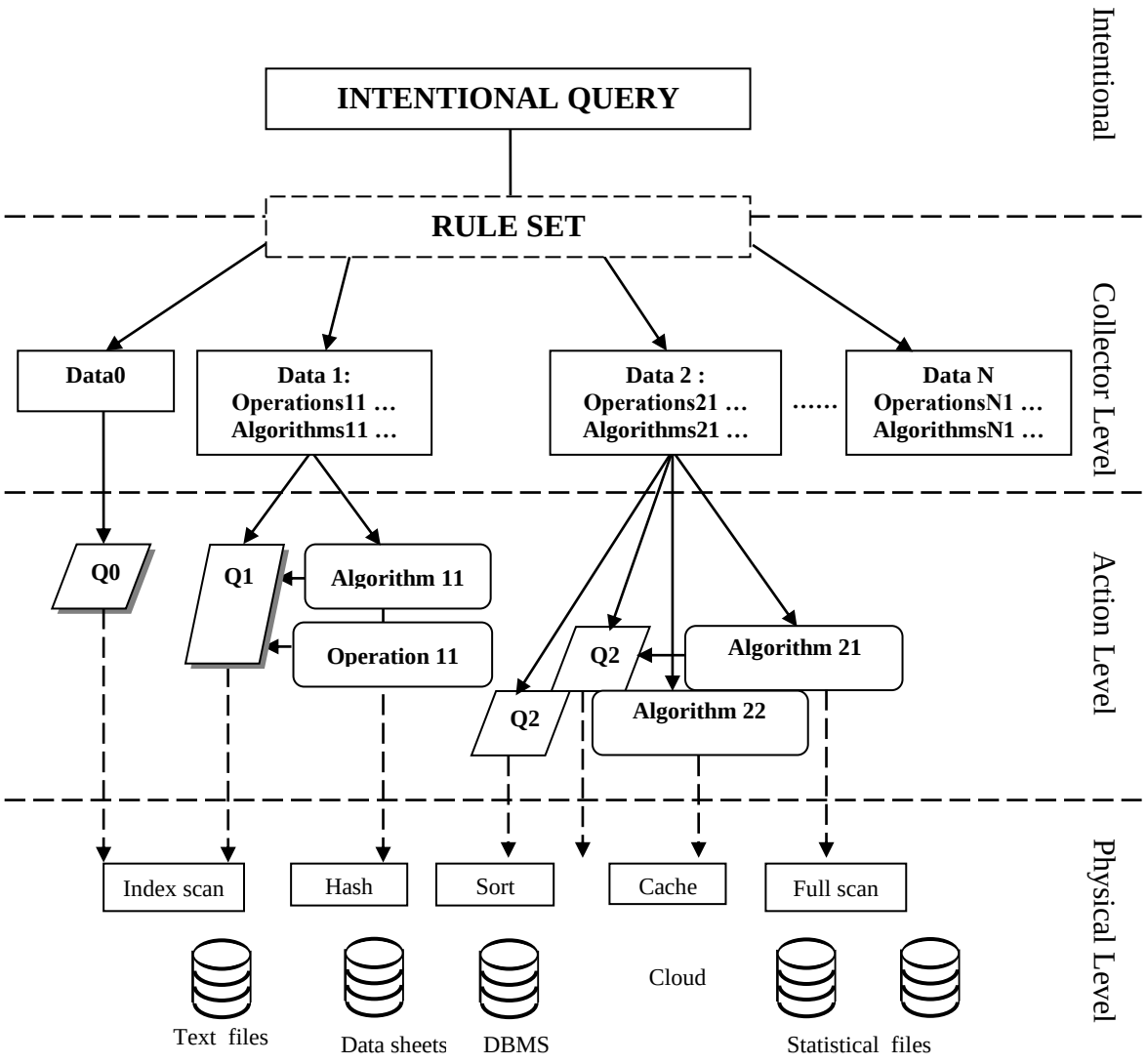


Fig 2.2 Schema valutazione query intenzionale.

2.8 Rappresentazione dei risultati

Dopo aver costruito l'insieme dei dati ottenuti dal database ed aver applicato i diversi algoritmi occorre costruire una rappresentazione dei dati e dei risultati ottenuti.

La rappresentazione dei risultati si basa ancora sul modello che comprende cubo, query sui cubi, risultati algoritmi. Le principali componenti del modello di rappresentazione dei risultati degli algoritmi sono:

- **Schema di clustering, (Clustering)**. Comprende un insieme di cluster, ciascuno con un proprio centroide rappresentativo, oltre ad una un'indicazione dell'insieme dei valori rappresentati ed per ogni componente il contributo allo specifico cluster.
- **Andamento (Trend)**. Suddivide ciascuno dei valori delle misure in base ad una serie storica, andamento temporale e/o stagionalità.
- **Valore anomalo (Outlier)**. Individuazione dell'insieme dei valori anomali rispetto all'andamento generale per caratterizzare criticità o incongruenze.
- **Componenti superiori (Top-k).** Individuare i valori principali o dominanti per caratterizzare sinteticamente l'insieme dei dati analizzati.

3 PROTOTIPO DEL MODELLO

Il prototipo del modello è stato realizzato a partire dalle necessità dell'utente e poi sviluppato secondo il modello MVC (Model View Controller). Lo schema del prototipo è rappresentato in fig. 3.1.

Le necessità dell'utente come verrà dettagliato in seguito (Fig. 3.1- Casi d'uso) comprendono la formulazione delle intenzioni, la visualizzazione dei dati raccolti e delle rappresentazioni grafiche ottenute dall'elaborazione sui dati.

Per soddisfare le necessità dell'utente il sistema deve: analizzare l'intenzione dell'utente, formulare le query necessarie per il recupero dei dati e costruire il dashboard per visualizzare i dati e i grafici. Inoltre il sistema deve mettere a disposizione comandi per la manipolazione dei dati mediante algoritmi predefiniti per arricchire la dashboard con nuovi dati e/o grafici.

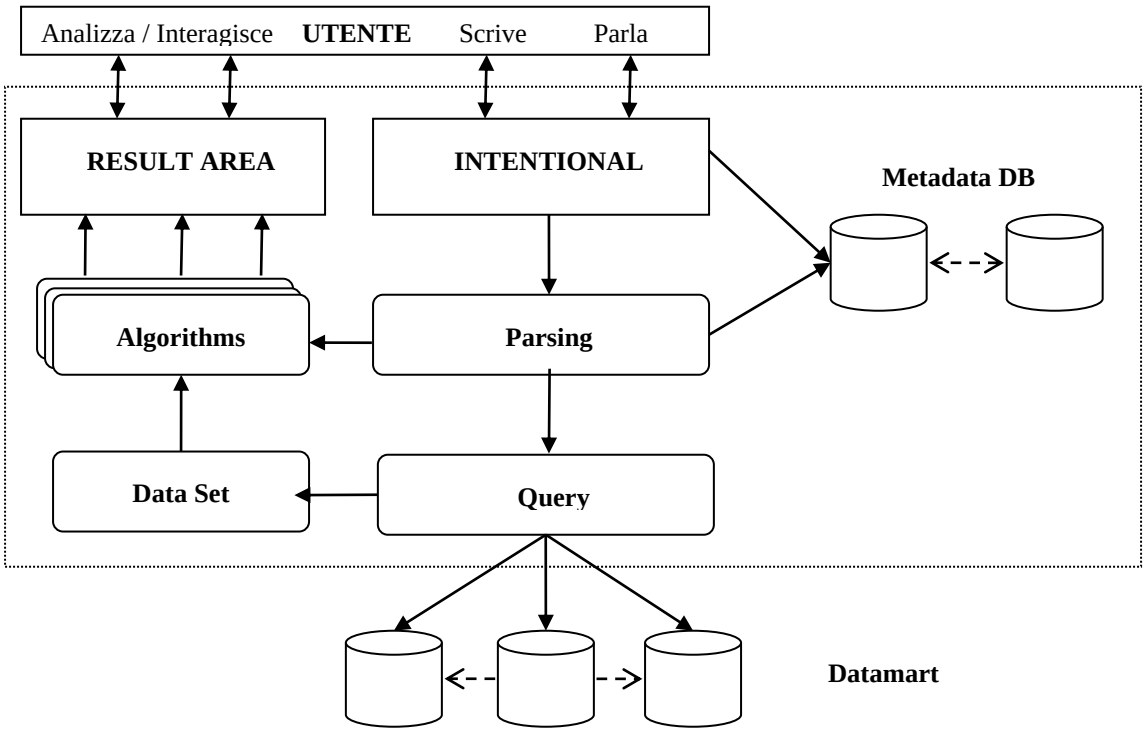


Fig 3.1 – Schema prototipo IAM

3.1 Compiti dell'utente

L'utente deve avere la possibilità di formulare l'intenzione secondo i modelli per gli operatori intenzionali previsti. Dopo aver formulato la sua intenzione, l'utente deve vedere i dati, che saranno visualizzati in forma tabellare. L'utente deve anche poter visualizzare i valori più significativi dei dati raccolti (valori anomali o valori predominanti).
L'utente deve inoltre vedere i grafici corrispondenti ai principali indicatori che meglio rispondono alla sua richiesta. Questi indicatori si riferiscono principalmente alle funzionalità di: Pearson, Anova e Categorizzazione.
Le principali funzionalità dell'utente sono rappresentate nel diagramma di figura 3.1.

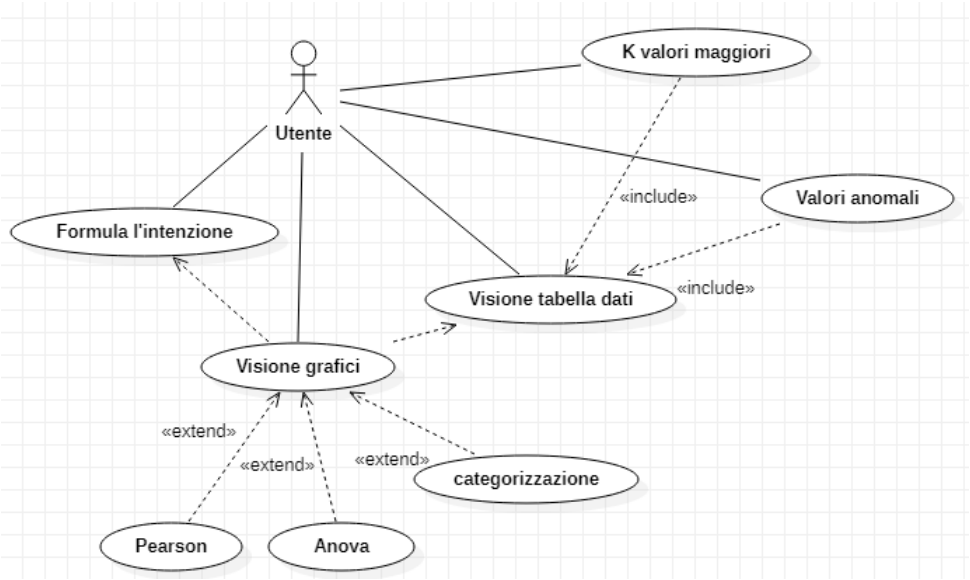


Fig. 3.2 schema delle azioni dell'utente

3.2 Formulazione delle intenzioni

La formulazione dell'intenzione prevede la realizzazione di più schemi di interazione con l'utente per il supporto di:

- Schema libero di scrittura e/o modifica dell'intenzione
- Scelta del modello di riferimento
- Scelta dell'intenzione di riferimento

- Selezione e sostituzione degli elementi del modello con i metadati dal relativo database
- Registrazione audio per la formulazione dell'intenzione con il parlato
- Memorizzazione delle intenzioni formulate con successo
- Componente di help con un database delle intenzioni

- **Schema libero di scrittura** e/o modifica della query, per formulare direttamente l'intenzione con la scrittura diretta della query intenzionale.

- **Scelta del modello di riferimento**, permette di compilare la richiesta intenzionale a partire da un modello sintattico selezionabile da una lista di modelli predefinita (lista modelli). Dopo aver selezionato il modello, questo potrà essere dettagliato e modificato con un editing libero o mediante la sostituzione delle componenti variabili con gli elementi scelti da liste selezionabili.

- **Scelta dell'intenzione di riferimento**, oltre alla lista dei modelli è previsto un elenco di query intenzionali di esempio che viene aggiornato ad ogni nuova query formulata. Tale elenco accumulando tutte le intenzioni eseguite con successo potrà essere particolarmente esteso e quindi dovrà essere filtrato in base al modello selezionato. Ad esempio selezionando un modello relativo al Describe verranno incluse nell'elenco delle query intenzionali solo quelle relative a tale operatore.

- **Selezione e sostituzione degli elementi dei metadati**. Durante la formulazione dell'intenzione il sistema assiste l'utente proponendo le varie opzioni selezionabili da liste costruite in base alle informazioni del conversational: lista cubi, lista misure, lista predicati disponibili, etc.

- **Registrazione audio** per la formulazione di query. Come ulteriore possibilità è stata realizzata l'assistenza vocale per permettere all'utente di formulare la sua intenzione attraverso il parlato. Considerando che le intenzioni sono frasi abbastanza semplici, l'interfaccia può essere adeguatamente controllata e corretta attraverso le strutture del conversational. La frase acquisita dal sistema vocale viene corretta con il matching di tutte le parole con un dizionario ricavato del database dei metadati.

L’approccio vocale permette un’interazione di alto livello, non solo nel tipo di formulazione delle richieste (richieste intenzionali), ma anche nel modo di essere espresse, non più con modalità di scrittura testuale, ma direttamente attraverso il parlato.

- **Componente di help**. Durante la formulazione delle intenzioni può anche essere prevista una componente di help per suggerire all’utente come iniziare o procedere nella sua analisi dei dati.
La struttura di help può essere realizzata con la memorizzazione delle diverse intenzioni formulate dagli utenti associandole alle diverse tipologie di analisi. Questa componente dovrà poi essere integrata con le operazioni (algoritmi) richiesti in fase di dashboard. Il tutto per arrivare a costruire un database delle intenzioni in grado di memorizzare i diversi modelli di analisi dei dati predefiniti e/o costruiti dall’utente durante la sua attività.

- **Database delle intenzioni.** Il database delle intenzioni deve poter raccogliere le esperienze ed il know-how dell’utente sull’analisi dei dati. Il sistema deve essere in grado quindi di mantenere le sequenze di esplorazione e manipolazione dei dati per riproporle in analisi simili. Questa struttura dovrebbe essere sufficientemente indipendente dai dati da poter essere utilizzata in diversi contesti e rappresentare quindi una sorta di modellizzazione delle diverse strategie di analisi dei dati.

3.3 Area di visualizzazione di dati e grafici

L’area di visualizzazione comprende le strutture per la visualizzazione dei dati, i comandi per la loro manipolazione mediante l’applicazione degli algoritmi ed i relativi grafici prodotti.

- Visualizzazione dati. La visualizzazione dei dati si basa su una rappresentazione tabellare sulla quale sono mostrati sia i dati ottenuti dalle query di base che i risultati prodotti dall’applicazione degli algoritmi ai dati. Sulla tabella vengono anche eseguite le evidenziazioni dei dati anomali e dei dati principali.

- Comandi applicazione algoritmi. Sull'area di visualizzazione è presente la lista dei comandi per applicare gli algoritmi ai dati selezionati. La lista degli algoritmi disponibili per ogni operatore intenzionale è predefinita, ma potrà essere modificata dall'utente a livello di configurazione per aggiungere nuovi algoritmi o eliminare quelli ritenuti non idonei per specifici operatori.

- Rappresentazioni grafiche. Sull'area di visualizzazione sono anche riportate le diverse rappresentazioni grafiche generate applicando i vari algoritmi. Le singole rappresentazioni possono essere di volta in volta richiamate e visualizzate nella loro completezza. I dati anomali vengono evidenziati in modo che l'utente possa focalizzarsi sulle eventuali anomalie rispetto a tutti gli altri dati.

- Salvataggio e caricamento. Sono previsti comandi per il salvataggio e il caricamento dell'area di visualizzazione per permettere all'utente di conservare e ripristinare la sua attività di analisi. Il comando di salvataggio permette la memorizzazione dell'area dei risultati con eventualmente associata una descrizione dell'utente. Con il comando di caricamento l'utente potrà selezionare la visualizzazione dei dati da ripristinare. In questo modo si vuole costruire una sequenza di rappresentazioni memorizzate che costituiscono un supporto conoscitivo dell'utente da utilizzare per le successive analisi dei dati. Le aree dei risultati possono essere memorizzate dall'utente assegnandone nomi evocativi. Queste strutture potranno poi essere ricaricate ed eseguite automaticamente. Lo scopo è quello di agevolare l'utente nel riutilizzo dell'analisi dei dati.

3.4 Parsing intenzione dell'utente

Il parsing dell'intenzione analizza la stringa che rappresenta l'intenzione espressa dall'utente individuando tutte le componenti necessarie per eseguire un'interrogazione completa alla sorgente dati. Attraverso tali componenti verranno poi generate le query al data warehouse.

Il processo di parsing comprende le seguenti fasi:

- Analisi sintattica dell’espressione intenzionale
- Costruzione delle strutture per la generazione delle query
- Componente di help come supporto all’utente per una migliore formulazione dell’intenzione corrente.

- Analisi sintattica dell’espressione intenzionale

L’analisi sintattica può essere realizzata attraverso una grammatica o con un parsing specifico (parsing dedicato). Può essere prevista anche una soluzione mista in cui la grammatica viene usata per la parte sintattica, mentre il parsing dedicato viene usato per analizzare aspetti più semantici ed individuando anche eventuali incongruenze nella formulazione dell’intenzione.

- Parsing con grammatica

Il parsing per l’analisi delle richieste intenzionali formulate dall’utente può essere realizzato mediante una grammatica per la sua verifica sintattica.

La grammatica è costruita a partire dai modelli sintattici degli operatori intenzionali ed estesa dinamicamente con le informazioni ottenute dal database conversational. Attraverso quest'ultimo sono definiti ed aggiunti tutti gli elementi riferiti alle misure, predicati e relative clausole di “using by” ed altro.

La grammatica per le due intenzioni Describe ed Explain è riportata nel capitolo 4 Grammatica parsing.

- Parsing dedicato

La non eccessiva complessità dei modelli intenzionali e la presenza di pochi elementi (operatori, misure, predicati) che assumono valori da liste predefinite permette di realizzare un parsing specifico garantendo anche una maggiore efficienza e una migliore adattabilità.

Dal parsing dedicato viene anche ricavata direttamente la struttura da utilizzare per la generazione delle query SQL. Attraverso un approccio dedicato è anche possibile ottenere un parsing in grado di evidenziare eventuali incongruenze della richiesta e suggerire alternative utili al raggiungimento dell'obiettivo dell'utente.

In questo modo viene potenziato il supporto all'utente (già previsto in fase di formulazione dell'intenzione) per guidarlo verso la formulazione di intenzioni più appropriate per la sua analisi.

- Componente di help come supporto all'utente

Il parsing comprende anche un componente di help come supporto all'utente per una migliore formulazione dell'intenzione corrente. Come precedentemente anticipato il parsing deve anche:
- mettere in evidenza le eventuali incongruenze della richiesta intenzionale
- suggerire alternative utili al raggiungimento dell'obiettivo dell'utente.
L'obiettivo del parsing è infatti anche quello di verificare eventuali incongruenze nella richiesta intenzionale suggerendo le azioni correttive o proponendo i modelli più adatti.

Esempi:
with foodmart store_sales for the_month=january using next(month) by store_name.
Il sistema suggerirà: Query to DB not succesfull Explain or Describe index not properly set.

with foodmart describe sum(store_cost), maximum(unit_sales) for the_year by store_name.
Il sistema suggerirà: Query to DB not successful For index not properly set.

with foodmart describe sum(store_cost), maximum(unit_sales) for the_year >= 1997 by.
Il sistema suggerirà: Query to DB not successful Group by level not found.

- Costruzione delle strutture per la generazione delle query

Dal parsing dell'intenzione sono poi generate le strutture per la costruzione delle query. Le strutture comprendono: selezione delle misure, specifica dei predicati, specifica del livello di raggruppamento.

La costruzione delle strutture per le query utilizza le informazioni del database conversational per convertire tutte le misure, dimensioni, cubi, etc... nelle relative strutture e denominazioni del

database dei dati. In questo modo il processo di costruzione risulta indipendente dalla struttura del database dei dati di riferimento.

3.5 Database dei metadati

Il database dei metadati costituisce lo schema di interfacciamento verso il data warehouse. Lo scopo è di realizzare un ambiente descrittivo e semantico per il sistema IAM. Come funzione descrittiva ha lo scopo di individuare i database, le relative tabelle dei fatti e le gerarchie. Inoltre elenca le misure, le dimensioni, i livelli e i predicati.
Questo database contiene tutte le informazioni dei metadati necessari al controllo e all'esistenza delle misure, delle operazioni applicabili alle misure e delle dimensioni espresse nelle intenzioni dell'utente. Esso può rappresentare dunque un modello semantico per separare la struttura logica dei dati dalla loro effettiva memorizzazione nell'ambiente di origine dei dati.

Attualmente ha una prevalente funzione descrittiva, potenzialmente potrebbe essere esteso con delle funzionalità più prettamente semantiche. Come funzionalità semantica dovrebbe permettere di definire nomi o alias alle varie grandezze per rendere le intenzioni IAM il più possibile indipendenti dalla struttura del data warehouse.

3.6 Generatore query

Terminata la fase di parsing con la generazione delle strutture di selezione dei dati, vengono poi generate le query alla sorgente dei dati. Il generatore di query produce una sequenza stringhe secondo il corretto formato SQL.

Le sequenze di stringhe che sono generate si presentano secondo il seguente schema:

Selezione di uno o più attributi dalla tabella dei fatti
Selezione di uno o più attributi dalla tabella delle dimensioni

Selezione di uno o più attributi dalla tabella delle dimensioni per specificare il o i livelli di raggruppamento.

3.7 Struttura del prototipo

Il Sistema è realizzato secondo il pattern MVC (Model View Controller) con il quale vengono strutturate ed organizzate le classi di tutto il sistema e la loro interazione.

Il **Model** svolge la funzione di elaborazione vera e propria e si compone delle seguenti parti:

- Parsing
- Generatore di query
- DB Conversational
- Esecuzione delle query
- Rappresentazione dei dati

La **View** rappresenta l'interfaccia grafica nei confronti dell'utente e realizza le funzioni di:

- Formulazione delle intenzioni
- Visualizzazione dei dati, risultati e dei grafici

Il **Controller** si occupa di acquisire l'input ed i comandi dell'utente, di controllare gli errori e di gestire la comunicazione tra Model e View. Il Controller comprende le funzionalità di:

- Gestione del processo di formulazione delle intenzioni
- Esecuzione del parsing
- Generazione ed esecuzione delle query
- Raccolta dei dati
- Costruzione del dashboard per la presentazione dati.

L'intero sistema è composto dalle classi dello schema di fig. 3.2 che ne descrive le loro relazioni, in termini di dipendenza, (freccia tratteggiata) e composizione, (rombo nero).

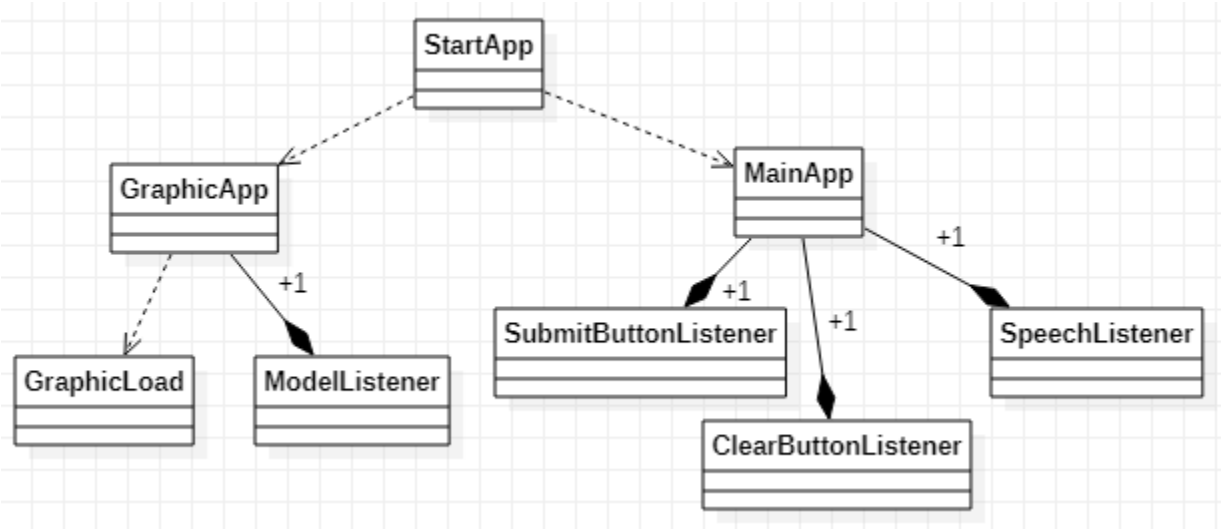


Fig. 3.3 Schema principale del diagramma della classi

La View è realizzata dalla classe GraphicApp che svolge le diverse modalità di input da parte dell'utente, della visualizzazione dei dati e della visualizzazione dei grafici finali. La View comprende anche funzioni di memorizzazione dei modelli e delle intenzioni formulate dall'utente durante la sessione.

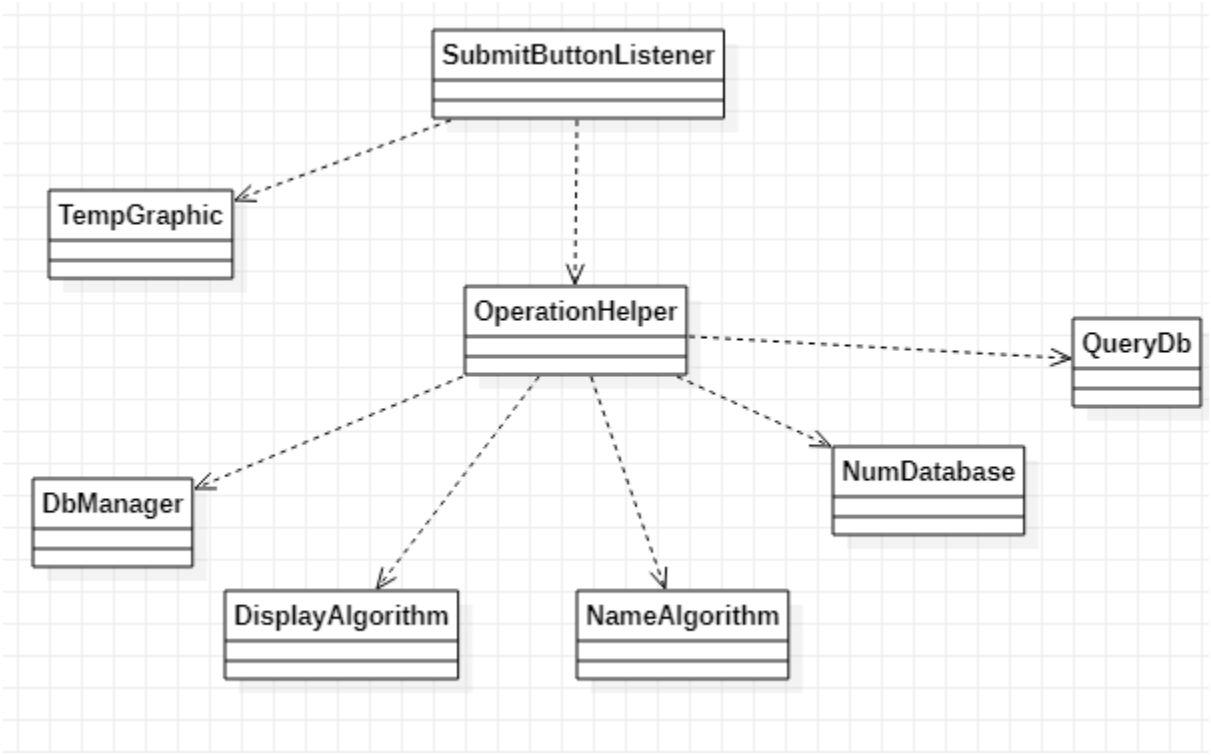


Fig. 3.4 Schema composizione classi del SubmitButtonListener

Il processo di elaborazione dell'intenzione dell'utente viene attivato dalla classe OperationHelper e svolto dalla classe Parser. L'OperationHelper prepara l'ambiente di lavoro del Parser avvalendosi di un insieme di classi (QueryDB, DbManager, ...) per recuperare tutte le informazioni necessarie.

Il Model comprende il parsing dell'intenzione dell'utente, il generatore di query e il database conversational. Il parsing analizza l'intenzione dell'utente per verificarne la correttezza sintattica, individuare le componenti della richiesta e realizzare le strutture necessarie per la generazione delle query.

La verifica dell'intenzione avviene utilizzando i metadati del database conversational per individuare le misure, i predicati e le dimensioni espresse dall'utente. Vengono quindi generate le strutture necessarie alla generazione delle query per recuperare i dati

Il Model realizza anche la struttura degli algoritmi che possono essere richiamati dall'utente sull'insieme dei dati recuperati dalle query.

Il Controller è realizzato dalla classe SubmitButtonListener, che gestisce le varie azioni di richiesta dell'utente (submit, chiamata di algoritmi e costruzione grafici).

3.8 Algoritmi di analisi dei dati e data mining

Gli algoritmi di analisi dei dati utilizzati negli operatori intenzionali Describe ed Explain sono dettagliati di seguito.

Algoritmi utilizzati per entrambi gli operatori

- **Top-k**, l'algoritmo individua i k elementi che hanno valore maggiore rispetto agli altri. Di fatto considera i primi k elementi dell'ordine fra tutti gli elementi.

- **Outlier**, algoritmo per individuare gli elementi che sono anomali ovvero con scostamento maggiore del doppio della deviazione standard.

Algoritmi per l'operatore Describe:

- **K-means**, algoritmo di clustering per categorizzare i dati, secondo i valori delle misure espresse nell'intenzione utente. L'obiettivo del Describe non è l'analisi, ma è quello di categorizzare i dati, appartenenti anche a una singola misura, l'algoritmo K-means permette di costruire i cluster per meglio descrivere il dataset. Vedere grafico 3.4

Algoritmi usati per l'operatore Explain:

- **ANOVA**, per valutare quanto i valori delle misure siano correlati fra loro e mostrare quanto la misura dell'Explain influenza la misura della correlazione.

- **Trend-detection**, per mostrare l'andamento delle vendite o dei costi nel tempo.

Indici per correlazione dei dati per l'operatore Explain:
- **Indice di Pearson**, per la correlazione lineare tra due insiemi di dati.

- **Indici ANOVA (F e p)**, indice di correlazione dove F è espresso come rapporto fra variabilità interna con la variabilità totale e p è il valore di significatività statistica. Il valore p aiuta a capire se la differenza tra il risultato osservato e quello ipotizzato è statisticamente significativa, cioè difficilmente spiegabile mediante la casualità dovuta al campionamento.

Per l'operatore Explain dal momento che prevede l'uso di più modelli ha senso spiegare la misura in questione con confronti piuttosto che categorizzazione, in quanto l'obiettivo non è descrivere o identificare un dato ma è piuttosto quello di analizzarlo.

Di seguito una breve descrizione degli algoritmi utilizzati più significativi

- K-means

L'algoritmo K-means è un algoritmo per suddividere un insieme di oggetti in k gruppi sulla base dei loro attributi. La complessità computazionale è di tipo NP-hard, ma tuttavia esistono diversi algoritmi euristici efficienti che convergono rapidamente a valore di ottimo locale.

L'algoritmo esegue una partizione delle n osservazioni in k cluster in modo che ogni osservazione appartenga al cluster con la media a lui più vicina. Ogni gruppo viene identificato mediante un centroide o punto medio e l'obiettivo dell'algoritmo è di minimizzare la varianza totale fra i gruppi.

Gli algoritmi K-means di tipo euristico seguono in genere la seguente procedura iterativa:

- inizialmente si creano k partizioni assegnando gli elementi iniziali ad ogni partizione, l'assegnamento può avvenire in modo random o usando calcoli euristici;
- si calcola il centroide di ogni gruppo e si costruisce una nuova partizione associando ogni elemento al gruppo che ha il centroide a lui più vicino;
- vengono ricalcolati tutti i centroidi dei nuovi gruppi;
- la procedura è ripetuta fino a quando l'algoritmo non converge, ovvero quando nessun elemento può essere spostato di gruppo.

- ANOVA

L'analisi della varianza (ANOVA, dall'inglese Analysis of Variance) indica un insieme di tecniche statistiche che permettono di confrontare due o più gruppi di dati prendendo in considerazione la variabilità interna a questi gruppi e la variabilità tra essi. L'ipotesi nulla solitamente prevede che i dati di tutti i gruppi abbiano la stessa origine, ovvero la stessa distribuzione stocastica, e che le differenze osservate tra i gruppi siano dovute solo al caso.

Si usano queste tecniche quando le variabili esplicative sono di tipo nominale (discreto). Nulla impedisce di usare queste tecniche anche in presenza di variabili esplicative di tipo ordinale o continuo, ma in tal caso sono meno efficienti delle tecniche alternative (ad esempio: regressione lineare).

L'analisi della varianza di un insieme composto da G gruppi viene calcolata scomponendo la varianza in 2 parti: varianza interna ai singoli gruppi (varianza within) e varianza fra i gruppi (varianza between).

$$\sigma^2 = \sigma_W^2 + \sigma_B^2$$

L'espressione può essere esplicitata mantenendo separate le due componenti nel seguente modo:

$$\sigma^2 = \sum_{g=1}^{G} \sigma_g^2 \frac{n_g - 1}{n - 1} + \sum_{g=1}^{G} (m_g - m)^2 \frac{n_g}{n - 1}$$

La varianza within (σ_W^2) è calcolata come media delle varianze parziali di ogni gruppo pesata rispetto alle loro frequenze relative, mentre la varianza between (σ_B^2) corrisponde alla varianza delle medie parziali pesate con le frequenze relative di ogni gruppo.

Dove:

- m_g sono le medie parziali di ciascun gruppo con n componenti,
- m rappresenta la media totale calcolata dalle medie parziali pesate con le frequenze di ogni gruppo (n_g / n-1 corrispondente al numero elementi del gruppo g / numero totale degli elementi n)

$$\sigma_g^2 = \frac{\sum_{j=1}^{n_g} \left[x_{gj} - m_g \right]^2}{n_g - 1} \qquad m_g = \sum_{j=1}^{n_g} \left[\frac{X_{gj}}{n_g} \right]$$

- Indice di Pearson

L'indice di Pearson è un coefficiente di correlazione di tipo lineare, detto anche indice di correlazione lineare. Esprime la relazione di proporzionalità tra 2 variabili statistiche con un valore $\in$ [-1;1]. I valori estremi (1 e -1) indicano una completa correlazione lineare, rispettivamente positiva e negativa, mentre il valore 0 indica la completa assenza di correlazione lineare. L'indice è calcolato come rapporto tra la covarianza delle 2 variabili e il prodotto delle deviazioni standard delle stesse variabili.

$$\rho(x, y) = \frac{\operatorname{cov}(X, Y)}{\sigma x \sigma y} \qquad r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

dove: $\bar{x}$ ed $\bar{y}$ sono rispettivamente la media dei valori X e Y.

- Trend detection

L'algoritmo di Trend-detection permette di individuare l'andamento di un valore di misura riferito ad una specifica finestra temporale. L'algoritmo individua una linea spezzata che rappresenta l'andamento dei dati in base ai valori della misura di riferimento.

Vengono utilizzati diversi metodi statistici per il trend detection quali:

- punteggio di Mann-Kendall è una valutazione statistica della presenza di una tendenza costante al rialzo o al ribasso nel conteggio nel tempo;
- modello basato sulla distribuzione di Poisson per valutare quanto sia anomalo un valore in una serie temporale confrontato con i valori precedenti. Permette di identificare aumenti (o diminuzioni) improvvisi che possono indicare la presenza di una tendenza.

Queste misure statistiche sono spesso combinate fra loro per determinare la presenza di una tendenza.

3.9 Interpretazione dei risultati

Si prendano in considerazione le seguenti query:

- with foodmart describe sum(store_sales), avg(store_cost) for the_year=1997 by brand_name size 4 (Vedere grafico della figura 3.4).
- with foodmart explain store_sales for promotion_name=wallet savers using correlation(unit_sales) by store_name (Vedere grafico 3.5).

- fvalue: 13.200461525319964

Il rapporto fra la variabilità dei gruppi e la variabilità all'interno di ciascun gruppo è 13.2. Siccome questo rapporto è diverso da 1, si riesce a scartare l'ipotesi nulla.

- pvalue: 0.00132340900510081495

Poiché il valore è inferiore a 0.05 possiamo concludere che fra le variabili ci sia una correlazione, ovvero le vendite complessive dipendono dalle vendite unitarie nei negozi.

- correlation: 0.7345900713733456

Poiché il valore è positivo, è possibile concludere che le due variabili siano direttamente proporzionali, ovvero al crescere delle vendite unitarie crescono anche le vendite complessive.

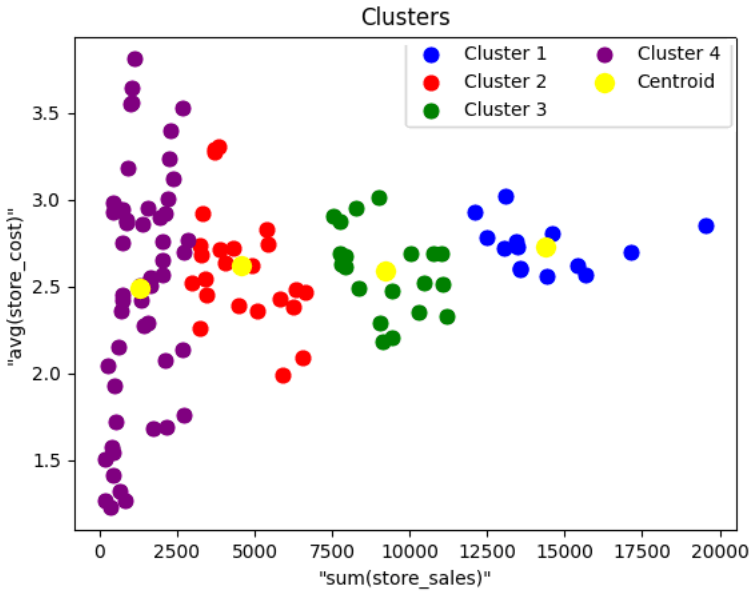


Fig. 3.5: K-means con 4 cluster

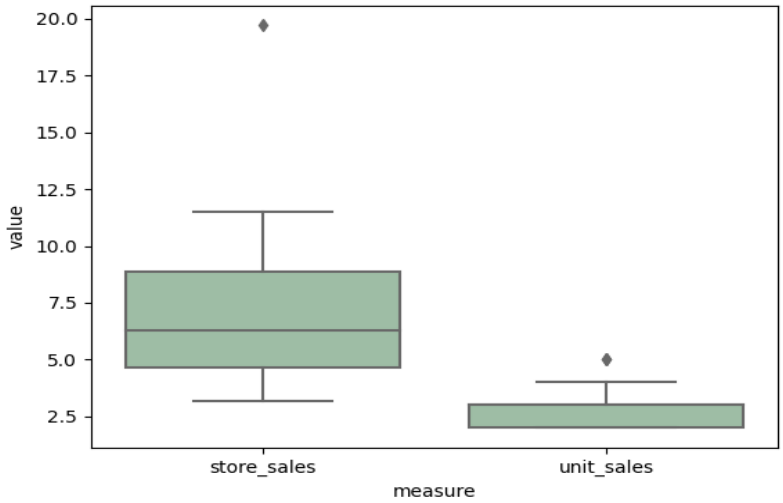


Fig. 3.6: Distribuzione dati anova

4 REALIZZAZIONE DEL PROTOTIPO

L'applicazione è stata sviluppata con il linguaggio di programmazione Java con IDE di sviluppo Eclipse. I database dei dati e di supporto sono gestiti in ambiente MySQL Workbench.
L'implementazione è fatta secondo modello MVC che realizza le componenti di Formulazione dell'intenzione, Parsing e costruzione Dashboard per la presentazione dei dati e risultati.

4.1 View (formulazione intenzione e visualizzazione risultati)

La View realizza l'interfaccia per la formulazione dell'intenzione, ed il livello di presentazione per la rappresentazione del dashboard con i comandi per l'attivazione degli algoritmi.

4.1.1 Formulazione intenzione

La formulazione dell'intenzione avviene attraverso una form di inserimento della stringa (TextArea) ed una serie di (ComboBox) per la scelta del modello, di un esempio di riferimento e dei selettori per la specifica o modifica dei diversi elementi che compongono l'intenzione.

La formulazione del'intenzione avviene attraverso le seguenti strutture:

- **Intentional request** (TextArea) per la scrittura e l'editing dell'intenzione dell'utente

- **Model list** (ComboBox) per la selezione del modello di riferimento. Esso contiene un elenco dei modelli disponibili per query intenzionali. La lista viene caricata leggendo le righe di un file esterno (ListModels.txt) il quale può essere modificato per aggiungere i modelli di nuove operazioni o modelli aggiuntivi.

- **Request list** (ComboBox) per la selezione delle query di esempio. La lista delle query di esempio viene anch'essa caricata da un file esterno (ListQuery.txt). Questa lista viene aggiornata automaticamente dopo ogni query effettuata con successo.

- **Conversational** (Panel) contenente tutti i selettori per impostare i parametri dei modelli delle query. Il pannello contiene selettori di tipo comboBox per impostare i seguenti parametri:
    - **Operator**: selettore per la scelta dell'operatore intenzionale da utilizzare nella query.
    - **Cube**: lista dei cubi disponibili. Per ogni cubo deve corrispondere il relativo database dei metadati dal quale selezionare gli elenchi dei selettori successivi.
    - **Measures**: elenco delle misure disponibili nel database di riferimento.
    - **Predicate**: elenco dei predicati applicabili.
    - **Levels**: elenco dei livelli applicabili per i raggruppamenti.

L'applicazione di un selettore determina una sostituzione nella stringa intenzionale (query) con il valore selezionato. Nel caso di una query costruita a partire da un modello, la corrispondente stringa di riferimento (identificata dalle parentesi angolate <xxx>) viene sostituita con il valore impostato. Esempio <cube> sostituito con 'foodmart' . Nel caso di una query contenente già un valore esplicitato, esso verrà sostituito con il nuovo valore selezionato. Esempio 'foodmart' sostituito con 'covid_mart'.

- Record audio. comandi (Buttons) per attivare e terminare il processo per formulare l'intenzione con modalità vocale. Il button Record attiva un thread di interpretazione vocale. Il thread interpreta l'audio dell'utente producendo la relativa stringa che viene aggiunta alle textarea di response. La frase verrà poi elaborata con una funzione di matching (tokenSimilarity, sviluppata da precedenti lavori) per eseguire la correzione di ciascun token della stringa vocale. Per ciascun token viene infine considerata una lista di espressioni ed elementi che vien definita internamente (attraverso enumerazioni) oppure da query al database dei metadati.

- Submit: terminata la formulazione dell'intenzione, essa viene lanciata tramite il comando Submit che avvia tutto il processo di elaborazione: parsing, query e creazione della dashboard. In caso di errore viene segnalato indicando la causa e l'origine sul'area dei messaggi.

- Area messaggi: TextArea per la visualizzazione degli eventuali messaggi di errore e per la visualizzazione del processo di elaborazione in corso.

4.1.2 Interazione vocale

Essendo il sistema IAM pensato per rispondere alle intenzioni dell'utente formulate in modo diretto attraverso un linguaggio naturale, si è voluto inserire un'interfaccia vocale per portare il sistema verso un approccio diretto con l'utente, anche se questo non era tra gli obiettivi del lavoro.

Sono state utilizzate API di Google, utilizzate con apposita chiave, per produrre l'intenzione mediante comando vocale. Un thread dedicato agisce sull'intenzione che si sta formulando in modo vocale e mostra di volta in volta l'interpretazione che il sistema dà alle singole parole pronunciate dall'utente.

La correzione delle parole avviene con il dizionario ricavato dai database conversational per le diverse tipologie di misure, dimensioni del database di dati.
Attualmente viene utilizzata la lingua inglese, ma utilizzando un'estensione del conversational. è anche possibile realizzare il supporto per lingue diverse quali ad esempio Italiano.
Esempio di intenzione vocale, (non corretta) :

which coming master explained for sale on the year 2021

Esempio di intenzione vocale, (corretta parzialmente) :

with covid_mart describe explained for sale on the year 2021

4.1.3 Costruzione area di visualizzazione

L'area per la visualizzazione dei dati, è il componente centrale della JFrame IAM, che rappresenta tutta l'applicazione.
Il componente principale dell'area di visualizzazione è costituito da una struttura tabellare (Table) e da un elenco di comandi (Button) oltre che da immagini (ImageIcon) dei grafici costruiti.
Nell'area di visualizzazione sono anche presenti i comandi di **save** e **load** per la memorizzazione e ripristino dei dati e grafici. L'operazione di salvataggio consiste nel salvare la query che ha

generato i dati con la sequenza degli algoritmi che sono stati applicati e i relativi grafici generati. Il ripristino determina l'esecuzione automatica della query e l'esecuzione di tutti gli algoritmi attivati.

- **Tabella dati**. I dati ritornati dalle query e dagli algoritmi vengono visualizzati in una apposita tabella. La tabella viene inizializzata con le label identificative sulle colonne e riempita con i singoli valori nelle righe. Per il riempimento vengono aggiunte righe che rappresentano le righe del dataset. Le celle della tabella vengono riempite singolarmente con i valori caricati dal file dati csv. Sulla tabella sono inoltre evidenziati i valori anomali evidenziando le righe corrispondenti.

- **Comandi algoritmi**. Sull'interfaccia di visualizzazione dei risultati vengono inseriti i pulsanti relativi agli algoritmi disponibili per l'operatore della query intenzionale. All'esecuzione di questi algoritmi vengono aggiunte nuove colonne per i nuovi valori calcolati, etichettate con il nome dell'algoritmo.

- **Grafici**. Sull'area dei risultati viene inoltre riportata la sequenza dei grafici in forma di icone che rappresentano tutti i grafici che possono essere visualizzati attivando i diversi algoritmi..

- **Comandi di salvataggio e ripristino**. All'area di visualizzazione sono infine aggiunti i comandi di save e load per il salvataggio della visualizzazione corrente comprensiva dei dati visualizzati e degli algoritmi applicati.

4.2 Model (Parsing, Query ed Algoritmi)

Il model comprende l'esecuzione del parsing , la generazione delle query e l'elaborazione degli algoritmi.

Il parsing esegue la traduzione dell'intenzione espressa dall'utente in una o più query SQL. Queste ultime sono poi eseguite per generare il dataset di riferimento. Al dataset sono applicati gli algoritmi di elaborazione per il dati mining dell'analisi vera e propria.

4.2.1 Parsing

Il processo di parsing viene attivato subito dopo il completamento dell'intezione da parrte dell'utente tramite il comando submit. Riceve in input la stringa che rappresenta l'intenzione e genera le strutture di identificazione dei token degli operatori.

Il parsing esegue la scansione della stringa intenzionale per analizzarla e verificare la query utilizando i metadati. Genera le strutture JSON dei token che verranno successivamente utilizzate per la generazione delle query SQL per l'interrogazione del database di riferimento.

In sintesi l'intero processo si svolge in due fasi:

- prima fase, da stringa intenzionale a JSON,
- seconda fase, da JSON a query SQL. (descritta in 4.2.2 Costruzione ed esecuzione delle query)

Le due fasi sono precedute da un passo preliminare per la costruzione delle liste di riferimento relative ai metadati.

- Costruzione liste di riferimento

Inizialmente sono definite e costruite tutte le liste di riferimento necessarie al processo di parsing e alla costruzione delle query SQL. Le liste sono costruite con enumerazioni interne al sistema e con interrogazioni al database dei metadata (conversational, covid_metadata, ..).

Le enumerazioni interne al sistema (Enum) definiscono le seguenti informazioni:

- elenco dei token delle query intenzionali (TokenOperator: with, explain, describe, using, ..)
- elenco dei database disponibili (EnumDataBase: foodmart, covid_mart, ..)
- elenco degli algoritmi disponibili nel sistema.
- elenco delle denominazioni temporali (mesi e giorni della settimana)

Attraverso interrogazioni al database dei metadati vengono costruite: le liste delle misure, le liste dei livelli e dei predicati relativi al database di origine dei dati.

Tutte le strutture delle liste e delle enumerazioni vengono costruite in fase di avvio dell'applicazione, in quanto sono utilizzate sia in fase di costruzione dell'intenzione da parte dell'utente che in fase di parsing vero e proprio.

- Trasformazione da stringa intenzionale a JSON.

La prima fase del parsing esegue l'analisi della stringa di input e produce le strutture JSON per la generazione delle query. La fase si compone delle seguenti operazioni

a. Identificazione del modello (Describe, Explain, ..)
b. Esame del modello (Describe o Explain)
c. Generazione strutture JSON

a. Identificazione del modello

Il Parsing lavora sulla stringa di input eseguendo la scansione dei diversi token della stringa ed eliminado i token elaborati.

Inizialmente esegue la ricerca del token di base "with" dalla stringa, successivamente esegue la ricerca del token di un operatore e l'individuazione del database (cubo) di riferimento.

L'individuazione del database di riferimento permette di eseguire la connessione al database (MetadataConnection) specificato dalla query intenzionale.

La ricerca del token di un operatore (describe, explain, ..) permette di selezionare la specifica sequenza di parsing da eseguire (Esame del Modello Describe | Explain).

b. Esame del modello (Describe, Explain)

Nel caso di operatore Describe viene eseguita la ricerca (operatingParsing) di tutte le componenti del modello sintattico relativo (USING, BY, FOR, ..) compresa l'individuazione di misure, predicati e livelli.

Nel caso di operatore Explain viene eseguita una scansione dedicata (findExplain) per individuare la struttura sintattica dell'Explain che risulta più complessa di quella dell'operatore Describe.

Dalla scansione della query intenzionale vengono individuati i predicati o selettori (SC - Selector Collection), le misure (MC – Measure Collector) e le operazioni di raggruppamento group by (GC – Group Collector).

- **Predicati o selettori**. Per ogni predicato o selettore (SC) viene individuata la lista dei comparatori (COP: EQ | GT | GE | LT | LE) presenti nell'intenzione, associandoli ai rispettivi attributi di riferimento (ATTR) ed ai valori esplicitati (VAL).
- **Misure**. Per ciascuna misura (MC) viene individuato il nome della misura (MEA) ed il tipo di operazione (OP). Esemipo avg(unit_sale) individua l'operatore di media applicato alla misura (unit_sale)
- **Group by**. In presenza di operazioni di raggruppamento (GC) vengono individuati i nomi dei diversi raggruppamenti dei livelli (nome-groupBy).

La mancanza degli elementi fondamentali (non opzionali) o la loro non corretta composizione determina l'interruzione del processo generando segnalazione di errore.

c. Generazione strutture JSON

Una volta individuati tutte le componenti del modello sintattico dell'operatore intenzionale vengono generate le relative strutture JSON. Nella struttura JSON sono riportati:

- i selettori o predicati (SC) con i valori specificati (VAL), i comparatori (COP) ed i nomi degli attributi (ATTRIB).
- le misure (MC) con gli operatori (OP) e il nome della misura.
- le operazioni di group by (GC) con il nome del livello di raggruppamento.

Esempio di struttura JSON:

```
{"SC": [( {  "VAL": "num-value",  "COP": "=",   "ATTR": "nome-attributo"   },)* ],
 "MC": [( { "OP": "tipo-operazione",  "MEA": "nome-misura"   },)*],
 "GC": [  "nome-groupBy"] }
```

4.2.2 Costruzione ed esecuzione delle query

La fase di costruzione ed esecuzione della query comprende due operazioni:

a. Costruzione delle query (trasformazione da JSON a query SQL)
b. Esecuzione delle query e generazione dati.

a. Costruzione delle query (trasformazione da JSON a query SQL)

In questa fase viene elaborato l'oggetto JSON creato dal parsing per generare le stringhe delle query SQL. La costruzione della query avviene esaminando tutto il JSON prodotto dal parsing e prelevandone le diverse componenti.

Sono individuate le misure che verranno selezionate nel "select" della query e le dimensioni su cui valutare il predicato di "where" della query SQL (QueryGeneratorChecker). Sono poi costruite le diverse parti della query ed infine generata la query SQL componendo le diverse strutture (createQuery).

Nella generazione delle query sono utilizzate (getData) le liste di misure, predicati, livelli della struttura dei metadati relative al cubo selezionato per verificare le misure e/o i pedicati/livelli espressi nell'intenzione dell'utente.

b. Esecuzione delle query e generazione dati

Le stringhe delle query SQL prodotte sono eseguite mediante l'interfacciamento con il database SQL (OperationHelper.writeFile).

Dall'esecuzione delle query (executeQuery) sono prodotti i dataset che vengono convertiti in file di formato csv da utilizzare nelle elaborazioni successive di visualizzazione e applicazione degli algoritmi.

4.2.3 Realizzazione algoritmi data mining (java e python)

Per la realizzazione degli algoritmi si è scelto di avere un sistema il più possibile aperto per sfruttare le potenzialità di altri ambienti, anche in vista di possibili sviluppi futuri. A tale

proposito si è scelto l'ambiente Python ricco di librerie con funzioni di analisi dei dati anche con approcci AI e particolarmente adatto all'elaborazione data mining.

Per quanto detto la realizzazione degli algoritmi avviene oltre che in ambiente java anche ambiente Python. In ambiente java sono stati implementati gli algoritmi più semplici ed immediati quali, top-k e l'outlier, mentre per quelli più complessi si sono utilizzate le librerie Python.

In ambiente java sono state realizzate due componenti per interfacciare e attivare tutti i moduli esterni (Python).

- Una prima compenente (PythonAlgorithm in java) si occupa di lanciare gli algoritmi propri dell'operatore explain: ANOVA, indice di Pearson e Trend detection.
- Una seconda compenente (PythonKmeans in java) usata per eseguire l'algoritmo K-means per categorizzare i dati raccolti.

Sono state realizzate due componenti separate, perché i file prodotti dai moduli esterni sono strutturalmente diversi; infatti nel caso dell'ANOVA e dell'indice di Pearson vengono prodotti solo risultati numerici, mentre nel caso del K-means sono prodotte strutture più complesse per rappresentare i cluster di raggruppamento dei dati.

Tutti i moduli esterni lavorano a partire dai dati (csv) e producono dati testuali (txt) con i relativi dati numerici e visualizzano i relativi grafici di rappresentazione.

- **Algoritmi e grafici prodotti da moduli esterni (Python)**

- Indice di Pearson

L'indice di **Pearson** viene calcolato con la funzione **pearsonr(dataX, dataY)** della libreia scipy.stats del Python. La funzione a partire dagli insiemi di dati (dataX, dataY) riferiti a due diverse misure calcola il coefficiente di Pearson che esprime il grado di correlazione fra gli insiemi di dati.

Il coefficiente viene calcolato con l'espressione definita nella sezione Pearson di 3.8 Algoritmi di analisi dei dati e data mining. Nel calcolo del valore viene assunto che i due set dati (dataX e dataY) siano normalmente distribuiti. Il calcolo è preceduto dalla visualizzazione del grafico della distribuzione dei dati con gli assi relativi alle due misure in questione (Fig. 4.1 Person index).

- ANOVA

L'ANOVA viene calcolata attraverso la funzione di libreiia **f_oneway(dataset)** della libreia scipy.stats. L'indice calcolato rappresenta il valore che rigetta o meno l'ipotesi nulla, rappresentata dal fatto che tutte le medie dei gruppi siano uguali fra loro. Quindi un valore diverso da zero esprime il fatto che almemo un gruppo ha media diversa da tutto il resto. Per **ANOVA** sono visualizzati i seguenti grafici:

- distribuzione dei dati con moda (Fig 4.2 Distribution Data)
- retta di regressione fra i quantili teorici ed i quantili predetti (Fig. 4.3 Regression line)
- frequenza dei residui (Fig. 4.4 Frequency Residuals).

- Trend-detection

Non viene utilizzata nessuna libreria, ma viene calcolato come differenza fra le sequenze dei valori delle due misure e generati i corrispondenti grafici. Per il trend-detection sono generati i grafici di:

- valore delle misure per i diversi predicati temporali. Ad esempio valore delle misure di store_sale e di cost con quantità dei dati e valori massimi raggiunti.. (Fig. 4.5 Andamento su diversi intervalli temporali).
- andamento come differenza tra i valori delle misure specificate in relazione a due intervalli temporali, precedente, successivo. (Fig. 4.6 Trend detection).

- K-means.

Il K-means viene calcolato con la funziione **KMeans(nCluster,labelMode,rndSstate)** della libreria sklearn.cluster. Dove nCluster è il numero di cluster da determinare, labelMode è la modalità di lacvoro e convergenza dell'algoritmo e rndState è la generazione random per la

determenizione iniziale dei centroidi dei cluster. Il risultati sono rappresentati con il grafico di Fig. 3.5: K-means con 4 cluster.

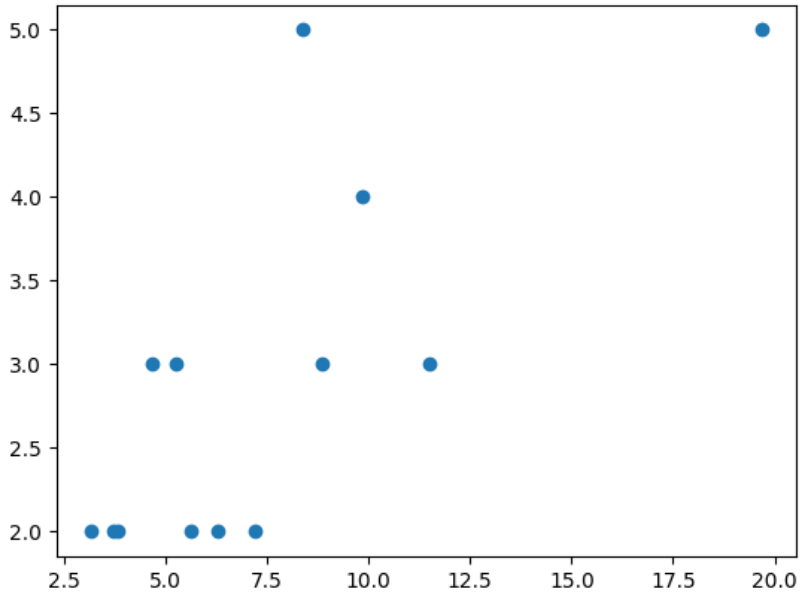


Fig. 4.1 Person index

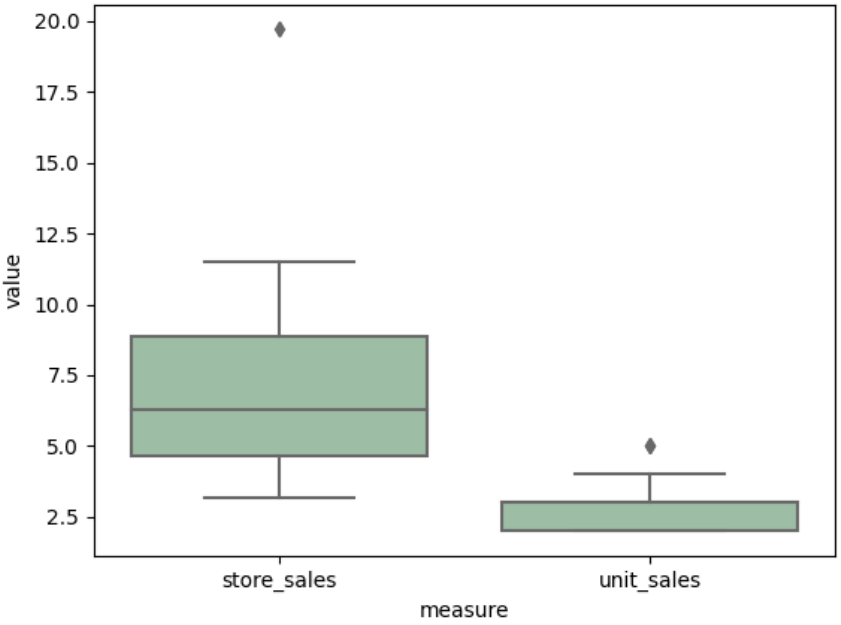


Fig. 4.2 ANOVA - Distribution Data

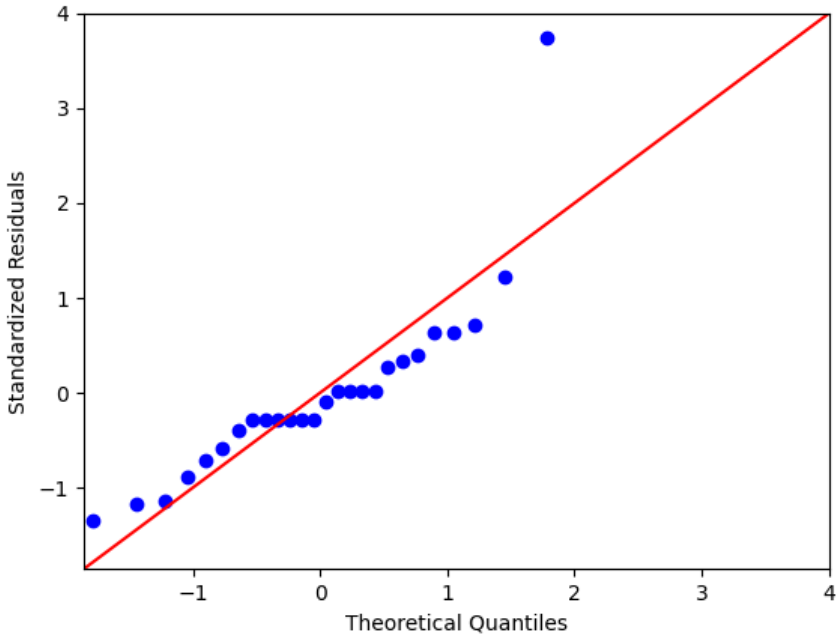


Fig. 4.3 ANOVA - Regression line

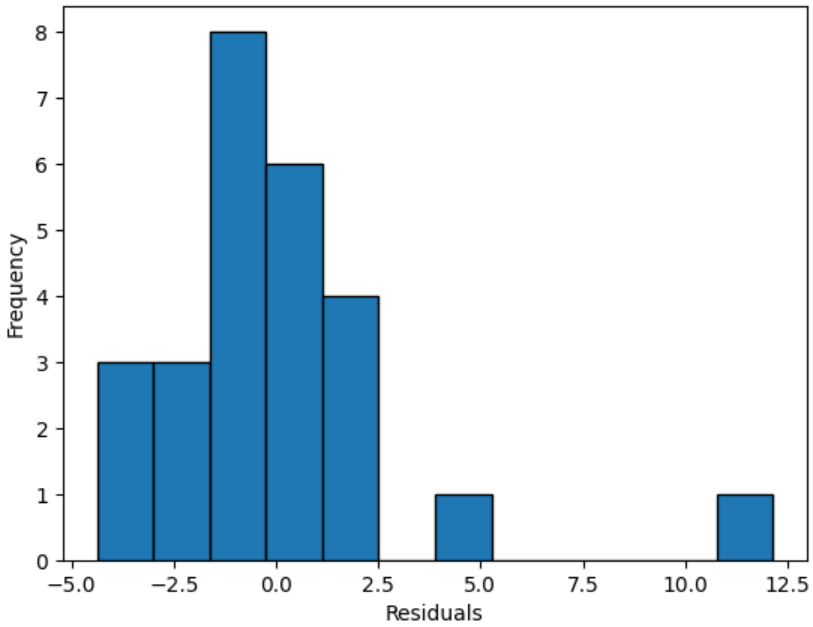


Fig. 4.4 ANOVA - Frequency Residuals

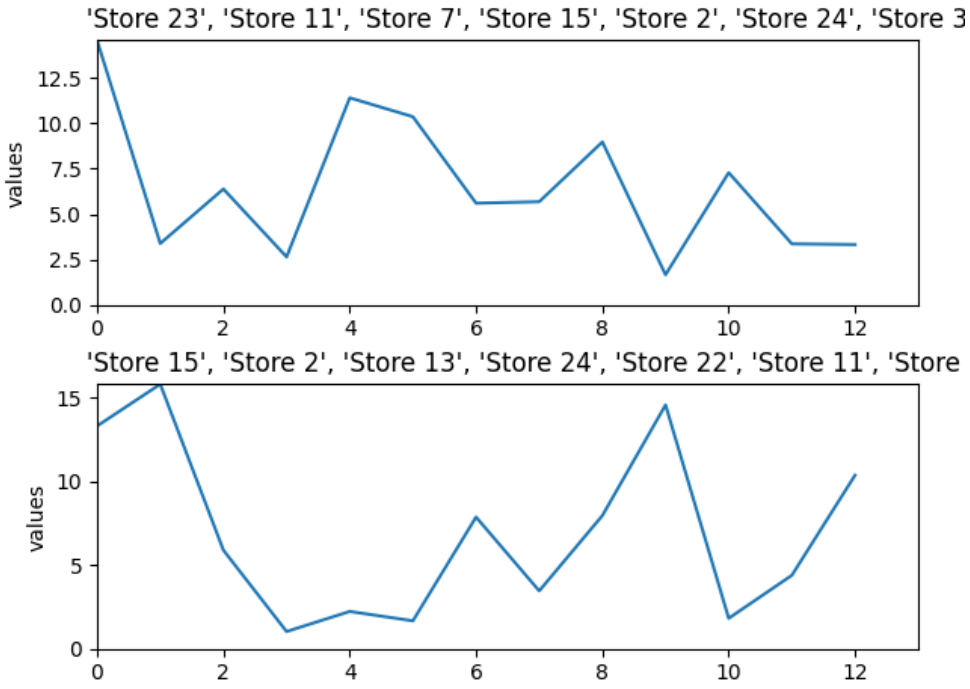


Fig. 4.5 Trend detection – andamento su diversi intervalli temporali.

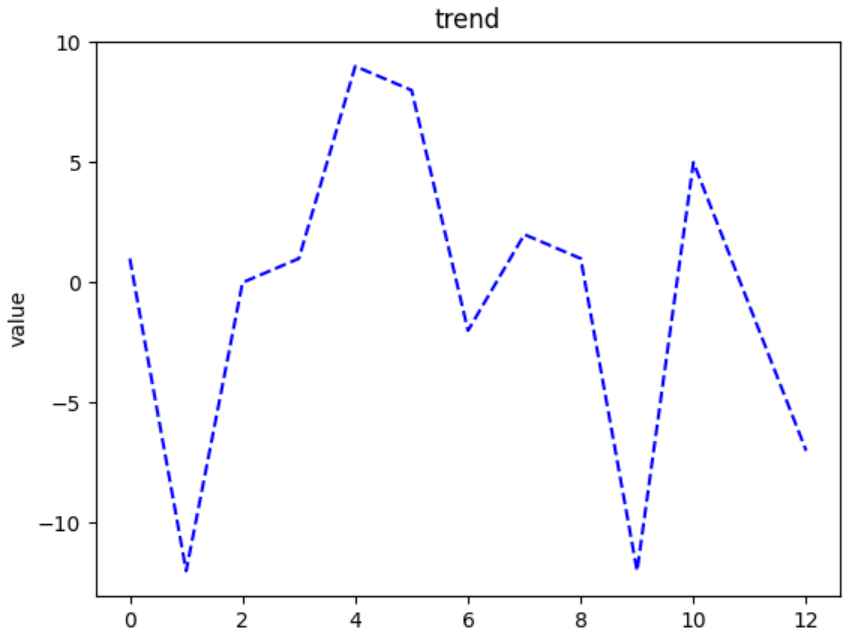


Fig. 4.6 Trend detection

4.3 Control (Schema di funzionamento)

Lo schema di funzionamento dell'applicazione definito dal Control è il seguente:

1. Avvio e visualizzazione interfaccia utente
2. Supporto alla formulazione dell'intenzione (supporto testuale, selezione e audio)
3. Esecuzione comando submit con avvio del processo di parsing, generazione ed esecuzione della query.
4. Raccolta e visualizzazione dati
5. Attivazione algoritmi
6. Aggiornamento visualizzazione dati e grafici.

Le diverse funzionalità sono contrallate dai listener delle componenti dell'interfaccia (View) con cui interagisce l'utente.
La struttura risulta estremamente semplificata in quanto tutta la complessità è demandata all'operazione di parsing che deve elaborare e fornire una risposta il più possibile esauriente all'intenzione formulata dall'utente.

4.4 Modello degli operatori

L'intenzione può essere formulata secondo il modello di Describe o secondo il modello di Explain.

L'intenzione secondo il modello di Describe è la seguente:

With cube **describe** <list measures> [for <list predicate>] by <list group-by> [size (k > 0)].

Le parentesi quadre indicano che questi campi della query non sono obbligatori, in quanto ritenuti non necessari, anche se un buon predicato di selezione consente di abbattere notevolmente i costi per l'estrazione dei dati.

L'intenzione del modello di Explain è la seguente:

With cube **explain** <list measures> [for <list predicate>] by <list group-by>
using <correlation <list measures>| next <year | month | week | day>>

4.5 Grammatiche per il parsing

Di seguito si riportano delle grammatiche per gli operatori intenzionali di Describe ed Explain, tali modelli possono essere utilizzati per la verifica sintattica con grammatica del processo di parsing.

Modelli di grammatica per il parsing relativa a Describe ed Explain.

```
describe : 'with' cube=id 'describe' measure=id (',' measure+=id)*
                ('for' selector=clause)?
                ('by' group=id)?
                ('using' models=MODEL_DES (',' models+=MODEL_DES)*)?
                ('size' k=INT)?
                EOF;

explain : 'with' cube=id 'explain' measure=id (',' measure+=id)*
                ('for' selector=clause)?
                ('using' models=MODEL_EXP (',' models+=MODEL_EXP)*)?
                ('by' group=id)?
                EOF;


id [String name] : ID { $name = $ID.text; };

clause  : condition (binary condition)*;
condition  : attr=ID op=comparator val=value
           | attr=ID in=IN '(' val=value (',' val+=value)* ')';

MODEL_DES: 'clustering' | 'top-k' | 'bottom-k' | 'outliers' ;
MODEL_EXP: 'correlation' | 'top-k'  | 'outliers' | 'next' | 'past';

value  : ID  | INT | DECIMAL | bool;
comparator  : EQ  | GT  | GE  | LT  | LE;
bool: TRUE | FALSE;
binary: AND | OR;

IN      : 'IN' | 'in';
AND   : 'AND' | 'and';
```

```
OR      : 'OR' | 'or';
NOT     : 'NOT' | 'not';
TRUE    : 'TRUE' | 'true';
FALSE: 'FALSE' | 'false';
EQ      : '=';
GT      : '>';
GE      : '>=';
LT      : '<';
LE      : '<=';

INT     : '-'? [0-9]+;
DECIMAL      : '-'? [0-9]+ '.' [0-9]+;
ID      : '\" [a-zA-Z0-9'_'\-'/' ] + '\"   |      [a-zA-Z0-9'_'\-'/']+ ;
```

4.6 Struttura del DB dei metadati

Il database dei metadati contiene i nomi delle tabelle, delle misure, delle dimensioni presenti nel database dei dati. In dettaglio le tabelle che costituiscono il database sono le seguenti:

- Tabella ‘table’, contiene i nomi ed i tipi delle tabelle dei fatti e delle dimensioni presenti nel database principale,
- Tabella ‘groupbyoperator’ tabella degli operatori, contiene i nomi ed i sinonimi dei principali operatori nel database.
- Tabella ‘measure’ contente tutte le misure del database
- Tabella ‘hierarchy’ delle gerarchie per specificare le principali tabelle dimensionali
- Tabella ‘language_predicate ‘ del lessico per i predicati, per specificare i termini di riferimento soprattutto del WHERE e del GROUP_BY.
- Tabella delle dimensioni del cubo, contenente il nome di tutte le dimensioni che identificano o aggiungono informazioni al fatto.

Le chiavi delle tabelle delle dimensioni che contribuiscono tutte assieme a formare la chiave della tabella dei fatti sono riferite alle seguenti tabelle: prodotti (product_id), tempo (time_id), clienti (customer_id), promozioni (promotion_id), magazzino (store_id).

4.7 Modalità di uso del sistema

In avvio la GUI del prototipo si presenta con tutte le tre componenti di base; area per la formulazione dell'intenzione, area visualizzazione risultati e area messaggi. (Fig. 4.7 Interfaccia iniziale).

- Formulazione intenzione

La formulazione dell'intezione viene realizzata attraverso la parte evidenziata in alto e comprende i seguenti componenti (Fig. 4.8 Formulazione Intenzione):

- Model List, selezione modelli per la scelta del modello di query intenzionale.
- Request List, selezione query per la scelta di una richiesta intenzionale da un elenco di query predefinito ed aggiornato dinamicamente con le nuove query realizzate.
- Intentional Request, per la formulazione della query intenzionale corrente con la scrittura e l'editing della stringa testuale.
- Selettori Metadata, una serie di selettori per la specifica delle componenti del modello e/o la modifica delle componenti delle richieste relative.
- Record Stop, per avviare la registrazione e formulare l'intezione in modalità vocale. La registrazione produce la stringa dell'intezione direttamente nell'area preposta (Intentional Request). Terminata la registrazione l'intezione può essere modificata con i selettori e/o modificata con editing.
- Submit. Attiva il processo di elaborazione con la generazione di tutte le strutture di visualizzazione dati e rappresentazioni grafiche dell'interfaccia.
- Algorithms. Successivamente l'elaborazione potrà essere prseguita e dettagliata con l'applicazione degli algorimi proposti attraverso appositi comandi (Fig. 4.9, Fig. 4.10 Risultati operatore Describe e Explain).

- Model List

La gestione dei modelli delle query mantiene un modello generale, la lista dei modelli di base e dei modelli più complessi per ciascuno degli operatori intenzionali. La scelta del modello rappresenta il punto di partenza dell'utente nel formulare la sua intenzione scegliendone uno fra

quelli proposti. La lista viene mantenuta su un file per essere facilmente modificabile ed adattabile dall'utente. I modelli presenti sono i seguenti:

modello generale: with <cube> <operator> <measure> [for <predicate>] by <level_groubBy>)

modello Describe: with <cube> describe <measure>, <measure>..., <measure>
[for <predicate>] by <level_groubBy>

modello Explain: with <cube> explain <measure> [for <predicate>] using
(correlation<measure> | (next | past)<year | month>) by <level>

- Request List

La selezione delle query permette di scegliere una query fra la lista delle query disponibili. La lista è aggiornata dinamicamente dopo ogni nuova query intenzionale formulata con successo.
Analogamente alla scelta del modello può essere scelto un esempio di query da formulare direttamente o da modificare per esprimerne una nuova.

Esempi di query formulate dall'utente:

with foodmart explain store_sales for the_month=january using next(month) by store_name
with foodmart describe sum(store_cost), maximum(unit_sales) for the_year >= 1997 by store_name
with foodmart explain store_sales for promotion_name=wallet savers using correlation(unit_sales) by store_name
with foodmart describe sum(store_cost), maximum(unit_sales) for the_year >= 1997 by brand_name
with foodmart describe sum(store_sales), avg(store_cost) for the_year=1997 by brand_name size 4
with foodmart explain store_sales for the_year=1997, store_state=wa using correlation(unit_sales) by store_name

Una volta selezionato il modello di partenza o una query dalla lista, l'utente può completare l'intenzione mediante una serie di selettori (comboBox) che permettono di sostituire le parti del modello con le relative espressioni.
Sono previsti 5 selettori: cubo, operatore, misura, predicato, clausola group_by e clausola using.

- Intentional Request

L’intenzione dell’utente può anche essere formulata scrivendo direttamente nell’area testo con tutte le opzioni di editing.

- Selettori metadata

In tutti i casi (sia che l’intezione sia iniziata con la scelta del modello, con la selezione di una query o con l’edit diretto o con comando vocale oppure una combinazione delle precedenti).

Possono essere utilizzati i selettori dei metatdata per comporre l’intenzione aggiungendo le singole componenti. I selettori possono anche essere utilizzati per completare o modificare la query scritta dall’utente.

I tipi di selettori previsti sono:

- Selezione Cubi (CubeComboBoxListener) permette di scegliere il cubo dati di riferimento tra quelli disponibili dal sistema.
- Selezione Misure (MeasureComboBoxListener) utilizzata per inserire e/o sostituire le misure nella query intenzionale. La lista dei valori utilizzabili viene determinata dal database dei metadati.
- Selezione Predicati (ComboBoxListener) usata per gestire inserimento e sostituzione dei predicati e dei selettori nella richiesta intenzionale. La lista dei predicati utilizzabili viene presa dal database dei metadati.

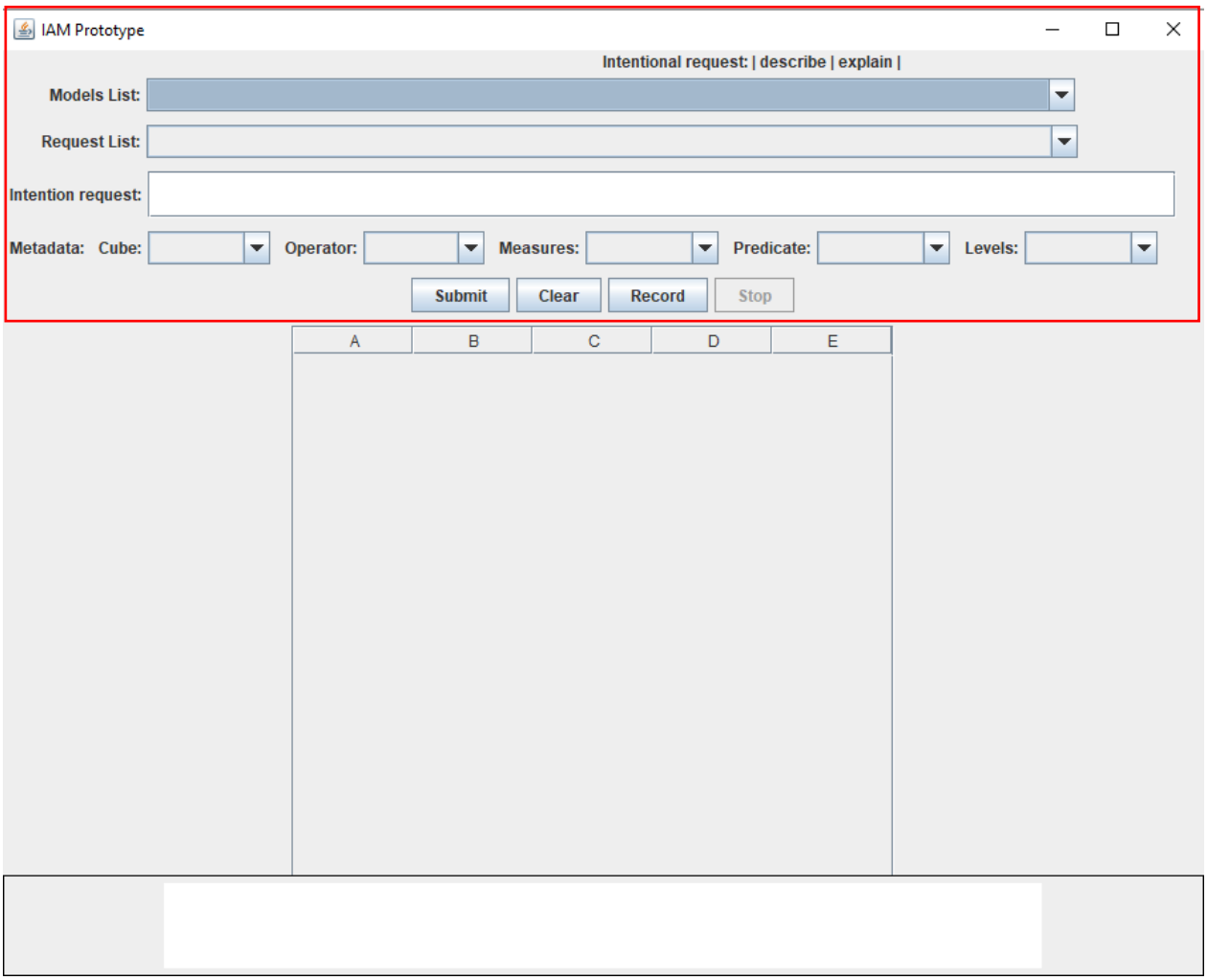


Fig. 4.7 Interfaccia iniziale

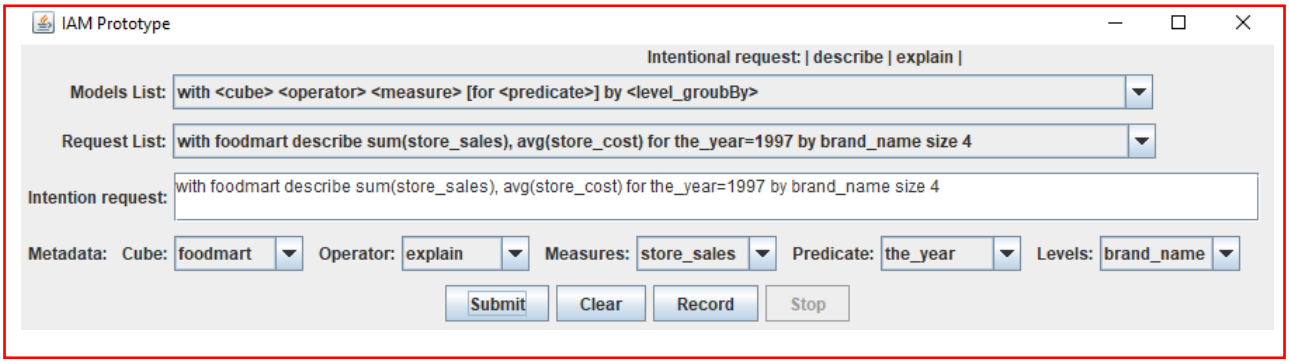


Fig. 4.8 Formulazione Intenzione

4.8 Database di prova

Il database di riferimento, (foodmart), si presenta secondo schema a stella. Di seguito la descrizione delle misure dello schema:

- **Store_sales**, misura aggregata che considera le vendite totali di uno specifico prodotto riferita ad uno specifico periodo in un determinato negozio;
- **Store_cost**, misura aggregata che considera i costi di produzione per uno specifico prodotto riferiti ad un particolare periodo in un determinato negozio;
- **Unit_sales**, prezzo di vendita unitario riferito ad un particolare prodotto

Di seguito la descrizione delle dimensioni dello schema:

- **Tempo** (principali: the_year, the_month, the_day)
- **Clientela** (principali: lname, fname, city)
- **Promozione** (principali: promotion_name, cost)
- **Magazzino** (principali: store_name, store_type, store_country)

4.9 Esempi di query e risultati

Di seguito sono riportati esempi di query per i due operatori intenzionali **Descibe** ed **Explain** implementati nel prototipo con le relative immagini dei risultati prodotti:

- with foodmart **describe** sum(store_sales), avg(store_cost) for the_year = 1997 by brand_name size 4 (Fig. 4.9 Risultati operatore Describe);
- with foodmart **explain** store_sales for promotion_name=wallet, savers using correlation(unit_sales) by store_name (Fig. 4.10 Risultati operatore Explain).

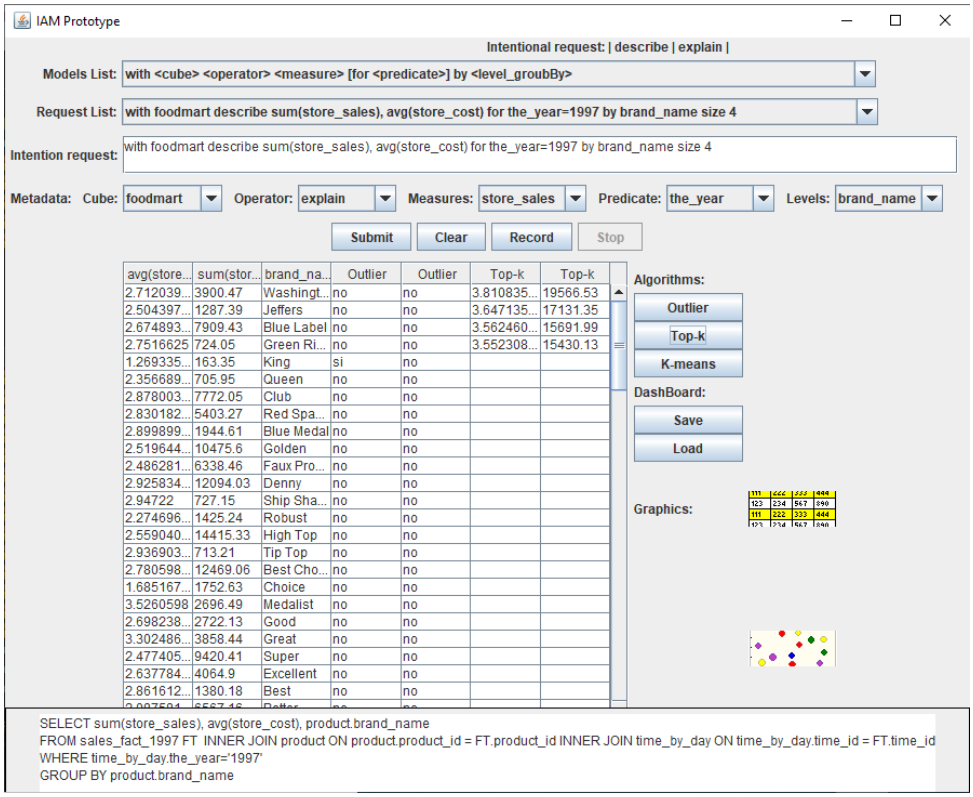


Fig. 4.9 Risultati operatore Describe

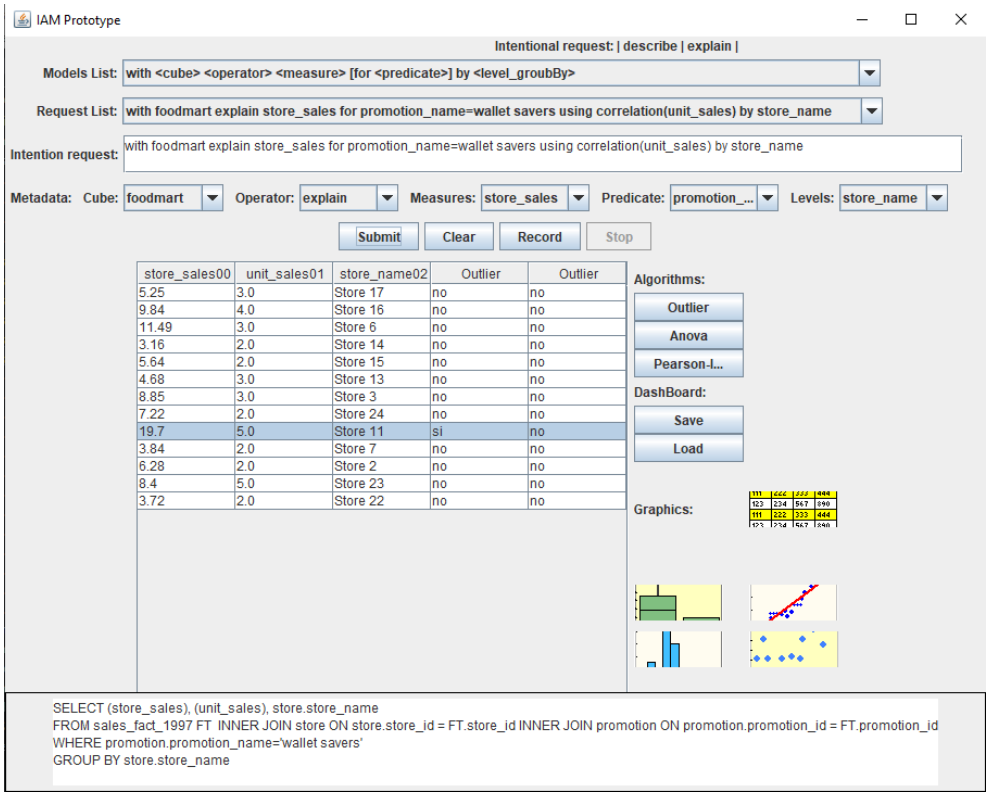


Fig. 4.10 Risultati operatore Explain

4.10 Pattern di progettazione e programmazione

I pattern di progettazione usati nella realizzazione del prototipo sono:

- **Pattern Observer**, usato per gestire i listener dei comandi di submit e visualizzazione degli algoritmi.
- **Pattern Generics** per definire tipi di dato parametrizzato. I tipi parametrizzati rivestono particolare importanza perché consentono di creare classi, interfacce e metodi per i quali il tipo di dato sul quale si opera può essere specificato come parametro.
- **Singleton** per l'interfaccia
- **Pattern concorrente**, pattern di programmazione concorrente, utilizzato per l'attivazione del thread che permette di registrare la voce dell'utente e di scrivere sulla JTextArea response la frase (intenzione) che l'utente ha espresso in modo vocale.
- **Uso di lambda,** uso di funzioni lambda per la gestione ed elaborazioni di liste.

4.11 Dettagli implementativi

I file del progetto (sviluppati in Java ed in Python), sono stati opportunamente rifattorizzati limitando la proliferazione del codice e riducendo le dimensioni dei moduli e delle classi.
Tutto il codice sviluppato è disponibile sul repository git al seguente indirizzo:
https://lorenzo95@bitbucket.org/lorenzo95/tesi.git

- Layout interfaccia

La classe StartApp è la classe di avvio dell'applicazione (IAM Prototipe) con la creazione del JFrame contenente l'interfaccia. L'interfaccia è realizzata, secondo uno schema *BorderLayout*, dove vengono costruiti i pannelli north, south e center.
Nel pannello nord, parte in alto nell'interfaccia, vengono inseriti i comboBox per la formulazione delle intenzioni, i tipi di modelli intenzionali e le query che l'utente potrà scegliere.

Nel pannello centro viene costruita l'area di visualizzazione dei risultati con la tabella per il caricamento dei dati ed i pulsanti per l'elaborazione degli algoritmi e le rappresentazione grafiche ottenute dall'applicazione degli stessi.
Nel pannello sud viene posizionata un'area dei messaggi (text area), per riportare le query formulate, i messaggi di elaborazione ed informazioni varie relative ai diversi valori calcolati (fattori di proporzionalità e correlazione fra i dati, etc).

- Interazione vocale

Il supporto per produrre l'intenzione mediante comando vocale è stato realizzato tramite uno specifico thread utilizzando le api di Google del Package com.darkprograms.speach.recognizer con le classi microfone, GSpeachDuplex, GSpechResponseListener e GoogleResponse.

- Descrizione formati dei dati

Di seguto sono indicati e descritti brevemente i formati dati utilizzati nel sistema.

- Formato query intenzionale

L'utente ha la possibilità di esprimere l'intenzione, in maniera vocale o in maniera testuale. In entrambi i casi l'intenzione è sempre memorizzata dal sistema come stringa alfanumerica.

- Formato parsing (JSON)

Il formato JSON (acronimo di JavaScript Object Notation) è un formato adatto all'interscambio di dati fra applicazioni client/server molto utilizzato nella programmazione web. Tale formato viene utilizzato dal parsing, per la costruzione delle strutture interne per identificare le diverse tipologie di token prodotti. Successivamente viene elaborato dal generatore di query SQL.

- Formato dati di intescambio (csv)

Il comma-separated values (abbreviato in CSV) è un formato di file basato su file di testo utilizzato per l'importazione ed esportazione di tabelle dati (ad esempio da fogli elettronici o database). Non esiste uno standard formale che lo definisce, ma solo alcune prassi più o meno consolidate.

Il formato viene utilizzato per lo scambio dati con i moduli esterni con la seguente rappresentazione:

<nome misura1>, ..., <nome misuraN>
val0 misura1, ..., val0 misuraN
..
valM misura1, ..., valM misuraN

- Formato di rappresentazione per la visualizzazione dei risultati

Per la rappresentazione dello stato di visualizzazione dei dati, risultati e grafici può essere previsto un semplice file testo contenente la query intenzionale con la sequenza degli algoritmi che sono stati applicati. In questo modo una rappresentazione è ricaricata rieseguendo in automatico tutto il processo di parsing, esecuzione query ed applicazione degli algoritmi.

Una forma più completa potrebbe prevedere la memorizzazione dei dati con gli algoritmi applicati, in questo modo i risultati possono essere riutilizzati direttamente senza dover riapplicare l'intero processo di query. In questa seconda ipotesi il sistema potrebbe funzionare anche senza interrogazioni alla sorgente dei dati.

5 CONCLUSIONI E POSSIBILI SVILUPPI

Il lavoro, iniziato da uno stadio estremamente semplificato, ha portato alla realizzazione di un prototipo con un ambiente strutturato in tutte le componenti di: formulazione dell'intenzione, elaborazione del parsing, esecuzione delle query, applicazione degli algoritmi e rappresentazione dei risultati. Il prototipo realizzato comprende la piena funzionalità degli operatori intenzionali di Describe ed Explain.

Durante la fase di progettazione sono state esaminate ed approfondite le tematiche e le proposte dei sistemi IAM analizzati nei loro principali concetti di base: operatori intenzionali, processo di raccolta dati, strutturazione del sistema. Inoltre è stata fatta una breve panoramica sui sistemi OLAP per evidenziare i limiti e le loro potenzialità.

- Possibili sviluppi

Come possibili sviluppi del prototipo, oltre ad implementare gli altri operatori intenzionali possono essere presi in considerazione i seguenti aspetti:

- **Estensione database metadati**. Prevedere un'estensione della funzionalità del database conversational per realizzare un modello semantico per avere una maggiore indipendenza dai dati di base (differenti denominazioni dei dati, definizione di alias, …).
- **Modellazione dinamica**. Permettere all'utente di definire dinamicamente i modelli e gli algoritmi che devono essere applicati alle misure. In presenza di più colonne di dati poter decidere il dataset sul quale applicare i diversi algoritmi. In pratica, una volta applicati determinati algoritmi o modelli, l'utente può decidere se legarli alle misure per poter essere applicate automaticamente nelle successive analisi.
- **Integrazione ambienti**. Una maggiore integrazione dei risultati forniti dagli ambienti esterni (ad esempio integrazione della visualizzazione dei grafici generati con Python).

- **Migliorare ed implementare le funzionalità** di salvataggio e caricamento dell'area di visualizzazione con l'obiettivo di rendere il sistema adatto a conservare le elaborazioni eseguite dall'utente.
- **Maggiore configurabilità ed adattabilità** ai diversi database, mantenendo valide le elaborazioni realizzate e le visualizzazioni dei risultati salvati dalle precedenti analisi.
- **Deploy applicazione con Tomcat**. Permettere la distribuzione dell'applicazione tramite Tomcat per poter proseguire gli sviluppi dell'applicazione come web – app.

6 BIBLIOGRAFIA

[1] P. Vassiliadis, P. Marcel, S. Rizzi. Beyond Roll-Up's and Drill-Down's: An Intentional Analytics Model to Reinvent OLAP Information Systems, 85:68–91, 2019, URL http://arxiv.org/abs/1812.07854.

[2] A. Chanson, N. Labroche, P. Marcel, S. Rizzi. Explaining multidimensional data under time constraint.

[3] A. Chédin, M. Francia, P. Marcel, V. Peralta, S. Rizzi. The Tell-Tale Cube. Published in the Workshop Proceedings of the EDBT/ICDT 2019 Joint Conference (March 26, 2019, Lisbon, Portugal) on CEUR-WS.org.

[4] O'Brien, J. A., & Marakas, G. M. (2009). Management information systems (9th ed.). Boston, MA: McGraw-Hill/Irwin.

Link:

https://olap.com/

https://en.wikipedia.org/wiki/K-means_clustering

https://en.wikipedia.org/wiki/Analysis_of_variance

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

https://docs.microsoft.com/it-it/azure/architecture/data-guide/relational-data/online-analytical-processing

https://www.ibm.com/support/knowledgecenter/SSWTQQ_2.0.3/solnguide/c_si_trendetection.html

https://www.tableau.com/