

POLITECNICO DI TORINO

DIPARTIMENTO DI INGEGNERIA GESTIONALE E DELLA
PRODUZIONE

Corso di Laurea in Ingegneria Gestionale
Classe L8 – Ingegneria dell'Informazione



Relazione dell'Esame Finale

Sviluppo di un software per la gestione dell'apertura di nuovi punti vendita

Relatore

Prof. Fulvio Corno

Candidato

Nicolò Flavio Campaniolo

A.A. 2018/2019

Copia integrale della proposta	3
Studente proponente.....	3
Titolo della proposta	3
Descrizione del problema proposto.....	3
Descrizione della rilevanza gestionale del problema.....	3
Descrizione dei data-set per la valutazione.....	3
Descrizione preliminare degli algoritmi coinvolti	4
Descrizione preliminare delle funzionalità previste per l'applicazione software.....	4
Descrizione dettagliata del problema affrontato	6
Descrizione del data-set utilizzato per l'analisi	9
Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati	11
Diagramma delle classi delle parti principali dell'applicazione	15
Alcune videate dell'applicazione realizzata	16
Risultati sperimentali ottenuti	19
Conclusioni.....	24

Copia integrale della proposta

Studente proponente

S226878 Nicolò Flavio Campaniolo

Titolo della proposta

Gestione dell'apertura di nuovi supermarket

Descrizione del problema proposto

Si intende realizzare un programma capace di definire il luogo ideale per l'apertura di nuovi punti vendita.

Il programma riceve dunque in input la quantità di supermercati che si intende aprire e restituisce un set di comuni, appartenenti alla medesima regione, dove si è valuta essere più idonea l'apertura di nuovi punti vendita.

Descrizione della rilevanza gestionale del problema

L'apertura di nuovi punti vendita è un problema di rilevanza gestionale. La cattiva gestione dell'investimento di un capitale può portare al fallimento dell'azienda ed è dunque necessario ottimizzare, dove possibile, tale gestione.

In tale problema si ipotizza il caso in cui un supermarket voglia lanciare una campagna regionale per l'apertura di nuovi punti vendita.

Tale algoritmo restituisce un set di comuni dove le condizioni si prestano migliori alla nascita di un nuovo supermarket.

Tuttavia, data la vastità del campo preso in considerazione, tale algoritmo si limita ad indicare esclusivamente il nome dei comuni maggiormente interessati, dove eventualmente sarà il caso di eseguire indagini di mercato più dettagliate.

Tale programma permette dunque di conoscere i principali rivali sul mercato in una determinata zona e le loro eventuali strategie di espansione.

Descrizione dei data-set per la valutazione

La risoluzione di tale problema avviene mediante l'utilizzo di due data-set. Nella fattispecie abbiamo un data-set contenente la lista dei supermercati presenti in Italia. Tale tabella

contiene la ragione, l'indirizzo, il CAP, la città, la provincia e la regione del supermercato. Un secondo data-set contiene invece un censimento riferito al 2011 per tutti i comuni italiani. Tengo a dire che tali dati, pur non essendo esattamente reali, sono comunque realistici.

Descrizione preliminare degli algoritmi coinvolti

In tale applicazione saranno presenti algoritmi di ricerca e ricorsivi. Gli algoritmi di ricerca semplici si occuperanno di interrogare il database per la restituzione di una lista di supermarket presenti nella stessa regione o aventi lo stesso nome.

Un algoritmo più complesso si occuperà della ricerca del set di comuni appartenenti alla stessa regione capace di ottimizzare l'apertura di nuovi punti vendita.

Tale algoritmo si occuperà di calcolare un 'punteggio' assegnato ad ogni comune che determini quanto tale comune si presti favorevole ad ospitare un nuovo punto vendita. Tale punteggio tiene conto del numero di supermarket per abitanti presenti all'interno del comune. Si tende dunque a prediligere i comuni nei quali la saturazione sia inferiore. Vengono inoltre scartati i comuni in cui il database non riconosce la presenza di supermarket, ad esempio perché aventi un numero troppo esiguo di abitanti. Tale algoritmo si preoccupa di verificare la presenza del supermercato che intende espandersi nei vari comuni e tenerne conto nella valutazione.

Per la realizzazione di un tale programma è necessario un algoritmo ricorsivo. Tale algoritmo calcola all'interno di ciascuna regione il set di comuni che ottimizzino il punteggio. Successivamente le confronta fra loro scegliendo la regione ottimale.

Per arrivare a tale soluzione l'applicazione riceve in input il nome della catena di supermarket che intende espandersi ed il numero di punti vendita che si intende aprire. In aggiunta l'utente può anche aggiungere una regione dove si intende aprire i punti vendita e tale algoritmo restituirà esclusivamente il set di comuni che massimizzi quella determinata regione

Descrizione preliminare delle funzionalità previste per l'applicazione software

L'utente si troverà davanti un'interfaccia dove avrà la possibilità di scegliere da una lista di supermarket quello interessato. Cliccando il bottone 'Cerca', riceverà in un'apposita area di testo il set di supermarket della catena selezionata, nei rispettivi comuni di ubicazione. In maniera analoga, avendo selezionato esclusivamente la regione, l'algoritmo restituisce nella medesima area, tutti i punti vendita presenti nella regione selezionata. È possibile anche scegliere contemporaneamente una regione ed una linea di supermarket. In tal caso verranno restituiti tutti i punti vendita di tale catena nella regione selezionata.

Dopo aver selezionato una catena di supermarket sarà disponibile il bottone ‘Espandi’. Nel caso non si sia selezionata alcuna regione, tale pulsante inizia la ricerca del set di comuni, appartenenti alla medesima regione, che ottimizzi l'apertura dei nuovi punti vendita. Tale risultato è assoluto. Nel caso si voglia un risultato relativo ad una regione in particolare sarà necessario selezionarla.

Il bottone ‘Reset’ ripulisce l'area di testo.

Descrizione dettagliata del problema affrontato

Il software tenta di trovare una soluzione al problema, non indifferente, dell'apertura di nuovi punti vendita. Tale problema ha una valenza aziendale, nei termini in cui l'apertura di un nuovo punto vendita rappresenta un investimento, e dunque un costo. È dunque necessario valutare in maniera analitica l'ubicazione dei nuovi punti vendita, nella fattispecie supermercati, o in generale punti di vendita alimentari al dettaglio. Chiaramente, data la vastità del campo scelto, da un punto di vista geografico, tale programma non pretende di trovare una soluzione esatta, assoluta e puntuale al problema. Tuttavia, potrebbe rappresentare un potente ed utile strumento per restringere fortemente il campo di ricerca ad una regione, o ad un insieme di comuni, dove eventualmente adottare una strategia di ricerca delle opportunità più dettagliata.

Tale strumento può essere utile anche per individuare le strategie espansive dei principali avversari. In ambito aziendale è stata spesso sottovalutata la reazione strategica avversaria, soprattutto in passato. Inoltre, mediante opportuni algoritmi di ricerca, è possibile conoscere la concorrenza, in una determinata regione o in un determinato comune, dando dunque l'opportunità di capire perché l'algoritmo restituisca tali comuni.

Concentrandoci sull'applicazione, vediamo come si presenta:

The screenshot shows a web application titled "Tesi_Supermarket". On the left, there is a map of Italy with its regions color-coded. Below the map, there are two radio buttons: "Ordina per comune" (selected) and "Ordina per nome catena". Below these are two buttons: "Cerca" and "Espandi". On the right side, there are several input fields and controls: a dropdown menu for "Scegli la regione:", a dropdown menu for "Scegli la catena:", a section "In alternativa :" with a radio button for "Ricerca avanzata" and a text input "Inserisci la catena cercata..", a text input "Inserisci il numero di punti vendita da aprire:" with a placeholder "Inserisci il numero di punti vendita..", a text input "Ricerca specifico comune:" with a placeholder "Inserisci un comune..", and a section "Maggiorazione del peso delle catene presenti" with radio buttons for "0" (selected), "25%", "50%", and "100%", and a checkbox "escludi presenti". At the bottom right, there is a "Reset" button.

Tralasciando momentaneamente i Radio Button, notiamo che è possibile interagire con l'applicazione essenzialmente mediante due pulsanti. Il pulsante 'Cerca' ci permette di visualizzare la lista di supermercati appartenenti alla medesima catena, scelta mediante menù a tendina, o appartenenti alla medesima regione, scelta anch'essa mediante menù a tendina.

Premendo il Radio Button 'Ricerca avanzata', sarà possibile effettuare una ricerca di supermercati la cui stringa passata nella sottostante Text Field sia contenuta all'interno della regione.

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa :

☒ Ricerca avanzata

Inserisci la catena cercata..

Inserisci il numero di punti vendita da aprire:

Inserisci il numero di punti vendita..

Ricerca specifico comune:

Inserisci un comune..

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%

☐ escludi presenti

Ordina per comune

Ordina per nome catena

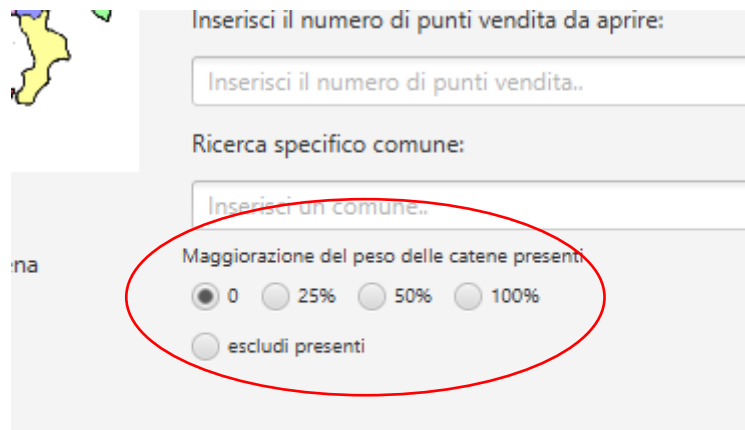
Cerca Espandi

Reset

Tale campo disattiva la funzionalità 'Espandi', disponibile esclusivamente per le catene di supermercati all'interno del menù a tendina, che corrispondono alle 30 più frequenti all'interno del database.

Sospendendone la descrizione dell'algoritmo, che verrà discussa in seguito, descriviamo la funzionalità principale. Mediante il pulsante 'Espandi', dopo aver selezionato la catena e il numero di punti vendita da aprire, viene visualizzato la lista di comuni che ottimizza l'apertura di tali punti. Eventualmente è possibile selezionare una regione, restringendo così il campo di ricerca.

Poniamo ora l'attenzione sui Radio Button evidenziati nella seguente figura:



Inserisci il numero di punti vendita da aprire:

Inserisci il numero di punti vendita..

Ricerca specifico comune:

Inserisci un comune..

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%


☐ escludi presenti

Tali comandi intendono dare la possibilità di scegliere la maggiorazione del peso di un supermercato, nel calcolo del punteggio, quando questo sia appartenente alla medesima catena che si intende espandere. Le scelte sono: 0, 25%, 50%, 100% ed escludi dove presente. Nel caso venga scelto 0 tutti i supermercati verranno visti in maniera uguale dall'algoritmo, che restituirà una lista di comuni nella quale risulta essere più conveniente l'apertura di un tot numero di supermercati, di qualunque catena. Nel caso in cui si scelga una percentuale il discorso cambia e, ad esempio impostando il 100%, un supermercato, appartenente alla stessa catena da espandere, verrà considerato avente lo stesso peso di due supermercati.

Tale scelta permette la realizzazione di un problema più reale, rappresentando la volontà di espansione geografica (in mercati nuovi).

Descrizione del data-set utilizzato per l'analisi

Il software, per il suo funzionamento, si serve di due data-set, contenuti all'interno del database "gestione_supermarket". Un primo, molto semplice, contiene 284 comuni, rappresentanti tutti i capoluoghi di provincia e i comuni aventi un numero di abitanti superiore a 30.000. Per ogni comune vi è il relativo numero di abitanti. La tabella che descrive tale situazione prende il nome di "abitanti_cpr".

#	Nome	Tipo di dati
 1	comune	VARCHAR
2	provincia	VARCHAR
3	regione	VARCHAR
4	abitanti	INT

Il nome del comune ('comune') rappresenta la chiave primaria che identifica il comune stesso. La tabella "elenco_supermarket" si presenta come segue. Un supermarket viene identificato attraverso la ragione, l'indirizzo e la città. Ciò avviene creando opportunamente dei metodi hashCode e equals in Java.

Per evitare di essere ridondante, in Java la classe supermarket è rappresentata dagli attributi:

- String ragione;
- String indirizzo;
- String cap;
- Comune città;

Viene dunque associato un oggetto comune a partire dalla città

#	Nome	Tipo di dati
1	ragione	TEXT
2	indirizzo	TEXT
3	cap	VARCHAR
4	città	VARCHAR
5	provincia	VARCHAR
6	regione	VARCHAR

La creazione del database è stata possibile grazie a:

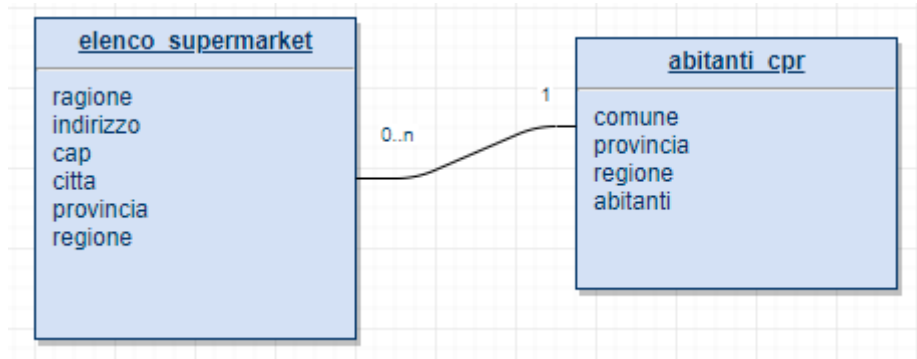
www.federweb.com

dove mi è stato possibile accedere ai supermercati presenti per ogni comune, scaricando l'apposito CSV.

Per quando riguarda la popolazione dei comuni, mi sono basato sui dati Istat del censimento del 2011.

<https://www.istat.it/>

Viene infine riportato il modello relazionale, dal quale si evince che in un comune possono esserci più supermercati, mentre un supermercato può essere presente in un solo comune.



Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati

L'applicazione presenta principalmente due tipologie di algoritmo.

Abbiamo in primis gli algoritmi di ricerca semplice. Tali algoritmi si occupano della ricerca di supermercati appartenenti a particolari tipologie all'interno del database. Nella fattispecie, attraverso il software è possibile ricercare:

- La lista di supermercati, contenuti nel database, appartenenti alla medesima regione, scelta mediante menù a tendina;
- La lista di supermercati, contenuti nel database, appartenenti alla medesima catena, scelta mediante menù a tendina;
- La lista di supermercati, contenuti nel database, appartenenti alla medesima catena e situati nella medesima regione, entrambe scelte dal menù a tendina;
- La lista di supermercati il cui nome della catena (o ragione) contenga la stringa di caratteri passata nell'apposita area di testo;
- La lista di supermercati il cui nome della catena (o ragione) contenga la stringa di caratteri passata nell'apposita area di testo, situati nella regione scelta mediante menù a tendina;
- La lista dei supermercati situati nel comune, il cui nome viene passato come parametro nell'apposito campo di testo;

Tale ricerca viene effettuata attraverso delle query rivolte al database dalle apposite classi DAO, secondo l'omonimo pattern. Il software non importa dunque tutta la lista dei supermercati al proprio interno, ma ne richiede una lista, a seconda delle esigenze richieste, al database, ogni volta che sia necessario. In tal senso il programma risulta essere, a mio avviso, più snello, dal momento che la ricerca risulta essere più rapida in SQL nel caso di un numero di punti vendita crescente. Vengono tuttavia importati tutti i comuni (nel database sono presenti 284 comuni) per facilitare l'associazione di un supermercato ad uno specifico comune.

Differente è la situazione nel caso degli algoritmi ricorsivi. Il programma si propone di ricercare il set di comuni nel quale l'apertura di nuovi punti vendita, di una catena specifica, risulti più conveniente. Viene inoltre richiesto che tali comuni appartengano alla medesima regione. Nel caso più generale l'algoritmo risulta essere il seguente:

```

public List<ComuneCandidato> espandi(int L, String regione, double perc){
    List<ComuneCandidato> result = new LinkedList<ComuneCandidato>();
    double punteggioMax = 0;
    for(String regione : this.getRegioni()) {

        List<ComuneCandidato> parziale = new LinkedList<ComuneCandidato>();
        Map<String, ComuneCandidato> mappaCandidati = daoC.getComuniCandidati(regione, regione, mappaComuni, perc);
        cerca(parziale, 0, mappaCandidati, L);
        double punteggio = calcolaPunteggio(parziale);
        if(punteggio>punteggioMax) {
            punteggioMax = punteggio;
            result = parziale;
        }
    }
    return result;
}

```

Valutiamo le varie parti dell'algoritmo. In un primo momento abbiamo la creazione della struttura dati che andrà a contenere la lista di comuni che ottimizza l'apertura di nuovi punti vendita. Per ogni regione, viene creato una soluzione parziale, inizialmente nulla, dove si andranno ad aggiungere, in maniera ricorsiva, i comuni candidati che massimizzano il punteggio. Tale passo verrà a breve spiegato in maniera adeguata. Infine, vi è un *if* che serve alla memorizzazione della regione e della lista di comuni associata ad essa, capace di massimizzare il punteggio.

Per comprendere meglio lo svolgimento dell'algoritmo introduciamo brevemente la classe *ComuneCandidato*. Tale classe si differenzia dalla classe *Comune*, dal momento che ad essa viene associato un punteggio dato dal rapporto tra il numero di abitanti ed il numero di supermercati presenti in esso, dove vi è una possibile maggiorazione del peso (o esclusione) dei supermercati della stessa catena che si vuole espandere. Tale maggiorazione (o esclusione) è scelta dall'utente mediante dei *RadioButton* visti in precedenza.

Descrizione classe *ComuneCandidato*:

```

public class ComuneCandidato implements Comparable<ComuneCandidato>{
    private Comune comune;
    private double punteggio;
    private int tuttiSuper;
    private int catenaSuper;
    private double perc;
    private double numSuper;

    public ComuneCandidato(Comune comune, int tuttiSuper, int catenaSuper, double perc) {
        this.comune = comune;
        this.tuttiSuper = tuttiSuper;
        this.catenaSuper = catenaSuper;
        this.perc=perc;
        this.numSuper = tuttiSuper + (perc)*catenaSuper;

        this.punteggio=comune.getAbitanti()/this.numSuper;
    }
}

```

Vi è inoltre un metodo per l'aggiornamento del punteggio:

```
public void aggiornaPunteggio() {  
    this.tuttiSuper++;  
    this.catenaSuper++;  
  
    this.numSuper = tuttiSuper + (perc)*catenaSuper;  
    this.punteggio = this.comune.getAbitanti()/this.numSuper;  
}
```

Chiaramente è facile evincere, che il livello di saturazione (supermercati per abitante) risulta essere inversamente proporzionale al punteggio.

Con questa conoscenza sarà più facile comprendere il funzionamento dell'algoritmo ricorsivo:

```
private void cerca(List<ComuneCandidato> parziale, int l, Map<String,ComuneCandidato> mappaCandidati, int lMax) {  
  
    if(l==lMax)  
        return;  
  
    //Ad ogni passo aggiungo il migliore possibile  
  
    ComuneCandidato elemento = cercaMassimo(mappaCandidati);  
    parziale.add(elemento);  
  
    cerca(parziale,l+1,mappaCandidati,lMax);  
}
```

Tale algoritmo esegue essenzialmente tre passi:

- Valuta se la soluzione parziale è completa;
- Chiama una funzione che restituisce il comune candidato, e dunque appartenente alla mappa mappaCandidati, che massimizza il punteggio e lo aggiunge al parziale. NB: i punteggi dei comuni candidati vengono aggiornati man mano che questi vengono aggiunti alla soluzione parziale;
- Infine, abbiamo la funzione ricorsiva alla quale viene passato come parametro il livello successivo;

Abbiamo dunque una soluzione ottima, data dai comuni che puntualmente massimizzano il punteggio.

Per capire completamente il funzionamento di tale software, credo sia il caso di prestare attenzione al seguente metodo presente all'interno della classe ComuniDAO. Tale metodo si occupa di restituire la lista di ComuniCandidati, che verranno memorizzati all'interno di una mappa.

```

public Map<String, ComuneCandidato> getComuniCandidati(String regione, String ragione,
    Map<String, Comune> mappaComuni, double perc) {

    Map<String, ComuneCandidato> result = new HashMap<>();

    String sql ="SELECT citta, COUNT(*) tutteRagioni, sum(case when ragione = ? then 1 else 0 end) "
        + "ragioneCercata FROM elenco_supermarket WHERE regione = ? GROUP BY citta ";

    try {
        Connection conn = ConnectDB.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);

        st.setString(1, ragione);
        st.setString(2, regione);

        ResultSet res = st.executeQuery();

        while (res.next()) {

            ComuneCandidato co = new ComuneCandidato(mappaComuni.get
                (res.getString("citta").trim()),res.getInt("tutteRagioni"),res.getInt("ragioneCercata"),perc);

            result.put((res.getString("citta").trim()), co);

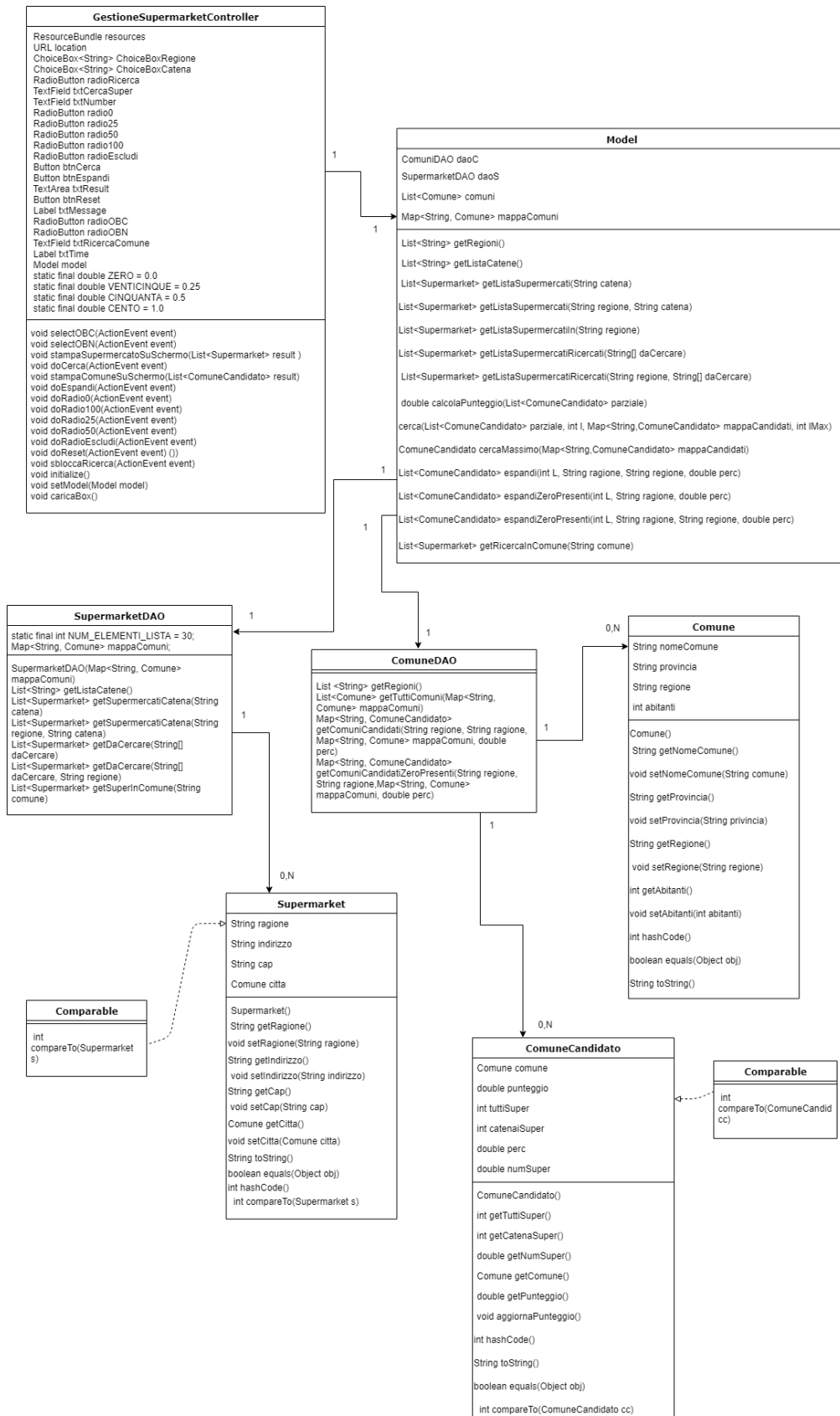
        }
        conn.close();

        return result;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

```

Tale metodo, comunicando con la tabella ‘elenco_supermarket’ attraverso una query, popola la mappa di ComuneCandidato avente come parametri il Comune, il numero di supermarket (tutti), il numero di supermarket della catena specificata come parametro del metodo e la percentuale di maggiorazione del peso anch’essa passata come parametro.

Diagramma delle classi delle parti principali dell'applicazione



Alcune videate dell'applicazione realizzata

Tesi_Supermarket

Scegli la regione: SICILIA

Scegli la catena:

In alternativa :

☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%

☐ escludi presenti

☐ Ordina per comune

☒ Ordina per nome catena

Iper Spaccio Strada Provinciale 54, 95045 MISTERBIANCO (CT) SICILIA

Iper Super Conveniente Via Due Fontane, 93100 CALTANISSETTA (CL) SICILIA

Iper le Dune Via Giorgio La Pira, 97100 RAGUSA (RG) SICILIA

Iperconveniente Via Caduti di Nassiriva, 91022 CASTELVETRANO (TP) SICILIA

Run time: 1677 mills

In questo paragrafo vediamo qualche videata dell'applicazione. Nella figura possiamo vedere come sia presentata la ricerca dei supermarket presenti nella regione Sicilia. Anche se non risulta bene dalla foto, essi sono ordinati per nome di 'ragione' crescente. Possiamo inoltre notare il messaggio, in basso a sinistra, contenente il 'Run time' espresso in millisecondi.

In maniera analoga possiamo vedere la ricerca di una specifica catena all'interno di una regione o in tutto lo Stato.

Tesi_Supermarket

Scegli la regione: **PIEMONTE**

Scegli la catena : **Naturasi'**

In alternativa :

☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☒ Ordina per comune
☐ Ordina per nome catena

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

Naturasi' Piazza Lorenzo Perosi 6, 15121 ALESSANDRIA (AL) PIEMONTE
 Naturasi' Via Tripoli 6, 13900 BIELLA (BI) PIEMONTE
 Naturasi' Via Guglielmo Marconi 12, 28100 NOVARA (NO) PIEMONTE
 Naturasi' Via Citta di San Francisco 1, 10064 PINEROLO (TO) PIEMONTE

Run time: 42 mills

Tesi_Supermarket

Scegli la regione:

Scegli la catena : **Naturasi'**

In alternativa :

☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☒ Ordina per comune
☐ Ordina per nome catena

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

Naturasi' Via Giobbe... PALERMO (PA) SICILIA
 Naturasi' Largo Cavalieri del Lavoro 66, 90146 PALERMO (PA) SICILIA
 Naturasi' Via la Spezia 72/A, 43125 PARMA (PR) EMILIA ROMAGNA
 Naturasi' Via Emilia Est 218, 43123 PARMA (PR) EMILIA ROMAGNA
 Naturasi' Viale dei Partigiani 84, 27100 PAVIA (PV) LOMBARDIA

Run time: 106 mills

Premendo il Radio Button 'Ricerca avanzata', notiamo che viene disabilitata l'area del software relativa all'espansione. Ciò avviene dal momento che l'algoritmo di espansione opera esclusivamente con le catene presenti nel menù a tendina, che corrispondono alle 30 catene più frequenti in Italia.

Tesi_Supermarket

Scegli la regione:

Scegli la catena : **Naturasi'**

In alternativa :

☒ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☒ Ordina per comune
☐ Ordina per nome catena

Maggiorazione del peso delle catene presenti

☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

Il Gigante Viale Lazio 4, 20089 ROZZANO (MI) LOMBARDIA
 Il Gigante Viale Ercole Marelli 19, 20099 SESTO SAN GIOVANNI (MI) LOMBARDIA
 Il Gigante Via Monti Vincenzo 49, 20099 SESTO SAN GIOVANNI (MI) LOMBARDIA
 Il Gigante Viale Italia 555, 20099 SESTO SAN GIOVANNI (MI) LOMBARDIA
 Il Gigante Corso Mortara 4, 10149 TORINO (TO) PIEMONTE

Run time: 45 mills

La funzionalità ‘Espandi’ si presenta essenzialmente in due varianti. In una prima variante, non viene scelta la regione da parte dell’utente, e sarà dunque l’algoritmo a scegliere la regione che massimizzi il punteggio. In una seconda variante, l’utente sceglierà la regione, forzando così l’algoritmo a restituire la lista di comuni che massimizzino l’apertura di nuovi punti vendita all’interno di quella regione.

The image shows two screenshots of the 'Tesi_Supermarket' application interface. Both screenshots show the same form with the following fields and options:

- Scegli la regione:** A dropdown menu.
- Scegli la catena:** A dropdown menu set to 'Naturasì'.
- In alternativa:** Radio buttons for 'Ricerca avanzata' and 'igant'.
- Inserisci il numero di punti vendita da aprire:** A text input field set to '10'.
- Ricerca specifico comune:** A text input field.
- Ordina per:** Radio buttons for 'Ordina per comune' and 'Ordina per nome catena'.
- Maggiorazione del peso delle catene presenti:** Radio buttons for '0', '25%', '50%', and '100%'. The '0' option is selected.
- escludi presenti:** A radio button.
- Cerca** and **Espandi** buttons.
- La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione: ...):** A scrollable list of results.
- Reset** button.
- Run time:** A label indicating the execution time.

In the left screenshot, the region is 'SICILIA' and the results list shows 'ADRANO (CT) SICILIA' repeated four times. The run time is 576 mills.

In the right screenshot, the region is 'LOMBARDIA' and the results list shows 'COLOGNO MONZESE (MI) LOMBARDIA', 'DESIO (MB) LOMBARDIA', 'DESIO (MB) LOMBARDIA', and 'BUSTO ARSIZIO (MB) LOMBARDIA'. The run time is 37 mills.

Infine, può risultare utile all’utente cercare di capire il perché dei dati restituiti dall’algoritmo. A tal proposito è possibile effettuare una specifica ricerca nel comune di interesse.

The image shows a screenshot of the 'Tesi_Supermarket' application interface. The form is filled with the following values:

- Scegli la regione:** A dropdown menu.
- Scegli la catena:** A dropdown menu set to 'Naturasì'.
- In alternativa:** Radio buttons for 'Ricerca avanzata' and 'igant'.
- Inserisci il numero di punti vendita da aprire:** A text input field set to '10'.
- Ricerca specifico comune:** A text input field set to 'ADRANO'.
- Ordina per:** Radio buttons for 'Ordina per comune' and 'Ordina per nome catena'. The 'Ordina per nome catena' option is selected.
- Maggiorazione del peso delle catene presenti:** Radio buttons for '0', '25%', '50%', and '100%'. The '0' option is selected.
- escludi presenti:** A radio button.
- Cerca** and **Espandi** buttons.
- La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione: ...):** A scrollable list of results.
- Reset** button.
- Run time:** A label indicating the execution time.

The results list shows the following information:

- Supermercati presenti a(d): ADRANO (abitanti : 35519)
- Alfo Motta S.r.l. Via Asiago 128, 95031 ADRANO (CT) SICILIA
- Crai Via Regina Margherita 188, 95031 ADRANO (CT) SICILIA
- Despar Via Statuto Lavoratori 95, 95031 ADRANO (CT) SICILIA
- Sangiorgio S.r.l. Via Paolo Vi, 95031 ADRANO (CT) SICILIA
- Tileri Commercio S.r.l. Via Commercio 110, 95031 ADRANO (CT) SICILIA

The run time is 41 mills.

Risultati sperimentali ottenuti

Consideriamo la seguente ricerca:

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa :
☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

Ordina per comune
☒ Ordina per nome catena

Cerca Espandi

Maggiorazione del peso delle catene presenti
☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione : SICILIA):

ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA

Reset

Run time: 516 mills

Ipotizziamo dunque di voler aprire 10 nuovi punti vendita Lidl. Il risultato ottenuto, vede la Sicilia come la migliore regione dove aprire questi nuovi 10 Lidl. La lista di comuni suggerisce di aprire:

- 6 punti vendita ad Adrano in provincia di Catania
- 4 punti vendita a Monreale in provincia di Palermo

Chiaramente risulta una decisione molto azzardata aprire 6 punti vendita in un'unica città, ma pone nell'utente la curiosità di sapere il perché l'algoritmo abbia scelto tali comuni.

Si può dunque pensare di fare un'analisi più dettagliata per vedere quali siano i supermarket presenti in tali comuni. Si ottiene dunque:

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa :
☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☐ Ordina per comune
☒ Ordina per nome catena

Maggiorazione del peso delle catene presenti
☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

-Supermercati presenti a(d): ADRANO (abitanti : 35519)
 Alfio Motta S.r.l. Via Asiago 128, 95031 ADRANO (CT) SICILIA
 Crai Via Regina Margherita 188, 95031 ADRANO (CT) SICILIA
 Despar Via Statuto Lavoratori 95, 95031 ADRANO (CT) SICILIA
 Sangiorgio S.r.l. Via Paolo Vi, 95031 ADRANO (CT) SICILIA
 Tilenni Commerciale S.p.A. Via Cappuccini 119, 95031 ADRANO (CT) SICILIA

Run time: 42 mills

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa :
☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☐ Ordina per comune
☒ Ordina per nome catena

Maggiorazione del peso delle catene presenti
☒ 0 ☐ 25% ☐ 50% ☐ 100%
☐ escludi presenti

-Supermercati presenti a(d): MONREALE (abitanti : 38068)
 Carrefour Via Circonvallazione, 77, 90046 MONREALE (PA) SICILIA
 Conad Via Ponte Parco 14, 90046 MONREALE (PA) SICILIA
 Di Matteo Srl Via Circonvallazione 77/3, 90046 MONREALE (PA) SICILIA
 Galliano Shopping S.r.l. Bivio di Cristina, Pioppo 90046 MONREALE (PA) SICILIA
 L'Orizzonte S.r.l. Via Ponte Parco 14, 90046 MONREALE (PA) SICILIA
 Porpora Group S.r.l. Via Venero 214, 90046 MONREALE (PA) SICILIA

Run time: 37 mills

Riportando i risultati in delle tabelle:

Adrano	35.519 abitanti
Alfio Motta S.r.l.	Via Asiago 128
Crai	Via Regina Margherita 188
Despar	Via Statuto Lavoratori 95
Sangiorgio S.r.l.	Via Paolo Vi
Tilenni Commerciale S.r.l.	Via Cappuccini 119

Monreale	38.068 abitanti
Carrefour	Via Circonvallazione, 77
Conad	Via Ponte Parco 14,
Di Matteo Srl	Via Circonvallazione 77/3
Galliano Shopping S.r.l.	Bivio di Cristina, Pioppo
L'Orizzonte S.r.l.	Via Ponte Parco 14
Porpora Group S.r.l.	Via Venero 214

Dai risultati si evince che la presenza di supermercati è molto bassa. Sembrano inoltre esserci dei negozi alimentari a gestione familiare, a giudicare dai nomi delle ragioni. Questo spiegherebbe il motivo dell'ideoneità all'apertura di nuovi punti vendita. Tuttavia, è necessario effettuare ulteriori ricerche di marketing. Si potrebbe vedere dove le persone abitualmente comprano prodotti alimentari e valutare la presenza di altri supermarket, eventualmente grossi, nell'immediata periferia, magari un altro comune non presente nel database.

Sebbene tale software non dia una verità assoluta, il suo scopo è stato raggiunto, dal momento che l'ambito di ricerca è stato notevolmente ristretto.

Vediamo ora un ultimo caso particolare. Ipotizziamo di voler aprire una catena Esselunga. Come molti sanno, tale catena non è presente nel meridione, quindi è prevedibile che l'algoritmo dia come risultato una regione meridionale.

Questa volta ci serviremo dei Radio Button per la maggiorazione del peso delle catene già presenti nella regione considerata.

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa:
☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☐ Ordina per comune
☒ Ordina per nome catena

Maggiorazione del peso delle catene presenti:
☒ 0 ☐ 25% ☐ 50% ☐ 100%

La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione: SICILIA):

ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA
ADRANO (CT)	SICILIA

Run time: 550 mills

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa:
☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

☐ Ordina per comune
☒ Ordina per nome catena

Maggiorazione del peso delle catene presenti:
☐ 0 ☐ 25% ☒ 50% ☐ 100%

La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione: PUGLIA):

BITONTO (BA)	PUGLIA
BITONTO (BA)	PUGLIA
BITONTO (BA)	PUGLIA
CORATO (BA)	PUGLIA

Run time: 533 mills

Come prevedibile, nel caso in cui la maggiorazione del peso sia nulla, ogni supermercato ha lo stesso peso ed è quindi normale che l'Esselunga sia trattato come un Lidl. Abbiamo dunque la stessa situazione vista in precedenza:

- 6 punti vendita ad Adrano in provincia di Catania
- 4 punti vendita a Monreale in provincia di Palermo

La situazione tuttavia cambia nel caso in cui aggiungessimo una maggiorazione del peso per le catene già presenti. In quest'ultimo caso abbiamo che la regione scelta è la Puglia con:

- 3 punti vendita a Bitonto in provincia di Bari
- 1 punto vendita a Corato in provincia di Bari
- 2 punti vendita a Grottaglie in provincia di Taranto
- 2 punti vendita a Martina Franca in provincia di Taranto
- 2 punti vendita a Trani in provincia di Barletta-Andria-Trani

La domanda che sorge spontanea è la seguente:

Perché, se in entrambe le regioni la presenza di Esselunga è nulla, la maggiorazione del peso modifica i risultati?

La risposta in realtà è molto semplice. Dopo che viene aggiunto un (possibile) punto vendita al comune candidato, che massimizza il punteggio, l'algoritmo tiene conto di tale aggiunta (diminuendo il punteggio). Pertanto, dal momento che gli unici due comuni che presentano

un esiguo numero di supermercati in Sicilia sono solo Adrano (CT) e Monreale (PA), l'algoritmo preferisce la Puglia dove è possibile distribuire i punti vendita in 5 Comuni, ottenendo un punteggio più alto.

Per risultare più chiaro, mostro cosa succede ai singoli punteggi (nella console).

ADRANO (CT)	SICILIA Punteggio: 7103.8	GROTTAGLIE (TA)	PUGLIA Punteggio: 4642.142857142857
ADRANO (CT)	SICILIA Punteggio: 5464.461538461538	BITONTO (BA)	PUGLIA Punteggio: 4019.785714285714
MONREALE (PA)	SICILIA Punteggio: 5438.285714285715	GROTTAGLIE (TA)	PUGLIA Punteggio: 3822.9411764705883
MONREALE (PA)	SICILIA Punteggio: 4478.588235294118	MARTINA FRANCA (TA)	PUGLIA Punteggio: 3768.0
ADRANO (CT)	SICILIA Punteggio: 4439.875	TRANI (BT)	PUGLIA Punteggio: 3721.7333333333333
MONREALE (PA)	SICILIA Punteggio: 3806.8	BITONTO (BA)	PUGLIA Punteggio: 3630.7741935483873
ADRANO (CT)	SICILIA Punteggio: 3738.842105263158	CORATO (BA)	PUGLIA Punteggio: 3431.785714285714
MONREALE (PA)	SICILIA Punteggio: 3310.2608695652175	TRANI (BT)	PUGLIA Punteggio: 3383.3939393939395
ADRANO (CT)	SICILIA Punteggio: 3229.0	MARTINA FRANCA (TA)	PUGLIA Punteggio: 3378.206896551724
GELA (CL)	SICILIA Punteggio: 3028.28	BITONTO (BA)	PUGLIA Punteggio: 3310.4117647058824

Si possono fare le seguenti considerazioni:

- Nonostante, inizialmente, il comune avente punteggio più alto sia Adrano (più del 50% più alto rispetto a Grottaglie), l'insistere ripetute volte su tale comune ne fa crollare il punteggio;
- Una visione del problema in tali termini è sicuramente più realistica, dove i supermercati vengono spalmati nei vari comuni;

Tuttavia, nel caso si volessero aprire pochi punti vendita, i punteggi dei comuni siciliani non crollerebbero così velocemente e sarebbero dunque la scelta ottima.

Tesi_Supermarket

Scegli la regione:

Scegli la catena:

In alternativa :

☐ Ricerca avanzata

Inserisci il numero di punti vendita da aprire:

Ricerca specifico comune:

Maggiorazione del peso delle catene presenti

☐ 0 ☐ 25% ☒ 50% ☐ 100%

☐ escludi presenti

☐ Ordina per comune

☐ Ordina per nome catena

La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione : SICILIA):

ADRANO (CT) SICILIA
ADRANO (CT) SICILIA
MONREALE (PA) SICILIA

Run time: 703 mills

Ultima considerazione riguarda l'efficienza dell'algoritmo. Si nota che la risposta rimane rapida anche nel caso (molto poco realistico) in cui una catena voglia aprire 1000 punti vendita.

Tesi_Supermarket

Scegli la regione:

Scegli la catena :

Esselunga

In alternativa :

☐ Ricerca avanzata

Inserisci la catena cercata..

Inserisci il numero di punti vendita da aprire:

1000

Ricerca specifico comune:

Inserisci un comune..

☒ Ordina per comune

☐ Ordina per nome catena

Cerca

Espandi

Maggiorazione del peso delle catene presenti

☐ 0

☐ 25%

☒ 50%

☐ 100%

☐ escludi presenti

La lista di comuni che ottimizza l'apertura di nuovi punti vendita (regione : LAZIO):

ALBANO LAZIALE (RM)

LAZIO

ALBANO LAZIALE (RM)

LAZIO

ALBANO LAZIALE (RM)

LAZIO

Run time: 1849 mills

Reset

Conclusioni

Siamo partiti dall'idea di un software capace di ottimizzare un problema di rilevanza aziendale, come l'apertura di nuovi punti vendita. Tengo a dire che tale software risulta essere leggermente diverso da quello proposto in partenza. Difatti, addentrandomi personalmente nella costruzione del database, a partire dai singoli comuni, ma soprattutto scrivendo il codice, mi domandavo cosa potesse volere l'utente finale. Attraverso ciò sono giunto al prodotto finale.

Uno dei principali limiti è dato dal fatto che il software non tiene conto della dimensione dei supermercati presente nel database. In tal senso si parte dal presupposto che ogni supermercato possa soddisfare la medesima domanda, cosa poco realistica. Per risolvere tale problema si dovrebbero recuperare dati più completi sulla dimensione dei supermercati, e dunque tenerne conto nel calcolo del punteggio.

Ovviamente anche nel database risiedono molti limiti. Questi sono giustificati dal fatto che, in qualità di studente che compie tale lavoro singolarmente, non ho potuto dedicare più tempo alla costruzione del database, né alla verifica completa della sua integrità. Tuttavia, con più di 16.000 punti vendita presenti, risulta sicuramente essere realistico.

Tale applicazione lavora a stretto contatto con il DB, mediante le classi DAO. Come citato in precedenza, reputo questa soluzione più snella. Infatti, nel caso in cui le dimensioni del DB aumentassero significativamente, le prestazioni del software non si modificherebbero eccessivamente, dal momento che tutte le operazioni di selezione sono lasciate alle query.

Concluderei dicendo che con tale applicazione non si pretende certo di prendere completamente le decisioni riguardo l'espansione aziendale, in termini territoriali, ma rappresenta un ottimo punto di partenza. Un utile strumento, attraverso il quale decidere dove indirizzare le ricerche di marketing più specifiche.



Quest'opera è distribuita con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

Copia della licenza consultabile al sito web: [Copia della licenza](#)