

MovieScore

Università degli Studi di Salerno
Laurea Magistrale in Informatica

Corso di Gestione Avanzata dei Dati

Docente:

Prof. Gennaro Costagliola

Studente:

Emanuele Russomanno

Anno Accademico 2016/2017



Specifica del problema

Capita spesso che prima di uscire per andare al cinema, si effettui una ricerca sui principali siti di cinema per:

- Avere una lista degli ultimi film attualmente in proiezione nelle sale.
- Scegliere tra questi il film che più si adatti ai nostri gusti.
- Verificare se il film scelto abbia ottenuto delle recensioni e dei rating positivi, così da evitare di rovinare la serata guardando un pessimo film.

Quindi, per evitare di sprecare denaro e tempo, ci assicuriamo che il film abbia avuto almeno un discreto consenso da parte di critica e spettatori, eseguendo un confronto dei rating su diversi siti che recensiscono film.

PROBLEMA: Effettuare questa operazione per ogni film può esser tedioso e time-consuming.

Descrizione funzionale di MovieScore

MovieScore consente di rendere questo processo molto più veloce e user-friendly, mettendo a disposizione informazioni e valutazioni relative al film, provenienti da diversi siti web, in un'unica schermata.

Prelievo dei film attualmente in proiezione nelle sale (fonte Mymovies), con possibile filtraggio per:

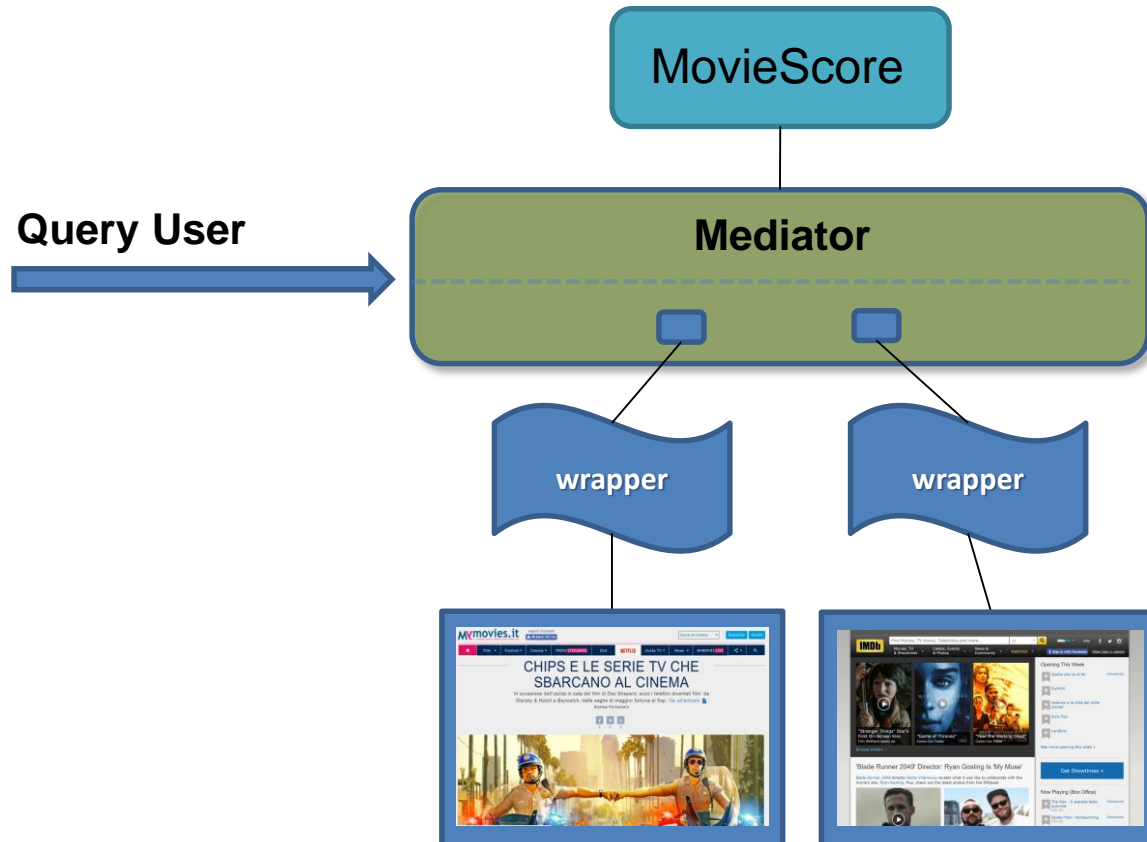
- Genere
- Durata
- Solo consigliati da Mymovies

Per ognuno dei film, verranno poi visualizzate delle informazioni attingendo anche alla seconda fonte (IMDB), tra cui:

- Rating:
 - IMDB
 - Mymovies
 - Metascore
- Mini-review:
 - Mymovies
- Trama:
 - IMDB

Architettura

- Ogni query inviata da parte dell'utente infatti corrisponde ad un'interrogazione dei dati in tempo reale direttamente sulle fonti, attraverso i wrapper che si pongono come strato intermedio, in modo da fornire al mediator un modello uniforme dei dati, sia in fase di interrogazione, sia in fase di restituzione dei risultati.



Fonti

Fonte	Metodo	Descrizione	Grado di volatilità
IMDB	API	Servizio per la ricerca di film presenti nel database.	Cambi Periodici
	Web Scraping	Sito web dedicato al cinema e alla TV. Rating dei film da parte degli utenti.	Cambi Periodici
Mymovies	Web Scraping	Sito web dedicato al mondo del cinema con recensioni, notizie e approfondimenti.	Cambi Periodici

Mymovies

Manual Wrapper



- Mymovies - *html*
 - Entrypoint: *http://www.mymovies.it/cinema/*
 - Xpath:
 - `//div[@class='padr padl']/div[2]/span[starts-with(text(),". $genere .")]/../..`
 - DOM:
 - `titolo = $element->getElementsByTagName("a")[0]->nodeValue`
 - `Mrating = $element->getElementsByTagName("div")[7]->getAttribute("ng-bind")`
 - `mini_rev = $element->getElementsByTagName("div")[5]->nodeValue`
 - `url_locandina = $element->getElementsByTagName("a")[1]->getElementsByTagName("img")[0]->getAttribute("src")`
 - `consigliato = $element->getElementsByTagName("div")[10]->nodeValue`
 - `genere_time = $element->getElementsByTagName("span")[0]->nodeValue;`

IMDB

Manual Wrapper



- IMDB - API
 - Per la ricerca della pagina del film, su cui effettuare lo scraping, ho utilizzato l'API di IMDB che fa riferimento al link seguente:
 - Endpoint: http://www.imdb.com/find?ref_=nv_sr_fn&q=
 - Parametri inviati tramite GET: *titolo del film*
 - Scraping
 - Xpath:
 - `//div[@id='main']/div/div[2]/table`
 - DOM:
 - `url = $element->item(0)->getElementsByTagName("a")[0]->getAttribute("href");`

IMDB

Manual Wrapper



- IMDB - *html*
 - Una volta ottenuta la pagina del film, estraggo le informazioni:
 - Xpath:
 - `scoreIMDB = //div[@id='title-overview-widget']/div[2]/div[2]/div/div[1]/div[1]/div[1]/strong/span/text()`
 - `scoreMetaScore = //div[@class='titleReviewBarItem']/a/div/span/text()`
 - `director = //span[@itemprop='director']/a/span`
 - `anno = //div[@id='titleYear']/a`
 - `url_locandina = //div[@class='poster']/a/img/@src`
 - `trama = //div[@class='summary_text']`

Schema Globale e Locale

InfoMovieInTheater(nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato)



InfoMovieSolo(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)

Mymovies(nomeFilm, genere, durata, scoreMM, consigliato, review)



IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)



GlobalAsView

1) **InfoMovieInTheater** (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato) :-

Mymovies(**nomeFilm**, genere, durata, scoreMM, consigliato, review),

IMDB(**nomeFilm**, trama, regista, anno, scoreIMDB, scoreMS, locandina)

2) **InfoMovieSolo**(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina) :- IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)

LocalAsView

Mymovies(nomeFilm, genere, durata, scoreMM, consigliato, review) :-
InfoMovieInTheater (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato)

IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina) :-
InfoMovieInTheater(nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato)

IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina) :-
InfoMovieSolo(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)

Query

1. **OnlyDrama&ShortMovie** (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato) :-
InfoMovieInTheater (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato), **InfoMovieSolo** (nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina),
genere='Drammatico', durata <= 100.

```
SELECT all
FROM InfoMovieInTheater, InfoMovieSolo
WHERE InfoMovieInTheater.genere='Drammatico' and InfoMovieInTheater.durata <= 100
InfoMovieInTheater.nomeFilm = InfoMovieSolo.nomeFilm
```

2. **FindMovie**(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina) :-
InfoMovieSolo (nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina), nomeFilm = 'Avatar'

```
SELECT all
FROM InfoMovieSolo
WHERE InfoMovieSolo.nomeFilm = 'Avatar'
```

Query

3. **OnlyRecomm**(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)
:- **InfoMovieInTheater** (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato), **InfoMovieSolo** (nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina), consigliato = 'Si'

SELECT all

FROM InfoMovieInTheater, InfoMovieSolo

WHERE InfoMovieInTheater.consigliato = 'Si', InfoMovieInTheater.nomeFilm = InfoMovieSolo.nomeFilm

GAV Query Unfolding

Query1 - OnlyDrama&ShortMovie (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato)

OnlyDrama&ShortMovie (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato):- InfoMovieInTheater (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato), InfoMovieSolo (nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina), genere='Drammatico', durata <= 100.

Unfolding + simplification

OnlyDrama&ShortMovie (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato) :- Mymovies(nomeFilm, genere, durata, scoreMM, consigliato, review), IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina), ~~IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)~~, genere='Drammatico', durata <= 100.

LAV bucket algorithm

OnlyDrama&ShortMovie (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato) :- InfoMovieInTheater (nomeFilm, genere, durata, review, regista, anno, trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato), InfoMovieSolo (nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina), genere='Drammatico', durata <= 100.

	InfoMovieInTheater	InfoMovieSolo
BUCKET	Mymovies(nomeFilm, genere, durata, scoreMM, consigliato, review)	IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)
	IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina)	

LAV bucket algorithm

Riformulazione 1

R1: **Mymovies**(nomeFilm, genere, durata, scoreMM, consigliato, review),
IMDB(nomeFilm, trama, regista, anno, scoreIMDB, scoreMS, locandina),
genere='Drammatico', durata <= 100

R1exp: **InfoMovieInTheater** (nomeFilm, genere, durata, review, regista, anno,
trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato),
genere='Drammatico', durata <= 100

Query Containment:

R1exp(nomeFilm, genere, durata, review, regista, anno, trama, locandina,
scoreMM, scoreIMDB, scoreMS, consigliato)

\subseteq

OnlyDrama&ShortMovie (nomeFilm, genere, durata, review, regista, anno,
trama, locandina, scoreMM, scoreIMDB, scoreMS, consigliato)

Tecnologie



PHP	XPath
DOM	HTML 5



Grazie per l'attenzione