

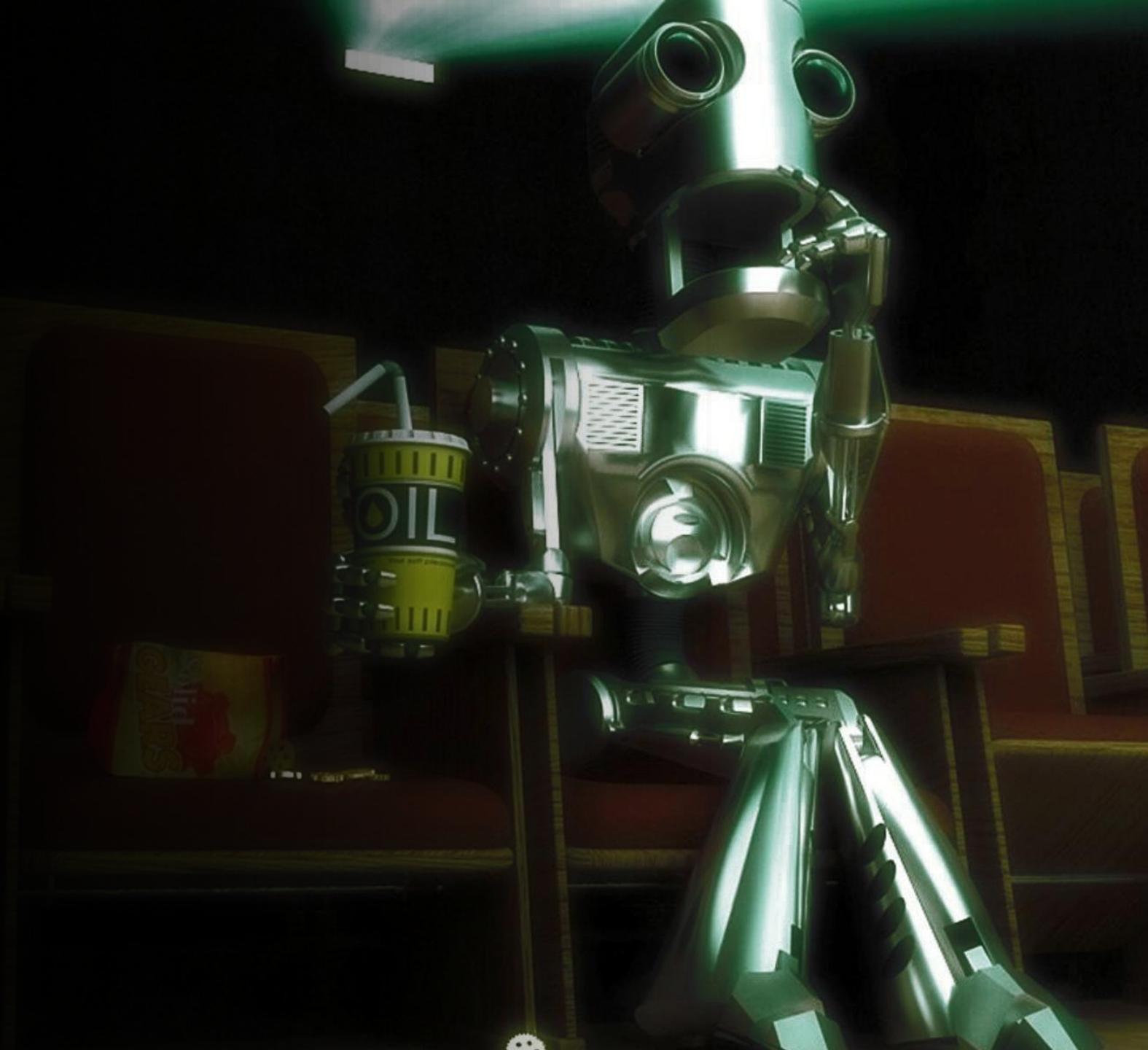


# PisaFlix

---

Progetto per il corso di *Data Mining  
and Machine Learning*

*Stefano Petrocchi*



# PisaFlix

L'applicazione prevede un *database* contenente informazioni generali su pellicole uscite nelle sale internazionali

I visitatori possono effettuare *ricerche* di film in forma anonima, è inoltre possibile *visualizzare* per ognuno una pagina contenente *informazioni* generali, trama e commenti da parte di utenti iscritti

Se si è *registriati* è infatti possibile:

- *Commentare* le pagine dei film
- Aggiungere film alla lista dei *preferiti*

(Sono presenti altre funzionalità non elencate per concisione)



# Scopo

Ampliamento del progetto di *Large Scale and Multi-Structured Databases* «*PisaFlix*» con l'aggiunta di moduli per:

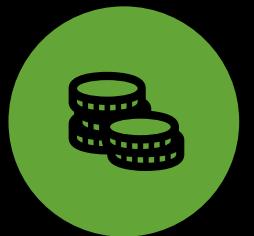
- **Safe search:** permette di filtrare contenuti per adulti attraverso una soglia mobile di *adultness* consentita
- **Suggerimento di contenuti:** nuovi contenuti vengono suggeriti all'utente in base ai suoi film preferiti

# Dataset Utilizzati

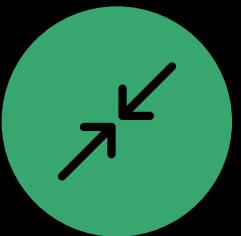
- Dataset di oltre *33000* film con trama, attori principali, registi, genere, titolo e data di uscita, prelevati da *Wikipedia*
- Dataset di circa *5000* pellicole con titolo, data di uscita e rating **MPAA**, prelevati da *IMDB*



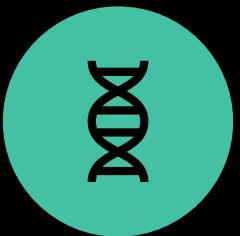
# Procedimento



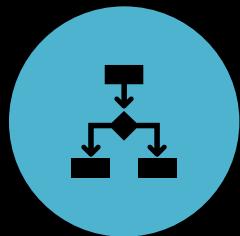
VETTOORIZZAZIONE DEI CONCETTI PRESENTI ALL'INTERNO DEI PLOT MEDIANTE L'UTILIZZO DI *TOKENIZER* E *STEMMER*



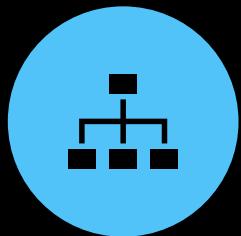
DIMENSIONALITY REDUCTION ESEGUITA CON LA TECNICA *TF-IDF*



OTTIMIZZAZIONE DEI PARAMETRI PER LA *TF-IDF* UTILIZZANDO UN ALGORITMO DI EVOLUZIONE DIFFERENZIALE



CLASSIFICAZIONE CON L'UTILIZZO DI UN *REGRESSORE LOGISTICO*

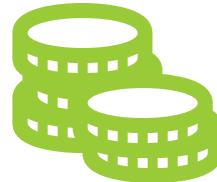


CLUSTERING MEDIANTE UN ALGORITMO *AGGLOMERATIVO GERARCHICO*

# Tokenization e Stemming

Una stessa tecnica di preprocessing per il clustering e la classificazione è stata utilizzata sulle **trame** dei film:

- Le parole presenti all'interno delle trame sono state suddivise mediante un processo di **tokenization**, ogni token è costituito da uno fino a tre **n-grammi**
- I *token* così ottenuti sono stati poi filtrati eliminando tutte le **stop words** e i **nomi** propri di persona presenti tra di essi
- I rimanenti token sono infine stati elaborati ricavando le radici delle parole utilizzando uno **snowball stemmer**



# Dimensionality Reduction

Il numero di termini ottenuti nei passi precedenti risulta troppo elevato per una classificazione o clustering, per questo sono state testate le seguenti tecniche di riduzione della dimensionalità:

- *Select k-best supervisionato* con metodo:
  - $\chi^2$
  - *Mutual-information*
- *Term frequency-inverse document frequency*



# Tempi di elaborazione

Il metodo *mutual-information* è stato scartato in quanto prevedeva tempi di risposta nettamente maggiori rispetto agli altri algoritmi, che ne avrebbero impedito un'ottimizzazione efficace (sull'hardware a mia disposizione):

- $\chi^2$ : meno di due minuti
- Tf-idf: meno di due minuti
- Mutual-info: più di venti minuti (dieci volte superiore)

# $\chi^2$ Feature Selection



Il metodo effettua una selezione **supervisionata** grazie a delle **etichette** che indicano se il film è adatto a bambini o per adulti



Il valore di  $\chi^2$  per un termine è proporzionale alla differenza tra la **frequenza attesa** di quel termine (la frequenza media del termine) e quella **realmente osservata** per le etichette



Vengono poi selezionati i **primi  $k$  termini** con  $\chi^2$  più elevato



I valori dei termini rimasti sono poi **normalizzati** (da notare che una distanza *euclidea* su valori normalizzati equivale ad una *cosine distance* sui valori originali)

# *Tf-idf* Feature Selection



Questo metodo al contrario del precedente **non è supervisionato**



Il valore assegnato ad un termine è **proporzionale alla frequenza nel documento** (in proporzione della dimensione del documento), **ma inversamente proporzionale alla frequenza nella collezione**



I termini sono dunque filtrati in base ad una soglia di **frequenza massima e minima** tra tutti i documenti (sopra o sotto una certa soglia non sono utili per una classificazione o clustering)



Vengono infine selezionati i **K** termini rimanenti **con valore più elevato**

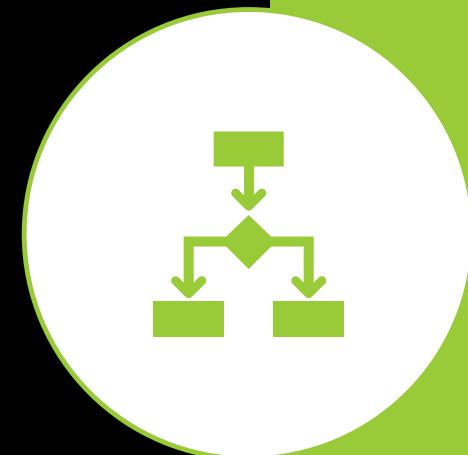
# Classificazione

L'obiettivo della classificazione è quello di attribuire un valore ai film in base alla quantità di contenuti per adulti al loro interno, indicata nell'applicazione dal coefficiente di «*adultness*»

La classificazione deve perciò fornire la *probabilità* che un film sia per adulti e *non* una classificazione netta

In questo modo l'utente, con l'ausilio di uno *slider*, può settare a piacimento un livello massimo di *adultness*

Solo i contenuti con *adultness* minore del livello settato saranno mostrati nelle ricerche o tra i film consigliati



# Integrazione dei Dataset

Il dataset di **Wikipedia** e quello di **IMDB** sono stati uniti effettuando un join con titolo e data di uscita



Il dataset così ottenuto contava circa **3000** entrate, dopo aver eliminato le entrate con valori di **MPAA** ambigui, i rimanenti valori **MPAA** sono stati riassunti in due macrocategorie «*per adulti*» e «*per bambini*»



Il dataset è stato poi bilanciato effettuando un ***undersampling*** sulla categoria «*per adulti*»



Il dataset finale contiene una collezione **bilanciata** di **1400** film di cui l'accuratezza è stata accertata con ispezione visiva



# Confronto tra Classificatori

I vari classificatori sono stati confrontati eseguendo una *stratified 10-fold cross validation*, per ogni classificatore è stata calcolata:

- *Accuratezza* media  $\pm$  margine di confidenza
- *Area Under the Curve*  $\pm$  margine di confidenza
- *Confusion Matrix* totale

# Classificatori con Preprocessing Tf-idf

Classificatore	Accuratezza Media	AUC	Confusion Matrix (Adulti   Bambini)
Logistic Regression	$83.2\% \pm 5.4\%$	$90.3\% \pm 3.2\%$	576   124 (Adulti Veri) 111   589 (Bambini Veri)
Gaussian Naïve Bayesian	$77.8\% \pm 6.5\%$	$84.6\% \pm 6.8\%$	550   150 (Adulti Veri) 161   539 (Bambini Veri)
Quadratic Discriminant	$70.6\% \pm 5.6\%$	$76.2\% \pm 6.8\%$	450   250 (Adulti Veri) 161   539 (Bambini Veri)
Decision Tree (Entropy)	$72.9\% \pm 5.1\%$	$72.9\% \pm 5.1\%$	519   181 (Adulti Veri) 198   502 (Bambini Veri)
Support Vectors (Dispendioso)	$83.2\% \pm 4.6\%$	$90.8\% \pm 3.0\%$	578   122 (Adulti Veri) 113   587 (Bambini Veri)
K-nearest Neighbors	$72.1\% \pm 4.3\%$	$79.7\% \pm 7.2\%$	578   122 (Adulti Veri) 286   432 (Bambini Veri)
Random Forest (Gini)	$83.9\% \pm 4.7\%$	$91.1\% \pm 2.8\%$	595   105 (Adulti Veri) 121   579 (Bambini Veri)
Random Forest (Entropy)	$84.2\% \pm 4.2\%$	$91.2\% \pm 2.5\%$	592   108 (Adulti Veri) 113   587 (Bambini Veri)
ADAboost (Logistic Regression)	$80.6\% \pm 4.9\%$	$88.1\% \pm 4.5\%$	547   153 (Adulti Veri) 119   581 (Bambini Veri)
ADAboost (Decision Tree Entropy)	$72.8\% \pm 7.8\%$	$72.8\% \pm 7.8\%$	532   168 (Adulti Veri) 213   487 (Bambini Veri)

Parametri utilizzati:  $\min\_df=0.1$ ,  $\max\_df=0.9$ ,  $\max\_features=500$

# Classificatori con Preprocessing $\chi^2$

Classificatore	Accuratezza Media	AUC	Confusion Matrix (Adulti   Bambini)
Logistic Regression	$86.1\% \pm 4.6\%$	$93.6\% \pm 3.0\%$	601   99 (Adulti Veri) 96   604 (Bambini Veri)
Gaussian Naïve Bayesian	$83.2\% \pm 3.9\%$	$89.1\% \pm 5.3\%$	536   164 (Adulti Veri) 71   629 (Bambini Veri)
Quadratic Discriminant	$81.3\% \pm 6.6\%$	$82.1\% \pm 7.2\%$	629   71 (Adulti Veri) 191   509 (Bambini Veri)
Decision Tree (Entropy)	$78.9\% \pm 6.7\%$	$78.9\% \pm 6.7\%$	541   159 (Adulti Veri) 137   563 (Bambini Veri)
Support Vectors (Dispendioso)	$86.7\% \pm 4.0\%$	$94.4\% \pm 2.7\%$	638   62 (Adulti Veri) 124   576 (Bambini Veri)
K-nearest Neighbors	$61.8\% \pm 5.4\%$	$79.2\% \pm 2.5\%$	171   529 (Adulti Veri) 6   694 (Bambini Veri)
Random Forest (Gini)	$87.4\% \pm 3.1\%$	$94.2\% \pm 3.6\%$	619   81 (Adulti Veri) 96   604 (Bambini Veri)
Random Forest (Entropy)	$87.6\% \pm 4.0\%$	$94.1\% \pm 3.0\%$	619   81 (Adulti Veri) 93   607 (Bambini Veri)
ADAboost (Logistic Regression)*	$87.3\% \pm 3.7\%$	$94.2\% \pm 2.2\%$	611   89 (Adulti Veri) 89   611 (Bambini Veri)
ADAboost (Decision Tree Entropy)	$79.3\% \pm 2.9\%$	$79.3\% \pm 2.9\%$	539   161 (Adulti Veri) 129   571 (Bambini Veri)

Parametro utilizzato:  $K=500$

\* Produce confini decisionali troppo netti per lo scopo della classificazione

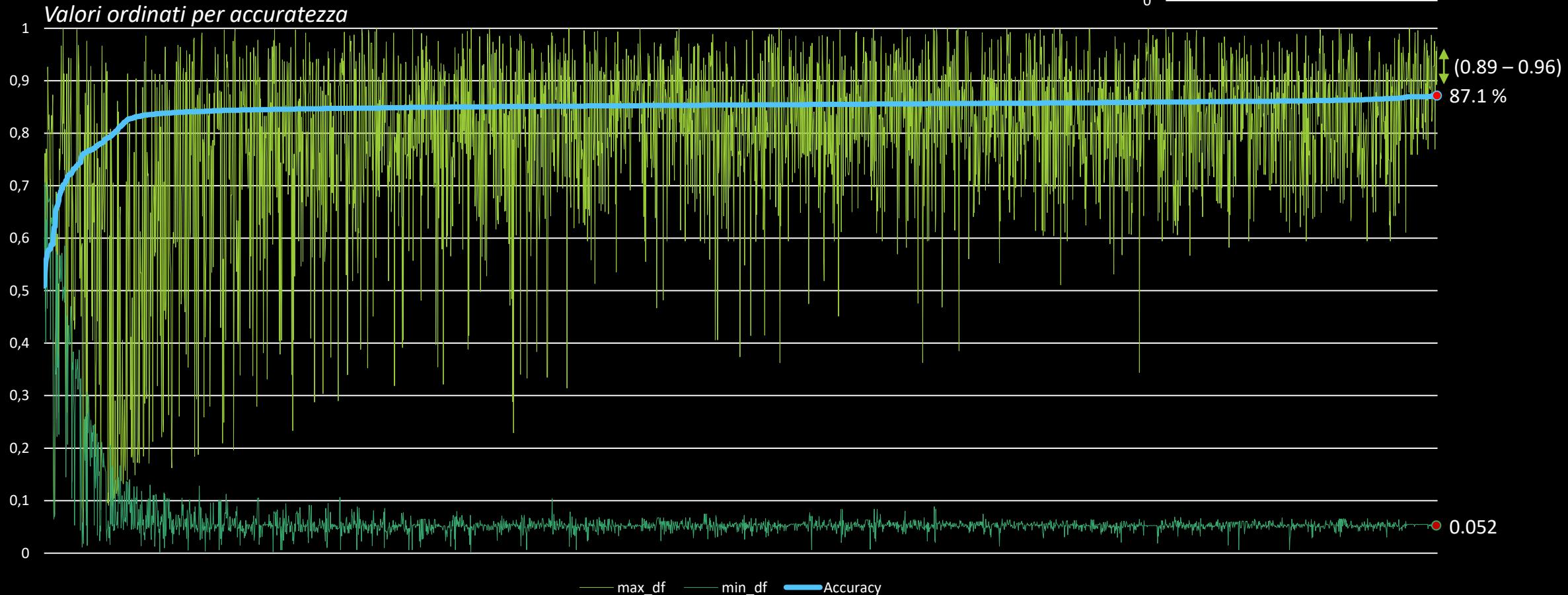
# Ottimizzazione dei parametri

I parametri utilizzati per il preprocessing sono stati ottimizzati utilizzando come funzione obiettivo la media dell'*error rate* di *dieci stratified 10-fold cross validation* (in modo da stabilizzare i risultati):

- Per  *$\chi^2$*  è stato utilizzato un ottimizzatore *brute force*, scelto in base al numero finito di possibili candidati per il parametro k
- Per *Tf-Idf* è stato utilizzato un algoritmo di *evoluzione differenziale* per trovare un minimo locale tra le infinite combinazioni dei tre parametri

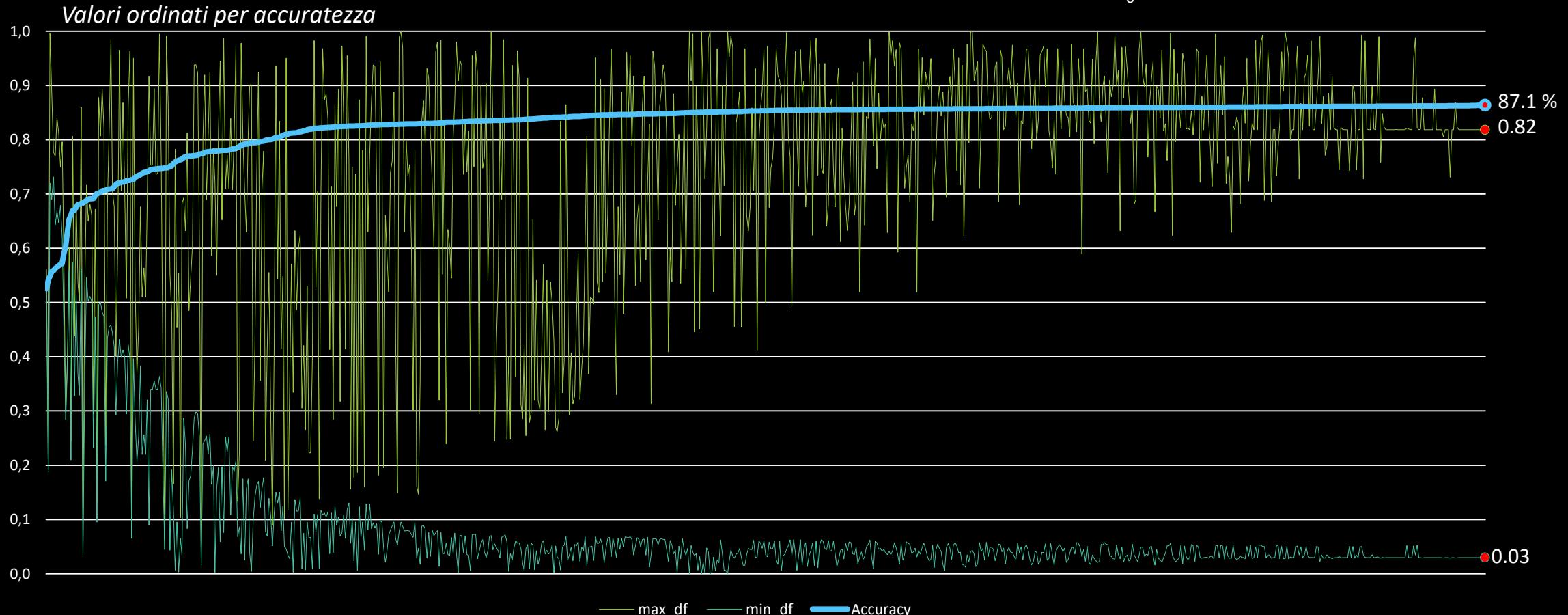


# Risultati Evoluzione Differenziale *(Random Forest con Entropia)*



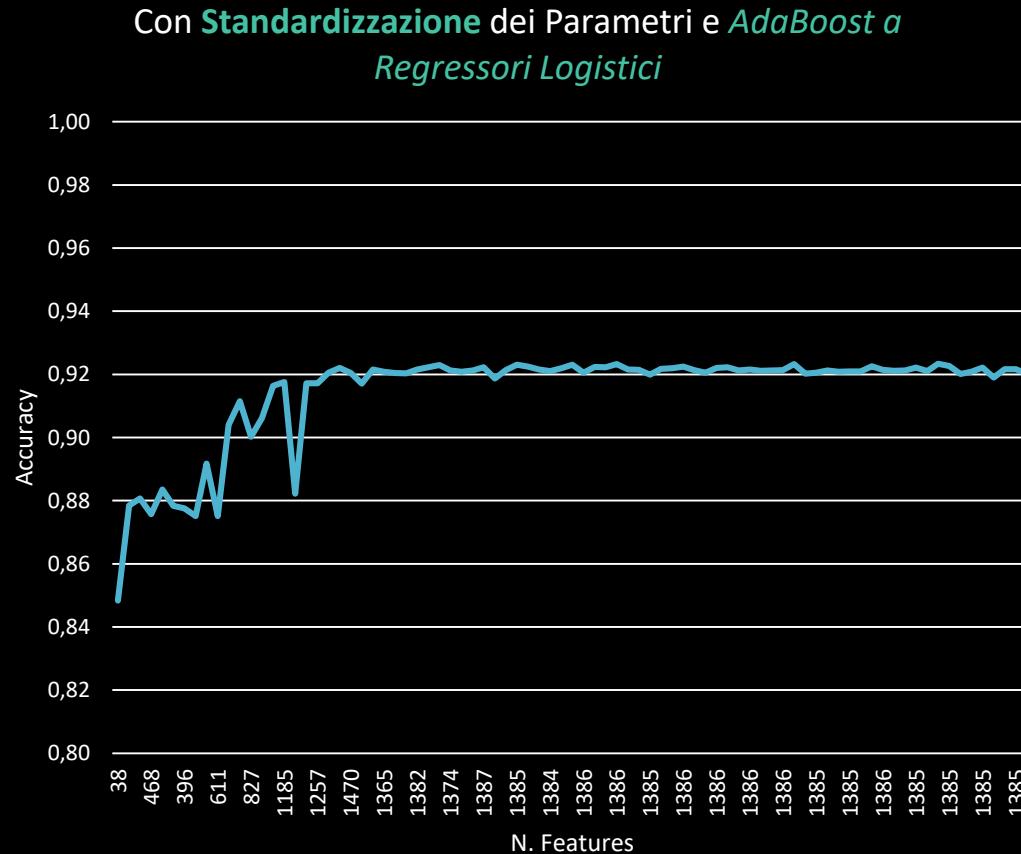
Parametri ottimi: `min_df=0.052, max_df=0.96, max_features=772` (Acc. 87%)

# Risultati Evoluzione Differenziale (*Regressione Logistica*)

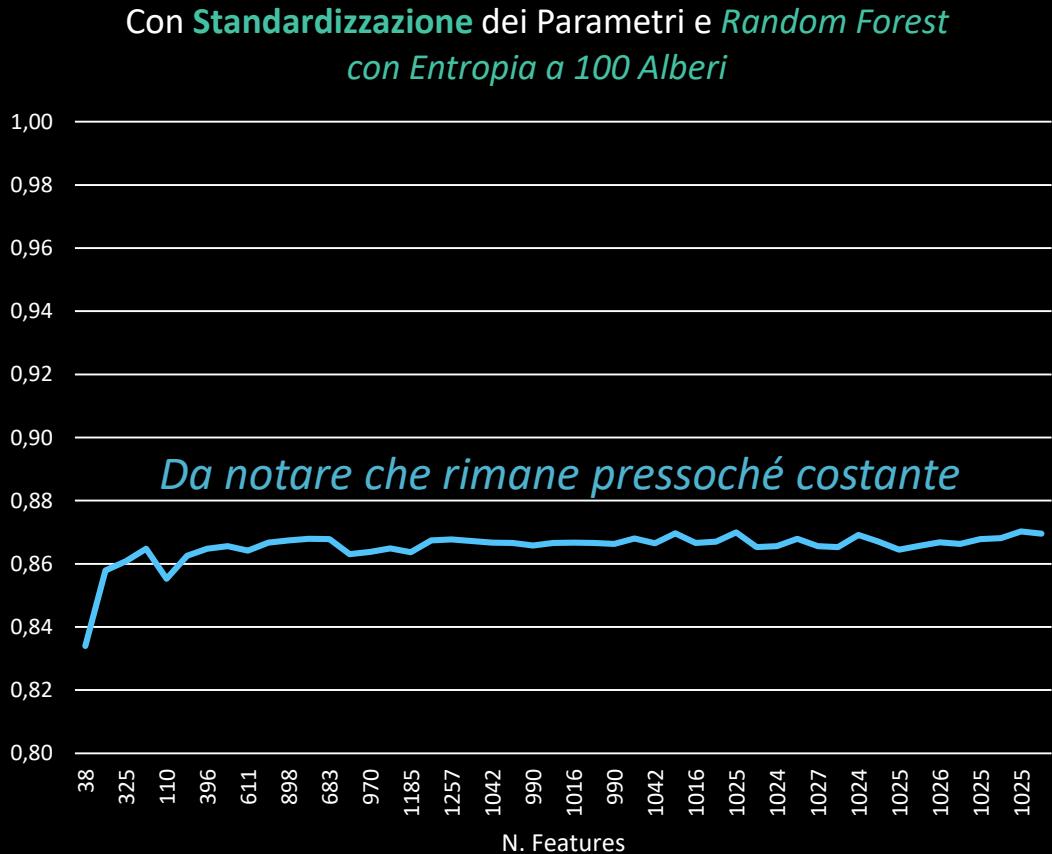


Parametri ottimi:  $\min\_df=0.03$ ,  $\max\_df=0.82$ ,  $\max\_features=1196$  (Acc. 86,5%)

# Risultati Ottimizzazione Brute Force



Parametro ottimo:  $K=1385$  (Acc. 92%)



Parametro ottimo:  $K > 500$  (Acc. 87%)  
(Risultati analoghi si hanno effettuando una normalizzazione L2 o tf-idf)

*Tag cloud*  
dei termini  
selezionati  
con  $\chi^2$

Da notare come i termini siano violenti, sessualmente esplicativi o riferiti al mondo della droga, tutte categorie prese in considerazione nel rating MPAA

## *Tag cloud* dei termini selezionati con *tf-idf*

Con *tf-idf* invece risultano  
molto più generici

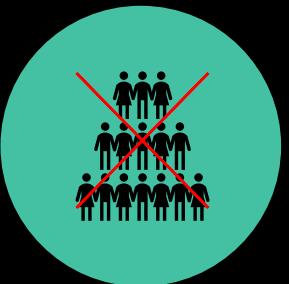
# Problematiche



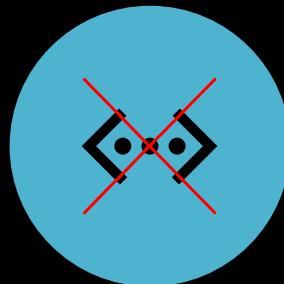
Utilizzare lo **stesso vocabolario** prelevato soltanto dai film etichettati **non porta a soluzioni generali** altrettanto **buone**, in quanto alcuni film presentano descrizioni brevi, che non contengono le parole del vocabolario



Il **classificatore random forest** è **parziale** verso quei film con una descrizione corta, dando un valore di **adultness** estremamente basso



Il classificatore **adaboost a regressori logistici** ottimizzato produce **confini decisionali estremamente netti** (0.999, 0.001) non adatti allo scopo della classificazione



I **support vectors** non sono pensati per generare una probabilità, il processo con cui viene calcolata in modo artificiale richiede un **tempo proibitivo**

# Conclusione

Invece di riutilizzare lo stesso vocabolario per tutti i film, questi vengono **prelevati a piccoli gruppi** e il vocabolario viene **ricalcolato** sull'insieme degli stessi e di quelli etichettati

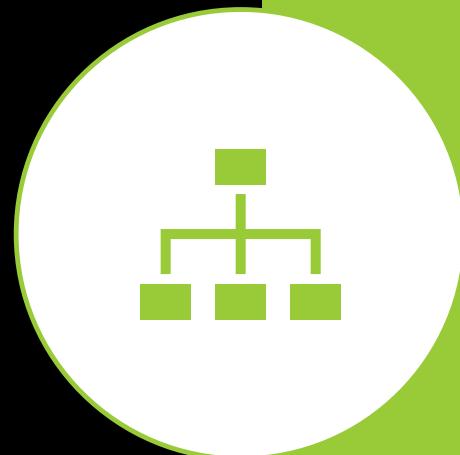
Non può perciò essere utilizzato un metodo ***supervisionato*** per la ***vettorizzazione e feature selection***, perciò viene utilizzato il metodo ***tf-idf*** che presentava comunque una precisione analoga

Come classificatore viene utilizzato un ***regressore logistico***, scartando i classificatori con le problematiche precedentemente elencate, che presenta comunque una precisione paragonabile

# Clustering

L'obiettivo del clustering è quello di ottenere un numero limitato di film suggeriti partendo dai preferiti dell'utente:

- Il numero di cluster da creare sarà perciò  $\frac{n}{k}$  dove  $n$  è il numero totale di film e  $k$  il numero di suggeriti medi per film
- È essenziale un algoritmo che permetta di selezionare il numero di cluster e che abbia risultati equilibrati
- Per ogni cluster è inoltre prelevato il termine con maggior peso tra i valori dei centri, insieme a quei termini il cui peso non è inferiore a metà di quello principale, questi servono per caratterizzare i vari cluster ottenuti



# Confronto tra Algoritmi di Clustering

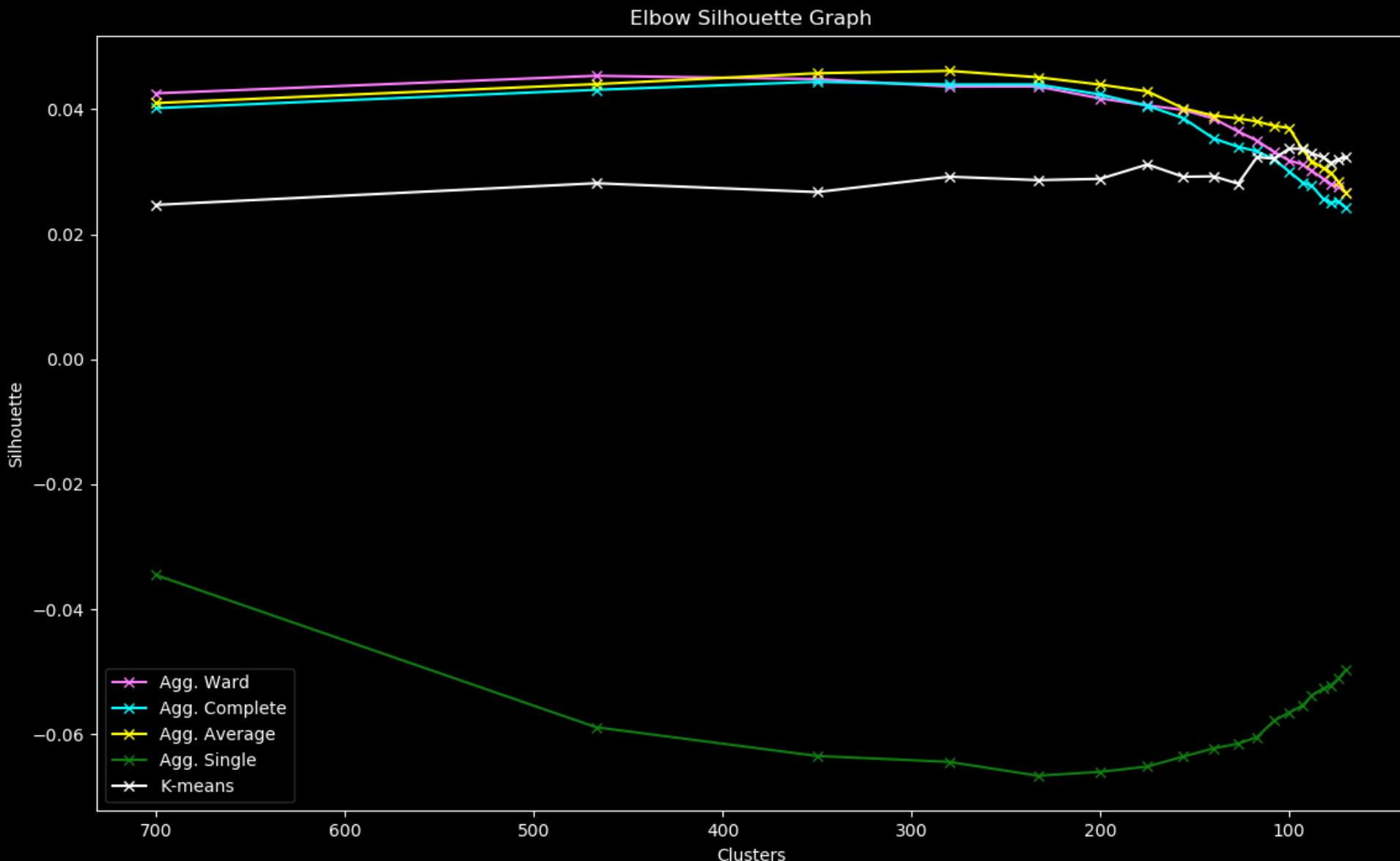
I vari algoritmi di clustering sono stati confrontati in modo da scegliere il più adatto

A tale scopo sono stati confrontati con un valore di  $\frac{n}{k}$  compreso tra due e venti (film suggeriti per preferito) usando i seguenti criteri:

- Silhouette, per verificare la qualità dei cluster
- Numero massimo di film in ogni cluster, in modo da verificarne un'equilibrata distribuzione (il numero medio sarà naturalmente  $\frac{n}{k}$ )

Per riuscire ad effettuare i vari test in maniera efficiente è stato utilizzato lo stesso campione di 1400 film della classificazione e non l'intero dataset di 33 mila pellicole

# Silhouette

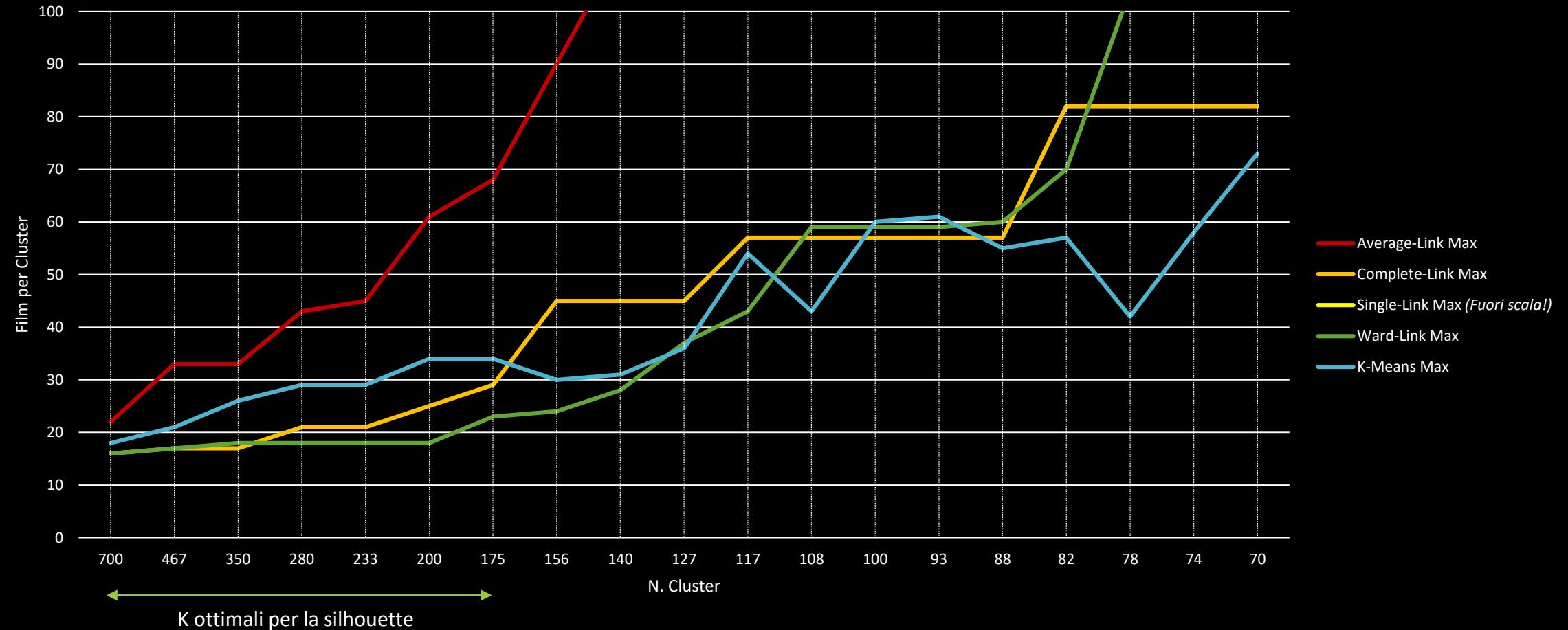


Dal confronto tra le **silhouette** dei vari algoritmi emerge come sia massima per un numero medio di film in ogni classe compreso tra due e otto

Per quei valori gli algoritmi di clustering **gerarchico agglomerativo** con *ward-link*, *average-link* e *complete-link* si equivalgono

L'algoritmo **K-means** risulta avere un valore nettamente minore, mentre il **single-link** addirittura negativo

# Distribuzione



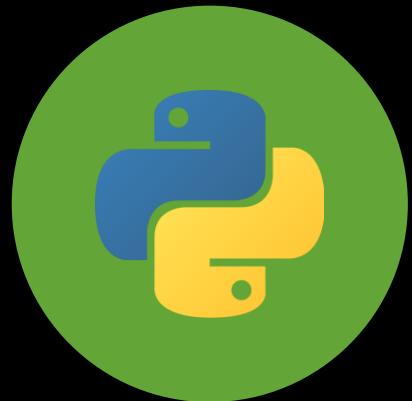
# Conclusione

Il metodo con le caratteristiche migliori risulta essere quello *gerarchico agglomerativo con complete-link* e  $\frac{n}{k} = 4$  (k=350) in quanto:

- Ha un valore di *silhouette* tra i migliori e comunque maggiore di 0.4
- Mantiene il numero massimo di film per cluster sotto ai venti, garantendo una distribuzione omogenea dei consigli sui film

Nel clustering del dataset completo sono aggiunte alla trama anche le parole che descrivono il genere, i registi e gli attori dei film

# Implementazione



È STATO UTILIZZATO IL LINGUAGGIO **PYTHON** ED IN PARTICOLARE LA LIBRERIA **SKLEARN** PER ESEGUIRE LE OPERAZIONI DI MACHINE LEARNING



IL LINGUAGGIO **JAVA** È STATO UTILIZZATO PER IMPLEMENTARE LE FUNZIONALITÀ DELL'APPLICAZIONE E L'INTERFACCIA GRAFICA



UN DBMS **MONGODB** È STATO UTILIZZATO PER MEMORIZZARE I RISULTATI DELLE OPERAZIONI DI MACHINE LEARNING E RENDERLI DISPONIBILI SULL'APPLICAZIONE

# Classificazione (Classe «Classifier.java»)



Il processo **termina** quando tutti i film sono stati classificati

# Clustering (Classe «Clusterizer.java»)

Tramite la classe Java viene eseguito l'**accesso al server MongoDB** e scaricato l'intero dataset dei film su un file csv (o un campione per volta in modo simile al classificatore)

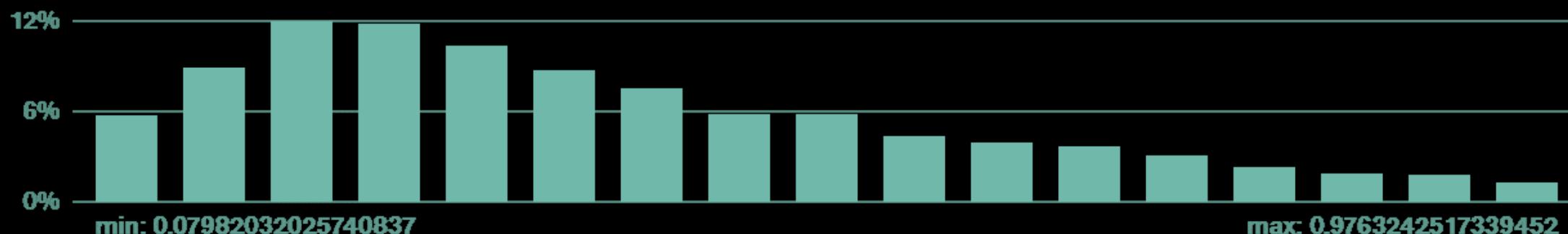
La classe esegue uno script *Python* (*clusterizer.py*) che si occupa di eseguire le **operazioni di preprocessing** (vettorizzazione *tf-idf*) sul contenuto del file csv, **calcolare i cluster** e i **termini** più rappresentativi degli stessi ed infine **salvare i risultati** su di un file *txt*

La classe java infine **recupera i risultati** dal file *txt* e **aggiorna il database** con le nuove assegnazioni dei cluster ed i nuovi tag per ogni film elaborato

# Risultati della Classificazione

Dalle statistiche prelevate dal database è possibile osservare come le probabilità siano state assegnate secondo una **curva gaussiana** spostata verso valori più vicini a film per bambini

I valori utilizzati dall'applicazione sono **normalizzati min-max** a tempo di esecuzione in modo da presentare un range di *adultness* tra 0% e 100% senza dover ogni volta aggiornare l'intero database



Valor Medio: 0.39 | Valore Minimo: 0.03 | Valore Massimo: 0.98  
Nota: L'estremo sinistro contiene un intervallo maggiore rispetto agli altri



Impostando la barra della safe search al 95% tutti i film con adulthood normalizzata maggiore di 0.05 non vengono mostrati nelle ricerche

Si osservi come tutti i film selezionati siano chiaramente film per bambini

Logged as: admin

[Logout](#)

[Show Profile](#)

# Interfaccia dell'Applicazione

Operazione completata

Recent  Doraemon ... 2017 Adultness: 4%	Recent  Monster H... 2015 Adultness: 3%	Recent  Born in Chi... 2017 Adultness: 3%	Recent  My Little P... 2017 Adultness: 4%	Recent  Magic to ... 2011 Adultness: 3%	Recent  My Big Bos... 2014 Adultness: 4%	Recent  Kit Kittred... 2008 Adultness: 3%	Recent  The Swan P... 2012 Adultness: 2%	Recent  Frozen 2013 Adultness: 4%
Recent  Madagasca... 2012 Adultness: 1%	Recent  Growing U... 2017 Adultness: 2%	Recent  All's Well, E... 2009 Adultness: 4%	Recent  Meow 2017 Adultness: 2%	Recent  Yona Yona ... 2009 Adultness: 1%	Recent  Brave 2012 Adultness: 3%	Recent  Clifford's R... 2004 Adultness: 1%	Recent  Magic Mag... 2003 Adultness: 1%	Recent  Moon Castl... 2011 Adultness: 4%
Recent  Thru the M... 2015 Adultness: 2%	Recent  Teacher's P... 2004 Adultness: 3%	Recent  Magic Tree... 2012 Adultness: 2%	Recent  The Adventu... 2005 Adultness: 3%	Recent  Captain Un... 2017 Adultness: 1%	Recent  Doraemon ... 2008 Adultness: 0%	Recent  Doraemon ... 2007 Adultness: 2%	Recent  Mary and t... 2017 Adultness: 3%	Recent  Barbie: The... 2012 Adultness: 4%
Recent  Thru the M... 2015 Adultness: 2%	Recent  Teacher's P... 2004 Adultness: 3%	Recent  Magic Tree... 2012 Adultness: 2%	Recent  The Adventu... 2005 Adultness: 3%	Recent  Captain Un... 2017 Adultness: 1%	Recent  Doraemon ... 2008 Adultness: 0%	Recent  Doraemon ... 2007 Adultness: 2%	Recent  Mary and t... 2017 Adultness: 3%	Recent  Barbie: The... 2012 Adultness: 4%



Modificando il codice dell'applicazione sono stati ricercati tutti i film con un *adultness* normalizzata maggiore di 0.95

Si può osservare come essi siano chiaramente film per adulti

Logged as: admin

[Logout](#)

[Show Profile](#)

	Bullet to th... 2013 Adultness: 98%		Killers 2014 Adultness: 96%		Beeba Boys 2015 Adultness: 97%		The Killer I... 2010 Adultness: 96%		Rampage: ... 2014 Adultness: 96%		Shagird 2011 Adultness: 97%		Memoir of ... 2017 Adultness: 96%		Dark Blue 2003 Adultness: 98%		Gangs of ... 2012 Adultness: 97%
	Perrier's Bo... 2009 Adultness: 96%		Crank 2006 Adultness: 97%		The Salton ... 2002 Adultness: 98%		Harry Brown 2009 Adultness: 97%		Killer Joe 2012 Adultness: 98%		My Bloody ... 2009 Adultness: 96%		Brooklyn's ... 2010 Adultness: 98%		Taking Lives 2004 Adultness: 97%		Header 2006 Adultness: 98%
	Mystic River 2003 Adultness: 97%		SPL: Sha P... 2005 Adultness: 96%		Dark Blue 2002 Adultness: 98%		Collateral 2004 Adultness: 97%		The Bourne... 2004 Adultness: 97%		Rakta Chari... 2010 Adultness: 98%		Narc 2002 Adultness: 98%		Heat, TheT... 2013 Adultness: 96%		Horrible Bo... 2011 Adultness: 97%



Effettuando una ricerca su *Harry Potter* si può notare come il rating si evolve nella successione dei film della serie: la distribuzione di *adultness* in questi film è un'ottima dimostrazione della bontà del modello

Gli ultimi due film in particolare «*I Doni della Morte*» risultano avere un valore di *adultness maggiore* rispetto ai primi, infatti trattano tematiche più crude, con assidui riferimenti ad assassinii e morti

Username

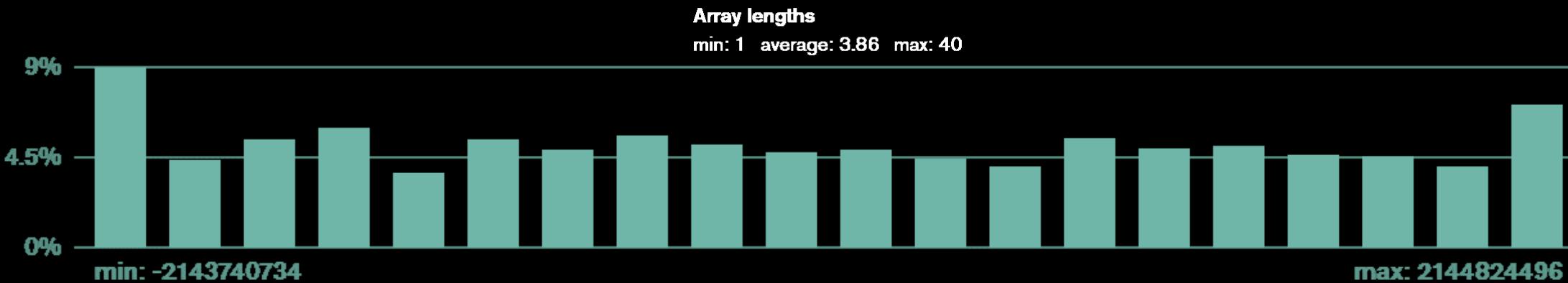
Password

 Login Register

# Risultati del Clustering

I risultati del clustering sull'insieme completo di *33 mila* film risultano ottimi, come evidenziato dalle **statistiche di MongoDB**:

- I film risultano **distribuiti omogeneamente** nei vari cluster
- Il numero medio di **tag** rappresentativi per ogni cluster è circa **quattro**



Nota: Per come è costruito l'istogramma nelle statistiche di MongoDB gli estremi tendono ad avere un range più ampio, la stessa cosa accade diminuendo l'intervallo



Aggiungendo il film «*Saw*» tra i preferiti vengono automaticamente suggeriti tutti i *sequel*, ciò indica che il processo di clustering li ha assegnati ad uno stesso cluster

Inoltre i tag «*die*», «*find*» e «*game*» sembrano ben rappresentare il cluster

Logged as: SawFan

[Logout](#)

[Show Profile](#)

\*\*\* Suggested \*\*\*



Jigsaw  
2017  
#die #find #game

Recent



Solo  
2017  
#father #kill

Favourite



Saw  
2004  
#die #find #game

\*\*\* Suggested \*\*\*



Saw III  
2006  
#die #find #game

Recent



Jomonte S...  
2017  
#busi

Recent



Bird  
2017  
#admit #chang

Recent



2:22  
2017  
#day #letter

\*\*\* Suggested \*\*\*



Saw IV  
2007  
#die #find #game

\*\*\* Suggested \*\*\*



Saw VI  
2009  
#die #find #game

Recent



Gun Shy  
2017  
#action #kidnap

Recent



Raju Gari G...  
2017  
#action #horror

Recent



J Revolusi  
2017  
#action #sister

Recent



Forgotten  
2017  
#action #mystery

Recent



First Kill  
2017  
#kidnap #mystery

Recent



The Belko ...  
2017  
#arriv #attempt

Recent



Godzilla: Pl...  
2017  
#group #kill

\*\*\* Suggested \*\*\*



Saw V  
2008  
#die #find #game

Recent



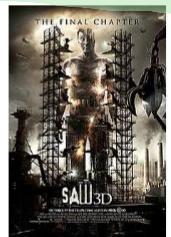
Ancien and...  
2017  
#car

\*\*\* Suggested \*\*\*



Saw II  
2005  
#die #find #game

\*\*\* Suggested \*\*\*



Saw VII  
2010  
#die #find #game

Recent



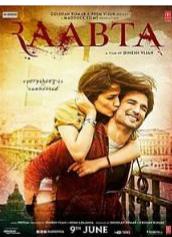
Andhhagadu  
2017  
#eye

Recent



Olanlar Oldu  
2017  
#comedi

Recent



Raabta  
2017  
#happen #kidnap

Recent



Pitch Perfe...  
2017  
#group #music

Recent



Angels We...  
2017  
#come #man

\*\*\* Suggested \*\*\*



The Zooke...  
2017  
#arriv #ask

\*\*\* Suggested \*\*\*



The Hungre...  
2012  
#die #find #game

CLASSIFICATO  
IN BASE AL GENERE

LA TRAMA  
È SIMILE!



Aggiungendo ai preferiti quattro film della saga «[Star Wars](#)» oltre ai sequel vengono suggeriti anche altri titoli a tema fantascientifico come «[Transformer](#)» e «[Alien](#)»

Logged as: StarWarsL...

[Logout](#)[Show Profile](#)

<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>
Srikanth, S... 2004 #armi #kidnap	Star Wars: ... 2013 #battl #destroy	Prometheus 2012 #kill #remain	Soldier 1998 #kill #remain	Tirante lo Bl... 2006 #armi #battl	Megaforce 1982 #battl #destroy	Jason X 2002 #kill #remain	Galaxy of T... 1981 #kill #remain	Reptilian 1999 #battl #destroy
<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>Favourite</b>	<b>*** Suggested ***</b>	<b>Favourite</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>
Navy SEALs 1990 #kill #remain	Yonggary 1999 #battl #destroy	Siege of the... 1963 #armi #battl	Star Wars: ... 2008 #armi #kidnap	Return of t... 1983 #battl #death	Star Wars: ... 2012 #armi #battl	Godzilla vs.... 1973 #battl #destroy	Excalibur 1981 #armi #battl	Spider-Man... 2007 #battl #death
<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>	<b>Favourite</b>	<b>*** Suggested ***</b>	<b>Favourite</b>	<b>*** Suggested ***</b>	<b>*** Suggested ***</b>
The Iron D... 1934 #armi #battl	Transforme... 2009 #battl #destroy	Mars Attac... 1996 #kill #remain	Alien 1979 #kill #remain	Star Wars E... 1977 #battl #death	Star Wars: ... 1999 #armi #battl	Rogue One... 2016 #kill #remain	Star Wars: ... 2017 #battl #destroy	Thor: Ragn... 2017 #battl #destroy

*The End*