Project Ανάκτησης Πληροφορίας

Ονοματεπώνυμο: Γκανάς Ιωάννης

Αριθμός Μητρώου: 1053577

Περιβάλλον υλοποίησης

Το περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση του project είναι το Anaconda Spyder 4.0.1 ενώ η γλώσσα προγραμματισμού είναι η Python 3.7.6 64-bit. Για την εγκατάσταση τους ακολουθήθηκαν οι οδηγίες εγκατάστασης από τις επίσημες ιστοσελίδες τους χωρίς να υπάρξει κάποιο πρόβλημα. . Παράλληλα χρησιμοποίησα την elasticsearch 7.10.0 καθώς και την kibana 7.10.0. Τα δύο τελευταία εργαλεία δεν χρειάζονται εγκατάσταση αλλά με το κατέβασμα τους μπορούν να χρησιμοποιηθούν αμέσως.

Λογισμικό

Windows 10 Education

Βιβλιοθήκες που χρησιμοποιήθηκαν

- **elasticsearch**: Η βιβλιοθήκη elasticsearch χρησιμοποιήθηκε σε όλα τα ερωτήματα του project. Η εγκατάσταση της έγινε πληκτρολογώντας conda install elasticsearch στο Anaconda Prompt. Στο ερώτημα 1 χρησιμοποιήθηκε για να δημιουργήσει το index στην elasticsearch καθώς και να φορτώσει τα δεδομένα των ταινιών σε αυτό. Στα υπόλοιπα ερωτήματα χρησιμοποιήθηκε για την αναζήτηση ταινιών με βάση την μετρική ομοιότητας BM25.
- **Pandas**: Η βιβλιοθήκη χρησιμοποιήθηκε για όλα τα ερωτήματα. Η εγκατάσταση της έγινε με την εντολή conda install pandas μέσω του Anaconda Prompt. Η Pandas χρησιμοποιήθηκε κυρίως για τις δυνατότητες που παρέχει για την διαχείριση δομών δεδομένων όπως τα dataframes. Οι δυνατότητες που χρησιμοποίησα αφορούν την εύκολη διαχείριση ενός πίνακα προσθέτοντας η αφαιρώντας στήλες η σειρές, την δυνατότητα για εύκολή εύρεση και αντικατάσταση στοιχείων με συγκεκριμένες τιμές (π.χ NaN), τον εύκολο υπολογισμό μεσών όρων κατά άξονες, την μετατροπή των dataframe

στην μορφή που χρειαζόμουν, τον συνδυασμό dataframes , η μετατροπή σε one hot encoding κ.α. .

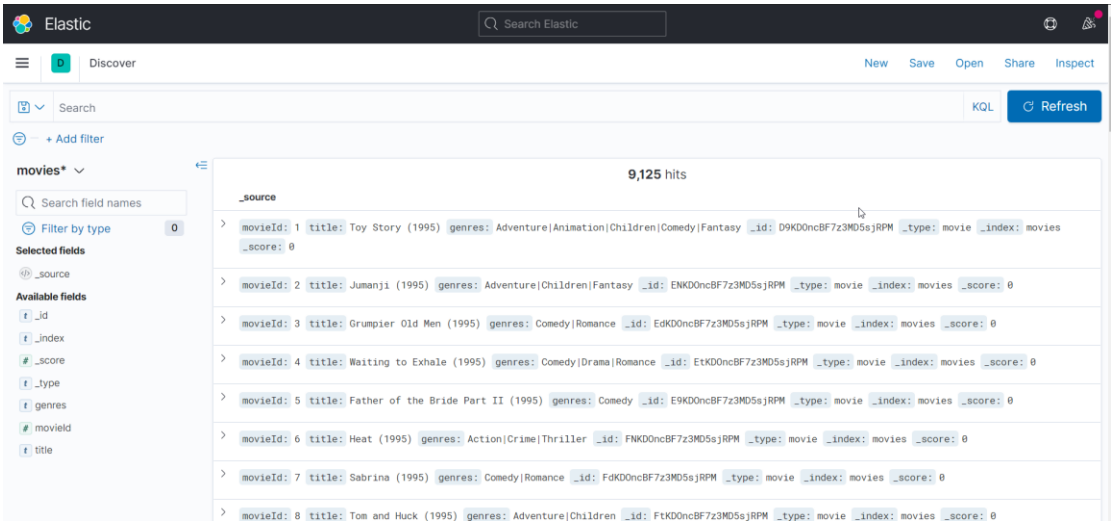
- **Math** : Η βιβλιοθήκη χρησιμοποιήθηκε στα ερωτήματα 2,3,4. Η εγκατάσταση της έγινε με την εντολή conda install math μέσω του Anaconda Prompt. Την χρησιμοποίησα για να ελέγξω το αν κάποιες τιμές ήταν NaN μέσω της συνάρτησης της isnan(arg) που επιστρέφει True or False.
- **ScikitLearn:** Η βιβλιοθήκη αυτή χρησιμοποιήθηκε στο ερώτημα 3. Η εγκατάσταση της έγινε μέσω του Anaconda Prompt με χρήση της εντολής conda install scikit-learn. Η βιβλιοθήκη αυτή παρέχει πολλές δυνατότητες για επίλυση προβλημάτων με τεχνικές Machine Learning. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε για την υλοποίηση του αλγορίθμου Kmeans. Η εισαγωγή του έγινε με την εντολή: from sklearn.cluster import KMeans
- **NumPy**: Η βιβλιοθήκη NumPy χρησιμοποιήθηκε στα ερωτήματα 3,4. Η εγκατάσταση της έγινε μέσω του Anaconda Prompt με την εντολή conda install numpy. Από την βιβλιοθήκη αυτή χρησιμοποιήθηκε η numpy.random.seed() έτσι ώστε να κάνω seed την γεννήτρια τυχαίων αριθμών.
- **Keras:** Η βιβλιοθήκη αυτή χρησιμοποιήθηκε μόνο στο ερώτημα 4. Η εγκατάσταση της έγινε στο Anaconda Prompt με την εξής διαδικασία:
    - Αρχικά , δημιουργήθηκε ένα Environment στην GPU του υπολογιστή μέσω της εντολής conda create --name PythonGPU
    - Έπειτα ενεργοποιήθηκε το περιβάλλον: activate PythonGPU΄
    - Η εγκατάσταση του Keras έγινε στο νέο περιβάλλον: conda install -c anaconda keras-gpu
    - Τέλος , απενεργοποιείται το Environment ου δημιουργήθηκε : conda deactivate

  Η βιβλιοθήκη Keras χρησιμοποιήθηκε για την δυνατότητα που παρέχει για την δημιουργία Νευρωνικών. Για το συγκεκριμένο πρόβλημα , εισήχθησαν :

    - from keras.models import Sequential: Χρησιμοποιήθηκε για την δημιουργία ενός ακολουθιακού νευρωνικού δικτύου.
    - from keras.layers import Dense: Χρησιμοποιήθηκε για τον προσδιορισμό των παραμέτρων του Νευρωνικού (Διάσταση Εισόδου , Κομβόι , Κρυφά επίπεδα , Συνάρτηση Ενεργοποίησης).
- **genism**: Η βιβλιοθήκη αυτή χρησιμοποιήθηκε μόνο στο ερώτημα 4. Η εγκατάσταση της έγινε μέσω του Anaconda Prompt με την εντολή conda install -c anaconda gensim. Για το συγκεκριμένο πρόβλημα εισήχθησαν:
    - from gensim.models.doc2vec import Doc2Vec, TaggedDocument: Χρησιμοποιήθηκαν για την δημιουργία word embeddings για κάθε τίτλο.

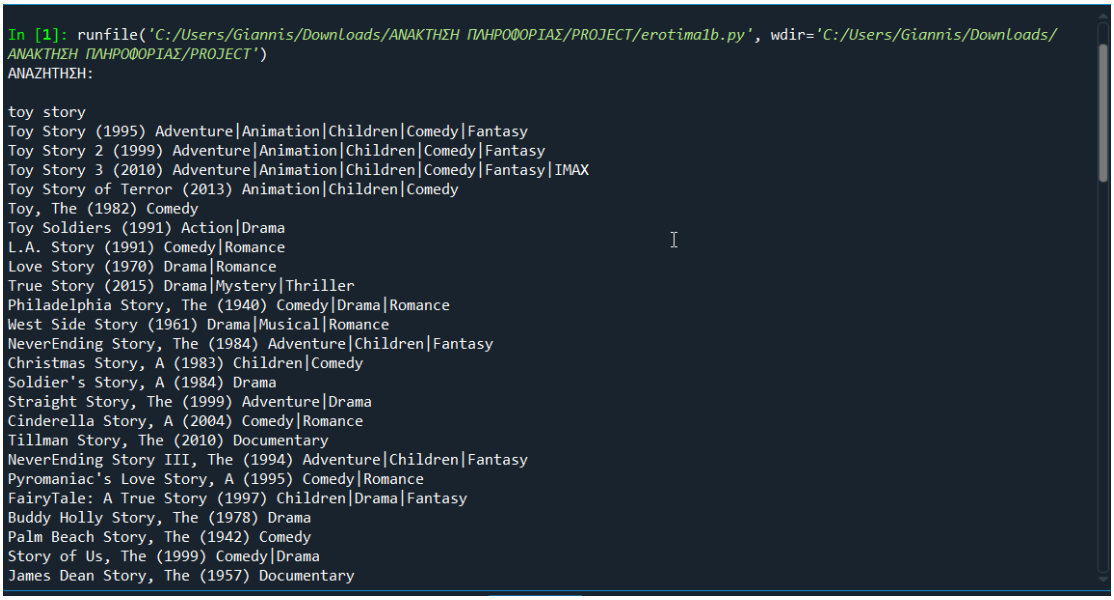
Σύντομη περιγραφή της διαδικασία υλοποίησης.

- ***Ερώτημα 1.α:*** Σε αυτό το ερώτημα μας ζητείται να εγκαταστήσουμε στο σύστημα μας την Elasticsearch και να γράψουμε ένα μικρό πρόγραμμα το οποίο θα διαβάζει τις εγγραφές που περιέχονται στο αρχείο movies.csv και θα τις εισάγει στην Elasticsearch. Αφού δημιούργησα ένα instance της Elasticsearch (es) , στην συνέχεια δημιούργησα ένα index με το όνομα movies. Παράλληλα διάβασα το movies.csv και το έβαλα σε ένα dataframe ενώ τέλος φόρτωσα μέσω της bulk τα δεδομένα αυτά στο index. Για να βεβαιωθώ ότι τα δεδομένα ανέβηκαν με επιτυχία στην Elasticsearch χρησιμοποίησα το kibana. Στην παρακάτω εικόνα φαίνεται πως τα δεδομένα έχουν ανέβει σωστά αφού βλέπω 9.125 εγγραφές ταινιών όσες δηλαδή είναι και οι ταινίες που βρίσκονται στο movies.csv



- ***Ερώτημα 1.β***: Σε αυτό το ερώτημα μας ζητείται να γράψουμε ένα δεύτερο πρόγραμμα το οποίο θα δέχεται ως είσοδο (είτε ως όρισμα γραμμής εντολών είτε κατά τη διάρκεια της εκτέλεσής του) ένα αλφαριθμητικό και θα επιστρέφει την λίστα των ταινιών που ταιριάζουν με αυτό διατεταγμένη σε φθίνουσα σειρά σύμφωνα με τη μετρική ομοιότητας της Elasticsearch (BM25). Εδώ μετά την δημιουργία του instance της elasticsearch έκανα αναζήτηση στο index μου με ένα match query όπου το πεδίο title παίρνει την τιμή που δίνει ο χρήστης. Έτσι μου επιστράφηκε λίστα ταινιών με βάση την BM25. Στην παρακάτω εικόνα φαίνεται το αποτέλεσμα που πήρα για την

αναζήτηση toy story.



- ***Ερώτημα 2***: Σε αυτό το ερώτημα μας ζητήθηκε να φτιάξουμε μια νέα μετρική η οποία θα συνυπολογίζει εκτός από την BM25, την μέση βαθμολογία που έχει πάρει μια ταινία καθώς και την βαθμολογία που της έχει δώσει ο χρήστης. Αφού φόρτωσα το ratings.csv στο οποίο περιέχονται τα απαραίτητα δεδομένα, στην συνέχεια με την βοήθεια συναρτήσεων της pandas (merge,groupby,mean) σχημάτισα ένα dataframe στο οποίο για κάθε ταινία που επέστρεψε το match query από την elasticsearch περιλαμβάνεται το score σύμφωνα με το BM25 (orderRating), η μέση βαθμολογία της ταινίας (meanRating) καθώς και η βαθμολογία που της έχει δώσει ο χρήστης που κάνει την αναζήτηση. Η μετρική που έφτιαξα (metric) συνυπολογίζει με τα ίδια βάρη και τα 3 παραπάνω και γίνεται ταξινόμηση των αποτελεσμάτων της αναζήτησης κατά φθίνουσα σειρά με βάση αυτή. Η παραπάνω τακτική δεν είναι και η καλύτερη δυνατή με βάση τα αποτελέσματα που έλαβα. Το βασικό της πρόβλημα ήταν ότι οι χρήστες έχουν βαθμολογήσει μόνο ένα μικρό υποσύνολο από τις ταινίες με αποτέλεσμα να υπάρχουν πολλές NaN τιμές. Έτσι ουσιαστικά δεν συμμετείχε σε ικανοποιητικό βαθμό στην τελική μετρική η αξιολόγηση που έχει δώσει ο χρήστης. Στην παρακάτω εικόνα φαίνεται το αποτέλεσμα που πήρα για την αναζήτηση toy story με

user_id = 1.

title	genres	iM25ratin	eanRatin	serRating	metric
Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	13.7628	3.87247	nan	17.6352
Toy Story 3 (2010)	Adventure|Animation|Children|Comedy|Fantasy|IMAX	12.4133	4.07143	nan	16.4847
Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy	12.4133	3.844	nan	16.2573
Toy Story of Terror (2013)	Animation|Children|Comedy	11.3048	4	nan	15.3048
Toy Soldiers (1991)	Action|Drama	8.26436	4	nan	12.2644
Toy, The (1982)	Comedy	8.26436	2.7	nan	10.9644
Brandon Teena Story, The (1998)	Documentary	4.51641	5	nan	9.51641
Philadelphia Story, The (1940)	Comedy|Drama|Romance	4.95926	4.35135	nan	9.31061
L.A. Story (1991)	Comedy|Romance	5.49841	3.69231	nan	9.19072
Christmas Story, A (1983)	Children|Comedy	4.95926	4.04667	nan	9.00593
West Side Story (1961)	Drama|Musical|Romance	4.95926	3.98889	nan	8.94815
Tokyo Story (Tôkyô monogatari) (1953)	Drama	4.51641	4.25	nan	8.76641
Greatest Story Ever Told, The (1965)	Drama	4.14616	4.5	nan	8.64616
Soldier's Story, A (1984)	Drama	4.95926	3.66667	nan	8.62593
Straight Story, The (1999)	Adventure|Drama	4.95926	3.57895	nan	8.53821
Palm Beach Story, The (1942)	Comedy	4.51641	4	nan	8.51641
Wristcutters: A Love Story (2006)	Drama|Fantasy|Romance	4.51641	4	nan	8.51641
Cinderella Story, A (2004)	Comedy|Romance	4.95926	3.5	nan	8.45926
New Police Story (Xin jing cha gu shi) (2004)	Action|Crime|Drama	3.32775	5	nan	8.32775
Buddy Holly Story, The (1978)	Drama	4.51641	3.75	nan	8.26641
NeverEnding Story, The (1984)	Adventure|Children|Fantasy	4.95926	3.30263	nan	8.26189
Love Story (1970)	Drama|Romance	5.49841	2.75	nan	8.24841
Glenn Miller Story, The (1953)	Drama	4.51641	3.66667	nan	8.18307
Family Band: The Cowsills Story (2011)	Documentary	4.14616	4	nan	8.14616
Not Quite Hollywood: The Wild, Untold Story of Ozploitation! (2008)	Documentary	3.12231	5	nan	8.12231

- ***<u>Ερώτημα 3</u>***: Στο ερώτημα αυτό μας ζητείται να επιλύσουμε το πρόβλημα που παρατηρήθηκε στο προηγούμενο ερώτημα χωρίζοντας τους χρήστες σε συστάδες (clusters) σύμφωνα με τον τρόπο που βαθμολογούν. Αρχικά, αφού φορτώσω τα δεδομένα σε dataframes, υπολογίζω για κάθε χρήστη τον μέσο όρο κάθε είδους (genre) οπότε παράγεται ένα dataframe 671 X 20 (πλήθος χρηστών X πλήθος ειδών). Επειδή όμως υπάρχουν πολλές NaN τιμές δηλαδή είδη που δεν έχουν βαθμολογηθεί από κάποιους χρήστες, αντικαθιστούμε αυτές τις τιμές με τον μέσο όρο του αντίστοιχου είδους. Στο παραπάνω σύνολο δεδομένων που προέκυψε (df4) εφαρμόζω τον αλγόριθμο KMeans για K = 8 και συσταδοποιώ τους χρήστες. Στην συνέχεια βρίσκω την μέση βαθμολογία κάθε συστάδας και με αυτή αντικαθιστώ τα NaN στο df8. Τέλος η μετρική ομοιότητας elasticsearch , της μέσης βαθμολογίας και η μετρική με την οποία θα γίνει τελικά η ταξινόμηση υπολογίζονται όπως στο ερώτημα 2. Με την μεθοδολογία που ακολουθήθηκε υπάρχει βελτίωση όντως αφού οι περιπτώσεις στις οποίες δεν υπάρχει βαθμολογία χρήστη για κάποια ταινία έχουν μειωθεί αισθητά. Στην παρακάτω εικόνα φαίνεται το

αποτέλεσμα που πήρα για την αναζήτηση toy story με user_id = 1.

title	genres	ıM25ratin	ıeanRatin	ıserRatinc	metric
Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	13.7628	3.87247	2	19.6352
Toy Story 3 (2010)	Adventure|Animation|Children|Comedy|Fantasy|IMAX	12.4133	4.07143	2	18.4847
Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy	12.4133	3.844	1.25	17.5073
Toy Story of Terror (2013)	Animation|Children|Comedy	11.3048	4	nan	15.3048
L.A. Story (1991)	Comedy|Romance	5.49841	3.69231	4.5	13.6907
Hachiko: A Dog's Story (a.k.a. Hachi: A Dog's Tale) (2009)	Drama	3.12231	4.5	5	12.6223
Christmas Story, A (1983)	Children|Comedy	4.95926	4.04667	3.5	12.5059
Toy, The (1982)	Comedy	8.26436	2.7	1.5	12.4644
Toy Soldiers (1991)	Action|Drama	8.26436	4	nan	12.2644
NeverEnding Story, The (1984)	Adventure|Children|Fantasy	4.95926	3.30263	3	11.2619
Anvil! The Story of Anvil (2008)	Documentary|Musical	4.14616	3.125	3.5	10.7712
Walk Hard: The Dewey Cox Story (2007)	Comedy|Musical	3.83202	3.75	3	10.582
Perfume: The Story of a Murderer (2006)	Crime|Drama|Thriller	3.83202	3.22222	3	10.0542
Spider Baby or, The Maddest Story Ever Told (Spider Baby) (1968)	Comedy|Horror	2.94077	3.5	3.5	9.94077
Brandon Teena Story, The (1998)	Documentary	4.51641	5	nan	9.51641
Philadelphia Story, The (1940)	Comedy|Drama|Romance	4.95926	4.35135	nan	9.31061
Dodgeball: A True Underdog Story (2004)	Comedy	4.14616	3.33784	1.5	8.984
West Side Story (1961)	Drama|Musical|Romance	4.95926	3.98889	nan	8.94815
Tokyo Story (Tôkyô monogatari) (1953)	Drama	4.51641	4.25	nan	8.76641
True Story (2015)	Drama|Mystery|Thriller	5.49841	2.16667	1	8.66507
Greatest Story Ever Told, The (1965)	Drama	4.14616	4.5	nan	8.64616
Soldier's Story, A (1984)	Drama	4.95926	3.66667	nan	8.62593
Straight Story, The (1999)	Adventure|Drama	4.95926	3.57895	nan	8.53821
Palm Beach Story, The (1942)	Comedy	4.51641	4	nan	8.51641
Wristcutters: A Love Story (2006)	Drama|Fantasy|Romance	4.51641	4	nan	8.51641

- ***Ερώτημα 4*** : Στο ερώτημα αυτό μας ζητείται να συμπληρώσουμε τις βαθμολογίες που λείπουν με την βοήθεια νευρωνικού δικτύου με το οποίο θα προσπαθήσουμε να προβλέψουμε τις βαθμολογίες που θα έβαζε ο χρήστης στις ταινίες που δεν έχει βαθμολογήσει. Αρχικά η μετρική ομοιότητας elasticsearch , η μέση βαθμολογία , και η μετρική με την οποία θα γίνει τελικά η ταξινόμηση υπολογίζονται όπως στα προηγούμενα ερωτήματα. Στην συνέχεια αρχίζω την διαδικασία για να προετοιμάσω το νευρωνικό δίκτυο και να βρω τις βαθμολογίες χρήστη για κάθε ταινία. Το νευρωνικό δίκτυο που θέλω να φτιάξω ουσιαστικά θα παίρνει σαν είσοδο το διάνυσμα μιας ταινίας και θα προβλέπει στην έξοδο τι βαθμολογία θα της έδινε ο χρήστης. Πρώτο βήμα είναι η μετατροπή των τίτλων σε word embeddings. Χρησιμοποιώντας την doc2vec βιβλιοθήκη μπορώ να παράξω ένα διάνυσμα από ένα document (document είναι οποιαδήποτε αλληλουχία λέξεων π.χ. τίτλος ταινίας). Στην συγκεκριμένη περίπτωση το διάνυσμα που παράγω επιλέγω να έχει μέγεθος 200. Μετά την εκπαίδευση του μοντέλου

και αφού πάρω την λίστα με τα word embeddings όλων των τίτλων πρέπει να μετατρέψω στην μορφή one hot encoding τα είδη των ταινιών, πράγμα που επιτυγχάνεται με την βοήθεια συναρτήσεων της pandas. Έτσι το input vector του νευρωνικού δικτύου έχει μέγεθος 220 (200 από το word embedding και 20 από το one hot encoding -υπάρχουν 20 διαφορετικά είδη). Πέρα όμως από το διάνυσμα εισόδου που ετοιμάστηκε είτε για την εκπαίδευση (x_train) είτε για πρόβλεψη (x_test), πρέπει να φτιαχτεί και το διάνυσμα εξόδου. Οι πιθανές βαθμολογίες είναι από 0.5 έως 5, οπότε υπάρχουν 10 πιθανές κλάσεις που καθεμιά αντιστοιχεί σε μία βαθμολογία οπότε η έξοδος θα είναι ένα διάνυσμα 10 θέσεων. Για αυτό τον λόγο μετατρέπω τις βαθμολογίες που έχει δώσει ο χρήστης στην μορφή one hot encoding δημιουργώντας με αυτό τον τρόπο το y_train. Στην συνέχεια δημιουργώ ένα νευρωνικό δίκτυο με 2 κρυφά επίπεδα που αποτελούνται από 32 νευρώνες και το εκπαιδεύω με τα x_train, y_train. Τέλος προβλέπω τις κλάσεις για κάθε διάνυσμα του x_test παράγοντας έτσι το y_test. Με την βοήθεια ενός λεξικού που έφτιαξα είμαι σε θέση να αντιστοιχήσω κάθε κλάση στην βαθμολογία της. Με τον παραπάνω τρόπο καταφέρνω να γεμίσω τις NaN και να παραχθεί το df9. Με το νευρωνικό προκύπτει ότι δεν υπάρχει κανένα NaN στη στήλη userRating πλέον οπότε κρίνεται καλύτερος από όλους τους παραπάνω τρόπους. Στην παρακάτω εικόνα φαίνεται το

αποτέλεσμα που πήρα για την αναζήτηση toy story με user_id = 1.

title	genres	IM25ratin	eanRatin	serRatinc	metric
Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	13.7628	3.87247	2	19.6352
Toy Story 3 (2010)	Adventure|Animation|Children|Comedy|Fantasy|IMAX	12.4133	4.07143	2	18.4847
Toy Story of Terror (2013)	Animation|Children|Comedy	11.3048	4	3	18.3048
Toy Story 2 (1999)	Adventure|Animation|Children|Comedy|Fantasy	12.4133	3.844	2	18.2573
Toy Soldiers (1991)	Action|Drama	8.26436	4	4	16.2644
Toy, The (1982)	Comedy	8.26436	2.7	3	13.9644
Brandon Teena Story, The (1998)	Documentary	4.51641	5	4	13.5164
New Police Story (Xin jing cha gu shi) (2004)	Action|Crime|Drama	3.32775	5	4	12.3278
Philadelphia Story, The (1940)	Comedy|Drama|Romance	4.95926	4.35135	3	12.3106
L.A. Story (1991)	Comedy|Romance	5.49841	3.69231	3	12.1907
Family Band: The Cowsills Story (2011)	Documentary	4.14616	4	4	12.1462
Not Quite Hollywood: The Wild, Untold Story of Ozploitation! (2008)	Documentary	3.12231	5	4	12.1223
Tillman Story, The (2010)	Documentary	4.95926	3	4	11.9593
Police Story (Ging chaat goo si) (1985)	Action|Comedy|Crime|Thriller	3.83202	4	4	11.832
Redemption: The Stan Tookie Williams Story (2004)	Crime|Documentary|Drama	3.83202	4	4	11.832
Chinese Ghost Story, A (Sinnui yauwan) (1987)	Action|Fantasy|Horror|Romance	3.83202	4.25	3.5	11.582
Palm Beach Story, The (1942)	Comedy	4.51641	4	3	11.5164
James Dean Story, The (1957)	Documentary	4.51641	3	4	11.5164
Cinderella Story, A (2004)	Comedy|Romance	4.95926	3.5	3	11.4593
Dragon: The Bruce Lee Story (1993)	Action|Drama	4.14616	3.25	4	11.3962
Riki-Oh: The Story of Ricky (Lik Wong) (1991)	Action|Crime|Thriller	3.32775	4	4	11.3278
Tokyo Story (Tôkyô monogatari) (1953)	Drama	4.51641	4.25	2.5	11.2664
Benny Goodman Story, The (1955)	Drama|Musical	4.51641	3.5	3	11.0164
Christmas Story, A (1983)	Children|Comedy	4.95926	4.04667	2	11.0059
West Side Story (1961)	Drama|Musical|Romance	4.95926	3.98889	2	10.9482