

# Συστήματα Διαχείρισης Δεδομένων Μεγάλης Κλίμακας

## Αναλυτική Αναφορά Εργασίας

**Ομάδα 10: Αλέξανδρος Βεντούρας [itp21106@hua.gr](mailto:itp21106@hua.gr)**

### 1. Εισαγωγή:

Η παρούσα εργασία εκπονήθηκε σε Java με Maven, σε λειτουργικό σύστημα Windows 10, με χρήση του Eclipse IDE. Στο παρόν αρχείο αναλύεται ο παραχθείς κώδικας, καθώς και τα διάφορα αποτελέσματα που προκύπτουν από την εκτέλεσή του. Για τις οδηγίες εκτέλεσης του προγράμματος, ανατρέξτε στο αρχείο `readme.docx`.

### 2. Πρώτο Μέρος:

Στο πρώτο κομμάτι της εργασίας (αρχείο `MoviesRDDOnly.java`), οι απαντήσεις στα ζητούμενα της εργασίας, υλοποιήθηκαν μόνο με `JavaRDDs` και `JavaPairRDDs`. Αρχικά υπάρχουν ορισμένες κλάσεις, οι οποίες χρησιμοποιούνται προκειμένου να γίνει ένα parsing των δεδομένων, σε μορφή κατάλληλη για περαιτέρω επεξεργασία. Αυτός ο τρόπος γραφής προτιμήθηκε έναντι της χρήσης inline συναρτήσεων, καθόσον γνώμη του γράφοντος είναι ότι έτσι αυξάνεται η αναγνωσιμότητα του κώδικα.

Αρχικά (γραμμές 29-37) υπάρχει η κλάση `SplitRatings`, η οποία χρησιμεύει στην επεξεργασία των δεδομένων που βρίσκονται στο αρχείο `ratings.dat`. Συγκεκριμένα, η περιεχόμενη της συνάρτηση παίρνει ως είσοδο ένα `String` (το οποίο δεν αποτελεί παρά μια γραμμή του αρχείου), και βάσει του διαχωριστικού συμβόλου (`::`), εξάγονται το δεύτερο και το τρίτο στοιχείο της γραμμής, τα οποία εν προκειμένω είναι το `id` της αξιολογούμενης ταινίας, και η αξιολόγηση. Αυτά τα στοιχεία μετατρέπονται σε `Integer` και `Float`, αντίστοιχα, εισάγονται σε ένα νέο `Tuple2`, και επιστρέφονται.

Έπειτα (γραμμές 39-46) υπάρχει η κλάση `SplitMovies`, η οποία χρησιμεύει στην επεξεργασία των δεδομένων που βρίσκονται στο αρχείο `movies.dat`. Παρόμοια με πριν, εξάγονται από την είσοδο της συνάρτησης, το πρώτο και το δεύτερο στοιχείο της γραμμής, τα οποία εν προκειμένω είναι το `id` της ταινίας και ο τίτλος της. Το πρώτο στοιχείο μετατρέπεται σε `Integer`, (το δεύτερο παραμένει ένα `String`), αντίστοιχα, εισάγονται σε ένα νέο `Tuple2`, και επιστρέφονται.

Στη συνέχεια, (γραμμές 48-66) υπάρχει η κλάση `Calculate Average`, η οποία χρησιμεύει στον υπολογισμό του μέσου όρου των `ratings` για κάθε ταινία. Συγκεκριμένα, η περιεχόμενη της συνάρτηση παίρνει ως είσοδο ένα `Tuple2<Integer, Iterable<Float>>` (το οποίο περιέχει το `id` της ταινίας, και ένα `Iterable` με όλες της αξιολογήσεις της). Από αυτή τη δομή, υπολογίζεται ο μέσος όρος των στοιχείων του `Iterable`, και επιστρέφεται μαζί με το `id` της ταινίας, σε ένα νέο αντικείμενο `Tuple2`, τύπου `<Integer, Float>`.

Τέλος, (γραμμές 69-75) υπάρχει η κλάση `ConvertToSimplePair`, η οποία χρησιμεύει στη μετατροπή μίας δομής `Tuple2<String,Tuple2<String,Float>>`, σε `Tuple2<String,Float>`. Συγκεκριμένα, η περιεχόμενη της συνάρτηση παίρνει ως είσοδο την πρώτη δομή, και επιστρέφει μόνο το δεύτερο στοιχείο της, τύπου `Tuple2<String,Float>`.

Στις επόμενες δύο γραμμές (77-78), γίνεται αρχικοποίηση του `SparkConference` και του `JavaSparkContext`, όπως υποδείχθηκε στο εργαστήριο, και εν συνεχεία αρχικοποιούνται τα δύο RDDs που περιέχουν τις γραμμές των αρχείων `ratings.dat` και `movies.dat`.

#### I) Πρώτο υποερώτημα:

Έπειτα, ξεκινάει το κομμάτι της εργασίας το οποίο αφορά το πρώτο υποερώτημα (25 περισσότερα αξιολογημένες ταινίες – γραμμές 84 - 129). Καταρχήν, εξάγονται τα χρήσιμα πεδία για τα `ratings` και τα `movies` (γραμμές 85 – 89), σε δύο `JavaPairRDDs`, με τη βοήθεια δύο εκ των προαναφερθεισών συναρτήσεων.

Στις επόμενες γραμμές γίνεται η διαδικασία του `map` και του `reduce`, έτσι ώστε να εξαχθεί τον αριθμό των φορών που έχει αξιολογηθεί κάθε ταινία (γραμμή 93), και στη συνέχεια να προκύψει το συγκεκριμένο άθροισμα (γραμμή 94) για κάθε ταινία.

Στη συνέχεια, προκειμένου να ταξινομηθούν σε φθίνουσα σειρά οι ταινίες (γραμμή 101), βάσει του πόσες φορές έχει κάθε μία αξιολογηθεί, πρέπει να αντιστραφεί το `key` και το `value`, έτσι ώστε από `<movieId,#ofAppearances>` να γίνει `< ofAppearances,movieId>`. Αυτό γίνεται στις γραμμές 96 – 98.

Ακολούθως, εξάγονται τα 25 πρώτα στοιχεία αυτού του `JavaPairRDD`, και τοποθετούνται σε μία λίστα (η οποία περιέχει αντικείμενα τύπου `Tuple2<Integer,Integer>` - γραμμή 105). Αυτό γίνεται γιατί δε βρέθηκε τρόπος να εξάγονται τα  $n$  πρώτα στοιχεία ενός `JavaPairRDD`, με τη μορφή ενός άλλου `JavaPairRDD`. Έπειτα, ξαναποθηκεύονται με χρήση της `parallelizePairs()` σε ένα `JavaPairRDD` τα στοιχεία αυτά, και ξανααντιστρέφονται το `key` και το `value` (γραμμές 107 – 112).

Στη συνέχεια, εκτελείται το `join` (με το `default inner join`) μεταξύ των δύο `PairRDDs` (γραμμές 115-6), έτσι ώστε να προκύψει το ζητούμενο υποσύνολο των αρχικών δεδομένων, και να συνδεθούν τα πλήθη αξιολογήσεων ταινιών με τους τίτλους τους. Έπειτα, στις γραμμές 118 – 129, τυπώνεται το αποτέλεσμα και αποθηκεύεται σε ένα `txt` (για τα αποτελέσματα της εκτέλεσης, βλέπετε επόμενη παράγραφο της αναφοράς).

#### II) Δεύτερο υποερώτημα:

Έπειτα, υπάρχει η υλοποίηση της λύσης του δεύτερου υποερωτήματος (κωμωδίες με έστω και μία βαθμολογία τουλάχιστον 3, γραμμές 131 – 152). Εδώ, επιλέγονται από το RDD που περιέχει τις γραμμές του `movies.dat` αρχείου, μόνο όσες αφορούν κωμωδίες (γραμμή 133), και με τη βοήθεια της `SplitMovies()` επιλέγονται τα κατάλληλα πεδία (γραμμή 136). Στη συνέχεια, από το `JavaPairRDD` που περιέχει τα χρήσιμα πεδία των αξιολογήσεων, επιλέγονται μόνο όσες σειρές

αφορούν αξιολογήσεις ταινιών πάνω από ή ίσες με 3 (γραμμή 139), και εκτελείται το join μεταξύ των δύο PairRDDs (γραμμή 142), έτσι ώστε να προκύψει το ζητούμενο υποσύνολο των αρχικών δεδομένων, και να κρατηθούν μόνο οι αξιολογήσεις που αφορούν κωμωδίες.

Κατόπιν, μετριέται το πλήθος των διαφορετικών ids που βρίσκονται στο αποτέλεσμα του join (γραμμή 145), και έτσι προκύπτει το ζητούμενο του συγκεκριμένου υποερωτήματος (πόσες κωμωδίες έχουν βαθμολογηθεί έστω και μία φορά με βαθμό τουλάχιστον 3). Έπειτα, στις γραμμές 147 – 152, τυπώνεται το αποτέλεσμα και αποθηκεύεται σε ένα txt.

### III) Τρίτο υποερώτημα:

Έπειτα, υπάρχει η υλοποίηση της λύσης του τρίτου υποερωτήματος (top 10 ρομαντικές ταινίες, λαμβάνοντας υπόψη μόνο τα ratings που έγιναν Δεκέμβριο, γραμμές 155 – 217). Εδώ, επιλέγεται από το RDD που περιέχει τις γραμμές του movies.dat αρχείου, μόνο όσες αφορούν ρομαντικές ταινίες (γραμμή 157), και με τη βοήθεια της SplitMovies() επιλέγονται τα κατάλληλα πεδία (γραμμή 160). Στη συνέχεια, από το JavaPairRDD που περιέχει τα χρήσιμα πεδία των αξιολογήσεων, επιλέγονται μόνο όσες σειρές αφορούν αξιολογήσεις ταινιών που έγιναν έναν οποιονδήποτε Δεκέμβριο (γραμμές 163 - 167), και με τη βοήθεια της SplitRatings() επιλέγονται τα κατάλληλα πεδία (γραμμή 170). Για να εξαχθεί ο μήνας από το υπάρχον timestamp, γίνεται χρήση της συνάρτησης getDateFromTimestamp().

Έπειτα, εκτελείται μια groupByKey(), έτσι ώστε να αποκτηθεί ένα RDD που περιέχει το Id κάθε ταινίας και όλες τις δεκεμβριάτικες αξιολογήσεις που το αφορούν (γραμμή 173), και αμέσως μετά υπολογίζεται ο μέσος όρος όλων των αξιολογήσεων για κάθε ταινία (γραμμή 176). Ακολούθως, εκτελείται το join μεταξύ των δύο PairRDDs (γραμμή 142), έτσι ώστε να προκύψει το ζητούμενο υποσύνολο των αρχικών δεδομένων, και να μείνουν μόνο οι αξιολογήσεις που αφορούν ρομαντικές κωμωδίες και έγιναν έναν κάποιο Δεκέμβριο.

Στη συνέχεια, εκτελείται μια mapToPair με χρήση της κλάσης ConvertToSimplePair, προκειμένου να κρατηθούν μόνο τα ωφέλιμα στοιχεία από το join (την δυάδα <romanceld, romancdRating> - (γραμμή 183).

Στη συνέχεια, προκειμένου να ταξινομηθούν σε φθίνουσα σειρά τα id των ρομαντικών ταινιών (γραμμή 190), βάσει του ποια έχει μεγαλύτερο μέσο όρο αξιολογήσεων, πρέπει να αντιστραφεί το key και το value, έτσι ώστε από <romanceld, romancdRating> να γίνει < romancdRating, romanceld>. Αυτό γίνεται στις γραμμές 186-7.

Ακολούθως, λαμβάνονται τα 10 πρώτα στοιχεία αυτού του JavaPairRDD, και τοποθετούνται σε μία λίστα (η οποία περιέχει αντικείμενα τύπου Tuple2<Float,String> - γραμμή 193). Έπειτα, ξαναποθηκεύονται με χρήση της parallelizePairs() σε ένα JavaPairRDD τα στοιχεία αυτά, και ξανααντιστρέφονται το key και το value (γραμμές 195 – 200). Έπειτα, στις γραμμές 202 – 216, τυπώνεται το αποτέλεσμα και αποθηκεύεται σε ένα txt.

Τέλος, ακολουθούν οι υλοποιήσεις της συνάρτησης getDateFromTimestamp (γραμμές 226 – 230), η οποία χρησιμοποιεί την getDateFromTimestamp (γραμμές 219 – 224), προκειμένου να εξαχθεί η ημερομηνία από το timestamp που υπάρχει στα αρχεία. Στο τέλος του αρχείου, υπό τη μορφή comments, υπάρχουν διάφορα links προς sites τα οποία μου χρησίμευσαν έτσι ώστε να επιλύσω διάφορα προβλήματα τα οποία προέκυψαν κατά την συγγραφή του κώδικα.

#### IV) Αποτελέσματα:

Στο πρώτο υποερώτημα (25 περισσότερες αξιολογημένες ταινίες), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

```
|((1196,(23091,Star Wars: Episode V - The Empire Strikes Back (1980)))
(608,(23794,Fargo (1996)))
(296,(34864,Pulp Fiction (1994)))
(592,(26996,Batman (1989)))
(588,(23531,Aladdin (1992)))
(480,(32631,Jurassic Park (1993)))
(780,(26042,Independence Day (a.k.a. ID4) (1996)))
(380,(25381,True Lies (1994)))
(260,(28566,Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)))
(356,(34457,Forrest Gump (1994)))
(32,(24397,12 Monkeys (Twelve Monkeys) (1995)))
(1,(26449,Toy Story (1995)))
(589,(28948,Terminator 2: Judgment Day (1991)))
(457,(28951,Fugitive, The (1993)))
(593,(33668,Silence of the Lambs, The (1991)))
(377,(23748,Speed (1994)))
(150,(27035,Apollo 13 (1995)))
(50,(24037,Usual Suspects, The (1995)))
(318,(31126,Shawshank Redemption, The (1994)))
(110,(29154,Braveheart (1995)))
(590,(25912,Dances with Wolves (1990)))
(1210,(25098,Star Wars: Episode VI - Return of the Jedi (1983)))
(2571,(23229,Matrix, The (1999)))
(527,(25777,Schindler's List (1993)))
(47,(22521,Seven (a.k.a. Se7en) (1995)))
```

Στο δεύτερο υποερώτημα (κωμωδίες με έστω και μία βαθμολογία τουλάχιστον 3), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

```
|3680 comedies were rated with 3 or more points.
```

Στο τρίτο υποερώτημα (top 10 ρομαντικές ταινίες, λαμβάνοντας υπόψη μόνο τα ratings που έγιναν Δεκέμβριο), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

```
(Some Girls (a.k.a. Sisters) (1988),5.0)
(Parallel Sons (1995),5.0)
(Summertime (1955),4.8333335)
(Thousand Clowns, A (1965),4.6666665)
(Breathing Room (1996),4.642857)
(Captives (1994),4.631579)
(Crude Oasis, The (1995),4.5555553)
(Parineeta (2005),4.5)
(Puffy Chair, The (2006),4.5)
(Sometimes Happiness, Sometimes Sorrow (Kabhi Khushi Kabhie Gham) (2001),4.5)
```

### 3. Δεύτερο Μέρος:

Στο δεύτερο κομμάτι της εργασίας (αρχεία `BigDataDataframes.java`, `Movie.java` και `Rating.java`), έγινε η υλοποίηση με `JavaRDDs` και κυρίως, με `Dataframes`. Αναφορικά με τα δύο αρχεία κλάσεων, αυτά προστέθηκαν στο δεύτερο κομμάτι της εργασίας, καθώς η μέθοδος `createDataframe()`, απαιτεί να δηλωθεί η κλάση των δεδομένων που θα υπάρχουν μέσα στο `Dataset` που κατασκευάζει. Επομένως, δεν ήταν δυνατό να υπάρχουν τα δεδομένα με την πρότερη μορφή τους. Έτσι, αυτά τα δύο αρχεία προέκυψαν απλώς χρησιμοποιώντας τις έτοιμες φόρμες που το Eclipse παρέχει για τη δημιουργία κλάσεων. Η δε συνάρτηση που βρίσκεται στο αρχείο `Rating.java`, και υπολογίζει το μήνα του `rating`, είναι η ίδια ακριβώς με αυτή που βρίσκεται στο τέλος του κώδικα του προηγούμενου ερωτήματος.

Έτσι, η ανάλυση της υλοποίησης θα επικεντρωθεί στο αρχείο `BigDataDataframes.java`. Αρχικά, όπως και στο προηγούμενο κομμάτι της εργασίας, υπάρχουν δύο κλάσεις, οι οποίες χρησιμοποιούνται προκειμένου να γίνει ένα parsing των δεδομένων, σε μορφή κατάλληλη για περαιτέρω επεξεργασία.

Πρώτα, (γραμμές 25-34) υπάρχει η κλάση `ParseRatings`, η οποία χρησιμεύει στην επεξεργασία των δεδομένων που βρίσκονται στο αρχείο `ratings.dat`. Συγκεκριμένα, η περιεχόμενη της συνάρτηση παίρνει ως είσοδο ένα `String`, και από αυτό εξάγονται το δεύτερο, το τρίτο και το τέταρτο στοιχείο της γραμμής, τα οποία εν προκειμένω είναι το `id` της αξιολογούμενης ταινίας, η αξιολόγηση, και το `timestamp`. Τα δύο πρώτα στοιχεία μετατρέπονται σε `Integer` και `Float`, αντίστοιχα, και έπειτα και τα τρία μαζί χρησιμοποιούνται για να δημιουργηθεί ένα `instance` τύπου `Rating`, το οποίο και επιστρέφεται.

Έπειτα (γραμμές 36-45) υπάρχει η κλάση `SplitMovies`, η οποία χρησιμεύει στην επεξεργασία των δεδομένων που βρίσκονται στο αρχείο `movies.dat`. Παρόμοια με πριν, εξάγονται από την είσοδο της συνάρτησης, το πρώτο, το δεύτερο, και το τρίτο στοιχείο της γραμμής, τα οποία εν προκειμένω είναι το `id` της ταινίας, ο τίτλος της, και τα ή το είδος της. Το πρώτο στοιχείο μετατρέπεται σε `Integer`, (το δεύτερο και το τρίτο παραμένουν `String`), αντίστοιχα, και έπειτα και τα τρία μαζί χρησιμοποιούνται για να δημιουργηθεί ένα `instance` τύπου `Movie`, το οποίο και επιστρέφεται.

Στις επόμενες τρεις γραμμές (47-49), γίνεται αρχικοποίηση του SparkConference, του JavaSparkContext και του SparkSession, και εν συνεχεία αρχικοποιούνται τα δύο RDDs που περιέχουν τις γραμμές των αρχείων ratings.dat και movies.dat., γίνεται το parsing των σειρών αυτών ώστε να εξαχθούν τα χρήσιμα στοιχεία κάθε γραμμής (γραμμές 51 – 56), και τέλος δημιουργούνται τα δύο Datasets με το σύνολο των δεδομένων (γραμμές 58 – 60).

#### I) Πρώτο υποερώτημα:

Έπειτα, ξεκινάει το κομμάτι της εργασίας το οποίο αφορά το πρώτο υποερώτημα (25 περισσότερες αξιολογημένες ταινίες – γραμμές 62 - 93).

Καταρχάς, δημιουργείται ένα νέο Dataset, το οποίο περιέχει τα ids των ταινιών που έχουν αξιολογηθεί, ομαδοποιημένα μέσω μιας group by, και με επιπλέον συνάρτηση την count(), για να επιστραφεί το πλήθος των εμφανίσεων κάθε id. Έπειτα, τα δεδομένα ταξινομούνται με φθίνουσα σειρά βάσει του προαναφερθέντος πλήθους, και κρατάμε μόνο τα πρώτα 25 (γραμμές 63-4).

Στη συνέχεια, εκτελείται το join (με το default inner join) μεταξύ των δύο Datasets (γραμμές 66 - 67), έτσι ώστε να συνδεθούν τα πλήθη αξιολογήσεων ταινιών με τους τίτλους τους. Έπειτα, στις γραμμές 69 - 71, τυπώνεται το αποτέλεσμα και το σύστημα αποθηκεύει σε έναν φάκελο τα ίδια αυτά αποτελέσματα.

#### II) Δεύτερο υποερώτημα:

Έπειτα, υπάρχει η υλοποίηση της λύσης του δεύτερου υποερωτήματος (γραμμές 73 - 93). Εδώ, από το Dataset που περιέχει τα αντικείμενα τύπου Rating, επιλέγονται μόνο όσες σειρές αφορούν αξιολογήσεις ταινιών πάνω από ή ίσες με 3, αφού πρώτα γίνει ένα group by ανά movielfid, κρατώντας παράλληλα τη μέγιστη αξιολόγηση για κάθε ταινία (γραμμές 74-5). Στη συνέχεια, επιλέγονται από το Dataset που περιέχει τα αντικείμενα τύπου Movie, μόνο όσα αφορούν κωμωδίες (γραμμή 77), και εκτελείται το join μεταξύ των δύο Datasets (γραμμές 79 - 80), έτσι ώστε να προκύψει το ζητούμενο υποσύνολο των αρχικών δεδομένων, και να μείνουν μόνο οι αξιολογήσεις που αφορούν κωμωδίες.

Κατόπιν, μετράται το πλήθος των αντικειμένων που βρίσκονται στο αποτέλεσμα του join (γραμμή 82), και έτσι προκύπτει το ζητούμενο του συγκεκριμένου υποερωτήματος (πόσες κωμωδίες έχουν βαθμολογηθεί έστω και μία φορά με βαθμό τουλάχιστον 3). Έπειτα, στις γραμμές 84 - 93, τυπώνεται το αποτέλεσμα και αποθηκεύεται σε ένα txt.

#### III) Τρίτο υποερώτημα:

Έπειτα, υπάρχει η υλοποίηση της λύσης του τρίτου υποερωτήματος (γραμμές 96 - 109). Εδώ, από το Dataset που περιέχει τα αντικείμενα τύπου Rating, επιλέγονται μόνο όσα αφορούν αξιολογήσεις ταινιών που έγιναν έναν οποιονδήποτε Δεκέμβριο, και γίνεται group by βάσει του movielfid, κρατώντας το μέσο όρο των αξιολογήσεων για κάθε συγκεκριμένη ταινία (γραμμές 97

– 98). Στη συνέχεια, επιλέγονται από το Dataset που περιέχει τα αντικείμενα τύπου Movie, μόνο όσα αφορούν ρομαντικές ταινίες (γραμμή 100).

Ακολούθως, εκτελείται το join μεταξύ των δύο Datasets (γραμμές 102-3), έτσι ώστε να προκύψει το ζητούμενο υποσύνολο των αρχικών δεδομένων, και να μείνουν μόνο οι αξιολογήσεις που αφορούν ρομαντικές κωμωδίες και έγιναν έναν κάποιο Δεκέμβριο.

Τέλος, τα δεδομένα ταξινομούνται με φθίνουσα σειρά βάσει του μέσου όρου των αξιολογήσεων (γραμμή 105), και, στις γραμμές 107-9, επιλέγονται οι δέκα πρώτες και τυπώνεται το αποτέλεσμα, το οποίο το σύστημα αποθηκεύει σε έναν φάκελο.

#### IV) Τέταρτο υποερώτημα:

Έπειτα, υπάρχει η υλοποίηση της λύσης του τέταρτου υποερωτήματος (γραμμές 112 - 124). Εδώ, από το Dataset που περιέχει τα αντικείμενα τύπου Rating, επιλέγονται μόνο όσα αφορούν αξιολογήσεις ταινιών που έγιναν έναν οποιονδήποτε Δεκέμβριο, και εκτελείται group by, με επιπλέον συνάρτηση την count(), για να επιστραφεί το πλήθος των εμφανίσεων κάθε movieId, μαζί με αυτό. Στη συνέχεια, ταξινομούνται με φθίνουσα σειρά τα δεδομένα βάσει του πλήθους των αξιολογήσεων, και επιλέγονται τα 10 πρώτα (γραμμές 114-5).

Μετά, εκτελείται το join μεταξύ των δύο Datasets (γραμμές 117-8), έτσι ώστε να συνδεθούν τα πλήθη αξιολογήσεων ταινιών με τους τίτλους τους. Τέλος, στις γραμμές 120 – 122 τυπώνεται το αποτέλεσμα, το οποίο το σύστημα αποθηκεύει σε έναν φάκελο.

#### IV) Αποτελέσματα:

Στο πρώτο	Toy Story (1995)
υποερώτημα (25	12 Monkeys (Twelve Monkeys) (1995)
περισσότερο	Seven (a.k.a. Se7en) (1995)
αξιολογημένες	Usual Suspects, The (1995)
ταινίες), ο υπό	Braveheart (1995)
ανάλυση κώδικας	Apollo 13 (1995)
παράγει τα εξής	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
αποτελέσματα:	Pulp Fiction (1994)
	Shawshank Redemption, The (1994)
	Forrest Gump (1994)
	Speed (1994)
	True Lies (1994)
	Fugitive, The (1993)
	Jurassic Park (1993)
	Schindler's List (1993)
	Aladdin (1992)
	Terminator 2: Judgment Day (1991)
	Dances with Wolves (1990)
	Batman (1989)
	Silence of the Lambs, The (1991)
	Fargo (1996)
	Independence Day (a.k.a. ID4) (1996)
	Star Wars: Episode V - The Empire Strikes Back (1980)
	Star Wars: Episode VI - Return of the Jedi (1983)
	Matrix, The (1999)

Στο δεύτερο υποερώτημα (κωμωδίες με έστω και μία βαθμολογία τουλάχιστον 3), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

---

3680 comedies were rated with 3 or more points.

Στο τρίτο υποερώτημα (top 10 ρομαντικές ταινίες, λαμβάνοντας υπόψη μόνο τα ratings που έγιναν Δεκέμβριο), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

Some Girls (a.k.a. Sisters) (1988)  
Parallel Sons (1995)  
Summertime (1955)  
Thousand Clowns, A (1965)  
Breathing Room (1996)  
Captives (1994)  
Crude Oasis, The (1995)  
Parineeta (2005)  
Sometimes Happiness, Sometimes Sorrow (Kabhi Khushi Kabhie Gham) (2001)  
Bells Are Ringing (1960)

Στο τέταρτο υποερώτημα (ρομαντικές ταινίες με τις περισσότερες αξιολογήσεις, λαμβάνοντας υπόψη μόνο τα ratings που έγιναν Δεκέμβριο), ο υπό ανάλυση κώδικας παράγει τα εξής αποτελέσματα:

---

Toy Story (1995)  
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)  
Pulp Fiction (1994)  
Jurassic Park (1993)  
Silence of the Lambs, The (1991)  
Fargo (1996)  
Independence Day (a.k.a. ID4) (1996)  
Star Wars: Episode V - The Empire Strikes Back (1980)  
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)  
Star Wars: Episode VI - Return of the Jedi (1983)