Οδηγίες

Στο φάκελο με το όνομα pp2_iis21098 υπάρει τα παρακάτω αρχεία:

- ratings.csv το οποίο είναι το αρχείο με τα αρχικά δεδομένα.
- recommender_system.py το οποίο είναι ο κώδικας του recomended system
- weighted_average.csv το οποίο είναι το αρχείο που κρατάει τα αποτελέσματα για 5 διαφορετικές τιμές του k με την μέθοδο **1.**
- adj.csv το οποίο είναι το αρχείο που κρατάει τα αποτελέσματα για 5 διαφορετικές τιμές του k με την μέθοδο **2.**
- common_users.csv το οποίο είναι το αρχείο που κρατάει τα αποτελέσματα για 5 διαφορετικές τιμές του k με την μέθοδο **3.**
- varians.csv το οποίο είναι το αρχείο που κρατάει τα αποτελέσματα για 5 διαφορετικές τιμές του k με την μέθοδο **4.**
- display_results.py το οποίο χρησιμοποιείται για τον υπολογισμό των μέσων όρων των αποτελεσμάτων για κάθε μέθοδο, καθώς και την οπτικοποιήση τους.

Το recommender_system.py έχει τρέξει για 5 διαφορετικές τιμές του k και τα δεδομένα αποθηκεύονται στα αρχεία που αναφέρθηκαν πιο πριν. Για να γίνει αυτό θα πρέπει στον ίδιο φάκελο που είναι και το παραπάνω πρόγραμμα να υπάρχουν τα αρχεία αυτά με τα ίδια ονόματα και να έχουν ως πρώτη σειρά την παρακάτω:

k,MAE,PasP,PasN,NasN,NasP,MAP,MAR
όπου

- k = ο αριθμός των πιο κοντινων γειτονων
- MAE = Mean Absolute Error
- PasP = Positives predicted as Positives
- PasN = Positives predicted as Negatives
- NasN = Negatives predicted as Negatives
- NasP = Negatives predicted as Positives
- MAP = Macro Average Precision
- MAR = Macro Average Recall

Το display_results.py πάιρνει τα δεδομένα που είναι αποθηκευμένα στα αρχεία που αναφέρθηκαν προηγουμένως και αφού υπολογίσει τους μέσους όρους για τα αρχεία ξεκινάει η οπτικοποίηση. Για να γίνει αυτό πρέπει προφανώς το παραπάνω πρόγραμμα να είναι στον ίδιο φάκελο με τα αρχεία.

Πρώτα θα εμφανιστεί το confusion matrix για την μέθοδο **1**, αφού κλείσετε αυτό το παράθυρο θα εμφανιστεί το confusion matrix για την μέθοδο **2**, αφού κλείσετε αυτό το παράθυρο θα εμφανιστεί το confusion matrix για την μέθοδο **3**, αφού κλείσετε αυτό το παράθυρο θα εμφανιστεί το confusion matrix για την μέθοδο **4** και τέλος αφού κλείσετε και αυτό το παράθυρο θα εμφανιστεί ένα διάγραμμα το οποίο θα δείχνει τα υπόλοιπα αποτελέσματα για όλες τις μεθόδους.

Περιγραφή Του Κώδικα

Στην αρχή το csv αποθηκεύεται ως ένα data frame από το οποίο τυχαία (Αλλά κάθε φορά τα ίδια) επιλέγονται τα training δεδομένα και τα testing δεδομένα.

Ακόμη με βάση το αρχικό data frame δημιουργείται μια μήτρα με το όνομα user_movie_matrix που έχει ως rows τα userId (δηλαδή τους users), ως columns τα movieId (δηλαδή τις ταινίες) και ως values τα αντίστοιχα ratings ενός user σε μια ταινία. Επίσης δημιουργείται και μια δεύτερη μήτρα η οποία είναι ακριβώς ίδια με την προηγούμενη και ονομάζεται actual_movie_matrix.

Στην συνέχεια γίνεται σε και τις δύο μήτρες αντικατάσταση των ratings που υπάρχουν στο test set με τις τιμές nan (not a number).

Το επόμενο βήμα είναι στην μήτρα user_movie_matrix να αφαιρέσω για κάθε σειρά τον μέσο όρο των μη nan ratings για κάθε μη nan rating της σειράς (πρώτο βήμα για Pearson).

Έπειτα υπολογίζεται και αποθηκεύεται το cosine similarity για κάθε ταινία με κάθε άλλη ταινία (ο υπολογισμός γίνεται με βάση την μήτρα user_movie_matrix).

Στο επόμενο βήμα για κάθε αντικείμενο στο test set (Σχήμα αντικειμένου: userId, moviesId, rating) υπολογίζονται και αποθηκεύονται οι k κοντινότερες ταινίες για τις οποίες ο χρήστης έχει δώσει rating.

Μετά γίνονται τα 4 διαφορετικά prediction με βάση τις μεθόδους 1, 2, 3, 4.
Για την μέθοδο:

- 3:
  com_count = count_common_users(actual_movie_matrix, movieId, nearest_movie_id)
  user_rating = actual_movie_matrix.loc[userId, nearest_movie_id]
  common_users_weighted_sum += com_count * user_rating
  common_users_sum_of_weights += com_count
    - Όπου com_count είναι το αποτέλεσμα που επιστρέφει η συνάρτηση count_common_users και είναι το σύνολο των users που έχουν βαθμολογήσει και τις 2 ταινίες
    - user_rating είναι η βαθμολογία που έχει δώσει ο χρήστης για την τρέχουσα κοντινή ταινία και την πάιρνουμε από την actual_movie_matrix καθός οι βαθμολογίες τις δεν έχουν αλλάξει.
    - common_users_weighted_sum είναι ο αριθμητής για την πρόβλεψη
    - common_users_sum_of_weights είναι ο παρονομαστής για την πρόβλεψη

- 4:
  variance = np.var(movie_ratings)
  user_rating = actual_movie_matrix.loc[userId, nearest_movie_id]
  var_common_users_weighted_sum += variance * user_rating
  var_common_users_sum_of_weights += variance
    - Όπου variance είναι το αποτέλεσμα της np.var(movie_ratings) η οποία υπολογίζει το viariance της τρέχουσας κοντινής ταινίας
    - user_rating είναι η βαθμολογία που έχει δώσει ο χρήστης για την τρέχουσα κοντινή ταινία και την πάιρνουμε από την actual_movie_matrix καθός οι βαθμολογίες τις δεν έχουν αλλάξει.
    - var_common_users_weighted_sum είναι ο αριθμητής για την πρόβλεψη
    - var_common_users_sum_of_weights είναι ο παρονομαστής για την πρόβλεψη

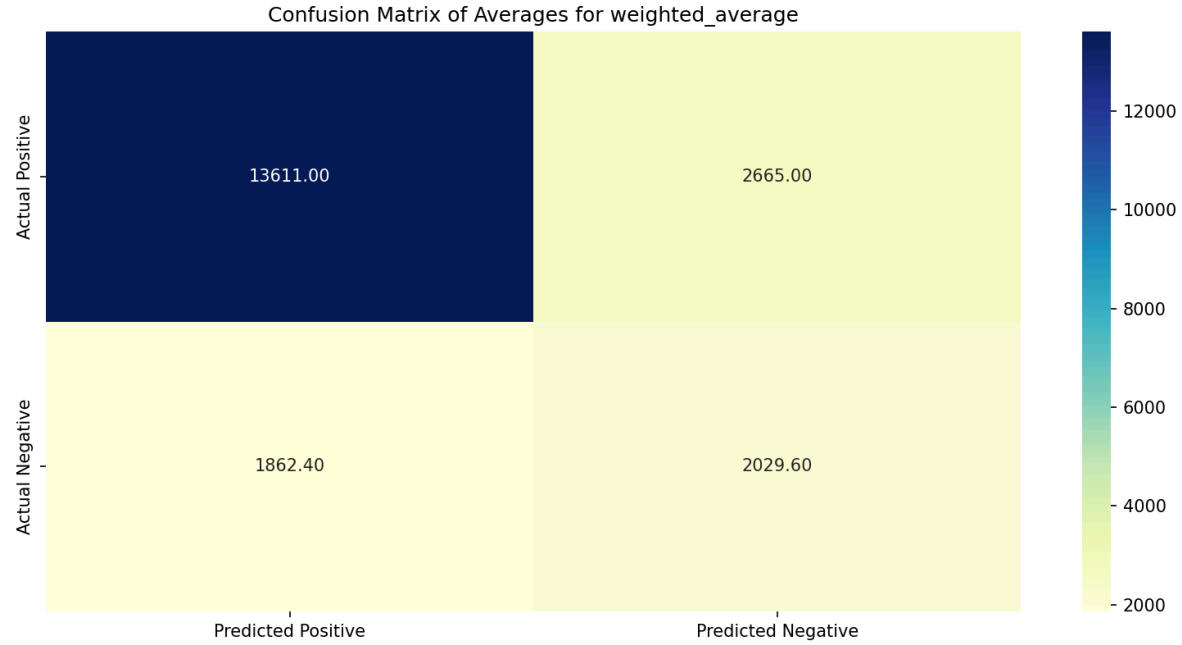
Για κάθε μια από τις τέσσερις παραπάνω μεθοδούς αν η πρόβλεψη είναι μικρότερη του 1 αποθηκευεται ως 1 και αν είναι μεγαλύτερη του 5 αποθηκεύεται ως 5.

Και στο τελευταίο βήμα υπολογίζονται με βάση τις προβλέψεις οι παρακάτω μετρικές για κάθε μέθοδο:

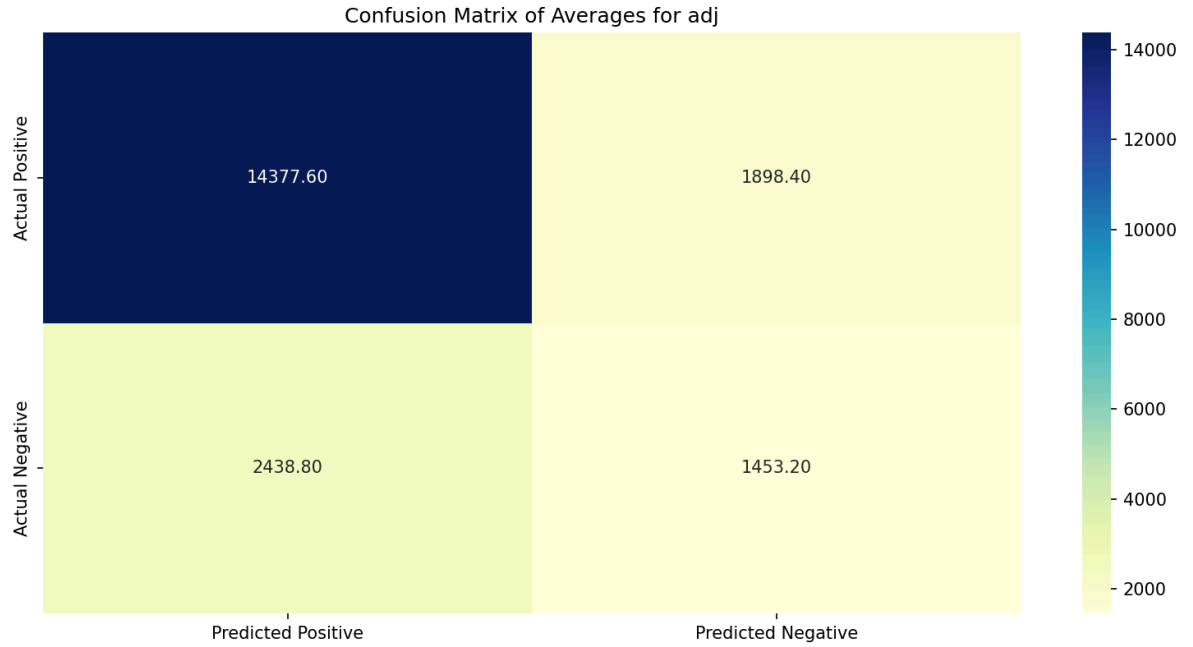
- Mean Absolute Error
- Macro Average Precision
- Macro Average Recall
- Confusion Matrix

Αποτελέσματα

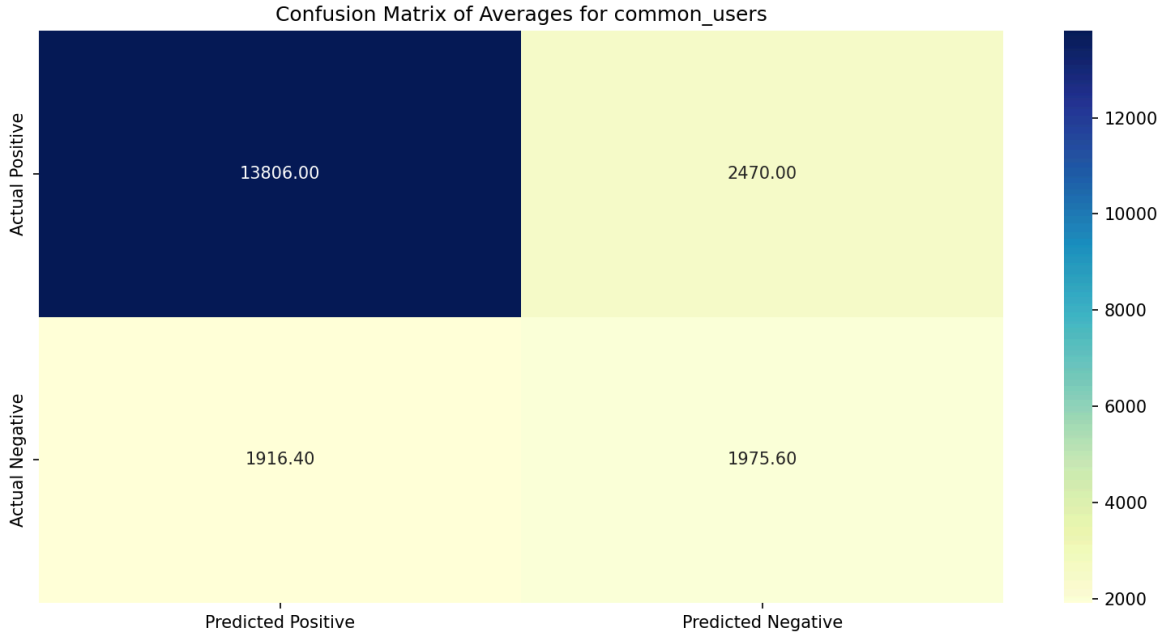
Confusion Matrix για την μέθοδο 1:



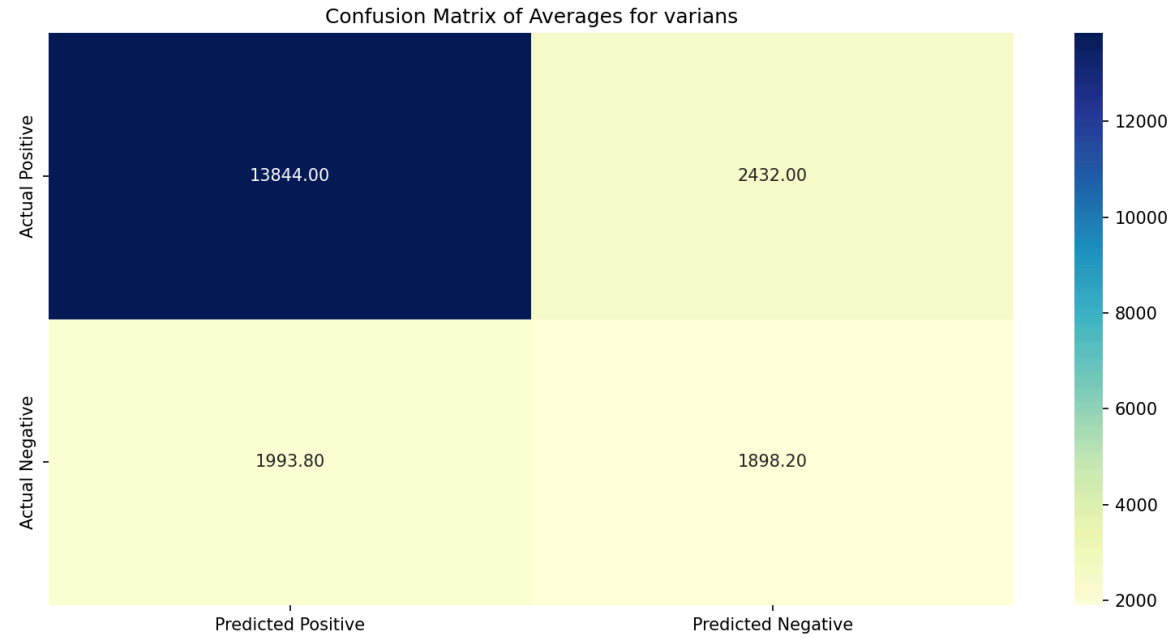
Confusion Matrix για την μέθοδο 2:



Confusion Matrix για την μέθοδο 3:



Confusion Matrix για την μέθοδο 4:



Σχεδιάγραμμα για MAE, MAR, MAP κάθε μοντέλου: