

RAPPORT PROJET LANGAGE R



**Tuteur académique**

Mr LALANNE

**Etablissement / Formation**

ESME SUDRIA / MASTER 2

Mardi 16 Janvier 2018

MANKOURI Jalil

YOUNES Théo

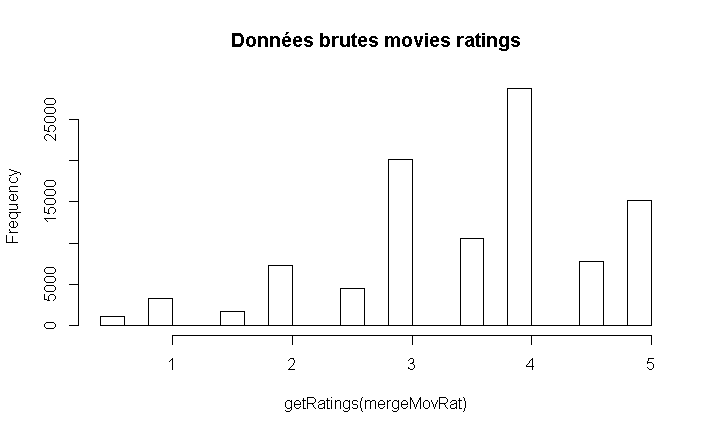
A travers ce mini compte-rendu, nous verrons comment créer un moteur de recommandation. Nous utiliserons des données qui font des recommandations sur les films. Nous utiliserons le paquet "recommenderlab" pour construire plusieurs moteurs différents.

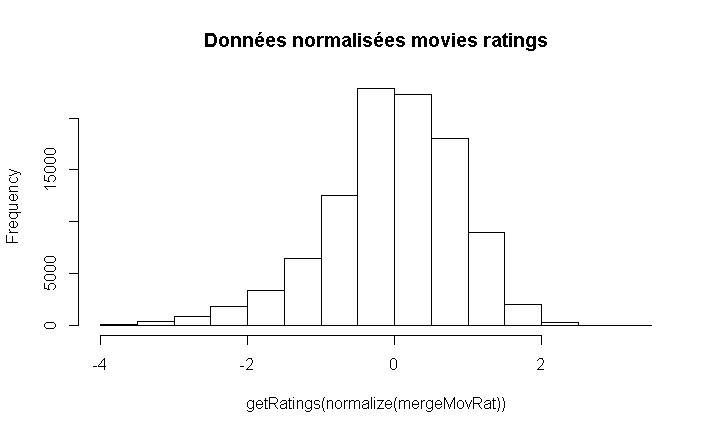
En effet avec le data set que nous avons utilisés, nous utiliserons les différentes données « ratings », « movies » et « tags ». Les évaluations fournissent les classements des films, les films fournissent les noms des films et les « tags » permettront d’identifier les différents types de films. Nous utiliserons la fonction "recommanderlab" qui dispose de cinq techniques différentes afin de développer des moteurs de recommandation (IBCF, UBCF, POPULAR, RANDOM, & SVD).

**Explication du code source**

Dans cette partie, on chargera les fichiers CSV (ratings, movies et tags). On fusionne les deux fichiers csv « movies et ratings » en prenant l'attribut « movieId » comme jointure grâce à l’aide de la fonction « merge ». Il nous faut ensuite transformer le type data.frame en RealRatingMatrix (utile pour la suite de notre analyse).

A l'aide de deux histogrammes, nous affichons dans le premier les données brutes tandis que dans le second les données normalisées de la fusion entre movies et ratings.



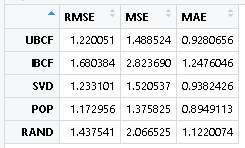


Pour notre analyse prédictive, nous créerons un schéma d’évaluation sur le dataset movies/ratings (mergeMovRat) en utilisant la méthode de la validation croisée. On décide de ne prendre que 80% (0.8) du set de données d'entrainement afin de prédire la partie restante. Il faut préciser à la fonction ci-dessous quelle est la note à partir de laquelle on considère qu'il s'agit d’une bonne notation (>=4), le paramètre k représente le nombre de plis pour la validation croisée (10).

Pour la réalisation de notre étude, nous utiliserons les deux fonctions "recommender" et "getdata" afin d'obtenir l'ensemble de données en utilisant les 5 techniques de modélisations (UBCF, IBCF, SVD, POPULAR, RANDOM).

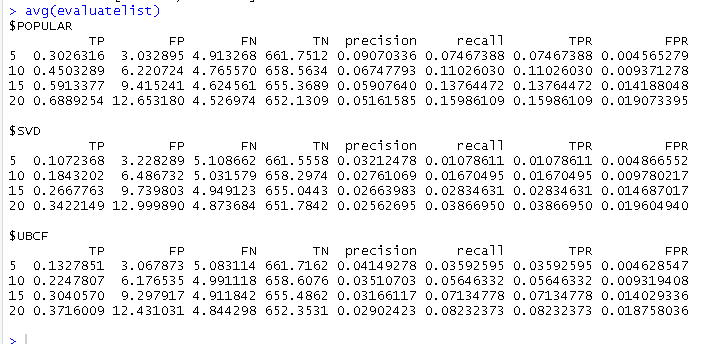
Maintenant que nous avons construit nos modèles pour chaque techniques (voir plus haut), il nous faut lancer une prédiction à l'aide de la fonction "predict" en complément de la fonction "getdata" avec pour argument "type" les notations (ratings).

Sur les prédictions générées, nous calculons la précision des 5 techniques de modélisations. Ici le taux d'erreur sera calculé.

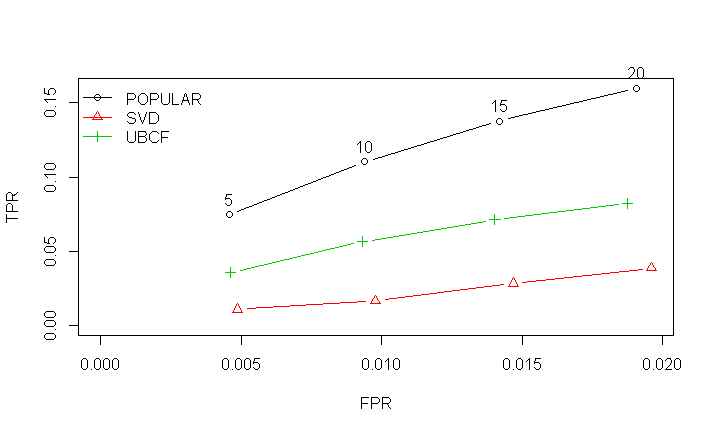
 Il nous faut ensuite combiner les résultats précédents ensemble afin de déterminer les techniques de modélisations les plus forts.

On observe que les techniques de modélisations IBCF et RAND sont les moins précises. Nous allons donc nous concentrer sur les trois autres techniques de modélisations.

Sur ces trois techniques de modélisations, on évalue sur plusieurs varaibles (ex : TP True Positive, FP False Positive, FN False Negative, TN True Negative, Precision, recall).

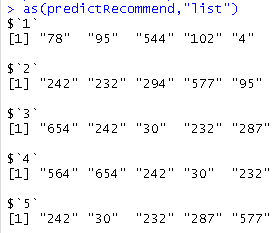


Cette visualisation nous permet de conclure que la technique de modélisation la plus optimale est la "POP", en regardant la valeur pour 5 films, POPULAR est la plus élevé avec 0.30 sur les "TP" (SVD avec 0.10, UBCF 0.13).



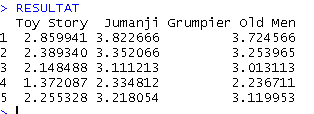
Sur la technique de modélisation "POPULAR", nous allons construire des recommandations cette fois-ci individuelles.

On réalise un tirage pour les 5 premières recommandations pour les 5 premiers évaluateurs afin de construire une liste.



Les numéros correspondent au identifiants (id) des films du dataset ratings d'origine.

Nous prédisons les notes probables que les 5 premiers évaluateurs pourraient attribuer à chaques films.

Sur les 5 premiers évaluateurs et les 3 trois premiers films du dataset ratings Toy Story, Jumanji et Grumpier.