**NAME : V92 RANK :1 RMSE:0.78**

**Approach:**

* Load test and train data , merge it to extract the user behavior.
* Use users past ratings and additional metadata to predict ratings for new users

1. **Collaborative filtering**

Uses users past ratings to predict the ratings using matrix factorization technique particularly Repeated Matrix Reconstruction. It works on principle of low rank approximation using SVD.

1. **Content based**

Uses genre and tags in addition to userID and Movie ID to predict the ratings using random forest regressor and ridge regression

**Methodology:**

**Content based recommender using genre and tags:**

1. After the data was loaded, the tags file was correlated with related tag files to get one file where all the tags and tag Id was arrranged according to its movieId.
2. Also, the genre file was transformed and merged to show which movieID (rated by users) has which type of genre in one file.
3. The genre and the training and testing file was merged to get all the features together .

eg: cv = CountVectorizer(tokenizer=space\_tokenizer)

vec\_genre\_matrix = cv.fit\_transform(df\_genre['genre']).toarray()

vec\_genre\_data = pd.DataFrame(vec\_genre\_matrix, columns=cv.get\_feature\_names())

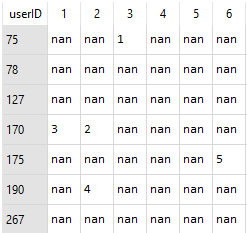
1. On the processed data , random regressor and ridge regression models were applied which gave me RMSE of 1.96 and 1.06 respectively.

model = Ridge(alpha=0.8, solver = "lsqr", fit\_intercept=False)

**Collaborative filtering using repeated Matrix reconstruction:[1][2]**

* Ratings can be predicted based on how certain users rate certain movies. Low rank approximations are well suited for this problem as it decomposes the rating matrix into small groups which can be combined to predict rating for missing user movie combination.
* After The train and test set is loaded, they are concatenated in one matrix after the table is pivoted to get userID as row number and movieID as column numbers while ratings in the matrix.

matrix = pd.concat([train,test]).pivot('userID','movieID','rating')



* A Low rank approximations is used to predict the missing ratings from the users. for that firstly Movie ratings mean is taken in the next step and in new matrix users ratings are shifted relative to each movie's mean movie rating which makes all unrated movie entries as 0.

zero\_mean = matrix-mean

mean1 = zero\_mean.fillna(0)

* All the nan are replaced with zeroes and all negative ratings are covered.
* SVD is used to form low rank approximation.

svd = TruncatedSVD(n\_components=20,random\_state=42)

mean1svd=pd.DataFrame(svd.inverse\_transform(svd.transform(mean1)), columns=mean1.columns,index=mean1.index)

* A convergence condition is put to iterate the procedure till it converges.
* To preserve the known ratings all ratings in the matrix are reset to the known values.
* A combination of 10 iterations and n\_components = 15 gives the best combination of RMSE

Reference:

[1]The singular Value Decomposition and Low-Rank Matrix Approximations: Recovering missing entries via the SVD

[2]Mining of massive datasets