

# CS F469 Movie Recommender System

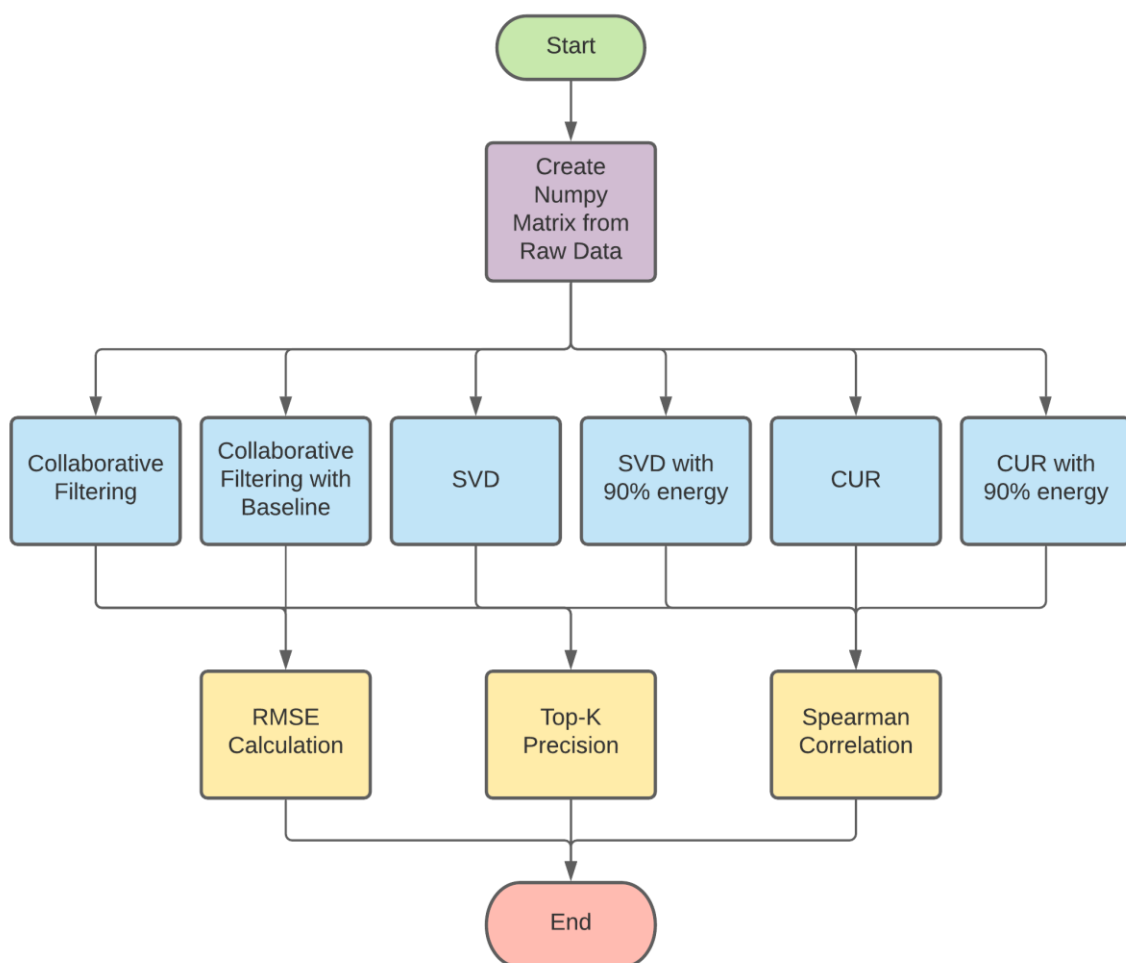
This project is a recommender system based on the MovieLens dataset built using

1. Collaborative filtering
2. SVD - Singular Value Decomposition
3. CUR

## Requirements

1. Numpy
2. Pandas

## Flowchart



# Collaborative Filtering (CF)

The implemented algorithm is an approach to recommendation systems where we estimate the rating a user might give to an item based on the ratings the user has given to other items. This is done by computing the similarity between two items and then using the score as weights to the ratings already present.

## Pearson Correlation Coefficient

```
method similarities.pearson_sim
```

The similarity metric being used here is the Pearson Correlation Coefficient, given by the Cosine similarity between two standardized vectors (i.e., rows corresponding to items).

$$\text{sim}(x, y) = \frac{\sum (r_{xs} - \mu_x) (r_{ys} - \mu_y)}{\sqrt{[\sum (r_{xs} - \mu_x)^2 \times \sum (r_{ys} - \mu_y)^2]}}$$

## Item-Item CF

```
class collaborate.Collaborate
```

The actual rating is estimated here and the input matrix is filled.

To estimate rating for a given (user, item) pair:

$$r_{xi} = \frac{\sum \text{sim}(i,j) \cdot r_{xj}}{\sum \text{sim}(i,j)}$$

A common practice to get a better estimate is to account for baseline offset.

$$r_{xi} = b + \frac{\sum \text{sim}(i,j) \cdot (r_{xj} - b_x)}{\sum \text{sim}(i,j)}$$

where  $b$  is given by,

$$b = \mu + b_x + b_i$$

$\mu$ : Overall Mean

$b_x$ : Deviation for user  $x$

$b_i$ : Deviation for item  $i$

# CUR

CUR Algorithm is a close estimate to SVD Algorithm.

# Evaluation Measures

## Contents

1. RMSE - Root Mean Square Error
2. Precision in Top K
3. Spearman Correlation

## RMSE - Root Mean Square Error

Formula:  $\sqrt{\frac{\sum (\text{predicted} - \text{actual})^2}{n}}$

## Precision in Top K

Gives an estimate of how many of the predicted ratings are present in the top K ratings of the user since only the good one's count in the error measure.

## Spearman Correlation

Formula:  $1 - \frac{\sum (\text{diff}(\text{predicted} - \text{actual})^2)}{n(n^2 - 1)}$

# Preprocessing

Dataset: <http://files.grouplens.org/datasets/movielens/ml-100k.zip>

1. Column 1 - User ID
2. Column 2 - Movie ID
3. Column 3 - Rating

Other columns are ignored since they are irrelevant to the implemented algorithms.

Output: Numpy array of shape 963 \* 1682

# SVD - Singular Value Decomposition

Theory:

$$A = U * \text{Sig} * V'$$

A = Original Data Matrix ( Users \* Items)

U = Users to Concept matrix

V = Items to Concept matrix

Sig = Concept Strength matrix containing Eigen values in decreasing order