EX.NO.:2

DATE:

Implement dimension reduction techniques for recommender systems.

AIM:

To implement dimension reduction techniques for recommender systems.

ALGORITHM:

Singular Value Decomposition:-

1. Import the necessary libraries
2. Generate sample data matrix
3. Split the data into training and testing datasets
4. Perform Singular value decomposition
5. Reconstruct the Original data matrix from the reduced form
6. Calculate Mean Squared Error on the test set

Matrix Factorization:-

1. Import the necessary libraries
2. Load the dataset
3. Use the dataset to form a dataframe format
4. Use the SVD algorithm
5. Perform cross-validation
6. Print the CV

PROGRAM:

Singular Value Decomposition:-

```
import numpy as np
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from scipy.sparse import lil_matrix
# Generate sample user-item matrix (replace this with your data)
num_users = 100
num_items = 50
ratings = np.random.randint(1, 6, size=(num_users, num_items))
# Split data into training and testing sets
train_data, test_data = train_test_split(ratings, test_size=0.2, random_state=42)
# Perform Singular Value Decomposition
n_components = 10  # Set the number of components (latent factors)
svd = TruncatedSVD(n_components=n_components)
reduced_train_data = svd.fit_transform(train_data)
# Reconstruct the original matrix from the reduced form
reconstructed_train_data = np.dot(reduced_train_data, svd.components_)
# Calculate Mean Squared Error on the test set
mse = mean_squared_error(train_data, reconstructed_train_data)
print(f"Mean Squared Error: {mse}")
```

Matrix Factorization:-

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate
```

 # Load your dataset (replace this with your data)

 # Use the load_from_df method if your data is in DataFrame format

data = Dataset.load_builtin('ml-100k')

 # Use the SVD algorithm

algo = SVD()

 # Perform cross-validation

CV=cross_validate(algo, data, measures=['RMSE'], cv=5, verbose=True)

print(CV)

OUTPUT:

Singular Value Decomposition:-

Mean Squared Error: 1.0896028841627086

Matrix Factorization:-

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (test_set)	0.9306	0.9388	0.9348	0.9399	0.9365	0.9361	0.0033
Fit time	2.04	1.76	1.84	2.43	1.55	1.92	0.30
Test time	0.18	0.22	0.23	0.37	0.14	0.23	0.08

{'test_rmse': array([0.93062173, 0.93880913, 0.93483342, 0.93991998, 0.93651167]),

'fit_time': (2.036858320236206, 1.7616021633148193, 1.8352279663085938, 2.430487871170044, 1.5543479919433594),

'test_time': (0.1761641502380371, 0.22463703155517578, 0.22916793823242188, 0.3696250915527344, 0.14364123344421387)}

RESULT:

Thus, the implementation of dimension reduction techniques for recommender systems was executed successfully.