EX.NO.:1

Implement dimension reduction techniques for recommender systems

---

AIM:

To implement dimension reduction techniques for recommender systems.

ALGORITHM:

Step 1: Import Necessary Libraries.

Step 2: Load Dataset.

Step 3: Split Data into Training and Testing Sets.

Step 4: Apply SVD for Dimension Reduction.

Step 5: Evaluate and Recommend.

PROGRAM

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics.pairwise import cosine_similarity

# Load the dataset
# Ensure you have the movies_metadata.csv file in the same directory
data_path = "movies_metadata.csv"
movies = pd.read_csv(data_path, low_memory=False)

# Preprocessing: Filter required columns and drop missing values
movies = movies[['title', 'overview']].dropna()

# Step 1: Convert text data (overviews) to numerical features using TF-IDF
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_features=5000)
tfidf_matrix = tfidf_vectorizer.fit_transform(movies['overview'])

print(f"TF-IDF Matrix Shape: {tfidf_matrix.shape}")
```

```
# Step 2: Dimensionality reduction using Truncated SVD
n_components = 100  # Reduce to 100 dimensions
svd = TruncatedSVD(n_components=n_components)
reduced_matrix = svd.fit_transform(tfidf_matrix)

print(f"Reduced Matrix Shape: {reduced_matrix.shape}")

# Step 3: Compute similarity matrix
cosine_sim = cosine_similarity(reduced_matrix, reduced_matrix)

# Function to get movie recommendations
def get_recommendations(title, cosine_sim=cosine_sim, movies=movies):
    # Get the index of the movie that matches the title
    idx = movies.index[movies['title'] == title].tolist()
    if not idx:
        return f"Movie '{title}' not found in the dataset."
    idx = idx[0]

    # Get pairwise similarity scores for all movies
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Sort the movies by similarity score
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Get the top 10 most similar movies
    top_movies = sim_scores[1:11]

    # Return the titles of the top similar movies
    return movies['title'].iloc[[i[0] for i in top_movies]].tolist()
```

```
# Test the recommender system
movie_title = "The Dark Knight Rises"
recommendations = get_recommendations(movie_title)

print(f"\nMovies similar to '{movie_title}':")
for i, movie in enumerate(recommendations, 1):
    print(f"{i}. {movie}")
```

OUTPUT

```
TF-IDF Matrix Shape: (44506, 5000)
Reduced Matrix Shape: (44506, 100)

Movies similar to 'The Dark Knight Rises':
1. The Glass Menagerie
2. The Voice of Bugle Ann
3. We Bought a Zoo
4. Marguerite & Julien
5. Down and Out in Beverly Hills
6. First Family
7. Hum Saath Saath Hain
8. The Pornographers
9. Communion
10. The Substance of Fire
```

RESULT

Thus the implementation of the dimension reduction technique for the recommended system was executed and verified successfully.