EX. NO.:4

DATE:

Implement content-based recommendation systems.

AIM:

To implement content-based recommendation systems.

ALGORITHM:

1. Import the necessary libraries
2. Get the sample data
3. Get the sample user data
4. Convert text data into TF-IDF features
5. Calculate the cosine similarity between items and user preference
6. Perform the function to get content-based recommendations for a user (weighted sum of items, indices of items, return recommended items)
7. Print the content-based recommendations

PROGRAM:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
# Sample item data (replace this with your data)
items = pd.DataFrame({
    'item_id': [1, 2, 3, 4],
    'title': ['Action Movie', 'Comedy Movie', 'Drama Movie', 'Sci-Fi Movie'],
    'genre': ['Action', 'Comedy', 'Drama', 'Sci-Fi'],
    'description': ['Explosions and car chases.', 'Laughs and humor all the way.', 'Intense emotional scenes.', 'Futuristic technology and space adventures.']})
# Sample user preferences (replace this with your data)
user_preferences = {
        'Action': 5,
        'Comedy': 4,
        'Drama': 2,
        'Sci-Fi': 3
        }
```

```
# Convert text data (title, genre, description) to TF-IDF features
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
item_features = tfidf_vectorizer.fit_transform(items['title'] + ' ' + items['genre'] + ' ' + items['description'])
# Calculate cosine similarity between items and user preferences
cosine_similarities = linear_kernel(item_features, tfidf_vectorizer.transform([f"{pref}" for pref in
user_preferences.keys()]))
# Function to get content-based recommendations for a user
def get_content_based_recommendations(user_preferences, items, item_features, cosine_similarities,
n_recommendations=2):
    # Weighted sum of item similarities based on user preferences
    weighted_similarities = np.dot(cosine_similarities.T, list(user_preferences.values()))
      # Get indices of items with highest weighted similarity
    recommended_item_indices = weighted_similarities.argsort()[::-1][:n_recommendations]
    # Return recommended items
    recommendations = items.iloc[recommended_item_indices]
    return recommendations
# Example: Get content-based recommendations for the user
content_based_recommendations = get_content_based_recommendations(user_preferences, items,
item_features, cosine_similarities)
print("Content-Based Recommendations:")
print(content_based_recommendations[['item_id', 'title', 'genre', 'description']])
```

OUTPUT:

Content-Based Recommendations:

	item_id	title	genre	description
0	1	Action Movie	Action	Explosions and car chases.
1	2	Comedy Movie	Comedy	Laughs and humor all the way.

RESULT:

Thus the implementation of content-based recommendation systems was executed successfully.