EX_NO_10

April 7, 2024

[1]:
```
'''Ex.No:10 Recommender system
REG:URK22AI1048

Aim:

To design a content-based recommender system using python programming

Description

Recommender systems are among the most popular applications of data
science today. There are a lot of applications where websites collect data from␣
 ↪their
users and use that data to predict the likes and dislikes of their users. This␣
 ↪allows
them to recommend the content that they like. Recommender System is a software
system that provides specific suggestions to users according to their␣
 ↪preferences.
These techniques may provide decision-making capabilities to the user. Items␣
 ↪refer to
any product that the recommender system suggests to its user like movies, music,
news, travel packages, e-commerce products, etc.

Content-based recommenders Suggest similar items based on a particular
item. This system uses item metadata, such as genre, director, description,␣
 ↪actors,
etc. for movies, to make these recommendations. And to recommend that, it will
make use of the user&amp;amp; past item metadata. A good example could be YouTube,
where based on the history, it suggests new videos that can be potentially␣
 ↪watched.'''
```

[1]:
```
'Ex.No:10 Recommender system\nREG:URK22AI1048\n\nAim:\n\nTo design a content-
based recommender system using python programming\n\nDescription\n\nRecommender
systems are among the most popular applications of data\nscience today. There
are a lot of applications where websites collect data from their\nusers and use
that data to predict the likes and dislikes of their users. This allows\nthem to
recommend the content that they like. Recommender System is a software\nsystem
that provides specific suggestions to users according to their
```

preferences.\nThese techniques may provide decision-making capabilities to the user. Items refer to\nany product that the recommender system suggests to its user like movies, music,\nnews, travel packages, e-commerce products, etc.\n\nContent-based recommenders Suggest similar items based on a particular\nitem. This system uses item metadata, such as genre, director, description, actors,\netc. for movies, to make these recommendations. And to recommend that, it will\nmake use of the user& past item metadata. A good example could be YouTube,\nwhere based on the history, it suggests new videos that can be potentially watched.'

[2]:
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

[4]:
```
data = pd.read_csv('udemy.csv')
```

[6]:
```
data['course_title'] = data['course_title'].str.strip()
```

[ ]:
```
tfidf = TfidfVectorizer(stop_words='course_id')
tfidf_matrix = tfidf.fit_transform(data['course_title'])
```

[ ]:
```
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

[9]:
```
def recommend_courses(input_title, n):
    if input_title not in data['course_title'].values:
        return "Course not found. Please enter a valid course title."
    idx = data[data['course_title'] == input_title].index[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    top_n = sim_scores[1:n+1]  # Exclude the input course itself
    recommendations = [data.iloc[i[0]]['course_id'] for i in top_n]
    return recommendations
```

[ ]:
```
input_title = input("Enter a course name: ")
num_recommendations = int(input("Enter the number of recommendations: "))
recommended_courses = recommend_courses(input_title, num_recommendations)
if isinstance(recommended_courses, str):
    print(recommended_courses)
else:
    print(f"Top {num_recommendations} course suggestions for '{input_title}':")
    for i, course in enumerate(recommended_courses):
        print(f"{i+1}. {course}")
```

[ ]:
```
'''Result:
To design a content-based recommender system using python programming is␣
↪executed Successfully.
```

```
'''
```