← Go Back to Recommendation Systems

Course Content

New Package Introduction - Recommendation Systems Part 2

Now, let's go through some of the common functions used in this LVC case studies.

CoClustering

Clustering based recommendation system is one important way to build recommendation system using the concepts of clustering. **CoClustering** is a method in the Surprise library that is used to build the clustering based recommendation system in Python. To import this, the below code can be used:

```
from surprise import CoClustering
```

One sample application of this function is shown below:

```
# Using CoClustering algorithm
clust_baseline = CoClustering(random_state = 1)

# Training the algorithm on the train set
clust_baseline.fit(trainset)

# Let us compute precision@k, recall@k, and F_1 score with k = 10
precision_recall_at_k(clust_baseline)
```

To get more information about this function, you can refer to this link.

nltk

Content based recommendation system is a very important way to build recommendation system. It uses the features of the items to find similar items to recommend. The concepts of natural language processing can be used to prepare relevant data if the item data is in text format. **nltk (Natural Language Toolkit)** is a library that is used to do text preprocessing and prepare the numerical vectors required for building the recommendation system. To import this library, the below code can be used:

```
import nltk
```

Some of the helpful methods in nltk that is used for text preprocessing is as follows:

Word_tokenize

```
from nltk import word_tokenize
```

This is used to perform tokenization of a text document.

Stopwords

```
from nltk.corpus import stopwords
```

This is used to remove stopwords of a certain language from the given text.

To get more information about nltk, you can refer to this link.

TfidfVectorizer

One of the basic requirement while working with text data is to convert it into a numeric vector. The TfidfVectorizer is a way to create such vectors. It considers both the term frequency and the inverse document frequency to create this vector. Term frequency is the frequency of a word in a document while the inverse document frequency depends on the number of documents in which the considered word appears. The below code is used to import this method:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

The below code shows one sample application of the same:

```
# Creating the TF-IDF object
tfidf = TfidfVectorizer(tokenizer = tokenize)

movie_tfidf = tfidf.fit_transform(final_ratings['text'].values).toarray()
```

To get more information about this function, you can refer to this link.

Cosine_similarity

This is a method used to find the cosine similarity of two numeric vectors in python. This is a method from the sklearn library in python. To import it, the below code can be used:

```
from sklearn.metrics.pairwise import cosine_similarity
```

To get more information on this function, you can refer to this link.

Happy learning!

Previous

Next