Types of Recommender Systems

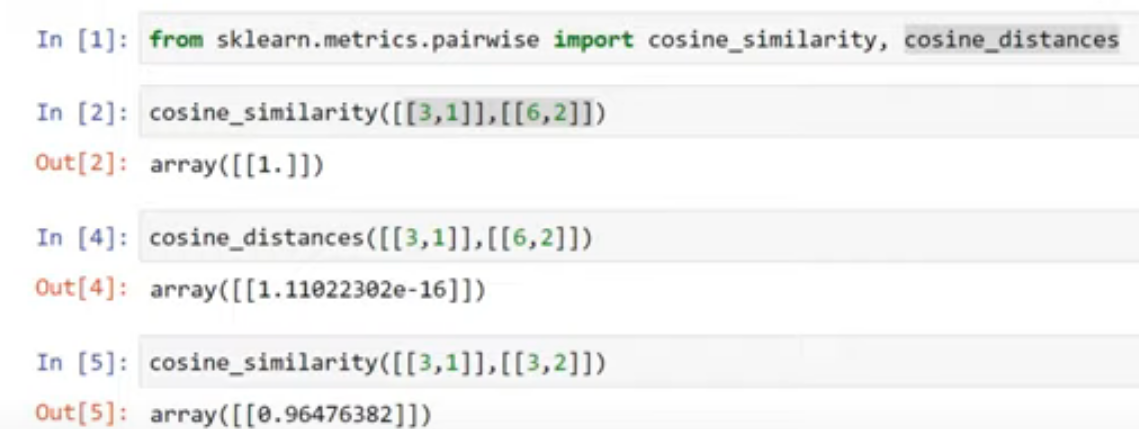
* Content based.
  + Based on similarity of content
* Collaborative filtering
  + Based on similar user interests
* Hybrid
  + Content + Collaborative

Project Flow:

Data 🡪 Pre-processing 🡪 Model building

Text

Description automatically generated



Vectorization

* Don’t use stop words
* How to decide how many top using words to use in a particular algorithm 🡪 That can be considered in hyper parameter tuning
* Mainly the vector will be sparse
* Remove activity / activities, love/ loving 🡪 Stemming
* Euclidean distance don’t work better if increasing dimensionality

text vectorization techniques

Text vectorization is the process of converting text data into numerical representations that can be processed by machine learning algorithms. There are several techniques for text vectorization, including:

* Bag-of-Words (BoW): BoW represents a document as a bag of its words, ignoring the order and context of the words. Each word is assigned a unique integer index, and the resulting vector for a document consists of the count of each word in the document.
* Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF represents each word in a document by its term frequency (number of times it appears in the document) multiplied by its inverse document frequency (logarithm of the total number of documents divided by the number of documents containing the word).
* Word Embeddings: Word embeddings represent each word as a dense vector in a high-dimensional space, where semantically similar words are close to each other. Popular techniques for generating word embeddings include Word2Vec and GloVe.
* Character-level embeddings: Character-level embeddings represent each character in a word as a dense vector in a high-dimensional space, and then combine them to form a word vector. This technique is useful for languages with complex morphology or for tasks like named entity recognition.
* Subword-level embeddings: Subword-level embeddings represent each word as a combination of character n-grams (e.g., "en", "gin", and "ing" for the word "engineering"). This technique is useful for handling out-of-vocabulary words and for capturing morphological information.
* Convolutional Neural Networks (CNNs): CNNs can be used for text classification tasks, where the input is a sequence of words represented by word embeddings. The CNN applies convolutional filters over the input sequence to detect patterns, and then aggregates the output of the filters to make a prediction.
* Recurrent Neural Networks (RNNs): RNNs can be used for sequence-to-sequence tasks, where the input and output are both sequences of words. RNNs can process input sequences of arbitrary length and capture the context of the input sequence. Popular RNN architectures include Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Curse of dimensionality

