Neshat Jalali Heravi January 2022
NLP Project Summary

Building a Content-based Movie Recommendation System

Abstract

In this project a content-based movie recommender system was built using Natural Language Processing. The dataset used have over 45,000 movie overviews. After the preliminary data cleaning and text preprocessing, TF-IDF vectorizer and the unsupervised Non-Negative Matrix Factorization (NMF) topic modeling algorithm was used to extract topics from the movies overview corpus. Then the cosine similarity metric was used among the movie topic profiles to create a content-based recommendation system.

Design

Recommender systems have a huge role in improving customer experience and increase in companies' revenue. The created recommendation system can be beneficial to new streaming services without sufficient user history to help them avoid the "cold start" problem. It can also be of use for established streaming services' consumers who are looking for movies based on their current topics of interests rather than their user history!

Data

The dataset for this project contains information of 45,466 movies featured in the Full MovieLens dataset (movies that are released on or before July 2017) and is available on Kaggle. After the preliminary data cleaning and text preprocessing 3 feature columns (title, released_date, and overview_lemm) and 38,148 rows where each representing a movie remained.

Algorithms

The data cleaning included dropping duplicate, missing entries, and short overviews since they did not include sufficient material for topic modeling. Then text preprocessing techniques (including removing numbers and punctuations, text tokenizing, parts of speech labeling and lemmatizing etc.) was performed.

Next, TF-IDF vectorizer was used to compare and test TruncatedSVD and NMF topic modelers. I decided to continue with the TF-IDF vectorizer and NMF topic modeler to create 20 topics using custom stopwords (in addition to English stopwords). Python's Wordcloud package was used to create visualization and review some sample topics. Finally, the cosine similarity metrics was used to build the recommendation system.

Tools

- Pandas and Numpy for data cleaning manipulation
- NLTK for text preprocessing
- Scikit-learn for preprocessing and modeling
- Wordcloud for text visualization

Communication

You can find the presentation slides and details on the analysis, and the notebook on my GitHub.