Final project topic

The Movies Dataset

EE 541: Spring 2024

Authors: Haodi Hu and Dr. Brandon Franzke

Dataset

These files contain metadata for all 45,000 movies listed in the Full MovieLens Dataset[2]. The dataset consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts and vote averages. This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website.

Refer to the website for further dataset introduction: Kaggle.

Data Files

This dataset consists of the following files:

`movies_metadata.csv`: The main Movies Metadata file. Contains information on 45,000 movies featured in the Full MovieLens dataset. Features include posters, backdrops, budget, revenue, release dates, languages, production countries and companies.

`keywords.csv`: Contains the movie plot keywords for our MovieLens movies. Available in the form of a stringified JSON Object.

`credits.csv`: Consists of Cast and Crew Information for all our movies. Available in the form of a stringified JSON Object.

`links.csv`: Contains the TMDB and IMDB IDs of all the movies featured in the Full MovieLens dataset.

`links_small.csv`: Contains the TMDB and IMDB IDs of a small subset of 9,000 movies of the Full Dataset.

`ratings_small.csv`: The subset of 100,000 ratings from 700 users on 9,000 movies.

Word Embeddings

Processing natural language text and extract useful information from the given word, a sentence using machine learning and deep learning techniques requires the string/text needs to be converted into a set of real numbers (a vector) — Word Embeddings. Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers.
In this dataset, although some word features already get encoded to certain numbers, e.g. genre, credits and keywords, you may still need to do wording embeddings on some features of words, e.g. title and overview, and you can even use better word embedding methods although they have been encoded.
In PyTorch, the class `torch.nn.Embedding` can be used for word embeddings. Please refer to the reference [3] for details about this function and [4] for more introduction and examples about word embedding in PyTorch.

Suggested Approach

1. Analyzing all the given data files and think about which features are relevant to your task. Do the word embeddings if you need.

2. Think about whether to do the feature engineering, e.g. normalizing or standardizing data, dimension reduction (PCA). Some movies do not have certain features, so you can throw the movies or set them as a fix value.

3. Split the dataset into training, validation and testing set.

4. Experiment with different machine learning and deep learning models and compare their performance.

Example Projects

Recommender systems: You are supposed to build a recommendation system which recommends movies to the user. Input of the system is a movie and the output is a recommendation list consisting similar movies to given movie.

Ratings Prediction: You can build a regression model of ratings prediction. You input some information (used as features, e.g. credits, keywords, budgets) about the movies and output the predicted ratings of them.

Rate of Return Prediction: You should build a regression system that helps investors decide whether to invest the given movies by predicting the rate of return (i.e. ratio of revenue and budget). The input should be movies' features (e.g. genre, cast and crew, keywords, runtime) and output is the predicted revenue.

Reference

[1] https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=ratings_small.csv

[2] https://grouplens.org/datasets/movielens/latest/

[3] https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

[4] https://pytorch.org/tutorials/beginner/nlp/word_embeddings_tutorial.html