Movie Recommendations

a project by:

Abdullah Adlouni
Ben Healy
Charles Dorman
David Shackelford

Project Description

Purpose

- To get a better understanding of movie ratings
  - Explore recommendation systems
- Combine multiple data sources to try and get useful information
- Find interesting data which shows relationships in what people enjoy watching

Major Questions

1. Can we make recommendations of movies based on prior customer ratings?
2. Can we create inferences based on attributes of a movie?
3. Can we observe trends in the movie industry and how people enjoy movies?

Datasets

Description of Datasets

- Using multiple data sources which contain multiple datasets
  - Netflix
  - IMDB
  - TMDB
- All datasets have been downloaded by each member
- Datasets are also accessible on AWS S3 database through boto3

Description of Datasets - Netflix

- https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data,
- Contains 17,770 movies rated by 480,189 users for a total of over 100 million ratings.

Description of Datasets - IMDB

- https://datasets.imdbws.com/
- Large datasets with information on 7,980,307 unique movies/shows/shorts objects and 11,906,873 uniques cast member objects
- 7 datasets
  - title.akas - regional title name, language, region, type (8 attributes total)
  - title.basics - secondary name, year released, runtime, genre (9 attributes total)
  - title.crew - directors, actors (3 attributes total)
  - title.episode - if title was a show, contains show information by episode (4 attributes total)
  - title principles - principle cast/crew including ordering and specific role (6 attributes total)
  - title.ratings - imdb rating and vote information (3 attributes total)
  - title.name.basics - names and information of crew including birth, death, and knownForTitles (6 attributes total)

Description of Datasets - TMDB

- https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata
- 2 datasets
  - TMDB-Credits - 4800 unique objects
    - 3 attributes - name, credits (casting, directing), unique identifiers
  - TMDB-Movies - 4800 unique objects
    - 20 attributes including popularity value, genres, revenue, runtime,

Prior Work

- The Netflix Prize: competition to predict user ratings for films based on previous ratings
- BellKor's Pragmatic Chaos team won
  - Improved Netflix's film recommendation algorithm by 10%
  - Used multiple ML algorithms sequentially and then blended their results
  - ML models were trained to minimize the RMSE on the probe data set
- Many sources for item-based collaborative filtering
  - https://medium.com/grabngoinfo/recommendation-system-item-based-collaborative-filtering-f5078504996a
  - https://medium.com/geekculture/overview-of-item-item-collaborative-filtering-recommendation-system-64ee15b24bb8
- https://www-cs.stanford.edu/people/nipunb/CS345a.pdf
  - Attempt to increase recommendation accuracy with supplemental data

Work Completed So Far

- Set up AWS integration
  - All members have access and can manipulate/upload/download csvs
- Matched movie titles between datasets using nlp
  - Combined title data with release date data to increase efficiency
  - Has some error, but most titles have a match
- User-user comparison
  - Effort to narrow down data
  - Filtered users based on movie overlap and rating correlation

Tools

Tools

Project Management

- Github
- AWS
- Discord

Data Analysis

- Pandas
- Numpy
- Regex
- Bash

Machine Learning

- NLTLK
- Scikit-Learn
- Statsmodels
- Tensorflow

Tools - Project Management

- Github
  - https://github.com/dash2927/ABCD
  - Will contain code and documentation
  - Each team member works on their own branch and pushes to the main branch
- AWS
  - Contains all data in one place through public S3 bucket (already set up)
  - Interfaces with python through boto3
  - Possibility of using additional resources if needed - AWS's relational database, EC2, etc.
- Discord
  - Communication channel between group members
  - Weekly sprints with video scrum meetings on Thursdays

Tools - Data Analysis

- Pandas
  - Data analysis and manipulation
  - Will explore data after importing through AWS
- Numpy
  - Python support for matrix and data arrays
  - Faster data manipulation than pandas
- Regex
  - Regular expression - search patterns in text
  - Will use for pattern matching for data integration
- Bash scripting
  - Will use for pipeline data manipulation

Tools - Machine Learning

- NLTLK
  - NLP (Natural Language Processing) toolkit for python
  - Will help us with title matching and possibly with getting attributes for recommendation
- Scikit-Learn
  - Machine learning library for python
  - Will be using mostly for decision tree and feature importance modules
- Statsmodels
  - Python library that contains statistical models
  - Will possibly use for vectorization during nlp
- Tensorflow
  - Possibility of using tensorflow if we decide that a NN model would fit better

Proposed Work

Cleaning

- Examine quality of all datasets
  - Missing values
  - Duplicated values
  - Wrong values (i.e. custom null values, strings in integer attributes)
- Normalize movie ratings
- Identify or remove outliers

Preprocessing

- Convert tab-spaced and comma-spaced datasets to consistent format
  - Should be easy to use within a pandas dataframe
- Remove attributes that give no added information
- NLP processing
  - Lemmatization
  - Stemming
  - Vectorizing
- Machine Learning preprocessing

Integration

- Match data based on movie titles
  - Will have to account for differences in title name, secondary title name
  - Can use either NLP or algorithm (subsequence matching)
- Make sure attributes are consistent between datasets
  - Decide on what information to leave out
- Export any integration changes to shared S3 database

Analysis

- Create a model for *item-based collaborative filtering*
  - Decision tree model which looks at item similarity
  - Can be based on multiple features found with dataset integration:
    - Title
    - Genre
    - Actors
    - Language
  - Use different attribute sets to observe how that affects recommendation
  - Can look at feature importance and what matters in deciding recommendation
- Create visuals on attribute trends, what people like and dislike

Evaluation

Evaluation

Recommendations of movies based on prior customer ratings?

- Create a working recommendation system (working meaning it will recommend a movie based on a set of preference features)
- Can calculate RMSE between our film recommendations and Netflix's Probe dataset

Can we create inferences based on attributes of a movie?

- Based on model, can use other attributes to improve on recommendation
- Calculated RMSE improves from previous recommendation model

Observe trends in the movie industry and what people enjoy?

- Visualizations on movie type, genre, director, actor, etc
- Visualizations on movie attributes over time

Thank you!