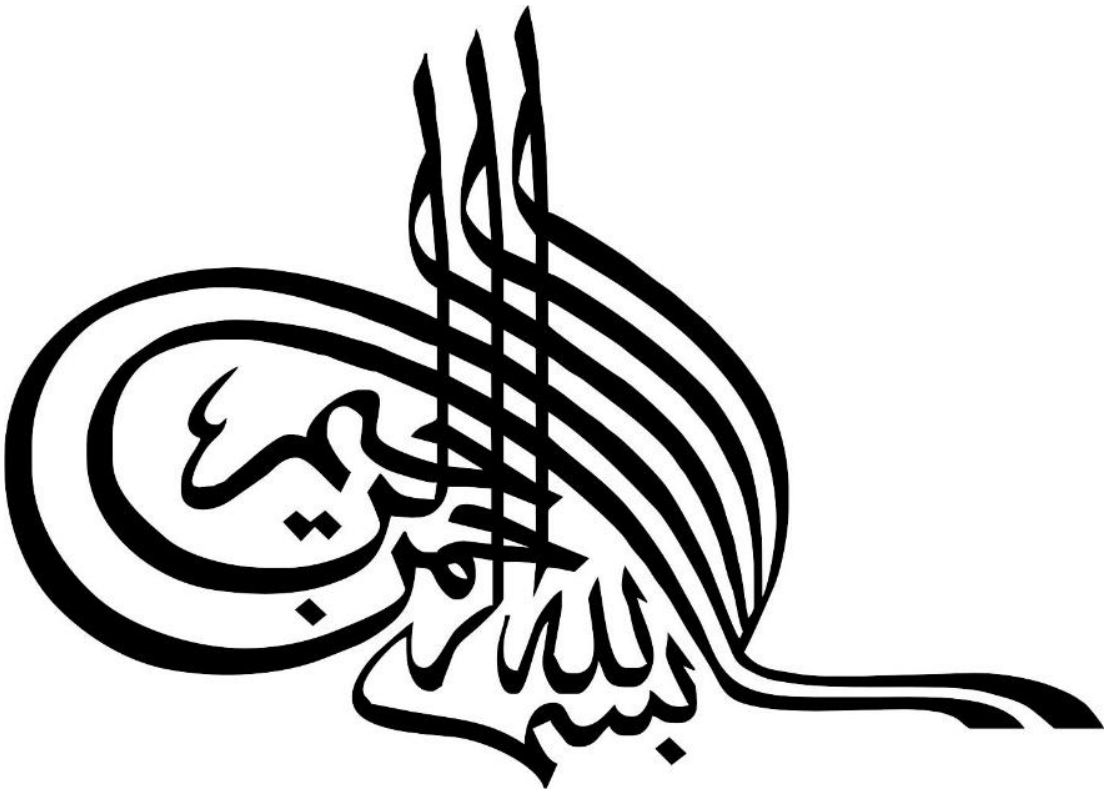


Movie Recommendation System

MAHDI HEIDARI & ZAHRA KOOHESTANI
G09

Dr.Ghadiri | Fundamental of Data Mining | 22 Apr 2021

Theoretical steps

T1) ارائه توضیحاتی در مورد موضوع پروژه, دلیل انتخاب این موضوع و هدف مورد انتظار نهایی گروه از انجام این پروژه

موضوع پروژه "سیستم پیشنهاد دهنده فیلم" است . علت انتخاب این پروژه روی آوردن جامعه به سمت و سوی تماشای فیلم در اوقات فراغت است چرا که تفریح ارزان و دردسترس است. اما از مشکلات این تفریح صرف زمان زیاد مخاطب برای انتخاب فیلم مورد علاقه است. لذا نیاز است تا سیستمی پیاده سازی شود تا براساس الگو های فرد ، فیلم های مورد علاقه شناسایی و به مخاطب ارائه شود. برای نمونه شناسایی علایق فرد در زمینه خصوصیات فیلم نظیر ژانر ، بازیگران ، کارگردان و... و نیز خصوصیات فرد نظیر سن ، تحصیلات و... و سپس پیشنهاد فیلم به فرد با توجه به این خصوصیات .

برای به روز بودن آرشیو فیلم های پیشنهادی نیز میتوانیم از api ها یا web scraping استفاده کنیم. (برای مثال خلاصه فیلم پیشنهاد شده را از یک سایت ایرانی به فرد نمایش بدهیم) و حتی میتوانیم در صورت عملکرد خوب سیستم ، نسخه گرافیکی بسازیم و حتی بتوانیم لینک دانلود فیلم یا زیرنویس و یا یک پلیر درون برنامه ای رو در اختیار فرد هم قرار بدهیم.

همچنین میتوانیم از مخاطبان بازخورد گرفته و سیستم را بهبود بخشید. (و احتمالا یک بخش پیشنهاد از سیستم لوکال برای زمان آفلاین هم اضافه کنیم تا از فیلم های موجود به خود فرد پیشنهاد بدهیم)

در کل میتوان با استفاده از این سیستم تماشای فیلم را برای افراد جامعه آسان تر و لذت بخش تر کرد و در زمان کمی فیلم های مورد علاقه خود را پیدا کنند که هدف از انجام این پروژه همین موضوع است.

T2) توضیحات مختصری از دیتاست انتخاب شده برای موضوع پروژه (سال ایجاد دیتاست, گردآورده دیتاست, منبع اصلی دیتاست و غیره) و همچنین در صورت موجود بودن پروژه های مشابه برای این دیتاست چند مورد از آنها را نام ببرید.(ارائه لینک کافی است)

دیتاست های این پروژه عبارتند از :

1. MovieLens Latest Datasets (https://grouplens.org/datasets/movielens)

27,000,000 ratings and 1,100,000 tag applications applied to 58,000 movies by 280,000 users. Includes tag genome data with 14 million relevance scores across 1,100 tags. Last updated 9/2018.

2. IMDb movies extensive dataset (https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset)

81k+ movies and 175k+ cast members scraped from IMD

در واقع برای این پروژه ما دو دیتاست فوق انتخاب کرده ایم که تا با ترکیب این دو ، دیتاست غنی تری داشته باشیم و ویژگی های بیشتری از فیلم ها داشته باشیم .

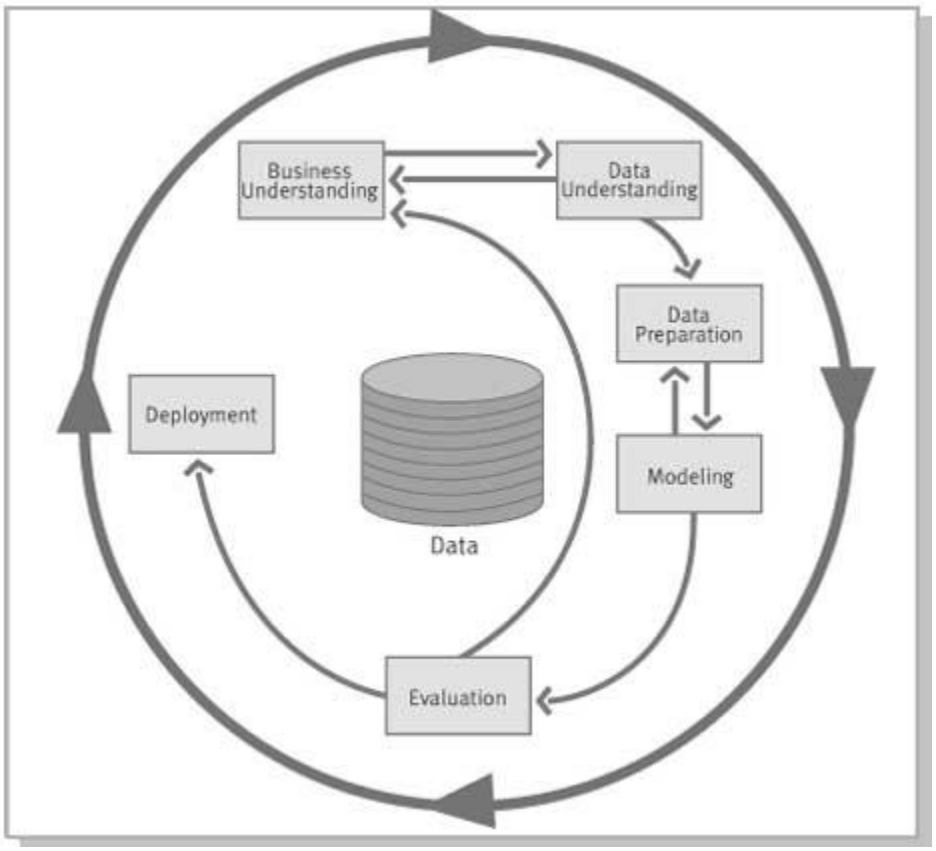
دیتاست اول از سایت Grouplens است و در سال 2016 ساخته شده است و در سال 2018 به روز رسانی شده است و 27 میلیون رکورد دارد.

دیتاست دوم از سایت Kaggle است که در سال 2019 توسط Stefano Leone ساخته شده منبع ااست و صلی آن از سایت IMDB است .

در مورد پروژه های انجام شده درمورد این دو دیتاست موردی یافت نشد ولی پروژه هایی در مورد دیتاست اول وجود دارد که در آدرس https://www.kaggle.com/grouplens/movielens-20m-dataset/code میتوان به دست آورد که چند مورد آن عبارت ند از :

1. Recommender System Deep Learning (https://www.kaggle.com/taruntiwarihp/recommender-system-deep-learning)
2. MovieLens Recommendation System (https://www.kaggle.com/akashsdas/movielens-recommendation-system)
3. Movie Recommendation System (https://www.kaggle.com/dikshabhati2002/movie-recommendation-system)

T3) تشریح کامل موضوع پروژه با استفاده از فرآیند CRISP-DM و ارائه توضیحات هر فاز فرآیند براساس موضوع پروژه



- Business Understanding

در طی چند دهه گذشته ، با ظهور یوتیوب ، آمازون ، نتفلیکس و بسیاری دیگر از خدمات وب از این قبیل ، سیستم های پیشنهادی جایگاه بیشتری در زندگی ما پیدا کرده اند. از تجارت الکترونیکی (پیشنهاد مقالاتی مورد علاقه خریداران) تا تبلیغات آنلاین (پیشنهاد مطالب مناسب به کاربران ، مطابق با ترجیحات آنها) ، امروزه سیستم های پیشنهادی دهنده بسیاری وجود دارند .

سیستم های پیشنهاد دهنده ، الگوریتم هایی هستند که هدف آنها پیشنهاد دادن موارد مرتبط به کاربران است .(مواردی مانند فیلم ، متن ، محصولات جهت خرید یا هر چیز دیگر بسته به صنعت)

سیستم های پیشنهاددهنده در بعضی از صنایع واقعاً حیاتی هستند زیرا در صورت کارآیی می توانند درآمد زیادی به وجود آورند یا راهی برای برجسته شدن در مقابل رقبا باشند. بذای مثال، چند سال پیش ، Netflix چالش هایی را ("Netflix prize") ایجاد کرد که در آن هدف تولید یک سیستم پیشنهاددهنده بود که عملکرد بهتری نسبت به الگوریتم فعلی را دارا دارد و یک جایزه 1 میلیون دلار برای برنده در نظر گرفتند.

درمورد دلایل فراگیر شدن سیستم های پیشنهاد دهنده میتوان به موارد زیر اشاره کرد :

1. کاربران به سادگی میتوانند موارد مورد علاقه خود را پیدا کنند.
2. به فروشندگان کالا در ارائه کالاهای خود به کاربر کمک می کند.
3. در جهت بهبود تعامل کاربران با وب سایت ها کمک میکند.
4. باعث تولید محتوای شخصی شده می شود.
5. محصولات که بیشترین ارتباط را با کاربران دارند، را به سادکی شناسایی میکنند.

از این رو سیستم پیشنهاد دهنده به خصوص در مورد فیلم بسیار مورد توجه است. چرا که تفریحی جذاب و رایگان برای اکثریت است.

- Data Understanding

برای داشتن یک سیستم پیشنهاد دهنده فیلم خوب دیتاست های مختلف و شرایط شان را بررسی کردیم که عبارتند از:

1. دیتاست مووری لنز که توسط وب سایت GroupLens جمع آوری شده است.
2. دیتاست فیلم های یاهو که به علت تحریم در اختیار ما قرار نگرفت. این دیتاست حاوی مقدار زیادی اطلاعات توصیفی در مورد بسیاری از فیلمهای منتشر شده قبل از نوامبر 2003 است ، از جمله بازیگران ، سازندگان ، خلاصه داستان ، ژانر ، میانگین امتیازات ، جوایز و غیره... این مجموعه داده می تواند به عنوان تست برای الگوریتم های یادگیری رابطه ای و داده کاوی و همچنین الگوریتم های ماتریسی و نمودار شامل PCA و الگوریتم های خوشه بندی باشد. اندازه این مجموعه داده 23 مگابایت است.
3. دیتاست CiaoDVD که دیتای DVD ها است و از وب سایت dvd.ciao.co.uk در دسامبر 2013 خریداری شده است.
4. دیتاست Netflix که در مسابقه جایزه Netflix استفاده شده است.
5. دیتاست MovieTweetings که مجموعه ای از داده های رتبه بندی فیلم های زنده است که از توییتر جمع آوری شده است.

در جدول زیر میتوان اطلاعات مربوط به این دیتاست هاشامل تعداد کاربران و فیلم ها و rating ها را مشاهده نمود : با بررسی این دیتاست ها به این نتیجه رسیدیم که دیتاست مووی لنز کامل تر و

Data Set	Users	Items	Ratings (Scale)
FilmTrust	1,508	2,071	35,497 --[0.5, 4.0]
CiaoDVD	17,615	16,121	72,665 --[1, 5]
MovieTweetings	70,994	37,506	910,396 --[0.0, 10.0]
MovieLens 100K	943	1,682	100,000 --[1, 5]
MovieLens 1M	6,040	3,706	1,000,209 --[1, 5]
MovieLens 10M	71,567	10,681	10,000,054--[1, 5]

جامع تر است و دیتای بهتری دارد . این دیتاست شامل 27 میلیون دیتا است و دارای دیتاست های links و movies و ratings و tags و genome-scores و genome-tags است . با بررسی این دیتاست ها به این نتیجه رسیدیم که tags و genome-scores و genome-tags برای کار ما مناسب نیست و از آن استفاده ای نکردیم . اما دیتاست links که دارای ستون های imdbId و tmdbId و دیتاست movies دارای ستون های genre و title و دیتاست ratings داری ستون های userId وmovieId و rating به علت اطلاعات مفید مورد استفاده قرار گرفت.

همچنین از دیتاست IMDB Movies نیز به عنوان دیتاست کمکی استفاده شد تا اطلاعات فیلم ها را به دست آوریم . این دیتاست دارای دیتاست های IMDb movies و IMDb names و IMDb ratings و IMDb title_principals است . که ازIMDb movies و IMDb ratings جهت بهره مندی از اطلاعات فیلم ها استفاده شد . در این دیتاست از ستون های genres و title وoriginal title و year وdate_published و genre و duration و country و language و director و writer و production_company و actors و description و avg_vote و votes و budget و usa_gross_income و worldwide_gross_income و metascore و reviews_from_users و reviews_from_critics و mean_vote و total_votes استفاده شد .

- Data Preparation

ابتدا دیتاست های links و movies و ratings و tags و genome-scores و genome-tags را از نظر دیتا ، تعداد داده های نال و ... بررسی کرده و مشاهده می شود دیتاست links و movies و ratings اطلاعات مفیدی دارند . همچنین دیتاستratings را از نظر تعداد فیلم ها ، تعداد کاربران و تعداد رای ها درهر سال به صورت تجمیعی به طور جداگانه ای بررسی کردیم . درنهایت دیتا های بین سال 1995تا 2005 را یک به یک جدا میکنیم و ذخیره میکنیم . همچنین همه این دیتاست ها را در حالت بالا تر 50 دصد آمار رای ها و نیز بالا تر از 75 درصد آمار رای ها ذخیره میکنیم . درنهایت حجم دیتاست ها در هر حالت را بررسی میکنیم .

سپس دارای دیتاست های IMDb movies و IMDb names و IMDb ratings و IMDb title_principals را از نظر دیتا ، تعداد داده های نال و ... بررسی کرده و مشاهده می شود دیتاست IMDb movies و IMDb ratings اطلاعات دلخواه ما را دارند .

در نهایت دیتاست های links و movies را بر اساس movieId مرج کرده و سپس imdbId را به فرمت صحیح خود (عددی با طول 7 رقم که ابتدای آن tt است) تبدیل میکنم . سپس حاصل را با IMDb movies براساس imdbId جدید مرج میکنیم. در نهایت با IMDb ratings مرج میکنیم و تحت دیتاست MoviesInfo ذخیره میکنیم.

سپس دیتاست MoviesInfo را خوانده و از نظر دیتا و تعداد داده های نال و... بررسی میکنیم . مشاهده می شود تعداد داده های نال بسیار زیاد است و تلاش کردیم از web scraping استفاده کنیم و اطلاعات را از این طریق کامل کنیم، حتی به طور کامل پیاده سازی شد اما بعدا به علت تغییرات در سایت IMDB Movie موفق به انجام این کارنشدیم. (چرا که همراه با هر تگ یک عدد رندم وجود داشت.)

ابتدا ستون director را بررسی میکنیم و میانگین mean_vote را برا فیلم های هر کارگردان به دست آورده و هر کارگردان را با توجه به این اعداد به چهار دسته ، دسته بندی میکنیم وبه هر کارگردان اعداد 1 یا 2 یا 3 یا 4 را اختصاص میدهیم.

در گام بعدی ستون های budget و usa_gross_income و worlwide_gross_income را عددی میکنیم. سپس داده های دیتاست را استاندارد سازی میکنیم.

حال به بررسی داده های نال میپردازیم که ستون های budget و usa_gross_income و worlwide_gross_income و metascore و reviews_from_users و reviews_from_critics و director_r دارای نال می باشد .حال از روش KNNImputer استفاده میکنیم تا داده های نال را پر کنیم . این روش بر اساس همسایه های شمترک مقادیر نال را پر میکند. در نهایت به بررسی دیتاست جهت خذف داده های پرت میپردازیم و داده های پرت را حذف میکنیم . با استفاده از heatmap میتوان مشاهده کرد که بعد از این اعمال باز هم وابستگی بین متغییر ها به نسبت قبل حذف شد .(روش KNNImputer این وابستگی را حفظ کرد.)

سپس ستون country را بررسی میکنیم و میدانیم هر کشور را به قاره ای مربوط است . حال به تعداد قاره ها ستون ایجاد کرده و برای هر کشورستون قاره مربوطه را یک میکنیم .

سپس ستون languageرا از نظر تعداد زبان های مجود بررسی میکنیم . مشاهد ه می شود بیشتر فیلم ها به زبان English است و بعد ازآن فیلم های به زبان French و Italian و German و Spanish و Russian بیشترین است . حال به همین تعداد ستون ایجاد میکنیم و برای هر فیلم با زبان خودش ستون مربوط ش را یک میکنیم . اگر در هیچ یک از حالات فوق نبود ، یک ستون Other ایجاد میکنیم و برای حالات غیر از حالات فوق آن را یک میکنیم.

سپس ستون genre را بررسی میکنیم و تعداد ژانر های موجود را به دست می آوریم که حدود 20 تا است . به تعداد ژانر ها ستون ایجاد میکنیم. حال این ستون ها را برای هر فیلمی با توجه به ژانری که دارد ، یک میکنیم.

درنهایت دیتاست را تحت نام Movies_metadata ذخیره میکنیم تا در مدل سازی ها استفاده کنیم .

- EDA

ابتدا MoviesInfo را خوانده و به بررسی این دیتاست و تعداد نال های هر ستون میپردازیم. همچنین تعداد نال های هر ستون و تعداد فیلم ها در هردهه را به دست می آوریم و نمودار pie chart آن را رسم میکنیم که مشاهده می شود در دهه 2010 بیشترین تعداد فیلم ها را داشته ایم .

همچنین Movies_metadata را خوانده و تعداد نال های آن را بررسی میکنیم که مشاهده می شود نال ندارد . سپس نمودار heatmap آن را رسیم و بررسی میکنیم.

سپس تعداد ژانر های موجود در دیتاست را به دست می آوریم که حدود 20 تا است . به تعداد ژانر ها ستون ایجاد میکنیم. حال این ستون ها را برای هر فیلمی با توجه به ژانری که دارد ، یک میکنیم. در ادامه تعداد فیلم های موجود در هر ژانر را به دست می آوریم و نمودارمیله ای آن را رسم میکنیم . مشاهده می شود از ژانر Drama بیشترین تعداد فیلم را داریم . سپس در هر دهه تعداد فیلم ها در هر ژانر را به دست آورده و سپس به فرمت درصدی تبدیل میکنیم و نمودار stackplot آن را رسم میکنیم .

همچنین نمودار WordCloud را برای ستون title رسم کرده و می توان مشاهده کلمات پرتکرار را مشاهده کرد .

حال دیتاست Rating از دیتاست مووی لنز را خوانده و به بررسی این دیتاست و تعداد نال های هر ستون میپردازیم میپردازیم . سپس این دیتاست را با MoviesInfo مرج میکنیم . حال یک میلیون اول دیتاست حاصل را جهت مدل سازی جدا میکنیم و رنج userId و movieId را در این حالت مشاهده می کنیم . سپس نمودار درصد تعداد کل رای در هر مقدار بین 0.5 تا 5 را بررسی میکنیم . پس از آن تعداد رای ها برای هر فیلم را به دست می آوردیم . و نمودار آن را رسم میکنیم. در نهایت تعداد رای ها برای هر کاربر را بررسی میکنیم و نمدار ش را رسم میکنیم. پس از بررسی مشاهده می شود که مناسب مدل سازی نیست پرا که پیوستگی لازم را ندارد. پس تصمیم گرفتین تا به جدا سازی دیتاست از یک رده سالی خاص بپردازیم.

- Modeling

برای مدل سازی از روش های مختلفی استفاده شده است . از روش های Suprise ، LightFM ، Keras ، Association Rules ، Regression ، Sequential Patterns و Simple Item Based Collaborative Filtering

- ✓ Suprise :یک recommender engine است .در این روش الگوریتم مختلف Suprise بررسی شد و پس از بررسی های مختلف به این نتیجه رسیدیم که SVD و SVD++ از همه الگوریتم ها بهتر است ولی سرعت SVD بالا تر است اگرچه نتیجه SVD++ بهتر است. با استفاده از SVD یک recommender engine طراحی کرده ایم .
- ✓ LightFM : در این روش از الگوریتم های Warp و Bpr استفاده شده است که برای مدل سازی سیستم های پیشنهاد دهنده مناسبند . در این مدل ابتدا دیتاست بین سال های 1995 تا 2001 را خوانده و به مدل ماترسی مناسب برای الگوریتم تبدیل میکنیم . سپس به دو روش فوق مدل سازی میکنیم . که این روش به شدت وقت گیر است.
- ✓ Keras : در این الگوریتم پس از جداسازی ستون های لازم به پیش بینی متغییر های mean_vote ، mean_vote_class در دیتاست Movies_metadata و rating در دیتاست مرج شده پرداختیم .mean_vote_class یک متغیری است که برای mean_vote بالای 6.5 برابر یک و در غیر این صورت صفر است . ابتدا با مدلی mean_vote (لایه های اولیه و پنهان با الگوریتم relu و لایه اخر با الگوریتم linear) و با مدل دیگری mean_vote_class (لایه های اولیه و پنهان با الگوریتم relu و لایه اخر با الگوریتم sigmoid) را پیش بینی کرده ایم . همچنین برای پیش بینی rating (لایه های اولیه و پنهان با الگوریتم relu و لایه اخر با الگوریتم linear) نیز از کراس استفاده شده است.
- ✓ Regression : در این الگوریتم پس از جداسازی ستون های لازم به پیش بینی متغییر های mean_vote ، mean_vote_class در دیتاست Movies_metadata و rating در دیتاست مرج شده پرداختیم در این قسمت برای پیش بینی mean_vote_class از Logistic

Regression و برای پیش بینی mean_vote از Poisson Regressor و نیز Linear Regression استفاده شده است. همچنین برای پیش بینی rating از Poisson Regressor و نیز Linear Regression استفاده شده است.

- ✓ Simple Item Based Collaborative Filtering : یک جدول pivot بین کاربر و فیلم ها براساس رای ها ایجاد می کند . حال از کاربر نام فیلم را گرفته و correlation آن فیلم با سایر فیلم ها بررسی کرده و فیلم های نزدیک را باز میگرداند.
- ✓ Sequential Patterns : به دنبال قانون هایی بودیم تا بررسی کنیم افراد بعد از هر فیلم چه فیلم هایی مشاهده کرده اند ولی در پایتون موفق به انجام این کار نشدیم . ولی در R تلاش کردیم و از کتابخانه های arulesSequences استفاده کردیم ولی به علت عدم زمان کافی این کارادامه نیافت.
- ✓ Association Rules :در این قسمت پس جداسازی دیتاست rating بین سال های 1995 تا 2001 ، یک threshold ای قرار داده شد تا کاربران و فیلم هایی باقی بمانند که حداقل 20 رای داده باشند و گرفته باشند تا مدل سازی و قانون های حاصل بهتر باشند . سپس movieId را ستون و userId را سطر قرار می دهیم و به حالت صفر و یکی تبدیل میکنیم . اینگونه عمل میکنیم که اگر کاربر فیلم را دیده باشد (رای داده باشد) در هر سل مقدار یک قرار می دهیم در غیر این صورت صفر است . حال از آن جایی روش fpgrowth سریع تر است (DFS) لذا از این روش استفاده می کنیم و frequent itemset ها را به دست می آوریم سپس قوانین را استخراج میکنیم .

همچنین لازم به ذکر است که از PCA استفاده کردیم تا تعداد متغییر های ورودی را کاهش دهیم ولی به علت عدم پوشش کامپوننت های حاصل ، این کار را انجام ندادیم .

- Evaluation

برای ارزیابی مدل های فوق به جز Association Rules از MSE و MAE و RMSE استفاده کرده ایم . علت این کار به دست آوردن و بررسی اختلاف رای های پیش بینی شده و رای های دیتاست بوده است.

- Deployment

در این فاز ما جهت پیش بینی یک تابع برای هر مدل پیاده سازی شد تا بتواند با گرفتن userId ویا MovieId (بسته به مدل و ساختار تابع) رای یا متوسط رای برای فیلم های ندیده و حتی دیده (جهت تست) پیش بینی کند .