فهرست مطالب

۱ مقدمه

هدف از این تکلیف آشنایی با کتابخانه های معروف پایتون در حوزه سیستم ها توصیه گر و استفاده از آن ها برای حل یک سری مسائل ساده در این حوزه است. در این تکلیف با کتابخانه های pandas, matplotlib, numpy و ... آشنا خواهیم شد که در رشته ی هوش مصنوعی و در سیستم های توصیه گر ابزار مفیدی هستند. با داشتن دانش درباره ی سیستم های توصیه گر بر اساس کاربر و بر اساس محصول میخواهیم در این تمرین از رای های داده شده به فیلم های سایت مووی لنز[۱] استفاده کنیم و میزان رای کاربر u را به فیلم i بر اساس الگوریتم های مختلف طراحی کنیم. ابتدا در بخش ۲ به معرفی مجموعه داده فیلم مووی لنز میپردازیم و هیستوگرام مربوط به رای های کاربران به فیلم ها را رسم میکنیم. سپس در بخش ۳ با تقسیم رای ها به ۵ قسمت تقسیم میکنیم و به پیاده سازی الگوریتم های سیستم های توصیه گر بر روی آن ها میپردازیم. سپس در بخش ۴ الگوریتم های پیاده سازی شده را ارزیابی کرده و نقش پارامتر های مختلف را در الگوریتم ها ارزیابی میکنیم. در نهایت به نتیجه گیری و توضیح کد های پیاده سازی شده میپردازیم.

۲ تحلیل دیتاست مووی لنز

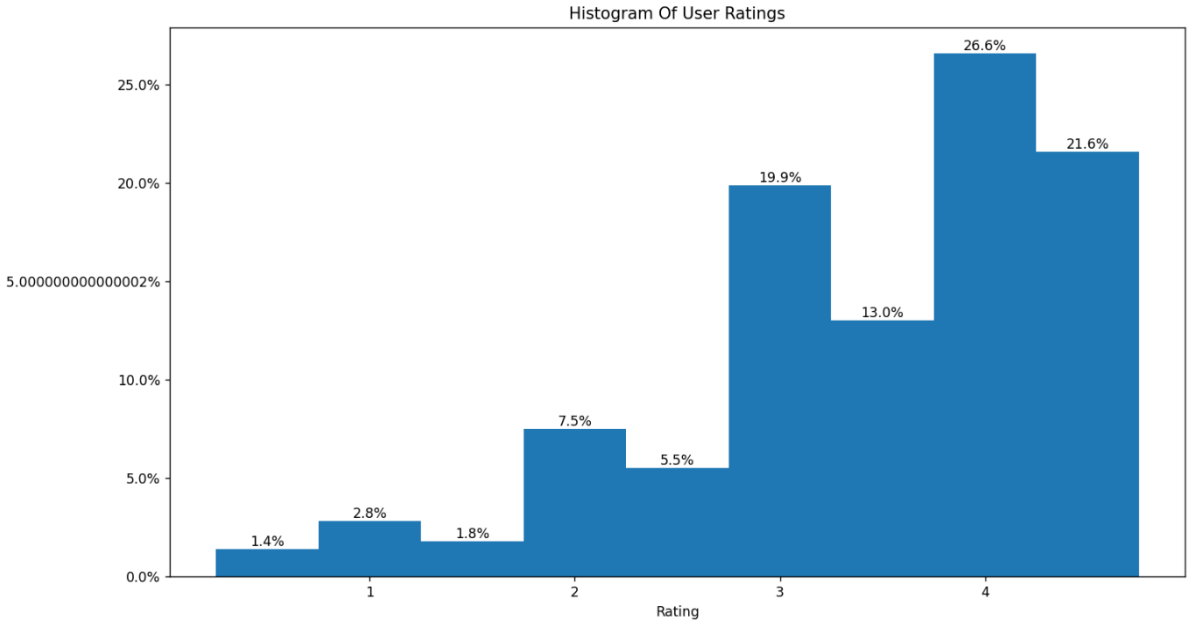
۲-۱ معرفی دیتاست

این مجموعه داده که برای کار های تحقیقات و توسعه در نظر گرفته شده است دارای ۱۰۰۸۳۷ رای کاربر است که به فیلم ها داده شده است. همچنین در این دیتاست در فایل links.csv هر کدام از فیلم ها به فیلم های سایت imdb و tmdb مپ شده اند.در فایل movies مشخصات هر فیلم و در فایل tags تگ های هر فیلم مشخص شده اند. رای های کاربران نیز در فایل ratings وجود دارند که هر کدام از رای ها دارای زمان هستند که میتوان آن ها را به ترتیب تاریخ و ساعت رای دهی مرتب و تقسیم بندی کرد

[۱] Movie Lens

۲-۲ هیستوگرام رای ها

در این قسمت هیستوگرام رای ها بر اساس رای دهی ۵ ستاره ای در تصویر ۱ رسم شده است. همانطور که در تصویر مشخص است، بیشترین میزان رای مربوط به امتیاز ۴ است. همچنین دیتاست به صورت بالانس شده نیست و تعداد رای های مثبت بسیار بیشتر از تعداد رای های منفی است.



تصویر ۱ هیستوگرام رای ها

۲-۳ نحوه تقسیم بندی دیتاست

برای تقسیم دیتاست به ۵ قسمت، باید رای های هر کاربر را به ۵ قسمت تقسیم کنیم و هر بار یک قسمت از آن را برای تست و مابقی را برای آموزش در نظر بگیریم. اگر این عمل را روی کل دیتاست انجام دهیم ممکن است **رای های برخی کاربران به طور کلی در مجموعه آموزشی وجود نداشته باشد** و نتوان شباهت آن کاربران را با بقیه کاربران محاسبه کرد. جهت انجام این کار ابتدا به هر رای یک برچسب اختصاص میدهیم که شماره آن فولد[۲] را نشان میدهد. برای هر کاربر ابتدا رای ها را بر اساس زمان و ساعت رای دهی مرتب میکنیم و سپس تقسیم بندی را انجام میدهیم.

۳ پیاده سازی الگوریتم های پیشبینی رای

در این بخش، نحوه عملکرد ۴ الگوریتم پیشبینی رای که بر روی داده های مووی لنز پیاده سازی شده اند، به تفکیک توضیح داده شده است. لازم به ذکر است میانگین گیری های مورد نیاز در الگوریتم های زیر به صورت ماسک شده[۳] انجام میشود و مقادیر صفر در آن محاسبه نمیشوند.

---

[۲] Fold
[۳] Masked

۳-۱ الگوریتم Per User Average

در این الگوریتم پیشبینی رای کاربر u به آیتم i برابر خواهد بود با میانگین تمامی رای های کاربر u به آیتم ها در داده های آموزشی. در نتیجه میتوانیم میانگین رای را برای تمامی کاربران در یک بردار[۴] ذخیره کنیم تا هر بار نیاز به محاسبه میانگین نباشد.

۳-۲ الگوریتم Per Item Average

در این الگوریتم پیشبینی رای کاربر u به آیتم i برابر خواهد بود با میانگین تمامی رای های داده شده به آیتم i در داده های آموزشی. در نتیجه میتوانیم میانگین رای را برای تمامی آیتم ها در یک بردار ذخیره کنیم تا هر بار نیاز به محاسبه میانگین نباشد.

۳-۳ الگوریتم Global Average

در این الگوریتم مقدار میانگین تمامی رای های موجود در ماتریس رای ها را به عنوان پیشبینی رای تمامی کاربران به تمامی آیتم ها در داده های آموزشی در نظر میگیریم.

۳-۴ الگوریتم User Based Collaborative Filtering

این الگوریتم بر اساس شباهت بین کاربران عمل میکند و برای پیاده سازی نیاز به محاسبه ماتریس شباهت دارد. برای محاسبه مارتیس شباهت روی قسمت بالای قطر اصلی حرکت میکنیم و هر بار مقدار شباهت را برای کاربران u و v محاسبه میکنیم. با داشتن مقادیر بالای قطر اصلی میتوانیم با کپی کردن این مقادیر با استفاده از ترانهاده آن مقادیر زیر قطر اصلی را هم به دست آوریم. ماتریس شباهت را تنها از روی داده های آموزشی محاسبه میکنیم. از معیار شباهت پیرسون[۵] برای محاسبه ماتریس استفاده میکنیم. با داشتن این ماتریس میتوانیم طبق فرمول مربوطه پیشبینی رای هر کاربر را به هر آیتم پیشبینی کنیم. این الگوریتم دارای دو پارامتر است:

- پارامتر k : تعداد همسایگان مشابه که در نتایج الگوریتم در نظر گرفته میشوند.
- پارامتر تتا : میزان حداقل شباهت همسایگان برای در نظر گرفتن آن ها در پیشبینی رای

۴ ارزیابی و نتایج آزمایشات

در این بخش به ارزیابی پیاده سازی الگوریتم های مختلف بر روی دیتاست مووی لنز میپردازیم. در ابتدا مجموعه رای ها را مطابق آنچه در بخش ۲-۳ گفته شد، به ۵ فولد افراز میکنیم. برای هر فولد آن را به عنوان مجموعه تست در نظر گرفته و بر روی ۴ فولد دیگر فرایند آموزش را انجام میدهیم. سپس نتیجه معیار های ارزیابی را بر روی این ۵ فولد

---

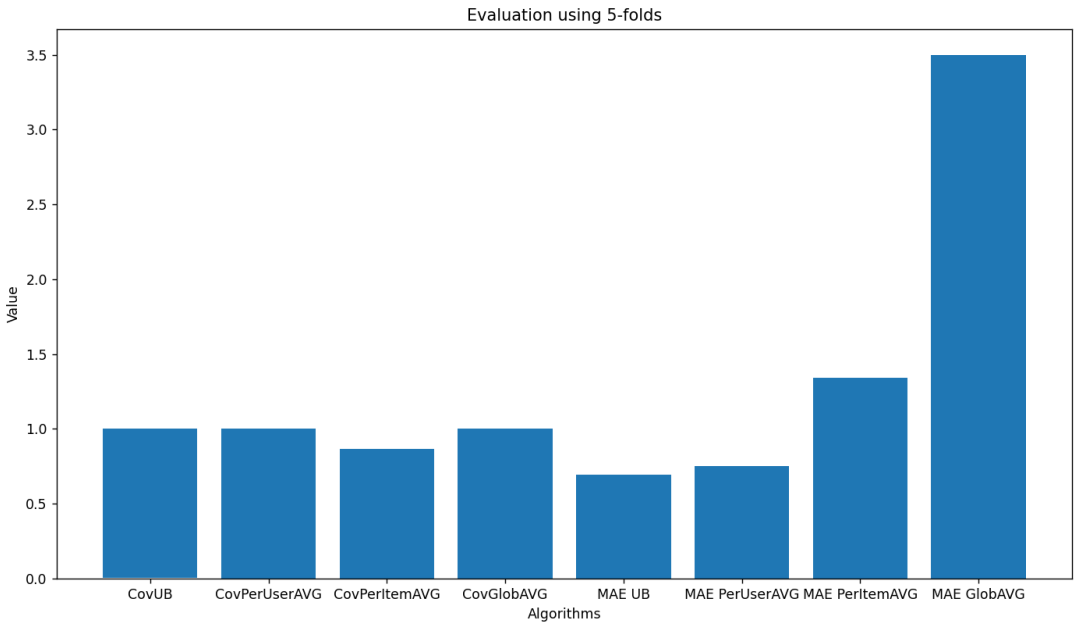
[۴] Vector

[۵] Pearson

میانگین میگیریم. در این مجموعه آزمایش از معیار های میانگین مطلق خطا و پوشش دهی استفاده کرده ایم. نتایج اجرای الگوریتم های یاد شده در بخش ۲ را مورد بحث قرار میدهیم.

۴-۱ نتایج و تحلیل پیاده سازی الگوریتم ها

نتایج محاسبه دو معیار ارزیابی معرفی شده بر روی الگوریتم های مختلف در جدول ۱ آمده است. همانطور که از نتایج هم مشخص است، **بهترین الگوریتم از نظر میزان خطا الگوریتم فیلتر جمعی بر اساس کاربران** است. چرا که با توجه به اینکه در آن پارامتر تتا برابر با ۰ در نظر گرفته شده است و تنها در میان کاربران مشابه میانگین گیری میکند و بدون دانش مانند الگوریتم های میانگین گیری عمل نمیکند. نکته قابل توجه این است که **میزان پوشش دهی فقط در الگوریتم میانگین آیتم ها برابر با ۱ نیست** چرا که الگوریتم میانگین سراسری حتی با وجود یک رای در ماتریس رای دهی هم مقدار آن قابل محاسبه است و الگوریتم های فیلتر جمعی هم اگر کاربر مشابه پیدا نکند میانگین رای کاربر را برمیگرداند و میانگین رای های کاربر هم با توجه به اینکه تمامی کاربر ها هم در مجموعه آموزشی حضور دارند و هم در داده های تست، برای هر کاربر قابل محاسبه هستند. اما برای مثال در الگوریتم میانگین آیتم ممکن است در یک فولد هیچ رای برای آن آیتم وجود نداشته باشد. همچنین بد ترین عملکرد مربوط به **میانگین سراسری** است چون بدون هیچ دانشی میانگین تمام رای ها را انتخاب کرده است.



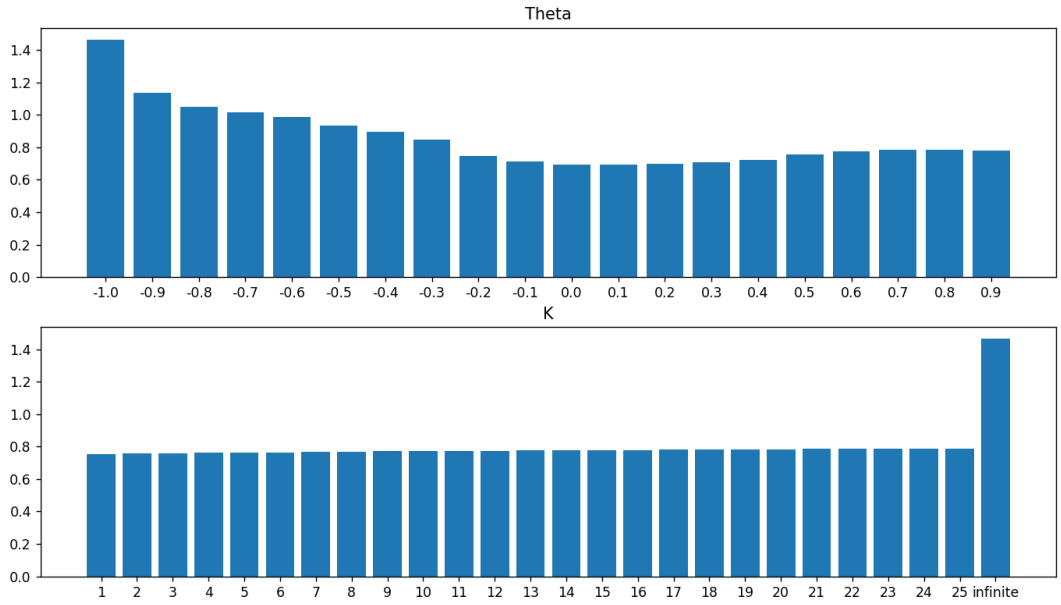
تصویر ۲ ارزیابی الگوریتم های پیشبینی

جدول ۱- نتایج حاصل از پیاده سازی الگوریتم ها

الگوریتم	معیار	مجموعه داده					
		دسته ۱	دسته ۲	دسته ۳	دسته ۴	دسته ۵	میانگین
Global AVG	میانگین مطلق خطا	۳.۵۹۹	۳.۵۲۱	۳.۴۶۸	۳.۴۵۴	۳.۴۴۸	۳.۴۹۸
	پوشش دهی (درصد)	۱	۱	۱	۱	۱	۱
Per Item AVG	میانگین مطلق خطا	۱.۳۹۲	۱.۳۲۳	۱.۳۰۶	۱.۳۳۲	۱.۳۴۷	۱.۳۴
	پوشش دهی (درصد)	۰.۸۶۱	۰.۸۷۳	۰.۸۶۲	۰.۸۶۳	۰.۸۶۵	۰.۸۶۵۵
Per User AVG	میانگین مطلق خطا	۰.۸۳۸	۰.۷۳۲	۰.۷۰۸	۰.۷۲۶	۰.۷۴۸	۰.۷۵
	پوشش دهی (درصد)	۱	۱	۱	۱	۱	۱
UserBasedCF	میانگین مطلق خطا	۰.۷۴۱	۰.۶۶۶	۰.۶۶۶	۰.۶۸۱	۰.۷۰۹	۰.۶۹۳
	پوشش دهی (درصد)	۱	۱	۱	۱	۱	۱

۴-۲ بررسی تاثیر اعمال پارامتر های مختلف

در تصویر ۳ تاثیر پارامتر های k و تتا را در اجرای الگوریتم فیلتر جمعی بر اساس کاربران میبینیم. نمودار میله ای بالا تاثیر پارامتر تتا را نشان میدهد. که میتوان دید خطا در -۱ بسیار زیاد است چون تمامی کاربران را به عنوان کاربران مشابه تشخیص میدهد و آن ها را انتخاب میکند اما خطا تا مقدار تتا برابر با ۰ نزولی است که این نشان میدهد در نظر گرفتن تتا های منفی تاثیر مخرب برای مدل دارد. از ۰ به بعد خطا به مقدار کمی صعودی میشود که این میتواند به دلیل کم شدن تعداد همسایگان باشد. در تحلیل پارامتر k میتوان گفت که تغییرات میزان خطا در k های مختلف از ۰ تا ۲۵ بسیار ناچیز است. به نظر میرسد در این دیتاست با داشتن ۶۱۰ کاربر تفاوت زیادی بین در نظر گرفتن تعداد کاربران کم دیده نمیشود. اما میتوان به وضوح دید که با در نظر گرفتن k=infinite خطا رشد قابل توجهی پیدا میکند که این نشان دهنده این موضوع است که اگر تعداد کاربران بسیار زیاد شود مدل نمیتواند عملکرد خوبی داشته باشد.



تصویر ۳ نمودار تاثیر پارامتر های k و تتا بر عملکرد الگوریتم فیلتر جمعی

۵ نتیجه گیری و جمع بندی

در این تمرین برای پیشبینی رای کاربر به آیتم از ۴ الگوریتم ساده استفاده شد که دیدیم با توجه به دیتاست که تعداد آیتم ها در آن بسیار بیشتر از تعداد کاربران بود، الگوریتم فیلتر جمعی بر اساس کاربران در صورتی که پارامتر k یا تتا برای آن به صورت مناسب انتخاب شود، بهترین عملکرد را در مقایسه با دیگر الگوریتم ها دارد. همچنین تاثیر پارامتر هایی که در فیلتر جمعی تعداد همسایگان را مشخص میکنند را بررسی کردیم و دیدیم که در ازای انتخاب همسایگان بسیار کم و بسیار زیاد میزان خطا زیاد است و در ازای انتخاب مقادیر مناسب برای تعداد همسایگان بهترین نتیجه را داریم. در نتیجه برای هر دیتاست با توجه به شرایط بهتر است الگوریتمی انتخاب شود که با دانش بیشتری عمل میکند و از اطلاعات بیشتری استفاده میکند و در نتیجه قادر است بهتر از سایر الگوریتم ها عمل کند.

۶ مستندات پیاده سازی

برای پیاده سازی تمرین از شی گرایی و کتابخانه های معروف پایتون استفاده شده است. برای لود کردن دیتاست از کتابخانه pandas و برای محاسبات از numpy و برای تقسیم فولد ها از scikit-learn استفاده شده است. همانطور که از ساختار کد ها مشخص است برای تمامی توابع نام مناسب انتخاب شده که از روی نام آن ها مشخص است که هر کدام چه کاری انجام میدهند اما با این وجود کد ها شامل کامنت توضیحات هستند. کلاس ها در پوشه Classes قرار گرفته اند که شامل موارد زیر است.

- Dataset : این کلاس برای لود کردن و عملیات روی دیتاست استفاده میشود.
- Collaborative : برای اعمال الگوریتم های مورد نیاز استفاده میشود و چون سه الگوریتم میانگین گیری زیر مجموعه الگوریتم Collaborative Filtering است، نام گذاری به این شکل انجام شده است. توابع این کلاس در واقع همان الگوریتم های مطرح شده در تکلیف است.
- Evaluation : این کلاس برای ارزیابی ها و محاسبات معیار های ارزیابی استفاده میشود.

همچنین در خارج از پوشه Classes فایل های پایتونی وجود دارد که کاربرد هر کدام در زیر ذکر شده است:

- Histogram : برای رسم هیستوگرام مربوط به دیتاست از آن استفاده میشود.
- Averages : با دریافت اندیس کاربر و اندیس آیتم پیشبینی رای را با استفاده از الگوریتم های میانگین گیری انجام میدهد.
- SaveSimilaritiesToFile : با خواندن فایل دیتاست شروع به ساخت ماتریس های شباهت برای فولد ها میکند و آن ها را در فایل های fold0.csv تا fold4.csv ذخیره میکند
- Evaluations : این فایل در نهایت با استفاده از کلاس Evaluation نمودار های میله ای خواسته شده را ترسیم میکند.

۷ پیوست

در مقایسه با نسخه های قبلی ارسال شده، نوع تقسیم دیتاست در این گزارش و کد نهایی اصلاح شده است که در نتیجه ی آن تمامی نتایج گزارش شده در دو نسخه قبلی فرستاده شده فاقد اعتبار هستند و صحیح نیستند. همچنین در بخش پیشبینی الگوریتم فیلتر جمعی یک باگ وجود داشت که در این نسخه نهایی اصلاح شده و رفع شده است. همچنین در این نسخه فایل های مربوط به ماتریس های شباهت ذخیره شده اند و از این لینک قابل دانلود هستند. برای اجرای فایل Evaluations.py نیاز است تا فایل های ماتریس های شباهت در کنارش در یک دایرکتوری قرار گیرند.