گزارش تمرین ۴ رایانش ابری

بردیا اردکانیان - ۹۸۳۱۰۷۲  
آراد فیروزکوهی - ۹۸۳۱۰۴۷

تمامی پروژه در یک نوت‌بوک در google collab در این لینک انجام شده. برای این کار در نوتبوک گوگل کولب را با کمک دستوراتی یک کلاستر اسپارک راه اندازی می‌کنیم. این کار تنها به جهت افزایش سرعت محاسبات می‌باشد و چون سیستم‌های ما ضعیفه =)

**گام محیط پیاده سازی**

جاوا و اسپارک دانلود و نصب شده، و دیتا از داخل گوگل درایو خوانده می‌شود. توجه کنید که دیتاست‌ها را در ادرسی در گوگل درایو قرار داده‌ام.



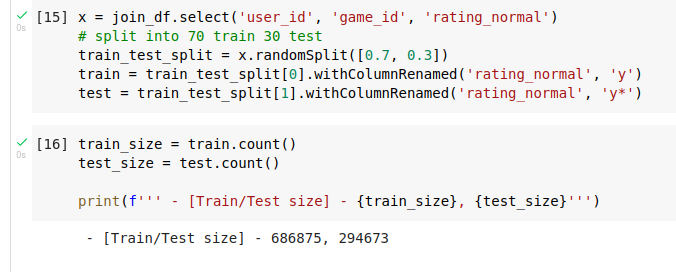
همانطور که مشاهده می‌کنید باینری‌های جاوا و اسپارک دانلود شده‌اند و در نهایت با کمک pip اسپارک را راه می‌اندازیم.

**گام داده‌گان**

در این گام دیتافریم‌های مربوطه به فایل games و rating خوانده و کش می‌شوند. توجه کنید اگر کش نکنیم سرعت action‌ها و transformation‌ها، مدل کردن و یادگیری مدل کمتر می‌شود. در مرحله بعدی می‌بایست ورودی‌های مدل را نرمالایز کنیم تا در بازه ۰ تا ۱ قرار گیرند. امتیاز داده شده از طرف کاربر بین ۱ تا ۵ و امتیاز داده شده به بازی از طرف منتقدین بین ۱ تا ۱۰۰ می‌باشد. پس به ترتیب دو ردیف جدید در دیتافریم آماده می کنیم که حاصل موارد فوق را تقسیم بر ۵ و ۱۰۰ ذخیره می‌کند تا در بازه ۰ و ۱ قرار بگیرند.

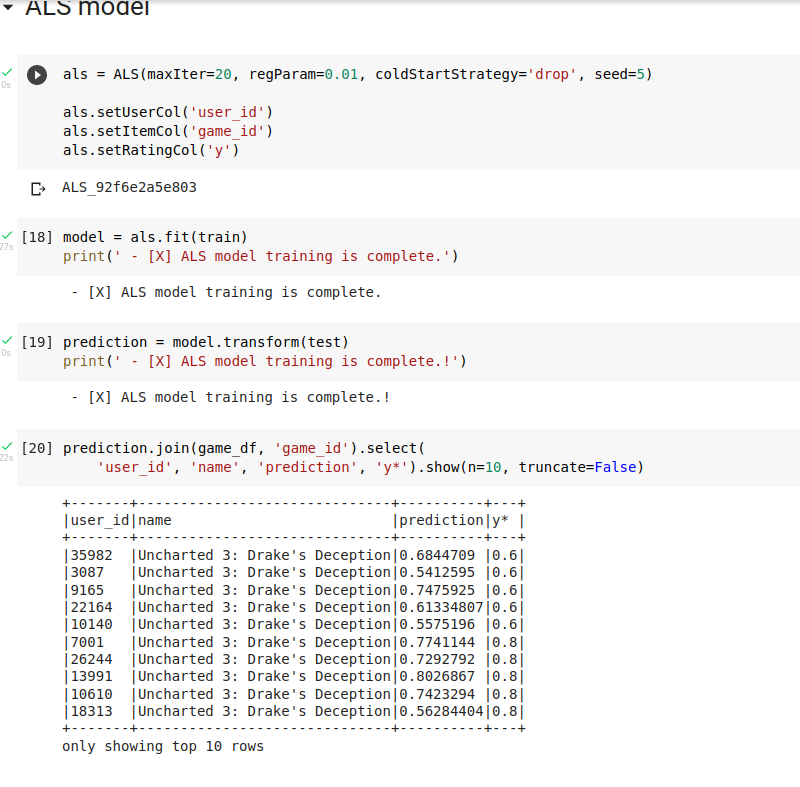
  
**آماده سازی داده اموزشی و تست**

در مرحله قبلی برای سادگی کار دو دیتافریم را بر اساس ردیف مشترک game\_id با یکدیگر ادغام می‌کنیم. برای آموزش مدل تنها به داده شناسه کاربر، شناسه بازی و امتیاز داده شده به بازی توسط کاربر نیاز داریم. دقیقا با همین سه ورودی می‌توانیم شباهت را بیابیم که کدام کاربر‌ها علایق مشترکی دارد یا به عبارتی به بازی‌های مشابه نمرات مشابهی می‌دهند. انتخاب ۳۰ ۷۰ تماما بر اساس تجربه بوده می‌تواند هر عددی باشد.



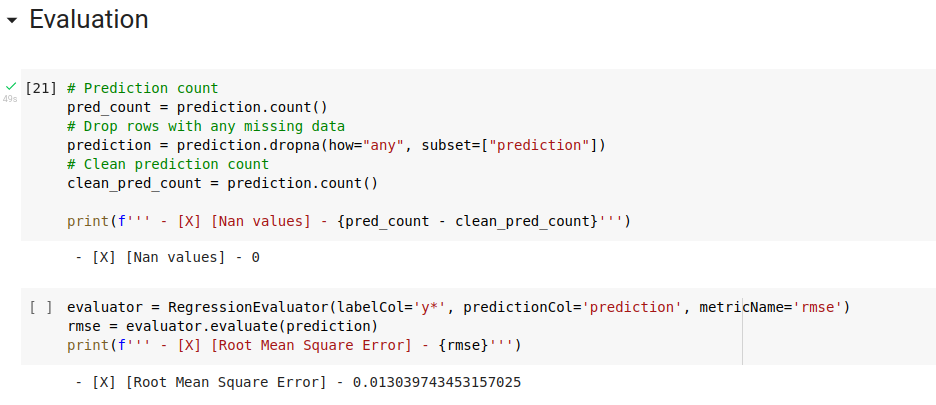
**آموزش مدل**

در نهایت یک مدل ALS آموزش می‌دهیم تا بتوانیم تخمینی بر امتیازات کاربران داشته باشیم. در این مدل، ابتدا ماتریس آیتم‌ها فریز شده و gradient descent بر روی کاربران اجرا شده، و سپس ماتریس کاربران فریز شده و gradient descent را روی ماتریس آیتم‌ها اعمال میکنیم. این کار باعث امکان موازی‌سازی الگوریتم و استفاده آن در دیتابیس‌های بزرگ و توضیع‌شده را می‌دهد. همچنین این الگوریتم باعث رفع اشکال‌هایی مانند cold start می‌شود. در نهایت خروجی مدل برای داده تست را مشاهده می‌کنیم.

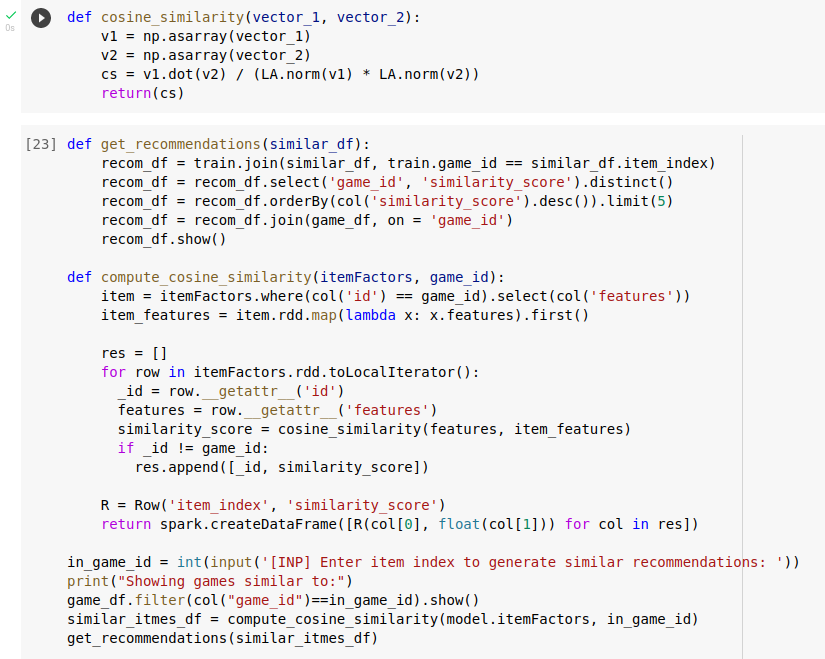


**ارزیابی مدل**

برای ارزیابی مدل از Root-mean square error استفاده شده است. می‌توان از sse یا هر تابع هزینه دیگری استفاده شود صرفا سلیقه ما rmse بوده. قبل از محاسبه rmse متوجه شدم بعضی از ردیف‌ها داده ندارند و باعث می‌شود عملیات به مشکل منطقی بخورد و exception دهد. برای همین تمامی ردیف‌هایی که این مشکل را دارند حذف می‌کنیم تا داده‌ها تمیز شوند.



در نهایت از این مدل به کمک شباهت کسینوسی مواردی که بیشترین شباهت را دارند را انتخاب می‌کنیم.



**شباهت کسینوسی**

تابع cosine\_similarity فرمول شباهت کسینوسی را بین دو بردار پیاده‌سازی می‌کند. تابع compute\_cosine\_similarity شباهت زوج موارد را بر اساس فیچر‌های مدل و تابع قبلی محاسبه می‌کند. در نهایت در تابع get\_recommendations بر اساس شباهت موارد را مرتب کرده و چند مورد شبیه‌تر را انتخاب می‌کنیم.

**خروجی**

همانطور که می‌بینید مدل به سادگی توانسته برای بازی company of heros و شباهت نمره داده شده توسط کاربران ۵ بازی دیگر را پیشنهاد دهد. فقط اون گیمری که هم chuchu rocket بازی می کنه هم hitman آدم جالبیه :)

