

چالش های پروژه

۱. چالش عملکرد در محاسبات امتیازدهی

چالش:

محاسبه مداوم میانگین امتیازات و تعداد زیاد آنها با کوثری های مکرر به دیتابیس، باعث کاهش عملکرد سیستم می شد.

راه حل:

پیاده سازی سیستم کش گذاری Redis برای آمار پست ها:

این راه حل، آمار پست ها (میانگین امتیاز و تعداد امتیازات) را به مدت ۵ ساعت (۱۸۰۰۰ ثانیه) در کش ذخیره می کند و فقط زمانی که کش منقضی می شود، مجدداً محاسبه می شود. و در این زمان هم هر درخواست جدیدی که بیاید به این عدد اضافه شده و در لحظه کاربر امتیاز های آپدیت شده رو می بیند اما به دیتابیس کوثری زده نمی شود و هر ۵ ساعت یکبار به دیتابیس کوثری زده می شود که به پرفورمنس کمک می کند.

۲. جلوگیری از تغییرات عمدی در امتیازدهی

چالش:

کاربران می توانستند در مدت زمان کوتاه، به صورت هماهنگ شده از شبکه های مجازی یا کانال ها به یک پست نمره خیلی پایین یا خیلی بالا بدهند.

راه حل:

پیاده سازی سیستم پردازش دسته ای امتیازها با وضعیت گذاری به صورت (در انتظار):

در این روش به جای اینکه هر رای به صورت مستقیم بعد از ایجاد روی امتیاز اصلی پست تاثیر بگذارد، امتیازهای جدید به کمک داشتن وضعیت pending در صف برای تایید می مانند و برای مثال در این کد عدد ۵۰۰ امتیاز در نظر گرفته شده برای هر پست که بعد از ۵۰۰ امتیاز به یک پست، تمام امتیازات آن پست با هم میانگین گرفته می شوند و با یک ضریب خطای ۰.۸ چک می شوند و اگر این میانگین نسبت به میانگین اصلی پست تفاوت بیش از ۰.۸ داشته باشد، مشکوک است و آن امتیاز ها دیگر valid نیستند. همچنین کرون جابی تعریف می شود که اگر تعداد امتیاز های پستی به ۵۰۰ نرسید هر شب برای مثال در ساعت ۲ یا ساعتی که سرور خلوت تر است به این پست ها نیز رسیدگی شود.

حالت دیگری هم می شد در نظر گرفت و آن اینکه امتیاز های پندینگ را هم تاثیر داد و بعد از حسابرسی اگر valid نبودند آن ها را حذف کرد. اما با این کار وضعیت به نمایش درآمدن پست ها برای کاربر اگر بر اساس میانگین امتیازات باشد حس خوبی را منتقل نمی کند و به عنوان مثال پستی که ارزش بالایی دارد برای مدتی به پایین می رود و پس از واریسی امتیازات و invalid شدن رای ها دوباره به بالا بازمی گردد که شاید خیلی جالب نباشد.

همچنین می توان این مسئله را به صورت خیلی دقیق تر به کمک بحث های دیتایی و به کمک دیتا آنالیز حل کرد و به کمک فرمول ها و همچنین نمودار نرمال خیلی نتیجه دقیق تری داشت که با توجه به زمان داده شده برای تسک پیاده سازی همچین ساختار هایی خیلی پیچیده و سخت می شد.

یکی دیگر از راهکار ها هم بررسی میانگین بر اساس بازه زمانی است که در بازه زمانی های مشخص مثلاً هر ۲ ساعت این سیستم ولیدیشن را اجرا کرد و آن را نسبت به رفتار کاربران در آن همان زمان در روز های گذشته برای آن پست در نظر گرفت و رفتار های مشکوک را بررسی کرد.

این مورد عضو چالش ها نبود اما با توجه به اینکه این سیستم برای یک شرکت در نظر گرفته شده است و مقیاس هم خیلی بزرگ است برای جدول امتیازات به جای آپدیت کردن هر row به ازای هر امتیاز یک row جدید ذخیره می کنم تا به فرایند تغییر امتیاز کاربران هم بتوان دسترسی داشت و همچنین به کمک فرایند soft delete و با داشتن فیلد is_active می توان آخرین امتیاز هر یوزر به هر پست را به دست آورد و آن را فقط برای محاسبات اصلی در نظر گرفت.