بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

فاز سوم پروژه درس سامانه‌های یادگیری ماشین

عنوان:

استقرار سیستم

پروژه SentiMovie

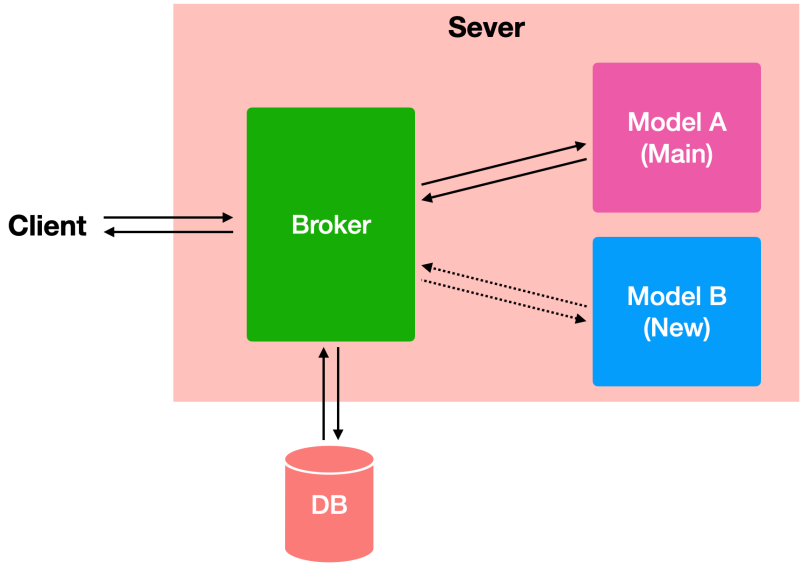
مدرس:

دکتر علی زارع‌زاده

نام و نام‌خانوادگی:

محمدحسین موثقی‌نیا

۲۰ مرداد ۱۴۰۲



شکل ۱: معماری سیستم

۱ مقدمه

در این پروژه بناست تا با استفاده از دیتاست مربوط به سایت rottentomatoes تلاش کنیم دو مدل یادگیری ماشین آموزش دهیم که یکی بناست به ازای هر جمله ورودی، احساسات مثبت یا منفی آن را تحلیل کند و به عنوان خروجی اعلام کند.

۲ معماری سیستم

معماری کلی استفاده شده در این پروژه در شکل ۱ آمده است. کارگزار[۱] وظیفه‌ی مدیریت درخواست‌ها و پایگاه داده‌ها را در اختیار دارد و با دریافت درخواست‌های تحلیل احساسات از جانب کاربر، آن را برای دو سرویس دیگر که هرکدام یک مدل تحلیل احساسات را پیاده‌سازی کرده‌اند ارسال می‌کند و با دریافت پاسخ از آن‌ها، درخواست کاربر را پاسخ می‌دهد. همچنین کارگزار کنترل پایگاه داده‌ها را در اختیار دارد و پاسخ درخواست‌های کاربران را در آن پایگاه داده ذخیره‌سازی می‌کند. قابلیت جستجو در این پایگاه داده نیز در اختیار مدیر سامانه قرار گرفته است. کاربر می‌تواند از طریق نقاط اتصال[۲] زیر سه دستور متفاوت را برای کارگزار ارسال نماید.

- نقطه‌ی اتصال /predict : سرویس تحلیل احساسات را به کاربر ارائه می‌دهد. کارگزار با دریافت این نوع درخواست، آن را به شکل همزمان برای دو مدل A و B ارسال می‌کند و پاسخ دریافت شده از

[۱] Broker
[۲] End-point

جانب مدل A را برای کاربر ارسال می‌کند. پاسخ ارسال شده از جانب مدل B نیز در پایگاه داده‌ها ذخیره می‌گردد ولی برای کاربر ارسال نمی‌شود. دلیل این نوع پیاده‌سازی در قسمت‌های بعدی آماده است.

- نقطه‌ی اتصال /predict/explainable : سرویس تحلیل احساسات توصیف‌پذیر را به کاربر ارائه می‌دهد. این سرویس مشابه سرویس سابق است. با این تفاوت که پاسخ برگشتی در قالب صفحه‌ی HTML خواهد بود که نشان می‌دهد هر بخش از متن ارسال شده در درخواست، به چه اندازه در تصمیم‌گیری نهایی مدل دخیل بوده است.
- نقطه‌ی اتصال /texts : سرویس مشاهده‌ی داده‌های درون پایگاه داده‌ها را به مدیر ارائه می‌دهد. مدیر سیستم با درخواست به این نقطه‌ی اتصال می‌تواند اطلاعات درخواست‌های ذخیره‌شده در پایگاه داده را به شکل کامل مشاهده نماید.

۳ استقرار

به منظور استقرار از سیستم هم‌روش استفاده شده است. الگوی استقرار مورد استفاده در این سیستم به صورت Dynamic deployment on server (server-less) می‌باشد. برای این منظور با استفاده از کوبرنتیز با محدودیت برای هرکدام از اپلیکیشن‌ها استفاده شده است که بین ۱ تا ۵ پاد مختلف می‌تواند برای هر کدام از اپلیکیشن‌ها بالا بیاید. همچنین نحوه پردازش ورودی‌ها دو نوع است، خدمات ارائه شده هم به صورت آنلاین و هم به صورت Batch می‌باشد که درصورتی که تعداد زیادی متن به عنوان ورودی داده شود، به منظور تحلیل مناسب به صورت Batch پردازش می‌شود.در غیر این صورت پردازش به صورت آنلاین خواهد بود. همچنین برای استراتژی استقرار از روش Silent استفاده شده است و مدل‌ها به صورت موازی خدمات داده و خروجی مدل دوم ذخیره سازی می‌شود ولی به کاربر نمایش داده نمی‌شود. دلیل انتخاب این نوع استراتژی سادگی آن و همچنین کم هزینه تر بودن آن بوده است. می‌توانستیم از روش قناری هم استفاده کنیم اما با توجه به این که پشتیبانی توسط هم‌روش انجام نمی‌شد به صورت Silent انجام شده است. همچنین دلیل انتخاب الگوی استقرار کم هزینه بودن و همچنین سادگی استقرار است و نکته مهم تر پرداخت به اندازه استفاده می‌باشد.

۴ اتوماسیون

به منظور پیاده‌سازی اتوماسیون، از Github Actions استفاده شده است. سیستم CICD این سامانه هنگامی که عمل push در گیت‌هاب پروژه انجام گیرد، به شکل خودکار فرایند خود را آغاز می‌کند. بعد از انجام تست‌های لازم روی کدهای پروژه، ابتداImageهای داکر لازم برای استقرار پروژه Build می‌شود. پس از ساخته‌شدن این Imageها، فرایند استقرار آن‌ها روی سرورهای هم‌روش صورت می‌گیرد تا استقرار سامانه‌ی بروزرسانی‌شده تکمیل شود.

۴-۱ End-to-End Testing

در قسمت تست این سامانه، به ازای هر کدام از سرویس‌های ارائه شده‌ی تحلیل احساسات، مجموعه‌ای از تست‌ها ایجاد شده است. در صورت وقوع خطا در هرکدام از بخش‌های فرایند تست، با یک Exception مواجه خواهیم شد که نشان‌دهنده‌ی عدم آمادگی برای استقرار آخرین ورژن کد است. برای اینکه این فرایند تست در اتوماسیون گنجانده شود، در قسمت CICD استقرار سامانه، یک job تحت عنوان test نوشته شده است که تمامی تست‌های لازم را روی کد انجام می‌دهد. بنابراین در صورت وقوع هرگونه مشکل در فرایند تست، به شکل خودکار از استقرار مجدد سامانه جلوگیری می‌شود.

۵ مانیتورینگ

برای رصد این سیستم تحلیل احساسات، از mlflow استفاده کرده‌ایم. ابتدا این نرم‌افزار مانیتورینگ را به یک Docker Container تبدیل کرده‌ایم. سپس با استفاده از کتابخانه‌ی واسط این نرم‌افزار برای زبان برنامه‌نویسی پایتون، اطلاعاتی مانند میزان مصرف منابع پردازشی مانند حافظه و پردازنده و همچنین میزان تاخیر پردازشی سیستم را اندازه‌گیری کرده‌ایم. سپس این مقادیر را به Container اجرا شده‌ی این نرم افزار ارسال می‌کنیم تا در پایگاه داده‌ی آن قرار گیرد. با ورود به پنل مدیریتی ارائه شده توسط این نرم‌افزار، می‌توان به نمودار این اطلاعات در طی زمان دسترسی داشت و معیارهای متفاوت را مورد بررسی قرار داد.

۶ چالش‌ها

- در صورتی که MLflow خاموش شود، تمامی اپلیکیشن ها ارور می‌دهند که امکان اجرا ندارند و دلیل این قضیه عدم ذخیره نتایج ران‌ها در حافظه ثابت بود. برای این منظور یک حافظه ثابت ساخته شد و تمامی ران‌ها روی آن ذخیره شد.

- زمانی که سیستم نمی‌توانست به مدل متصل شود، کلا ارور می‌داد و کرش می‌کرد، برای این منظور از کنترل خطا استفاده شد و تلاش دوباره انجام می‌شود.
- مشکل سرعت بسیار پایین اجرای سیستم اتوماسیون استقرار بر روی سرورهای گیت‌هاب، با استفاده از یک سرور شخصی که به عنوان اجرا کننده به گیت‌هاب ارائه شد سرعت ارتقا پیدا کرد.