باسمه تعالی

پیشنهاد پروژه

توسعه بستر MLOps

به سفارش

شرکت سامانه گستر سحاب پرداز

ارائه دهنده

شرکت فناوری اطلاعات آرمان دید نو

(آدین)

بهار ۱۴۰۱

فهرست مطالب

۱- مقدمه

سند حاضر به منظور ارائه یک پیوست فنی با هدف انجام پروژه «**توسعه بستر MLOps**» به سفارش شرکت سحاب پرداز تهیه شده است. بدین منظور در ابتدا تعاریفی از ماهیت MLOps و کارکردهای آن ارائه شده و دلیل استفاده از این بستر و همچنین مشتریان هدف آن مشخص شده است. سپس راهکار پیشنهادی شرکت آدین و جزئیات مربوط به معماری مفهومی، معماری فیزیکی، معماری منطقی و محدودیت‌ها و فرضیات تشریح شده و پس از آن، موارد مربوط به اجرای پروژه مانند روش تست و ارزیابی قابلیت‌ها، نحوه آموزش، نصب، راه‌اندازی و تحویل محصول نهایی و منابع مورد نیاز برای پروژه ارائه شده است. در نهایت نیز فازبندی اجرای پروژه توسط شرکت آدین، تحویل‌دادنی‌های هر فاز و زمانبندی تحویل خروجی‌ها آورده شده است. همچنین به پیوست این سند، شرحی از نیازمندی‌ها و نکات کلیدی فنی مورد نظر کارفرما درخصوص توسعه و پیاده‌سازی این بستر آمده است و که شامل برخی جزئیات مد نظر کارفرما درخصوص نحوه پیاده‌سازی و نمونه عملیاتی برای پیاده‌سازی پروژه می‌باشد..

۲- تعاریف

MLOps به مجموعه‌ای از فرایندها، ابزارها و شیوه‌ها جهت مدیریت چرخه توسعه مدل‌های یادگیری ماشین در یک محیط عملیاتی اشاره دارد. همچنین این چرخه شامل همکاری بین دانشمندان داده، مهندسان و تیم‌های DevOps است به‌نحوی‌که این اطمینان حاصل شود که مدل‌ها به طور مؤثر توسعه داده شده، مستقر شده و پایش و به‌روزرسانی می‌شوند. هدف MLOps افزایش سرعت، قابلیت اطمینان و مقیاس‌پذیری مدل‌های یادگیری ماشین و فرایند توسعه این مدل‌ها در تولید است؛ درحالی‌که خطرات ناشی از ریسک عدم موفقیت را نیز کاهش می‌دهد. همچنین به‌کارگیری MLOps فرایند مدیریت را ساده‌تر کرده، کیفیت را افزایش می‌دهد و استقرار مدل‌های یادگیری عمیق و یادگیری ماشین در محیط‌های تولید در مقیاس بزرگ را خودکار می‌کند. لذا می‌توان گفت یکی از اهداف MLOps، بهبود خودکارسازی و ارتقای کیفیت مدل‌های تولید و درعین‌حال توجه به الزامات تجاری و نظارتی است.

در ادامه این بخش، برخی موارد کلی و کلیدی در موضوع هوش مصنوعی، یادگیری ماشین و عملیات مربوط به توسعه مدل‌های یادگیری ماشین آورده می‌شود.

۲-۱- چالش‌های سیستم‌های هوش مصنوعی

مدیریت سیستم‌های هوش مصنوعی در مقیاس بزرگ کار آسانی نیست و در این مسیر چالش‌های مهمی وجود دارد که تیم‌ها باید با آن مواجه شوند. برخی از این چالش‌ها در کاربرد یادگیری ماشین (ML) در یک محصول مقیاس‌پذیر عبارت‌اند از:

- **کیفیت و کمیت داده**: اطمینان از وجود داده‌های آموزشی با کیفیت بالا برای آموزش مدل‌های ML یک چالش بزرگ است.
- **عملکرد مدل**: مدل‌های ML گاهی اوقات ممکن است برای تعمیم به داده‌های جدید مشکل داشته باشند و ممکن است در صورت استقرار در تنظیمات دنیای واقعی ضعیف عمل کنند.
- **تفسیرپذیری مدل**: مدل‌های ML می‌توانند پیچیده و غیر قابل درک باشند و تفسیر پیش‌بینی‌ها و تشخیص مشکلات و مقایسه کیفیت آن‌ها، متخصصین را به چالش می‌کشد.
- **منابع محاسباتی**: آموزش و به‌کارگیری مدل‌های ML بزرگ و پیچیده می‌تواند به مقدار قابل توجهی از منابع محاسباتی نیاز داشته باشد که می‌تواند در یک محصول مقیاس‌پذیر چالش‌برانگیز باشد.
- **استقرار مدل**: استقرار مدل‌های ML در یک محیط تولید و ادغام آن‌ها در یک محصول می‌تواند چالش‌برانگیز باشد و نیاز به بررسی دقیق زیرساخت‌ها، حریم خصوصی داده‌ها و امنیت دارد.
- **نگهداری مدل**: مدل‌های ML باید با تغییر داده‌ها و نیازمندی‌های محصول، به‌روزرسانی و بازآموزی شوند که تکرار متناوب این فرایند در یک محصول مقیاس‌پذیر پیچیده و چالش‌زا است.
- **بایاس**: مدل‌های ML گاهی اوقات ممکن است سوگیری‌های موجود در داده‌های آموزشی را تداوم بخشند و توجه به این مسائل هنگام توسعه یک محصول مقیاس‌پذیر ML بسیار مهم خواهد بود.

۲-۲- کاربردهای MLOps

MLOps به‌ویژه در مورد استقرار مدل‌های ML در تولید اهمیت دارد. زیرا به سازمان‌ها کمک می‌کند تا مطمئن شوند که مدل‌هایشان در طول زمان دقیق، قابل اعتماد و کارآمد هستند. به‌طورکلی، MLOps با

خودکار کردن بسیاری از مراحل مربوط به آموزش، استقرار و مدیریت مدل‌های ML به دانشمندان و مهندسان داده اجازه می‌دهد تا با همکاری یکدیگر به ارائه سریع‌تر و کارآمدتر مدل‌های یادگیری ماشین دست یابند.

لذا می‌توان گفت بیشترین سود دهی استقرار و استفاده از MLOps در مواقعی است که به‌روزرسانی و بازآموزش مدل‌ها با داده‌ها و یا فیچرهای جدید مورد نیاز بوده و استفاده از آخرین نسخه آموزش داده شده در خروجی‌ها مهم باشد. همچنین در مقیاس‌های بزرگ (به لحاظ حجم و پیچیدگی داده یا به لحاظ حجم و پیچیدگی مدل) استفاده از بستر MLOps موجب افزایش سرعت و بهبود عملکرد مهندسان یادگیری ماشین و دانشمندان علم داده خواهد شد.

برای رسیدن به اهداف فوق، تیم‌هایMLOps معمولاً ترکیبی از ابزارها، فرایندها و شیوه‌های زیر را استفاده می‌کنند:

- کنترل نسخه برای مدل‌های ML، فیچرها و مجموعه‌داده‌ها
- آزمون خودکار و ادغام مداوم
- کانتینرسازی و ارکستراسیون برای توسعه، آموزش و استقرار مدل
- نظارت و ثبت گزارش برای ردیابی عملکرد مدل در تولید
- حاکمیت و انطباق برای مدل‌ها

۳-۲- مزایای MLOps

به‌طورکلی، MLOps یک چارچوب قدرتمند است که می‌تواند به تیم‌ها کمک کند تا پروژه‌های یادگیری ماشینی را به طور مؤثر مدیریت و مقیاس کنند. از سوی دیگر عواملی وجود دارد که ممکن است MLOps بهترین گزینه برای حل برخی از مسائل باشد. در اینجا چند نمونه از این دلایل ذکر می‌شود:

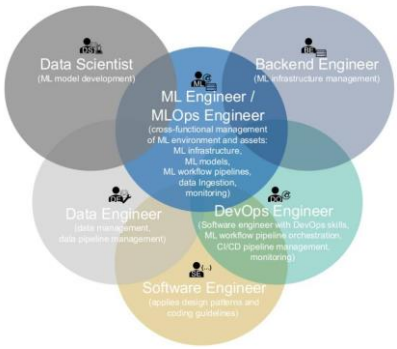
- **بهبود سرعت ایجاد و استقرار مدل**: در نتیجه مدیریت جامع چرخه زندگی یادگیری ماشین، به دلیل افزایش ارزش یادگیری ماشین.
- **زمان سریع‌تر جهت بازاریابی**: MLOps دانشمندان و مهندسان داده را قادر می‌سازد تا سریع‌تر تکرار کنند و مدل‌ها را سریع‌تر به تولید بفرستند. همچنین روش‌های یکپارچه‌سازی و تحویل مداوم، استقرار این سیستم‌ها را در تولید آسان‌تر می‌کند.

- **بهبود کیفیت مدل**: با خودکارسازی آموزش، اعتبارسنجی و استقرار مدل‌ها، احتمال خطای انسانی کاهش می‌یابد که منجر به عملکرد بهتر مدل می‌شود. همچنین، جهت اطمینان از ارائه پیش‌بینی‌های باکیفیت بالا، سیستم MLOps باید قادر به اندازه‌گیری model drift باشد. این امر احتمال دیدگاه‌های نادرست در فرضیه‌های مدل را کاهش می‌دهد.
- **همکاری بهتر**: MLOps همکاری بین دانشمندان داده، توسعه‌دهندگان و تیم‌های عملیاتی را تشویق می‌کند و تیم‌های متقابل را قادر می‌سازد تا به طور مؤثرتری با هم کار کنند. MLOps همکاری و شفافیت را در میان تیم‌های DataOps، مهندسین یادگیری ماشین، تحلیلگران تجاری/مدیران محصول، مهندسین تضمین کیفیت و مهندسین زیرساخت افزایش می‌دهد.
- **مقیاس‌پذیری**: خودکار کردن فرایند استقرار، مقیاس‌پذیری مدل‌ها را جهت رسیدگی به مقادیر بزرگ و افزایشی داده و جمعیت بیشتر کاربران آسان‌تر می‌کند.

۳- شرح فنی راهکار پیشنهادی توسعه بستر MLOps

۳-۱- فرضیات و محدودیت‌ها

در این بخش، به بیان فرضیات و محدودیت‌هایی که توسعه بستر MLOps با آن‌ها مواجه خواهد بود پرداخته شده است. این فرضیات و محدودیت‌ها با توجه به شرایط کنونی تیم‌های کارفرما و مجری مشخص شده‌اند و ممکن است در موارد بعدی از توسعه این بستر صادق نباشند.



شکل 1 – تخصص‌های مورد نیاز در توسعه MLOps

- **تنوع تخصص‌های مورد نیاز**: پیاده‌سازی یک بستر MLOps نیاز به بهره‌گیری از تخصص‌های مختلف و متنوع دارد که به نوبه خود در مقایسه با سایر انواع بسترها و محصولات نرم‌افزاری، قابل توجه است. تخصص‌های اصلی مورد نیاز جهت توسعه این محصول شامل مهندسی یادگیری ماشین، مهندسی داده، مهندسی DevOps و حتی برنامه‌نویسی و توسعه backend است. طبیعتا تیم مجری می‌بایست حائز تمامی این موارد باشد.
- **پیچیدگی یکپارچه‌سازی**: ادغام جریان‌های کاری ML در فرایندهای توسعه نرم‌افزار موجود منجر به تغییرات قابل توجهی در جریان‌های کاری و ابزارهای موجود بوده و تنها در صورتی ممکن خواهد بود که نرم‌افزارها و ابزارهای موجود، در تضاد با سرویس‌ها و مولفه‌های MLOps نباشند.
- **امنیت داده‌ها**: MLOps به مدیریت دقیق داده‌های حساس از جمله ذخیره‌سازی ایمن، کنترل دسترسی و نظارت بر استفاده از داده‌ها نیاز دارد.
- **امکان تامین منابع**: پیاده‌سازی MLOps نیازمند سرمایه‌گذاری قابل توجهی در فناوری، فرآیندها و کارکنان است.MLOps جهت استقرار مؤثر مدل ML در دنیای واقعی ضروری است، اما هزینه اولیه توسعه و راه‌اندازی آن به‌دلیل پیچیدگی و منابع مورد نیاز می‌تواند بسیار گران باشد. توانایی ایجاد و حفظ یک چرخه حیات MLOps به طور مؤثر به منابع کافی (افراد، زیرساخت‌ها و ابزارها) نیاز دارد. در برنامه حاضر، فرض بر امکان تامین منابع مورد نیاز (خصوصا سخت‌افزار و داده) جهت توسعه پروژه توسط کارفرما و بهره‌بردار است. طبیعی است هرگونه عدم موفقیت در تامین منابع مذکور، تحقق اهداف و شاخص‌های پروژه را با مخاطره مواجه و یا غیر ممکن می‌سازد.
- **استفاده از پردازنده‌های گرافیکی**: برای دستیابی به سرعت و کارایی معقول، مدل‌های می‌بایست با استفاده از پردازنده‌های گرافیکی مناسب آموزش داده شوند. البته به جهت صرفه‌جویی در هزینه‌ها و بنا بر صلاحدید کارفرما و بهره‌بردار، در حالت سرویس‌دهی، مدل‌ها می‌توانند از CPU به‌جای GPU استفاده کنند.
- **شاخص‌های مورد نیاز جهت محاسبه و تامین مقیاس‌پذیری**: روش‌های مختلفی برای پشتیبانی از حالت‌های مختلف استنتاج و پیش‌بینی در یادگیری ماشین وجود دارد. می‌توان به ازای هر مورد کاربرد، یک مدل مخصوص تولید کرد. از طرفی هم می‌توان تمام حالت‌های مختلف را توسط یک مدل (که با انواع داده‌ها و فیچرها آموزش داده شده) پاسخ داد. لذا تخمین دقیق

مواردی مانند «نرخ استخراج فیچرها»، «حجم فیچرها در feature storeها» و «تعداد مدل‌های مورد نیاز جهت ارائه پاسخ بهینه به نیاز مشتری» نیازمند شناخت ابعاد مختلف مسائل واقعی و داده‌های مربوط به آن‌ها است. از طرفی، این موارد، معماری استقرار سامانه و مقیاس‌پذیری آن را تحت تاثیر قرار خواهند داد. استفاده از پارادایم «یک مدل، یک سرور» یا رفتن به سمت «چند مدلی» در هر پاد، به‌کارگیری یا عدم به‌کارگیری زیرساخت‌های توزیع‌شده در سطح فایل‌سیستم یا پایگاه‌های داده، نحوه پیکربندی و معماری Kubernetes و سایر زیرساخت‌های DevOpsی پروژه و ... ازجمله این موارد هستند. با توجه به اینکه تا لحظه نگارش این سند، هنوز مسائل واقعی بهره‌برداران و داده‌های متناظر با آن‌ها مشخص نشده‌اند، امکان ارائه معماری دقیق فیزیکی و طرح استقرار و راه‌حل تامین مقیاس‌پذیری به‌صورت قطعی میسر نمی‌باشد.

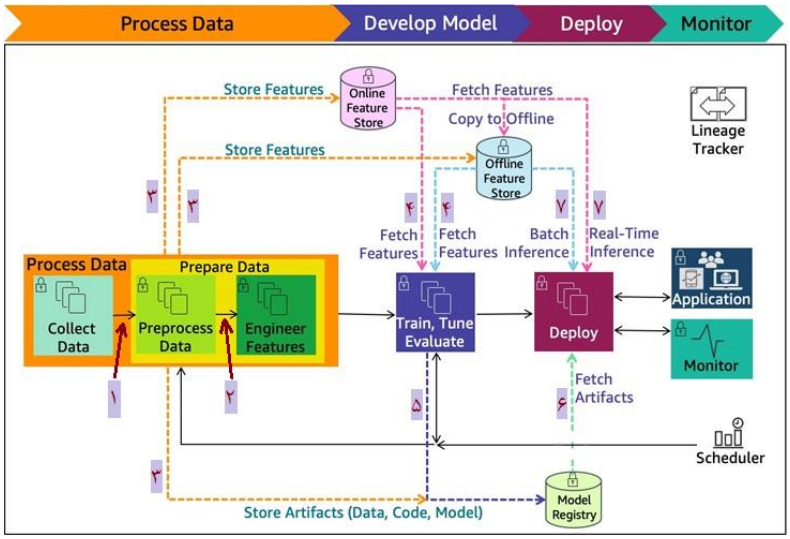
- **تخمین سخت‌افزار مورد نیاز در Production**: برای تخمین دقیق سخت‌افزار مورد نیاز در محیط Production، نیاز است علاوه‌بر موارد مذکور در بند فوق (شاخص‌های مورد نیاز جهت محاسبه و تامین مقیاس‌پذیری)، ابعاد دقیق نیازهای عملیاتی بهره‌بردار در حین استفاده از بستر MLOps نیز مشخص شود. این ابعاد شامل موارد زیر می‌باشند:
  - تعداد پایپ لاین‌ها
  - انواع مدل‌ها مورد نیاز جهت پشتیبانی
  - میانگین و بیشینه نرخ داده ورودی مورد استفاده
  - میانگین و بیشینه نرخ و تعداد سرو مدل (در ثانیه)
- **محدودیت دسترسی به محیط Production**: براساس اعلام کارفرما، به دلیل ملاحظات محرمانگی، امکان دسترسی به محیط Production برای تیم فنی مجری میسر نمی‌باشد. لذا تحویل در محیط Stage می‌بایست به گونه‌ای انجام گیرد که نمایندگان کارفرما قادر به نصب و راه‌اندازی محصولات توسعه داده شده در محیط Production باشند.
- **محدودیت انتخاب در فناوری‌ها**: تمامی فناوری‌های انتخابی در این طرح، می‌بایست با لحاظ کردن ملاحظات امنیتی و فناوری مد نظر کارفرما و بهره‌بردار انتخاب و استفاده شوند.

- **محدودیت دسترسی به داده‌های خام**: بر اساس اعلام کارفرما، امکان دسترسی به داده خام در محیط Production میسر نیست. لذا در این طرح، فرض شده که داده‌های مورد نیاز بهره‌بردار، به‌صورت آماده در اختیار مولفه ETL بستر MLOps قرار خواهد گرفت و نیازی به اتصال به ذیرساخت‌های ذخیره داده و واکشی داده از آن‌ها و تغییر داده در آن‌ها نمی‌باشد.
- **تخمین منابع مورد نیاز در Stage**: با توجه به مشخص نبودن ابعاد دقیق مسائل بهره‌بردار و عدم قطعیت محاسبات مربوط به منابع مورد نیاز؛ جهت تعیین منابع محیط Stage، فرض بر رعایت مقیاس حداقل ۲۰ یا ۲۵ درصد نسبت به Production است. بدیهی است هرچه میزان منابع Stage به Production شبیه‌تر و نزدیک‌تر باشد، نتایج آزمون‌های محیط Stage قابل‌اتکاتر و انتقال از Stage به Production سریع‌تر و کم‌هزینه‌تر خواهد بود.
- **انواع داده ورودی**: بر اساس اطلاعاتی که تا لحظه نگارش این سند موجود است، تمرکز مسائل بهره‌بردار فعلی روی داده‌های از جنس جدولی و متنی است. لذا تمامی موارد پیشنهادی در طرح بر اساس همین نوع داده طراحی و ارائه شده است. هرچند که ممکن است تا حد خوبی روی سایر انواع داده مانند تصویر، صوت و ... هم کارایی داشته باشد.
- **عدم قطعیت مصادیق فناوری**: باتوجه به ابهامات موجود در زمان نگارش این سند و همچنین تازگی و عدم بلوغ موضوع MLOps در کشور و حتی دنیا، تمامی مواردی که در بخش‌های مختلف سند درخصوص استفاده از فناوری‌ها، زیرساخت‌ها، معماری‌ها و ... آمده صرفا پیشنهادی بوده و بنا به شرایط و حسب اعلام تیم مجری و با توافق و تایید کارفرما قابل تغییر می‌باشند.
- **استفاده از زیرساخت‌های فعلی کارفرما**: طبق توافق انجام شده و با توجه به پیچیدگی‌های ذاتی MLOps و به جهت کاهش مخاطرات و پیچیدگی‌های پروژه، در ابتدای کار هیچ پیش‌فرض مشخص و محدودکننده‌ای درخصوص استفاده قطعی از زیرساخت‌های فعلی موجود در سبد محصولات کارفرما (اعم از نرم یا سخت) وجود ندارد. اما از طرفی طراحی‌ها و توسعه‌های انجام گرفته نباید متناقض با آن‌ها باشد به گونه‌ای که امکان یکپارچه‌سازی با سایر محصولات کارفرما را در آینده ناممکن سازد.
- **عدم لزوم بکارگیری تکنیک‌های یادگیری ماشین:** بسیاری از مسائل در فضای تولید، قابلیت حل به صورت الگوریتمیک را دارند و برای حل آن‌ها الزامی بر بکارگیری روش‌های یادگیری ماشین

نیست. همچنین، در برخی مسائل، تهیه داده‌های کافی برچسب‌خورده جهت آموزش مدل مقرون به‌صرفه نیست. بنابراین، عامل اصلی تعیین‌کننده در استفاده یا عدم استفاده از MLOps، اولا توجیه منطقی استفاده از یادگیری ماشین در حل با کیفیت مسئله نسبت به سایر روش‌های هوشمند و ثانیا امکان تهیه داده برچسب‌خورده به میزان کافی برای مسئله می‌باشد.

۲-۳- معماری منطقی

در این بخش به معرفی مولفه‌های اصلی در معماری یک بستر MLOps در سطح منطقی و مفهومی (بدون ذکر مصادیق فناوری‌ها و زیرساخت‌ها) پرداخته شده؛ همچنین جریان داده اصلی و سناریو تعامل مولفه‌ها شرح داده شده است. شکل 2 یک نمای کلی از مولفه‌ها، وظایف و مراحل اصلی موجود در مفهوم و منطق MLOps را نمایش می‌دهد. همین معماری و شِما، پایه و اساس پیشنهاد توسعه مندرج در این سند است.



شکل 2 – نمای کلی معماری منطقی بستر MLOps

۱-۲-۳- مولفه‌های اصلی

به‌طورکلی نمای اصلی در معماری منطقی MLOps شامل اجزا و مراحل زیر است:

- **آماده‌سازی داده‌ها**: این مولفه جهت انجام اولین مرحله در چرخه حیات یادگیری ماشینی است و شامل **جمع‌آوری، تمیزکردن و پیش‌پردازش** داده‌هایی است که برای آموزش و ارزیابی مدل‌های ML استفاده می‌شود. هدف از این مرحله جمع‌آوری و آماده‌سازی داده‌ها در قالبی است که به‌راحتی توسط مدل ML مصرف شود و اطمینان حاصل شود که داده‌ها از کیفیت بالایی برخوردار هستند. آماده‌سازی داده‌ها ممکن است شامل کارهایی مانند نرمال‌سازی داده‌ها، تشخیص نقاط پرت و مهندسی ویژگی‌ها باشد.
- **انباره داده آنلاین/آفلاین ویژگی‌ها:** ویژگی‌ها پس از استخراج، وارد انباره‌های داده می‌شوند. انباره داده آفلاین، تاریخچه مقادیر ویژگی‌ها را نگهداری می‌کند و در مرحله یادگیری مدل مورداستفاده قرار می‌گیرد. همچنین برای استنتاج به صورت دسته‌ای نیز مورد استفاده قرار می‌گیرد. انباره داده آنلاین، ورودی ویژگی‌های استخراج‌شده از داده‌های خام ورودی را نگهداری می‌کند و برای تامین انباره داده ویژگی‌های آفلاین مورداستفاده قرار می‌گیرد. همچنین بسته به نوع مدل، ممکن است در مرحله استقرار و سرو مدل نیز مورد بهره‌برداری قرار گیرد.
- **مدیریت کدهای منبع:** تمامی کدهای استفاده شده در چرخه MLOps از جمله کد منبع مدل‌های یادگیری ماشین، الگوریتم‌ها و خطوط لوله در یک مولفه با این نام ذخیره و نگهداری شده و بنا به نیاز در هر مرحله واکشی و استفاده خواهند شد. در شکل 2 این مولفه با نام Model Registry نمایش داده شده است.
- **انتخاب، آموزش، تنظیم و ارزیابی مدل:** در این مرحله، دانشمندان و مهندسان داده، معماری، الگوریتم و فراپارامترهای مدل ML مناسب را بر اساس ویژگی‌های داده‌ها و مسئله‌ای که باید حل شود، انتخاب می‌کنند. سپس با استفاده از داده‌های آماده شده و با تنظیم پارامترهای مدل برای به‌حداقل‌رساندن خطا بین خروجی‌های پیش‌بینی‌شده و خروجی‌های واقعی، اقدام به اجرای فرایند آموزش و تنظیم مدل می‌نمایند در نهایت به ارزیابی عملکرد مدل آموزش‌دیده با مقایسه

پیش‌بینی‌های آن با مجموعه‌ای از نتایج شناخته‌شده (که داده‌های آزمایشی نیز نامیده می‌شود)، و تعیین میزان تعمیم مدل به داده‌های نادیده جدید می‌پردازند.

- **یکپارچه‌سازی مداوم/ استقرار مستمر (CI/CD)** :این جزء ادغام و استقرار مدل‌های یادگیری ماشین را خودکار می‌کند. به عبارت دیگر، مدل‌های آموزش‌دیده را در یک محیط تولید مستقر می‌کند، جایی که می‌توان از آن‌ها برای پیش‌بینی استفاده کرد. این مرحله شامل در دسترس قراردادن مدل برای کاربران نهایی است، خواه از طریق استقرار مدل در یک محیط تولید یا با ارائه یک API برای دسترسی دیگران به مدل باشد.
- **نظارت مدل:** این مرحله شامل نظارت بر عملکرد مدل مستقر در طول زمان و انجام به‌روزرسانی‌ها یا تنظیمات موردنیاز برای حفظ دقت و عملکرد مدل است.این مؤلفه عملکرد مدل‌های مستقر شده را نظارت می‌کند و در مورد دقت و قابلیت اطمینان آن‌ها بازخورد ارائه می‌کند.
- **مدیریت مدل:** این مؤلفه چرخه عمر مدل‌های یادگیری ماشین از جمله نسخه‌سازی، به‌روزرسانی[1] و بازنشستگی مدل‌ها[2] در صورت لزوم را مدیریت می‌کند. همچنین مدل‌های آموزش‌دیده و فراداده‌های مربوط به مدل‌ها مانند نسخه، دقت و داده‌های آموزشی را ذخیره می‌کند.
- **نرم‌افزار نهایی:** این بخش که در شکل 2 با نام Application آمده است، محل استقرار کد یا API نهایی برای استفاده از مدل مستقر شده است. در واقع اینجا جایی است که درخواست کاربر را دریافت کرده و با استفاده از استنتاج به کمک مدل، جواب را برای کاربر ارسال می‌کند.

۲-۲-۳- جریان داده

جریان داده‌ای بین مؤلفه‌های اصلی معماری منطقی که در شکل 2 از روند کاری بستر MLOps ملاحظه می‌شود، در این بخش شرح داده خواهد شد. این جریان داده‌ای ناظر بر مراحل طی شده در MLOps از زمان

[1] بروزرسانی مدل بر اساس شرایط مشخص (مانند زمانبندی قبلی یا بر اساس پارامترهای مورد پایش) انجام می‌شود. در صورت بروز هر یک از شرایط تعیین شده، عملیات آموزش مدل مجدداً اجرا شده و نسخه جدید مدل مستقر می‌شود.

[2] این مرحله شامل بازنشستگی مدل‌هایی است که دیگر موردنیاز نیستند یا دیگر عملکرد خوبی ندارند، تا اطمینان حاصل شود که فقط مدل‌هایی با عملکرد خوب به کار گرفته می‌شوند.

ورود داده تا هنگام آموزش و استقرار مدل ML را شامل می‌شود. همان‌طور که در شکل 2 دیده می‌شود جهت تسهیل در فهم توضیحات، کلیه مراحل جریان داده اصلی شماره‌گذاری شده‌اند. نحوه عملکرد MLOps در هر یک از مراحل به شرح زیر است:

۱. ابتدا داده جمع‌آوری شده برای پیش‌پردازش ارسال می‌شود. این پیش‌پردازش می‌تواند در بستر اصلی MLOps، به‌صورت مستقل و یا در بستر داخلی Feature Store انتخابی مستقر شود.

۲. داده پس از پیش‌پردازش، برای استخراج ویژگی‌ها مورد بررسی قرار گرفته و ویژگی‌های مورد نیاز از آن استخراج می‌شوند.

۳. ویژگی‌های استخراج شده در دو انباره آنلاین و آفلاین ذخیره‌سازی می‌شوند. بسته به نوع پردازش داده (دسته‌ای یا جریانی)، ویژگی‌ها ممکن است ابتدا در انباره آنلاین ذخیره شده و بعد در انباره آفلاین کپی شوند (حالت جریانی)، یا از ابتدا مستقیماً در انباره آفلاین ذخیره شوند (حالت دسته‌ای). همچنین فراداده در مورد داده ورودی (مثلاً زمان دریافت ورودی، حجم داده، ...) در Model Registry ذخیره می‌شود.

۴. در مرحله آموزش مدل، ویژگی‌ها از انباره‌ها خوانده شده و مورد استفاده قرار می‌گیرند. اینجا نیز با توجه به دسته‌ای یا جریانی بودن پردازش، داده‌ها از انباره آفلاین یا آنلاین خوانده می‌شوند.

۵. پس از پایان فرایند آموزش، مدل ساخته شده در محل ذخیره‌سازی مدل‌ها قرار می‌گیرد.

۶. در مرحله استقرار، مدل مورد نظر از محل ذخیره‌سازی مدل‌ها بارگذاری می‌شود.

۷. هنگام استنتاج، بسته به نوع مدل (وابسته به داده، یا مستقل از داده) و همچنین نوع درخواست کاربر (تکی، یا دسته‌ای)، ممکن است نیاز به دسترسی به انباره داده آنلاین و یا آفلاین باشد و اطلاعات از انباره‌ها به محل استقرار مدل منتقل شوند.

۳-۳- معماری فیزیکی

منظور از معماری فیزیکی، نگاشت مولفه‌های معماری منطقی به فناوری‌های مشخص جهت توسعه یک نمونه از بستر MLOps است. انتخاب فناوری‌ها در معماری فیزیکی MLOps به نیازهای کاربر و پروژه‌های یادگیری ماشینی درحال‌توسعه بستگی دارد. معماری با توجه به نیاز سازمان، ممکن است نیاز به سطوح

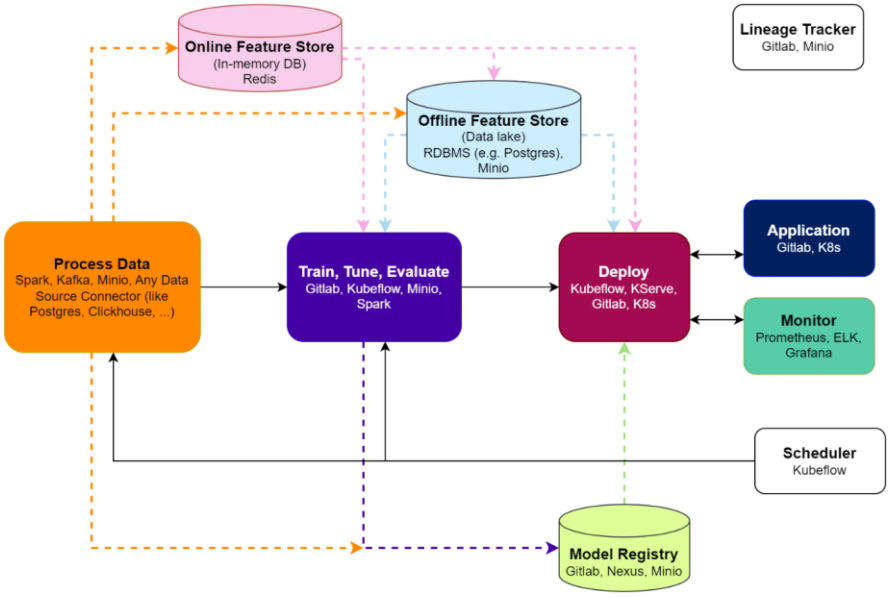
مختلفی از مقیاس‌پذیری انعطاف‌پذیری و کارایی داشته و قادر به رسیدگی به نیازهای جریان کار یادگیری ماشین و ادغام و استقرار مداوم مدل‌های یادگیری ماشین باشد. معماری فیزیکی MLOps به اجزا و فناوری‌های نرم‌افزاری مورداستفاده برای پیاده‌سازی معماری منطقی MLOps اشاره دارد و می‌تواند شامل موارد زیر باشد:

- **زیرساخت محاسباتی:** شامل سرورهای داخلی، ماشین‌های مجازی مبتنی بر ابر یا ترکیبی از اینهاست.
- **ذخیره‌سازی داده‌ها:** شامل پایگاه‌های داده، دریاچه‌های داده یا سیستم‌های فایل مورداستفاده برای ذخیره داده‌های آموزشی، داده‌های آزمایشی و خروجی‌های مدل است.
- **شبکه‌سازی:** شامل اجزای سخت‌افزاری و نرم‌افزاری که از ارتباط بین اجزای مختلف معماری MLOps پشتیبانی می‌کند.
- **نظارت:** شامل ابزارهای نظارت، تحلیلگرهای گزارش و داشبوردهایی که برای نظارت بر عملکرد مدل‌های یادگیری ماشین و گردش کار MLOps استفاده می‌شوند.
- **امنیت:** شامل فایروال‌ها، سیستم‌های کنترل دسترسی، رمزگذاری و سایر اقدامات امنیتی برای محافظت از داده‌ها و مدل‌های حساس درگیر در MLOps است.
- **ابزارها و پلتفرم‌ها:** شامل ابزارها و پلتفرم‌های نرم‌افزاری منبع‌باز و تجاری که برای یادگیری ماشین استفاده می‌شوند، مانند TensorFlow، PyTorch و غیره. برخی ابزار/پلتفرم برتر MLOps جهت مدیریت چرخه یادگیری ماشین عبارتنداز:
  - Kubeflow
  - MLFlow
  - TensorFlow Extended (TFX)
  - Amazon SageMaker
  - Azure Machine Learning
  - Google Cloud ML Engine
  - Data Version Control (DVC)

- H2O Driverless AI

- **خطوط لوله:** پلتفرمی برای ساخت، استقرار و مدیریت گردش‌کار یادگیری ماشینی سرتاسری است. این یک رابط بصری برای ایجاد و مدیریت گردش‌های کاری پیچیده و همچنین ابزارهایی برای آماده‌سازی داده‌ها، آموزش مدل و استقرار فراهم می‌کند.
- **سرو مدل**: بستری برای استقرار و مدیریت مدل‌های یادگیری ماشین در تولید است. این یک پلتفرم مقیاس‌پذیر و قابل‌اعتماد برای ارائه مدل‌ها به‌عنوان میکروسرویس، با پشتیبانی از استنتاج بلادرنگ و دسته‌ای فراهم می‌کند.
- **اپراتورهای آموزش مدل:** اپراتورهای آموزشی Kubeflow راهی برای استقرار و مدیریت مشاغل آموزشی توزیع شده در Kubernetes ارائه می‌دهند. آن‌ها به شما اجازه می‌دهند با استفاده از چارچوب‌های یادگیری ماشینی محبوب مانند TensorFlow، PyTorch و XGBoost، مدل‌ها را آموزش دهید.
- **متادیتا:** راهی برای ردیابی و مدیریت فراداده‌های مرتبط با گردش کار یادگیری ماشین ارائه می‌دهد. این به شما امکان می‌دهد اطلاعات مربوط به داده‌های مورداستفاده برای آموزش، مدل‌های آموزش‌دیده و فراپارامترهای مورداستفاده را ذخیره کنید.
- **نوت‌بوک‌های Jupyter:** نوت‌بوک‌های Kubeflow Jupyter راهی برای اجرا و مدیریت نوت‌بوک‌های Jupyter در Kubernetes ارائه می‌دهند. این به شما امکان می‌دهد گردش‌های کاری یادگیری ماشین خود را در یک محیط آشنا توسعه و آزمایش کنید.
- **تنظیم هایپرپارامترها:** یک چارچوب تنظیم است که به شما امکان می‌دهد فرآیند یافتن هایپرپارامترهای بهینه را برای مدل‌های یادگیری ماشین خود به طور خودکار انجام دهید. از انواع الگوریتم‌های تنظیم پشتیبانی می‌کند و می‌تواند با فریم‌ورک‌های معروف یادگیری ماشینی مانند TensorFlow، PyTorch و XGBoost استفاده شود.
- **داشبورد:** یک رابط مبتنی بر وب برای مدیریت و نظارت بر استقرار ارائه می‌دهد. این یک نمای جامع از تمام اجزای موجود در استقرار شما، و همچنین معیارها و گزارش‌های بلادرنگ برای هر جزء ارائه می‌کند.

برای معماری فیزیکی (مطابق با معماری منطقی ارائه شده در بخش قبل و نیز نیازها و مسائل مطرح شده توسط کارفرما)، نمونه‌ای مطابق شکل 3 پیشنهاد می‌شود که هر یک از فناوری‌های پیشنهادی قابل استفاده در هر مولفه نیز در آن آمده است. لازم به ذکر است که این پیشنهاد اولیه برای معماری فیزیکی است و با توجه به پیشرفت کار، هر یک از اجزای آن ممکن است در طول فرایند اجرا و پیاده‌سازی تغییر کنند.



شکل 3- معماری فیزیکی پیشنهادی برای MLOps

شرح پیشنهاد اولیه فناوری‌ها در لایه فیزیکی برای هر یک از مؤلفه‌های معماری منطقی در لایه فیزیکی به‌صورت زیر خواهد بود:

- **پردازش داده:** این بخش شامل جمع‌آوری داده، پیش‌پردازش داده و استخراج ویژگی‌ها است. برای جمع‌آوری داده، هر نوع منبع داده‌ای قابل پذیرش است. البته نگهداری داده ورودی در محدوده سامانه MLOps نیست و این سامانه تنها اتصال و خواندن داده از منابع داده‌ای متنوع مانند Postgres و Clickhouse را پشتیبانی می‌کند. این بخش شامل یک خطلوله برای پردازش داده است، که می‌تواند توسط Spark اجرا شود. نتایج مراحل مختلف در Minio ذخیره‌سازی می‌شود. نکته بعد این که پردازش داده در MLOps دو حالت دارد: دسته‌ای و جریانی. در حالت

جریانی، خروجی این بخش در Kafka قرار می‌گیرد و توسط Online Feature Store خوانده شده و برای آموزش استفاده می‌شود.

- **انباره ویژگی آنلاین:** ویژگی‌های ذخیره شده در انباره ویژگی آنلاین، عمدتاً در هنگام استنتاج مورد استفاده قرار می‌گیرند. ویژگی مورد نیاز این بخش، سرعت بالای پاسخگویی تکی است. در عین حال حجم داده مورد نیاز در این بخش نسبت به انباره ویژگی آفلاین به مراتب پایین‌تر است. در نتیجه بهترین گزینه برای این بخش، استفاده از یک پایگاه داده درون حافظه است که با توجه به حجم مورد نیاز مشتری، معمولاً Redis پاسخگوی نیاز مشتری خواهد بود. حتی در صورت نیاز به بالا بردن حجم داده مورد نیاز برای پاسخگویی به مشتری‌های مختلف در سرویس ابری، می‌توان از Redis cluster و یا شارد کردن داده روی چند سرویس مستقل Redis استفاده کرد. دیگر کاربرد این انباره، آموزش مدل به صورت جریانی می‌باشد.
- **انباره ویژگی آفلاین:** ویژگی‌های ذخیره شده در انباره ویژگی آفلاین، دو کاربرد دارند: برای آموزش مدل به صورت دسته‌ای، و همچنین برای استنتاج دسته‌ای. همچنین حجم داده بسیار زیادی باید در این انباره ویژگی ذخیره شود (توجه کنید که تاریخچه ویژگی‌های استخراج شده نیز در این انباره ذخیره می‌شوند). در نتیجه بهترین انتخاب برای این مؤلفه، استفاده از یک دریاچه داده (Data lake) است. با توجه به حجم داده مورد نیاز مشتری، می‌توان از یک RDBMS مانند Postgres برای حجم کم یا متوسط، و یا پایگاه‌داده‌های مقیاس‌پذیر مانند Clickhouse برای حجم بالا استفاده کرد. نکته قابل ذکر این است که در صورت ارائه خدمات MLOps به صورت ابری، قطعاً حجم داده بیشتر و بیشتر خواهد شد و در نتیجه ناگزیر به استفاده از فناوری‌هایی مانند Clickhouse خواهیم بود. اما در صورتی که مشتری علاقه‌ای به قرار دادن داده خود روی خدمات ابری نداشته باشد (به علت محرمانگی، یا حجم بالای داده، یا هر علت دیگر) و نیاز به نصب یک نسخه از سامانه بر روی زیرساخت مشتری باشد، در این صورت کاملاً محتمل است که نیازی به پیچیدگی نصب و تنظیم و راه‌اندازی و پایش و نگهداری زیرساختی مانند Clickhouse نباشد و با استفاده از پایگاه‌های داده ساده‌تر مانند Postgres بتوان با هزینه بسیار پایین‌تر نیاز مشتری را مرتفع نمود. نکته دیگر این که فراداده مربوط به ویژگی‌های ذخیره شده را در Minio ذخیره می‌کنیم.

- **محل ذخیره‌سازی مدل:** این بخش محل نگهداری نسخه‌های مختلف فراداده‌ها، مدل‌ها و کدها است، و هنگام استقرار، مدل از آن بارگذاری می‌شود. برای ذخیره‌سازی نسخه‌های مختلف مدل از Minio استفاده می‌شود، و تمام Artifact ها مانند image های داکر و پکیج‌های پایتون در Nexus ذخیره می‌شوند که مخصوص ذخیره و بازیابی انواع Artifact ها است. برای نگهداری و استفاده از کدها Gitlab گزینه مناسبی به نظر می‌رسد. همچنین برای ذخیره‌سازی فراداده مربوط به داده ورودی، از Minio استفاده می‌شود.
- **آموزش، بهینه‌سازی و ارزیابی:** این بخش متشکل از یک خط لوله برای مراحل مختلف آموزش مدل است. این خط لوله توسط gitlab-ci فراخوانی شده و اجرای مراحل داخل آن تماماً توسط Kubeflow مدیریت می‌شود، که برای ذخیره‌سازی داده بین مراحل خط لوله، از Minio استفاده می‌کند. همچنین دانشمند داده، می‌تواند برای آموزش مدل از Spark استفاده کند.
- **استقرار نهایی:** استقرار مدل توسط gitlab-ci در بستر Kubernetes انجام می‌شود. در اینجا API استنتاج توسط Kubeflow و KServe ارائه خواهد شد. مدل‌ها و داده‌های مورد نیاز، از مؤلفه‌های دیگر دریافت می‌شوند.
- **نرم‌افزار نهایی:** نرم‌افزاری که از API استنتاج استفاده می‌کند، توسط gitlab-ci در بستر K8s راه‌اندازی می‌شود.
- **پایش:** برای پایش مدل مستقر شده، از Prometheus، ELK و Grafana استفاده می‌کنیم که برای این کار کاملاً استاندارد هستند.
- **زمان‌بندی:** برای این مؤلفه، نیاز به استفاده از زیرساخت مجزا مانند Airflow یا Azkaban یا Luigi نیست، زیرا خود Kubeflow این بخش را داخل خود مدیریت می‌کند.
- **مدیریت تاریخچه:** برای مدیریت تاریخچه کدها، داده‌ها و مدل‌ها، از Gitlab و Minio استفاده می‌کنیم.

۴-۳- ویژگی‌های کارکردی

ویژگی‌های کارکردی MLOps را می‌توان به‌صورت زیر خلاصه کرد:

- **قابلیت مدیریت داده‌های ورودی و پردازش‌های اولیه** شامل اجرای مدیریت‌شده و تکرارپذیر خطلوله‌های ورود و پردازش اولیه داده و نیز کنترل نسخه‌های داده‌های ذخیره شده و کدهای آماده‌سازی داده و خطلوله‌ها.
- امکان **دسترسی به زیرساخت‌های مختلف ذخیره‌سازی داده** (مانند Object Storageها، فایل‌سیستم‌ها و پایگاه‌های داده‌ای) به‌صورت یکپارچه (مثلا با استفاده از Kubeflow) برای کاربران MLOps فراهم می‌شود.
- **حذف چرخه‌ها و محاسبات تکراری استخراج ویژگی‌ها** با به‌کارگیری مولفه‌های Offline/Online feature store و همچنین افزایش سرعت توسعه، استقرار و اجرای کدهای استخراج ویژگی با استفاده از مولفه مدیریت کدهای منبع.
- قابلیت **سرویس‌دهی مقیاس‌پذیر و اتکاپذیر مدل** از طریق ایجاد امکان استقرار و فراخوانی مدل‌های مختلف ML از طریق API با به‌کارگیری زیرساخت‌های مقیاس‌پذیر و کانتینرها (مانند Ducker و Kubernetes) به‌صورت تقریبا خودکار.
- قابلیت **تنظیم هایپرپارامترها** از طریق ایجاد مجموعه‌ای از ابزارها برای خودکارسازی فرآیند تنظیم پارامترها که به کاربران امکان می‌دهد به‌سرعت مجموعه بهینه‌ای از پارامترها را برای مدل خود پیدا کنند.
- قابلیت **اجرای مکرر فرایند آموزش و استقرارد خودکار مدل** با استفاده از ابزارهایی مانند Argo Workflows ارائه می‌شود و کاربران را قادر می‌سازد تا به‌راحتی گردش‌های کاری یادگیری ماشینی پیچیده را ایجاد، اجرا و مدیریت کنند.
- امکان **نسخه‌سازی مدل** از طریق ایجاد و ارائه یک سیستم کنترل نسخه برای مدل‌های یادگیری ماشین که به کاربران امکان می‌دهد نسخه‌های مختلف مدل‌های خود را پیگیری کنند و به‌راحتی بین آن‌ها جابه‌جا شوند.

۵-۳- ویژگی‌های غیر کارکردی

MLOps علاوه بر ارائه مجموعه متنوعی از ویژگی‌های کارکردی، برخی ویژگی غیر کارکردی را نیز ارائه می‌کند که برای بهره‌مندی از قابلیت اطمینان، مقیاس‌پذیری و امنیت گردش‌های کاری ML ضروری هستند. به‌طورکلی ویژگی‌های غیر کارکردی MLOps برای اطمینان از اینکه جریان‌های کاری ML می‌توانند به شیوه‌ای قابل‌اعتماد، ایمن و مقیاس‌پذیر مستقر و مدیریت شوند، حیاتی هستند. MLOps با ارائه این ویژگی‌ها، سازمان‌ها را قادر می‌سازد تا بر توسعه و استقرار مدل‌های ML باکیفیت بالا بدون نگرانی در مورد زیرساخت‌های اساسی و پیچیدگی‌های عملیاتی تمرکز کنند.

برخی از ویژگی‌های کلیدی غیرکاربردی MLOps عبارت‌اند از:

- **مقیاس‌پذیری**: به کاربران امکان می‌دهد تا بسته به تقاضا، جریان‌های کاری ML خود را افزایش یا کاهش دهند که کمک می‌کند تا اطمینان حاصل شود که گردش‌های کاری می‌توانند حجم زیادی از داده و قدرت پردازش را مدیریت کنند.
- **امنیت**: تعدادی از ویژگی‌های امنیتی، از جمله مکانیسم‌های احراز هویت و مجوز، رمزگذاری داده‌ها در حال انتقال و در حالت استراحت، و ذخیره امن اسرار و اعتبار را ارائه می‌دهد.
- **قابلیت اطمینان MLOps**: اصولا MLOps برای ارائه یک پلتفرم بسیار قابل اعتماد برای گردش کار ML طراحی شده است. این ویژگی‌هایی مانند تحمل خطا، خودترمیمی و مقیاس خودکار را ارائه می‌دهد که به اطمینان از اینکه جریان‌های کاری ML می‌توانند بدون وقفه یا خرابی کار کنند کمک می‌کند.
- **قابلیت حمل**: کاربران می‌توانند به‌راحتی گردش کار ML خود را از یک محیط به محیط دیگر منتقل کنند. این امر از طریق استفاده از Containerization و Kubernetes به دست می‌آید که یک پلتفرم ثابت برای اجرای گردش‌های کاری ML بدون توجه به زیرساخت‌های اساسی فراهم می‌کند.
- **تحمل خطا**: MLOps از زیرساخت‌های تحمل‌پذیر خطا که توسط فناوری‌هایی ازجمله Kubernetes ارائه می‌شوند استفاده می‌کند تا اطمینان حاصل کند که بارهای کاری یادگیری ماشین در صورت خرابی گره یا مشکلات شبکه مختل نمی‌شود.

- **سفارشی‌سازی**: MLOps با ارائه از طیف وسیعی از APIها و نقاط افزونه به کاربران این امکان را می‌دهد تا عملکرد خود را با سیستم‌های موجود ادغام کنند. لذا کاربران می‌توانند پلتفرم را با نیازهای خاص خودتنظیم و راه‌حل‌های سفارشی ایجاد نمایند که نیازهای منحصربه‌فرد آن‌ها را برآورده کند.
- **مانیتورینگ، نظارت و مدیریت لاگ‌ها**: منظور از مانیتورینگ، فرآیند ردیابی عملکرد مدل‌های ML در تولید است. نظارت، شامل جمع‌آوری داده‌ها در مورد پیش‌بینی‌های مدل، مقایسه آن‌ها با نتایج واقعی و شناسایی هرگونه اختلاف یا خطا است. مانیتورینگ به شناسایی مشکلات احتمالی سیستم، مانند جابه‌جایی مدل کمک می‌کند و پاسخ سریع برای حفظ عملکرد سیستم را ممکن می‌سازد. مانیتورینگ امکان می‌دهد آزمایش‌های مختلف را ردیابی و مقایسه کرده و عملکرد مدل‌های خود را در طول زمان پیگیری و تصمیم‌گیری آگاهانه در مورد اینکه کدام مدل‌ها را اجرا کنید، بگیرید. همچنین ابزارهای تجسم را برای کمک به شما در نظارت بر پیشرفت گردش کار یادگیری ماشینی، تجسم نتایج آزمایش‌های و اشکال‌زدایی مشکلات مدل‌های خود ارائه می‌دهد.

۳-۶- منابع موردنیاز

در این بخش، منابع مورد نیاز جهت اجرای پروژه در دو محیط Stage و Production مورد بررسی قرار گرفته و پیشنهاد مرتبط ارائه شده است.

ابتدا باتوجه به مسائل مندرج در بخش‌های قبلی سند، مانند معماری فیزیکی و ویژگی‌های کارکردی و غیر کارکردی، منابع حداقلی مورد نیاز جهت راه‌اندازی یک محیط Stage تخمین زده و ارائه شده است. سپس حداکثر ظرفیت منابع قابل اختصاص در محیط Production که توسط کارفرما و بهره‌بردار (طی جلسات فنی برگزار شده) اعلام شده است ذکر شده و در نهایت منابع مناسب برای محیط Stage (بر اساس تخمین‌های بخش اول و محدویت منابع Production در بخش دوم) آورده شده است.

۳-۶-۱- تخمین منابع مورد نیاز محیط Stage

بنابر بررسی‌های اولیه صورت گرفته، منابع مورد نیاز توسعه و راه‌اندازی MLOps براساس نوع و حجم داده ورودی و روش حل مسئله متغیر خواهد بود. از طرفی در زمان تدوین این سند، ابعاد مسائل و داده‌های متناظر با آن‌ها مشخص و نهایی نشده‌اند. علیرغم این نکته، براساس تجربیات و برداشت‌ها از بستر مسائل

پیش‌رو و حداقل سرویس‌های موردنیاز برای زیرساخت MLOps، منابع مورد نیاز برای ایجاد یک محیط Stage به‌ازای هریک از مولفه‌ها و بخش‌های پیشنهادی جهت توسعه MLOps، توسط تیم فنی محاسبه شده و مطابق جدول 1 پیشنهاد می‌شود. بدیهی است که این پیشنهاد در زمان اجرا و با توجه به اطلاعات دقیق موجود در آن مقطع قابل اصلاح است.

جدول 1- منابع مورد نیاز (و فرمول محاسبه) به ازای هر جزء از معماری

Service	CPU	RAM	HDD (Data)	SSD (OS + Data)	GPU
Minio	4	24	-	40 + 300	
Redis	8	48	100	40	
Postgres	4	16	-	40 + 400	
Kafka	4	16	100	40	
Spark	16	64	-	40	
Nexus	2	8	400	40	
Gitlab	4	24	100	40	
ELK	2*4 = 8	2*16 = 32	2*500 = 1000	40	
Prometheus	4	16	100	40	
K8s (2*master, 4*worker)	2*2 + 4*6 = 28	2*4 + 4*24 = 104	4*100 = 400	6*40 = 240	
Kubeflow	4*6 = 24	4*16 = 64	-	4*40 + 4*100 = 560	2*4090
Sum	**106**	**416**	**2200**	**1860**	**2*4090**

نکات در نظر گرفته شده برای استخراج میزان منابع مورد نیاز به این شرح است:

- **Minio**: این مؤلفه یک Object Storage است و کار پردازشی زیادی انجام نمی‌دهد. در نتیجه نیاز به پردازنده بالایی ندارد. همچنین از RAM به عنوان cache استفاده می‌کند، در نتیجه برای حجم متوسط داده، حجم بالایی از RAM نیاز ندارد. توجه کنید که Kubeflow برای ذخیره‌سازی داده‌های میانی خود (مانند خروجی‌های مراحل pipeline ها)، تماماً از minio استفاده می‌کند.
- **Redis**: این مؤلفه کل داده را در حافظه نگهداری می‌کند و به عنوان Online feature store استفاده می‌شود، در نتیجه حجم حافظه متناسب با حجم داده نیاز خواهد داشت. هارد در نظر گرفته شده، برای persist کردن داده redis اختصاص داده خواهد شد.
- **Postgres**: این مؤلفه برای ذخیره‌سازی حجم متوسط داده در Offline feature store استفاده می‌شود. به علت سادگی نصب و کاربرد، می‌توان برای جایی که حجم داده خیلی بالا نیست از آن استفاده کرد و در صورت نیاز به پردازش حجم بالاتر داده، از گزینه‌هایی مانند Clickhouse

بهره گرفت. این مؤلفه نیز از RAM بیشتر به عنوان cache استفاده می‌کند و پردازش معمول دارد. حجم دیسک نیز متناسب با حجم داده خواهد بود. برای بالاتر رفتن سرعت خواندن و نوشتن داده، از هارد SSD استفاده می‌کنیم.

- **Kafka**: این مؤلفه برای پردازش جریانی داده‌ها در MLOps استفاده می‌شود. در نتیجه به طور معمول retention بالایی نخواهد داشت و نیاز به حجم بالایی از دیسک ندارد. با توجه به حجم و نرخ ورودی داده جریانی، می‌توان منابع و همچنین تعداد instance های این مؤلفه را بالاتر برد.
- **Spark**: این مؤلفه ۲ کاربرد دارد: یکی برای پردازش داده‌ها و استخراج ویژگی‌ها، و دیگری هنگام آموزش مدل. اسپارک یک مؤلفه کاملاً پردازشی است، که کل داده ورودی را در حافظه نگهداری می‌کند. با توجه به این که به صورت خوشه بالا می‌آید و مقیاس‌پذیر است، میزان پردازنده و RAM مورد نیاز آن کاملاً بسته به حجم داده و میزان پردازش همزمان مورد نیاز، قابل افزایش است. فعلاً برای stage یک خوشه کوچک اسپارک فرض شده است.
- **Nexus**: این مؤلفه در Model Registry استفاده می‌شود و docker image ها و پکیج‌های پایتون و امثالهم در آن ذخیره می‌شوند (توجه کنید که هر بار تغییر در کد استنتاج، باعث ایجاد یک docker image جدید می‌شود که در مرحله استقرار از آن استفاده می‌شود)، فلذا بار زیادی ندارد و با توجه به مسائل اولیه، حجم متوسطی از دیسک نیاز خواهد داشت. در نتیجه یک مؤلفه حداقلی دیده شده است. در آینده با افزایش تعداد مسائل همزمان اجرا شده روی بستر MLOps، این مؤلفه نیز می‌تواند حجم بالاتری داشته باشد.
- **Gitlab**: این مؤلفه برای نگهداری کدها استفاده می‌شود، و همچنین فرایند CI/CD توسط gitlab-ci انجام می‌شود. در نتیجه با توجه به میزان همزمانی کاربران سامانه، تعداد runner ها می‌تواند افزایش پیدا کند. برای stage، یک گیت‌لب کوچک فرض شده است.
- **ELK**:با توجه به ذخیره‌سازی لاگ تمام مؤلفه‌ها در ELK، هارد HDD قابل توجهی برای آن در نظر گرفته‌ایم. در ادامه کار با اجرای کامل سامانه و مؤلفه‌های مختلف، با توجه به میزان لاگ تولید شده توسط آنها، ممکن است منابع این مؤلفه افزایش پیدا کند. همچنین می‌توان با hot و

cold کردن لاگ‌ها، از دیسک SSD با ظرفیتی کمتر از HDD مورد استفاده نیز برای ذخیره‌سازی لاگ‌های hot و بالا رفتن سرعت پردازش آن‌ها بهره برد.

- **Prometheus**: برای پایش کل سامانه، برای stage یک محیط متوسط فرض شده است. در صورت افزایش تعداد مؤلفه‌ها منابع این مؤلفه را نیز می‌توان افزایش داد.
- **K8s و Kubeflow**: منابع این دو مؤلفه با توجه به تجربه عملی تیم در بالا آوردن آن‌ها برای مقیاس Stage برآورد شده است.

۳-۶-۲- منابع محیط Production

منابع موردنیاز در Production به‌گونه‌ای در نظر گرفته شود که برای حل مسئله‌ای با ابعاد داده‌ای حداکثر ۱۰ برابر مسئله شاهکار مناسب باشد (طبق آخرین اعلام تیم بهره‌بردار، نرخ تولید داده مد نظر در مسئله شاهکار ۶۰میلیون رکورد در ماه می‌باشد). براساس محاسبات انجام شده و ابعاد ده برابری مسئله و علیرغم رشد ۵ برابری منابع در Production، کل ظرفیت ممکن در محیط Production که قبلا توسط کارفرما و بهره‌بردار اعلام شده بود و در جدول 2 آورده شده است، مورد استفاده قرار خواهد گرفت.

جدول 2 – منابع موجود و قابل استفاده در محیط Production

تعداد / نوع	نوع منبع	تعداد در هر واحد	مجموع
۲۰ سرور	CPU	۳۲ هسته	۶۴۰ هسته
	RAM	۱۲۸ گیگابایت	۲/۵۶ ترابایت
	HDD	۱ ترابایت	۲۰ ترابایت
	SSD	۵۰۰ گیگابایت	۱۰ ترابایت
۴ عدد	GPU RTX 4090		۴ عدد

۳-۶-۳- منابع محیط Stage

با توجه به محاسبات و توضیحات ارائه شده در بخش ۳-۶-۱-، و نیز توافق ضمنی صورت‌گرفته مبنی بر نسبت ابعاد محیط Stage به Production (که طی آن، ابعاد Stage معادل حداقل ۲۰ درصد محیط Production توافق شده است) منابع مورد نیاز در محیط Stage، مطابق با ... پیشنهاد می‌شود. لازم به ذکر است که تعداد GPU در فضای Stage حداقل ۲ مورد برای دو منظور استقرار و آموزش مدل به طور همزمان لحاظ شده است. براساس محاسبات انجام شده، بطور میانگین، حدود ۸۰ درصد حافظه و ۸۰ درصد پردازنده از منابع Stage مورد استفاده قرار خواهد گرفت.

جدول 3- منابع پیشنهادی جهت آماده‌سازی محیط Stage

تعداد / نوع	نوع منبع	تعداد در هر واحد	مجموع
۴ سرور	CPU	۳۲ هسته	۱۲۸ هسته
	RAM	۱۲۸ گیگابایت	۵۱۲ گیگابایت
	HDD	۱ ترابایت	۴ ترابایت
	SSD	۵۰۰ گیگابایت	۲ ترابایت
۲ عدد	GPU RTX 4090		۲ عدد

۴- برنامه توسعه و اجرا

۴-۱- نسخه‌بندی و تحویل نسخه‌ها

این پروژه در مجموع ۵ فاز و در بازه زمانی ۱۴ ماه تعریف شده است. در ادامه این بخش به صورت کامل، زمان‌بندی و خروجی‌های هر فاز تشریح شده است. همچنین زمان شروع این پروژه در خرداد ماه سال ۱۴۰۲ و نقطه پایان آن، ۱۴ ماه بعد در نظر گرفته شده است. تایید نهایی هر مورد از موارد بیان شده در جداول مربوط به تحویل‌دادنی‌های هر فاز، منوط به تایید نماینده اعلامی کارفرما در خصوص موضوع مربوط به هر بند می‌باشد. این فرد بایستی از سوی کارفرما به مدیر پروژه در معرفی گردد.

۴-۱-۱- فاز اول پروژه

فاز اول پروژه در قالب دو ماه قابل اجرا خواهد بود که شرح گام به گام اقدامات و تعهدات کارفرما و پیمانکار در این فاز در جدول 4 ارائه شده است. همچنین پیش‌نیاز هر یک از این موارد در ستون مربوط به پیش‌نیاز مشخص شده است. در این فاز، برخی مستندات همچون سند نیازمندی پروژه MLOps، سند معماری سامانه و سند طراحی آزمون به طور مجزا تحویل کارفرما خواهد شد. علاوه بر این، در حوزه توسعه محصول، پس از اعلام دقیق سخت‌افزار مورد نیاز برای پیاده‌سازی در محیط stage به کارفرما و آماده شدن این محیط از سوی کارفرما، سرویس kubeflow (که در حال حاضر به عنوان گزینه اصلی برای استفاده در MLOps می‌باشد) بر روی بستر کوبرنتیز پیاده‌سازی شده و شمای اولیه‌ای از ماژول‌های یادگیری ماشینی بر روی آن ارائه خواهد شد.

به موازات این فرآیند، درصورت آماده شدن داده‌های مسئله شاهکار، داده‌ها و مسئله به صورت دقیق مورد بررسی قرار خواهد گرفت و به صورت یک مسئله یادگیری ماشینی صورت‌بندی خواهد شد. پس از صورت‌بندی مسئله، داده‌ها پیش‌پردازش و آماده شده و در خارج از بستر MLOps، به صورت مجزا مورد تجزیه و تحلیل قرار خواهند گرفت و مدل مناسب برای حل این مسئله، پیاده‌سازی خواهد شد. پیاده‌سازی این مسئله در بستر MLOps در فازهای بعدی انجام می‌شود.

جدول 4- شرح فعالیت‌های مربوط به فاز اول پروژه

فاز اول				
شناسه فعالیت	شرح	مدت‌زمان (ماه)	پیش‌نیاز	مسئول
۱.۱	تدوین سند نیازمندی	۲		پیمانکار
۱.۲	ارائه سخت‌افزار موردنیاز محیط stage			پیمانکار
۱.۳	تحویل نمونه داده مسئله شاهکار			کارفرما
۱.۴	ارائه سند معماری سامانه			پیمانکار
۱.۵	تأمین سخت‌افزار بستر stage		۱.۱ و ۱.۴	کارفرما
۱.۶	تعریف دقیق صورت‌مسئله شاهکار		۱.۳	پیمانکار
۱.۷	نصب و راه‌اندازی kubeflow بر روی بستر کوبرنتیز به همراه ماژول‌های ML		۱.۵	پیمانکار
۱.۸	آماده‌سازی داده‌های اولیه تحویل داده‌شده		۱.۳	پیمانکار
۱.۹	توسعه مدل مسئله شاهکار		۱.۸	پیمانکار
۱.۱۰	ارائه سند طراحی آزمون پروژه			پیمانکار

در پایان فاز اول، یک نسخه از محصولات پروژه شامل موارد مندرج در جدول 5 به کارفرما تحویل خواهد شد.

جدول 5- تحویل‌دادنی‌های انتهای فاز اول

ردیف	عنوان تحویل‌دادنی	بستر تحویل مؤلفه‌ها	نوع
۱	سند نیازمندی	-	مستند متنی
۲	سند معماری سامانه	-	مستند متنی
۳	سند طراحی آزمون	-	مستند متنی
۴	سند صورت‌بندی تعریف مسئله شاهکار	-	مستند متنی
۵	مدل توسعه داده شده برای حل مسئله شاهکار، مستقل از MLOps	Stage	کد و نرم افزار
۶	نسخه اولیه بستر MLOps	Stage	کد و نرم افزار

۲-۱-۴- فاز دوم پروژه

فاز دوم پروژه در قالب سه ماه قابل اجرا خواهد بود که شرح گام به گام اقدامات و تعهدات کارفرما و پیمانکار در این فاز در جدول 6 ارائه شده است. همچنین پیش‌نیاز هر یک از این موارد در ستون مربوط به پیش‌نیاز مشخص شده است.

در این فاز، کارفرما می‌بایست بر روی مسئله دوم متمرکز شده و مسئله پیشنهادی خود را به تیم مجری اعلام نماید تا مراحل مربوط به صورت‌بندی مسئله در این فاز و توسعه مدل در فازهای بعد در زمان مقرر انجام شود. در صورت تحویل به موقع داده‌های نمونه و مسئله دوم از سوی کارفرما در این فاز، پیمانکار متعهد به ارائه صورت‌بندی مسئله دوم خواهد بود.

علاوه بر این، در این فاز، پایپلاین CI/CD برای مسئله شاهکار که در فاز قبلی خارج از بستر MLOps حل شده، ایجاد خواهد شد و مدل مسئله شاهکار در داشبورد Kubeflow با استفاده از مؤلفه Kserve پیاده‌سازی خواهد شد. بدین منظور، ماژول Kserve نیز ب صورت پیش‌نیاز بر بستر MLOps پیاده‌سازی خواهد شد.

جدول 6- شرح فعالیت‌های مربوط به فاز دوم پروژه

فاز دوم				
شناسه فعالیت	شرح	مدت‌زمان (ماه)	پیش‌نیاز	مسئول
۲.۱	ایجاد پایپ لاین CI/CD یادگیری مدل مسئله شاهکار	۳	۱.۸	پیمانکار
۲.۲	استقرار ماژول Kserve			پیمانکار
۲.۳	استنتاج مدل مسئله شاهکار در داشبورد Kubeflow با استفاده از مؤلفه Kserve		۲.۲	پیمانکار
۲.۴	اعلام مسئله دوم ML			کارفرما
۲.۵	تحویل داده مسئله دوم ML			کارفرما
۲.۶	تعریف دقیق صورت‌مسئله دوم ML		۲.۴	پیمانکار

در پایان فاز دوم، یک نسخه از محصولات پروژه شامل موارد مندرج در جدول 7 به کارفرما تحویل خواهد شد.

جدول 7- تحویل‌دادنی‌های انتهای فاز دوم

ردیف	عنوان تحویل‌دادنی	بستر تحویل مؤلفه‌ها	نوع
۱	تعریف دقیق صورت‌مسئله دوم ML	-	مستند متنی
۲	ایجاد پایپ لاین CI/CD یادگیری مدل مسئله شاهکار، بدون Feature store و با داده‌های مستقیم	Stage	کد و نرم افزار
۳	استقرار ماژول Kserve مسئله شاهکار	Stage	کد و نرم افزار
۴	استنتاج مدل مسئله شاهکار در داشبورد Kubeflow با استفاده از مؤلفه Kserve	Stage	کد و نرم افزار

۳-۱-۴- فاز سوم پروژه

فاز سوم پروژه در قالب سه ماه قابل اجرا خواهد بود که شرح گام به گام اقدامات و تعهدات کارفرما و پیمانکار در این فاز در جدول 8 ارائه شده است. همچنین پیش‌نیاز هر یک از این موارد در ستون مربوط به پیش‌نیاز مشخص شده است.

در این فاز، سه ماژول مانیتورینگ و نظارت در داشبورد، ماژول feature store و ماژول offline feature store بر بستر MLOps پیاده‌سازی خواهند شد. علاوه بر این، داده‌های مربوط به مسئله دوم پیش‌پردازش و آماده‌سازی شده و مدل مربوط به مسئله دوم نیز توسعه داده می‌شود.

جدول 8- شرح فعالیت‌های مربوط به فاز سوم پروژه

فاز سوم				
شناسه فعالیت	شرح	مدت‌زمان (ماه)	پیش‌نیاز	مسئول
۳.۱	مانیتورینگ و نظارت داشبورد Kubeflow	۳	۲.۱	پیمانکار
۳.۲	آماده‌سازی داده‌ها		۲.۲	پیمانکار
۳.۳	استقرار ماژول موردنیاز feature store		۳.۲	پیمانکار
۳.۴	Offline feature store		۳.۳	پیمانکار
۳.۵	توسعه مدل مسئله دوم ML		۲.۵	پیمانکار

در پایان فاز سوم، یک نسخه از محصولات پروژه شامل موارد مندرج در جدول 9 به کارفرما تحویل خواهد شد.

جدول 9- تحویل‌دادنی‌های انتهای فاز سوم

ردیف	عنوان تحویل‌دادنی	بستر تحویل مؤلفه‌ها	نوع
۱	پیاده‌سازی ماژول مانیتورینگ و نظارت داشبورد بستر MLOps	Stage	کد و نرم افزار
۲	feature store	Stage	کد و نرم افزار
۳	استقرار ماژول مورد نیاز Offline feature store	Stage	کد و نرم افزار
۴	توسعه مدل مسئله دوم ML	Stage	کد و نرم افزار

۴-۱-۴- فاز چهارم پروژه

فاز چهارم پروژه در قالب سه ماه قابل اجرا خواهد بود که شرح گام به گام اقدامات و تعهدات کارفرما و پیمانکار در این فاز در جدول 10 ارائه شده است. همچنین پیش‌نیاز هر یک از این موارد در ستون مربوط به پیش‌نیاز مشخص شده است.

در فاز چهارم، ماژول Online feature store نیز بر بستر MLOps پیاده‌سازی خواهد شد و مسئله اول به فضای production منتقل خواهد شد. البته این موضوع منوط به اختصاص زیرساخت production از سوی کارفرما خواهد بود. علاوه بر این، مستندات مربوط به نصب و استقرار مسئله شاهکار، راهنمای استفاده از مسئله شاهکار و مستند مربوط به نتایج طرح آزمون پیاده‌سازی شده برای مسئله شاهکار، ارائه خواهد شد.

به منظور پیاده‌سازی مسئله دوم بر بستر MLOps، پایپ لاین CI/CD یادگیری مدل مسئله دوم ایجاد خواهد شد و نتایج مربوط به مدل توسعه داده شده برای مسئله دوم در داشبورد Kubeflow با استفاده از مؤلفه Kserve ارائه خواهد شد. همچنین، در این فاز برای پایش مدل مستقر شده، از Prometheus، ELK و Grafana استفاده می‌کنیم و مدیریت تاریخچه و زمان‌بندی توسط خود Kubeflow این بخش را داخل خود مدیریت می‌کند.

جدول 10- شرح فعالیت‌های مربوط به فاز چهارم پروژه

فاز سوم				
شناسه فعالیت	شرح	مدت‌زمان (ماه)	پیش‌نیاز	مسئول
۴.۱	Online feature store برای مسئله شاهکار	۳	۳.۲	پیمانکار
۴.۲	ارائه سند نصب و استقرار مسئله شاهکار		۴.۱	پیمانکار
۴.۳	مستندات نحوه استفاده مسئله شاهکار		۴.۲	پیمانکار
۴.۴	اجرای آزمون‌های طرح آزمون		۴.۲	پیمانکار
۴.۵	ایجاد پایپ لاین CI/CD یادگیری مدل مسئله دوم		۳.۵	پیمانکار
۴.۶	استنتاج مدل مسئله دوم در داشبورد Kubeflow با استفاده از مؤلفه Kserve		۴.۳	پیمانکار
۴.۷	نظارت مدل مستقر شده مسئله شاهکار		۴.۳	پیمانکار
۴.۸	مدیریت تاریخچه مسئله شاهکار		۴.۳	پیمانکار
۴.۹	زمان‌بندی بروزرسانی مدل مسئله شاهکار		۴.۳	پیمانکار
۴.۱۰	انتقال مسئله اول به فضای Production		۴.۴	پیمانکار

در پایان فاز چهارم، یک نسخه از محصولات پروژه شامل موارد مندرج در جدول 11 به کارفرما تحویل خواهد شد.

جدول 11- تحویل‌دادنی‌های انتهای فاز چهارم

ردیف	عنوان تحویل‌دادنی	بستر تحویل مؤلفه‌ها	نوع
۱	سند نصب و استقرار مسئله شاهکار	-	مستند متنی آموزشی
۲	سند نحوه استفاده از مسئله شاهکار	-	مستند متنی آموزشی
۳	آموزش پیرامون مستندات ارائه شده بستر MLOps برای مسئله شاهکار	-	جلسات آموزشی مطابق نیاز کارفرما
۴	سند نتایج طرح آزمون پیاده‌سازی شده برای مسئله شاهکار	-	مستند متنی فنی
۵	اعمال آزمون‌های پیاده‌سازی شده برای مسئله شاهکار	Stage	کد و نرم افزار
۶	Online feature store برای مسئله شاهکار	Stage	کد و نرم افزار
۷	ایجاد پایپ لاین CI/CD یادگیری مدل مسئله دوم	Stage	کد و نرم افزار
۸	استنتاج مدل مسئله دوم در داشبورد Kubeflow با استفاده از مؤلفه Kserve	Stage	کد و نرم افزار
۹	نظارت مدل مستقر شده مسئله شاهکار	Stage	کد و نرم افزار

۱۰	مدیریت تاریخچه مسئله شاهکار	Stage	کد و نرم افزار
۱۱	زمان‌بندی بروزرسانی مدل مسئله شاهکار	Stage	کد و نرم افزار
۱۲	حل خودکار مسئله شاهکار در فضای production	Production	کد و نرم افزار

۵-۱-۴- فاز پنجم پروژه

فاز پنجم پروژه در قالب سه ماه قابل اجرا خواهد بود که شرح گام به گام اقدامات و تعهدات کارفرما و پیمانکار در این فاز در جدول 12 ارائه شده است. همچنین پیش‌نیاز هر یک از این موارد در ستون مربوط به پیش‌نیاز مشخص شده است. در فاز پایانی، ماژول‌های مانیتورینگ و نظارت داشبورد Kubeflow، ماژول Offline feature store و ماژول Online feature store برای مسئله دوم نیز پیاده‌سازی شده و مستندات مربوط به طرح آزمون مسئله دوم، سند نصب و استقرار مسئله دوم و سند نحوه استفاده از مسئله دوم بر بستر MLOps نیز ارائه خواهد شد و در نهایت این مسئله نیز به بستر production منتقل می‌شود.

جدول 12- شرح فعالیت‌های مربوط به فاز پنجم پروژه

فاز سوم				
شناسه فعالیت	شرح	مدت‌زمان (ماه)	پیش‌نیاز	مسئول
۵.۱	مانیتورینگ و نظارت داشبورد Kubeflow مسئله دوم	۳	۴.۴	پیمانکار
۵.۲	Offline feature store مسئله دوم		۴.۴	پیمانکار
۵.۳	Online feature store مسئله دوم		۴.۴	پیمانکار
۵.۴	اجرای آزمون‌های طرح آزمون مسئله دوم		۵.۳ و ۴.۲	پیمانکار
۵.۵	ارائه سند نصب و استقرار مسئله دوم		۵.۴	پیمانکار
۵.۶	مستندات نحوه استفاده مسئله دوم		۵.۴	پیمانکار
۵.۷	نظارت مدل مستقر شده مسئله دوم		۵.۳	پیمانکار
۵.۸	مدیریت تاریخچه مسئله دوم		۵.۳	پیمانکار
۵.۹	زمان‌بندی مسئله دوم		۵.۳	پیمانکار
۵.۱۰	انتقال مسئله دوم به فضای Production		۵.۴	پیمانکار

در پایان فاز پنجم، یک نسخه از محصولات پروژه شامل موارد مندرج در جدول ۱۳ به کارفرما تحویل می‌شود.

جدول ۱۳- تحویل‌دادنی‌های انتهای فاز پنجم

ردیف	عنوان تحویل‌دادنی	بستر تحویل مؤلفه‌ها	نوع
۱	سند نتایج آزمون‌های طرح آزمون مسئله دوم	-	مستند متنی فنی
۲	سند نصب و استقرار مسئله دوم	-	مستند متنی آموزشی
۳	سند نحوه استفاده مسئله دوم	-	مستند متنی آموزشی
۴	آموزش پیرامون مستندات ارائه شده بستر MLOps برای مسئله دوم	-	جلسات آموزشی مطابق نیاز کارفرما
۵	نتایج آزمون‌های مسئله دوم	-	کد و مستند متنی
۶	ماژول مانیتورینگ و نظارت داشبورد برای مسئله دوم	Stage	کد و نرم افزار
۷	ماژول Offline feature store بری مسئله دوم	Stage	کد و نرم افزار
۸	ماژول online feature store برای مسئله دوم	Stage	کد و نرم افزار
۹	نظارت مدل مستقر شده برای مسئله دوم	Stage	کد و نرم افزار
۱۰	مدیریت تاریخچه برای مسئله دوم	Stage	کد و نرم افزار
۱۱	زمان‌بندی برای مسئله دوم	Stage	کد و نرم افزار
۱۲	حل خودکار مسئله دوم در فضای production	Production	کد و نرم افزار

۴-۲- تست و ارزیابی MLOps

در این بخش، نحوه کلی آزمون هر یک از مؤلفه‌های معماری مشخص می‌شود. جزئیات آزمون‌ها و نحوه پیاده‌سازی در این بخش مطرح نیست و طبق زمانبندی در سند طراحی آزمون مشخص خواهد شد.

در پیشنهاد حاضر، چهار نوع آزمون برای مؤلفه‌های مختلف در نظر گرفته شده که در ادامه توضیح داده خواهند شد. در نهایت یک آزمون یکپارچگی کل سامانه نیز انجام خواهد شد که عملکرد کل سامانه را مستقل از عملکرد تک‌تک مؤلفه‌ها مورد بررسی قرار خواهد داد.

۴-۲-۱- آزمون عملکرد (Functionality test)

تعریف: هر مؤلفه باید بتواند مستقل از عملکرد بقیه مؤلفه‌ها با دریافت ورودی‌های مشخص در حیطه تعریف شده، آن‌ها را با پردازش کرده و خروجی‌های مدنظر را ایجاد کند.

نحوه پیاده‌سازی: نحوه پیاده‌سازی به صورت «آزمون واحد» خواهد بود. همچنین پیش‌نیاز این آزمون، ایجاد داده‌های آزمون طبیعی متناسب با هر مؤلفه است که عملکرد طبیعی آن مؤلفه مورد آزمایش قرار گیرد.

مؤلفه‌های درگیر در این آزمون: تمام مؤلفه‌ها

۴-۲-۲- آزمون تحمل‌پذیری خطا (Fault tolerance test)

تعریف: در این نوع آزمون، عملکرد مؤلفه در قبال رخداد خطا آزموده می‌شود. این خطا می‌تواند ناشی از وجود ورودی نامناسب باشد، یا ایجاد مشکل برای یکی از زیرسامانه‌های مؤلفه، یا ایجاد مشکل برای یکی از مؤلفه‌هایی که مورد استفاده این مؤلفه هستند (مانند پایگاه داده).

نحوه پیاده‌سازی: برای آزمودن عملکرد مؤلفه در قبال ورودی نامناسب، می‌توان آزمون‌های واحد با ورودی‌های نامناسب ایجاد کرد و عملکرد مؤلفه را در قبال آن‌ها سنجید. برای آزمودن عملکرد مؤلفه در هنگام رخداد مشکل برای زیرساخت‌ها، می‌توان برق یکی از سرورها یا اتصال شبکه آن را قطع کرد (توسط ابزار Chaos Monkey یا ابزار مشابه) و منتظر ماند تا فرایند با موفقیت به انتها برسد و در عملکرد مؤلفه هیچ اختلالی مشاهده نشود (بدیهی است که در این سناریو، افزایش زمان اجرا کاملاً طبیعی است). برای حالتی که مؤلفه دیگری که این مؤلفه به آن وابسته است دچار مشکل شود، می‌توان مؤلفه دیگر را شبیه‌سازی کرد و سپس آن را دچار مشکل نمود، یا می‌توان توسط روش‌های مانند استفاده از داکر، مؤلفه دیگر را بالا آورد و سپس در آن اختلال ایجاد کرد.

مؤلفه‌های درگیر در این آزمون: پردازش داده، انباره داده (آنلاین و آفلاین)، استقرار، نرم‌افزار نهایی

۴-۲-۳- آزمون سرعت (Performance test)

تعریف: زمان پاسخ مؤلفه‌ها باید از حد مشخص شده پایین‌تر باشد.

نحوه پیاده‌سازی: برای آزمون سرعت، می‌توان درخواست‌هایی به صورت همزمان برای مؤلفه ارسال کرد و زمان پاسخ را بررسی نمود. یا در صورتی که مؤلفه پردازشی است، می‌توان داده مشخصی را ورودی داد و زمان ایجاد خروجی را بررسی کرد. پیش‌نیاز این کار، ایجاد داده‌های واقعی است تا سرعت واقعی مؤلفه سنجیده شود.

مؤلفه‌های درگیر در این آزمون: تمام مؤلفه‌ها، به جز پایشگر، زمان‌بند، تاریخچه

۴-۲-۴- آزمون بار (Stress test)

تعریف: هدف از این آزمون، سنجش عملکرد مؤلفه زیر بار سنگین است. بنابراین مؤلفه را در طول مدت قابل توجهی زیر بار سنگین می‌گذاریم، و پس از آن بررسی می‌کنیم که آیا عملکرد آن در طول این مدت و بعد از آن دچار خدشه شده است یا خیر.

نحوه پیاده‌سازی: می‌توان توسط ابزارهایی مانند ab یا locust این آزمون را انجام داد. پیش‌نیاز این بخش، داشتن داده واقعی برای هر مؤلفه است.

مؤلفه‌های درگیر در این آزمون: پردازش داده، انباره داده (آنلاین و آفلاین)، استقرار، نرم‌افزار نهایی

۴-۲-۵- آزمون یکپارچگی (Integration test)

تعریف: هنگامی که یک سامانه از مؤلفه‌های مختلفی تشکیل شده، حتی در صورتی که تمام مؤلفه‌ها آزمون‌های مستقل خود را به خوبی بگذرانند، باز نیاز است تا ارتباط و اتصال این مؤلفه‌ها به یکدیگر مورد بررسی قرار گیرد. یکی از شایع‌ترین مشکلاتی که در ارتباط مؤلفه‌ها با یکدیگر رخ می‌دهد، تغییر در قالب ورودی/خروجی مؤلفه یا شیوه‌نامه ارتباطی مؤلفه‌ها است. با داشتن آزمون یکپارچگی، اطمینان حاصل می‌شود که بعد از ایجاد هر تغییر، کل سامانه به صورت یک‌پارچه به درستی کار می‌کند. با توجه به سنگین و زمان‌بر بودن آزمون‌های یکپارچگی، می‌توان اجرای آن‌ها را منوط به ایجاد تغییرات بزرگ‌تر نمود.

نحوه پیاده‌سازی: یک آزمون یک‌پارچه‌سازی برای کل سامانه نوشته خواهد شد، که با گرفتن ورودی خام مشخص در قالب مشخص، با اجرای کل فرایند، باید بتواند پاسخ درخواست کاربر را به درستی بدهد. همچنین در ادامه با شبیه‌سازی پیشامد شرایط فرایند بازسازی مدل، و دادن داده جدید ورودی، انتظار داریم مدل جدید زیر بار رفته و خروجی کاربر متناسب با داده جدید تغییر کند. در این آزمون جامع، حالت‌های مختلف عملکرد سامانه آزمایش خواهند شد (مانند پردازش دسته‌ای یا جریانی داده ورودی).

۴-۳- نصب و راه‌اندازی

برای این پروژه در دو فاز، نصب و راه‌اندازی در محیط Production در نظر گرفته شده است. دلیل این انتخاب، تکمیل کلیه مؤلفه‌های MLOps برای مدل حل شده مسئله اول و دوم در این دو فاز است. بر این اساس، تیم مجری متعهد است خروجی‌های این دو فاز از سیستم را پس از پایان هر فاز، در محیط Production راه‌اندازی، تحویل و آموزش دهد. لازم به ذکر است، کلیه مراحل نصب و راه‌اندازی در محیط Production و انتقال نسخه‌های نهایی تست و تاییدشده توسط کارفرما در محیط Stage، به صورت خودکار و با ابزارهایی همچون Ansible و Helm chart با هماهنگی با کارفرما در فاز سوم و پنجم صورت خواهد گرفت.

نکته: تحویل‌گیری نهایی سامانه منوط به یکپارچگی بخش‌های مختلف و تأیید کارکرد نهایی سامانه است. کلیه مراحل مندرج در مستندات توسط تیم کارفرما با حضور نماینده مجری، اجرا و صحت‌سنجی خواهد شد.

۴-۴- آموزش

مجری متعهد به برگزاری دوره آموزشی با شرایط ذیل برای یک تیم/نفر معرفی‌شده از سمت کارفرما است:

۱. آموزش جامع بهره‌برداری و استفاده از محصول پروژه (مواردی همچون نحوه نصب، راه‌اندازی، نگهداری، توسعه و نیز بهره‌برداری برای تجهیزاتی/سامانه‌هایی که نیاز به آموزش دارند) می‌بایست حتماً در طرح آموزش دیده شود.

۲. برگزاری جلسات آموزش مفاهیم کلی و نصب و مدیریت بستر MLOps که آموزش‌های راهبری و کاربری برای کارشناسان ذی‌ربط در محل مورد تأیید نماینده فنی سحاب در خصوص تکمیل فاز آموزش، باید به‌صورت کارگاه و یا کلاس آموزشی ارائه گردد. آموزش می‌بایست در دو سطح اپراتوری و مدیر سیستم برگزار شود.

۳. تأییدیه نماینده فنی کارفرما در خصوص تأیید صحت مستندات تولید شده در تمامی مراحل فوق باید صورت پذیرد.

۴-۵- برنامه کیفیت پروژه

Commented [n1]: با توجه به موارد بیان شده در جلسه روز شنبه مورخ ۳۰ اردیبهشت ۱۴۰۲ با مهندس نصیرنژاد، این بند بایستی پس از جلسه با مهندس اکبرزاده تکمیل شود و مبتنی بر نحوه ارزیابی کیفیت توسط شرکت محترم سحاب است.

۴-۶- برنامه ارتباطات پروژه

در این بخش، برنامه اجرایی نحوه ارتباط میان ذی‌نفعان پروژه در هر مرحله بیان شده است. در ابتدا در جدول زیر، نقش‌های ذی‌نفعان موجود در برنامه تعریف شده و مشخصات فردی صاحبان فعلی این نقش‌ها در زمان تدوین پروپوزال مشخص شده است. بدیهی است در صورت تغییر هر یک از این افراد، فرد جایگزین به شرکت محترم سحاب‌پرداز معرفی خواهد شد.

جدول ١٤- نقش‌های ذی‌نفعان درگیر در پروژه و افراد منتسب به هر نقش

نام فرد	نقش	شرح وظایف	ایمیل	شماره تماس
نوید محمدی	جانشین مدیرعامل (VP)	امور اداری، حقوقی و مالی قرارداد	Navid.znu@gmail.com	۰۹۱۲۶۴۴۲۰۵۳
مائده سادات طاهائی	مدیر پروژه MLOps	برنامه اجرایی پروژه، تحویل تحویل دادنی‌ها و اداره جلسات فنی	mstahaei@gmail.com	۰۹۱۲۴۴۵۱۶۲۶
سید امین میرزایی	راهبر فناوری	رعایت سطح کیفیت و الزامات فنی کارفرما و انتخاب و تایید فناوری‌های زیرساختی پروژه	sayyidjan@chmail.ir	۰۹۱۲۵۰۳۲۹۰۸

حال با توجه به نقش‌ها و افراد درگیر در این پروژه، نحوه ارتباط میان شرکت آدین به عنوان پیمانکار و شرکت سحاب‌پرداز به عنوان کارفرما، در هر مرحله از فازهای پیشرفت پروژه، در جدول زیر مشخص شده است.

جدول 15- برنامه ارتباطی میان پیمانکار و کارفرما

عنوان ارتباط	شرح ارتباط	برنامه زمانی	نحوه ارتباط	نتایج	مسئول
جلسه تحویل فاز	جلسه تحویل خروجی‌های هر فاز به منظور تسویه حساب فاز	منطبق بر جدول زمانبندی پروپوزال	جلسه حضوری یا آنلاین	تست خروجی منطبق با جدول تحویل‌دادنی هر فاز	VP

عنوان ارتباط	شرح ارتباط	برنامه زمانی	نحوه ارتباط	نتایج	مسئول
جلسه تحویل نسخه عملیاتی	تحویل نسخه بر روی سرور مبتنی بر فازهای قرارداد	منطبق بر جدول زمانبندی پروپوزال	جلسه حضوری یا آنلاین	تست عملیاتی خروجی‌های فنی نصب شده بر روی سرور در هر فاز	مدیر پروژه
گزارش پیشرفت فنی پروژه	جلسه مستمر برای ارائه گزارش پیشبرد فنی پروژه	دو هفته یکبار	جلسه حضوری یا آنلاین	گزارش پیشرفت پروژه از منظر فنی، بررسی بک‌لاگ‌ها و بررسی انطباق زمانی پیشرفت پروژه	مدیر پروژه
گزارش عملکرد پروژه	شرح فعالیت‌های انجام شده در یک ماه گذشته	ماهانه	ایمیلی	گزارش پیشرفت	مدیر پروژه
نیازمندی‌های خاص (درخواست جلسه ضروری)	برگزاری جلسه برای موارد ضروری غیرقابل پیش‌بینی از سوی کارفرما یا پیمانکار	مطابق با موارد اضطراری	جلسه حضوری یا آنلاین	رفع مشکل به وجود آمده به صورت اضطرار	VP

۴-۷- برنامه مدیریت مخاطرات پروژه

ریسک‌های این پروژه در قالب FMEA به صورت ریشه، علت و اثر در تصویر زیر نمایش داده شده است. در این مدل که یک قاعده مدیریت پیش‌بینانه ریسک است، دارای سه لایه اصلی شامل ریشه بروز ریسک، ریسک احتمالی و اثرات ناشی از بروز این ریسک، تقسیم بندی شده است. اثر نهایی تمامی این موارد، تاخیر در تحویل پروژه و تغییر زمانبندی تحویل فازها از سوی پیمانکار خواهد بود که پیمانکار هیچ دخل و تصری در این موارد نداشته و تمامی این ریسک‌ها ناشی از عملکرد کارفرما خواهد بود. نحوه برخورد با هر یک از حالات ریسک بیان شده در شمای کلان FMEA در ادامه تشریح شده است.

شکل 4- مدیریت مخاطرات پروژه بر مبنای FMEA

جدول زیر بیانگر میزان اولویت ریسک‌های محتمل در پروژه بر اساس محاسبه میزان RPN هر سناریوی پیش‌بینی شده در FMEA است. میزان RPN[1] عددی بین ۱ تا ۱۰۰۰ است که میزان اولویت ریسک را نشان می‌دهد. RPN از ضرب سه عدد شدت خطر (S)، احتمال وقوع (O) و میزان غیر قابل تشخیص بودن (D) به دست می‌آید. مواردی که RPN بالاتری دارند، بایستی با اولویت بیشتری بررسی شوند که می‌توانند در جلسات اضطراری بیان شده در برنامه ارتباطی

[1] risk priority number

کارفرما و پیمانکار، مطرح شوند. در این جدول، عامل بروز ریسک به عنوان عامل اصلی اولویت‌بندی شده است.

جدول 16- اولویت‌بندی ریشه بروز ریسک‌های پروژه

ریشه بروز ریسک	شدت خطر	احتمال وقوع	میزان عدم تشخیص	RPN	اولویت
عدم دسترسی به داده نمونه مسئله	۵	۴	۴	۸۰	۵
عدم تحویل به موقع داده‌های نمونه مسئله	۹	۷	۵	۳۱۵	۲
عدم دسترسی به موقع به زیرساخت مرحله stage	۷	۶	۵	۲۱۰	۳
عدم دسترسی به موقع به زیرساخت مرحله production	۷	۶	۵	۲۱۰	۴
هزینه بالای یکپارچه‌سازی MLOps با گرانیت	۱۰	۷	۵	۳۵۰	۱

برنامه ارائه شده به منظور مدیریت هر یک از این ریسک‌ها به شرح زیر خواهد بود:

- در صورت عدم دسترسی به داده نمونه مسئله و یا تاخیر در دسترسی به داده مورد نیاز برای صورت‌بندی مسئله، یا بایستی داده ماک تهیه شود و یا مسئله جدید با داده در دسترس از سوی کارفرما ارائه شود. طبیعتا در صورت بروز این مشکل، هزینه و زمان مورد نیاز برای این موضوع بایستی به زمان و هزینه پروژه اضافه گردد.
- در صورت فراهم نشدن به موقع زیرساخت مورد نیاز در دو محیط stage و یا production، زمان تحویل فازهای پروژه متعاقبا تا زمان فراهم شدن زیرساخت مناسب به تعویق خواهد افتاد.
- با توجه به الزامات بیان شده توسط کارفرما در جلسات تعریف و تدقیق پروژه، پیش‌فرض این هست که سرویس‌های گرانیت در قالب APIهای خوش‌تعریف، در دسترس و قابل‌استفاده خواهند بود. درصورت نقض این پیش‌فرض یا تحمیل هزینه پیاده‌سازی زیاد به بستر MLOps جهت یکپارچه شدن با گرانیت، می‌بایست مشکلات، چالش‌ها و هزینه‌های تحمیلی متناسب با آن‌ها در جلساتی میان تیم فنی مجری و کارفرما مطرح شده و در خصوص برآورد هزینه و زمان لازم برای یکپارچه‌سازی، توافق شود. سپس زمان و هزینه توافق شده، در قالب متمم به قرارداد اضافه گردد. همچنین درصورت عدم

توافق در زمان و هزینه ناشی از پیچیدگی‌های یکپارچه‌سازی MLOps و گرانیت، مجری موظف به تحویل MLOps به صورت مستقل می‌باشد.

- درصورت تاخیر کارفرما در نصب، راه‌اندازی و تحویل گرانیت، زمان تاخیر به زمان پروژه اضافه می‌شود. همچنین در صورتی که تاخیر به وجود آمده منجر به تحمیل کار اضافه به تیم مجری (در خصوص نصب MLOps با و بدون گرانیت) شود، جبران هزینه مذکور می‌بایست در قالب متمم در قرارداد لحاظ گردد.

۵- زمان‌بندی کلی و تحلیل هزینه پروژه

در این بخش طی دو جدول اصلی، زمانبندی کلی اجرای پروژه و محاسبه هزینه پروژه (مبتنی بر نیرو انسانی تیم اجرا)، ارائه شده است.

۵-۱- جدول زمانبندی کلی اجرا

زمانبندی کلی اجرای پروژه و موعدهای تحویل نسخه‌ها و فازها به شرح مندرج در جدول 17 است.

جدول 17- زمانبندی کلی اجرای پروژه و موعد تحویل فازها

فاز/نسخه	موعد تحویل (ماه از شروع پروژه)													
	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴
نسخه ۱ مطابق با مشخصات مندرج در جدول 5		✓												
نسخه ۲ مطابق با مشخصات مندرج در جدول 7					✓									
نسخه ۳ مطابق با مشخصات مندرج در جدول 9								✓						
نصب و راه‌اندازی نسخه ۳ در محیط بهره‌بردار									✓					
نسخه ۴ مطابق با مشخصات مندرج در جدول 11											✓			
نسخه ۵ مطابق با مشخصات مندرج در جدول ۱۳														✓
نصب و راه‌اندازی نسخه نهای در محیط بهره‌بردار														✓

۵-۲- تحلیل هزینه

با توجه به اینکه این پروژه ذاتا یک پروژه توسعه نرم‌افزار است، عمده هزینه آن مرتبط با بخش نیرو انسانی می‌باشد. لذا سایر هزینه‌ها مانند سربار اجرایی، تجهیزات مورد نیاز آزمایشگاه و محیط توسعه و ... به‌صورت سرشکن در هزینه معادل نفر/ساعت نیرو انسانی اختصاصی تیم توسعه محاسبه شده است. بر این اساس، هزینه انجام پروژه و محاسبات آن مطابق جدول 18 می‌باشد. همان‌طور که مشخص است، هزینه اجرای پروژه طی ۱۴ ماه شمسی با تیمی متشکل از ۸ نفر، معادل ۱۱،۰۱۶،۰۰۰،۰۰۰ (یازده میلیارد و شانزده میلیون) تومان براورد شده است.

جدول 18- محاسبه هزینه اجرای پروژه

تخصص/نقش	تعداد نفرات	نفر/ماه حضور در تیم	نفر/ساعت حضور در تیم
مدیر پروژه	۱	۱۴.۰	۲۵۲۰
مهندس یادگیری ماشین	۲	۲۱.۰	۳۷۸۰
مهندس DevOps	۳	۳۷.۰	۶۶۶۰
مهندس داده	۱	۱۴.۰	۲۵۲۰
راهبر فناوری	۱	۴.۷	۸۴۰
مجموع	**۸**	**۹۰.۷**	**۱۶،۳۲۰**
هزینه هر نفر/ساعت (هزار تومان)	۶۷۵		
هزینه کل پروژه (میلیون تومان)	**۱۱،۰۱۶**		

همچنین آدین آمادگی دارد درصورت نیاز کارفرما، به جهت تامین اطمینان ایشان از کیفیت اجرای پروژه، رزومه افراد کلیدی پروژه شامل مدیر، راهبر/راهبران فناوری و حداقل یکی از مهندسین ارشد تیم را به کارفرما ارائه و تایید ایشان را اخذ نماید.

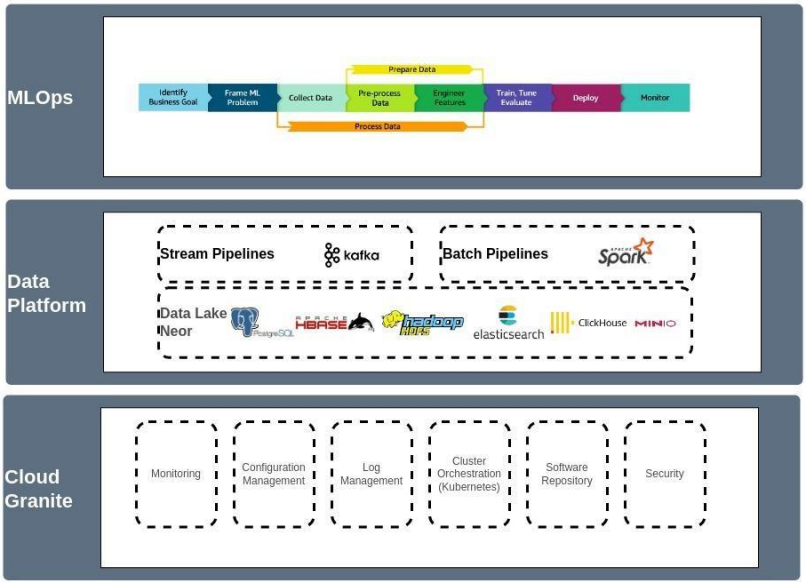
۶- پیوست: موارد فنی مورد توجه کارفرما

۶-۱- مقدمه

هدف از تدوین این پیوست، کمک به ایجاد دیدی شفاف و صریح بین کارفرما و پیمانکار در توسعه محصول MLOps می‌باشد. در این پیوست که محتوای آن از سوی کارفرما ارائه شده، ابتدا به بیان محدوده و دامنه مورد انتظار از این محصول خواهیم پرداخت و در ادامه نیازمندی‌های مورد انتظار در زمینه توسعه و استقرا محصول MLOps شرح داده خواهد شد.

۶-۲- محدوده و دامنه محصول

در حال حاضر برخی از محصولات از جمله دو محصول کلود گرانیت و دیتا پلتفرم توسط تیم فنی شرکت سحاب توسعه داده شده یا درحال‌توسعه می‌باشند؛ بنابراین این انتظار وجود دارد که محصول MLOps از طرفی سازگاری کافی با این دو پلتفرم را داشته باشد و از طرفی نیاز است که از سرویس و خدمات آن‌ها به‌خوبی بهره بگیرد. برای روشن‌شدن موضوع به‌اختصار به بیان ویژگی‌ها و دامنه این محصولات خواهیم پرداخت. در زیر شمای کلی از محصولات نمایش‌داده‌شده است.



۳-۶- زیرساخت رایانش ابری گرانیت

این پلتفرم وظیفه ایجاد بستر ابری را بر عهده دارد و برای این موضوع سرویس‌های مختلفی مانند مانیتورینگ، مدیریت لاگ و زیرساخت رایانش ابری (Kubernetes) را ارائه می‌کند؛ بنابراین محصول MLOps باید با این زیرساخت سازگار بوده و بتواند از سرویس‌های ارائه شده توسط آن استفاده نماید.

دیتا پلتفرم

این بستر از سه بخش اصلی زیر تشکیل شده است:

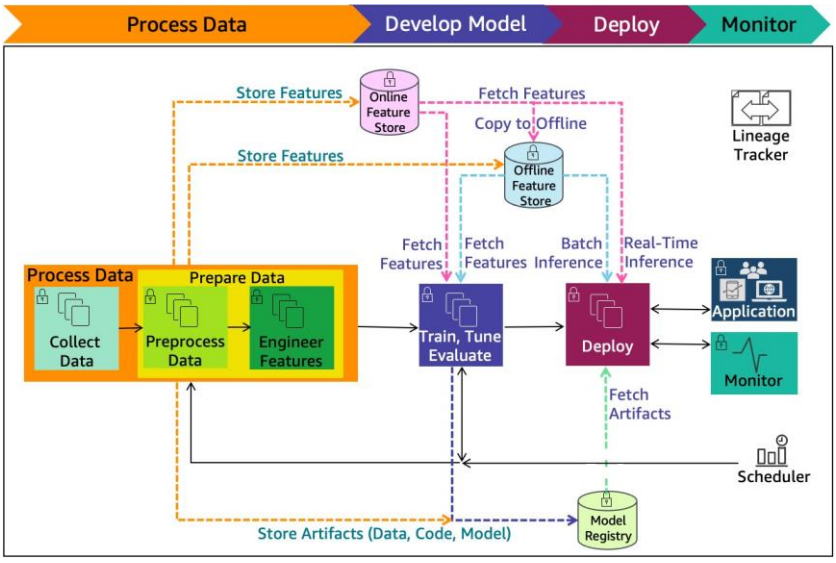
- دریاچه داده: این بخش وظیفه ارائه بستری برای ذخیره‌سازی داده با ویژگی‌های مختلف و برای استفاده‌های متفاوت را بر عهده دارد.
- پردازش جریانی: این بستر امکان پردازش جریانی بر روی داده با امکان تعامل با دریاچه داده را در اختیار کاربر قرار می‌دهد

- پردازش دسته‌ای: این بستر امکان پردازش دسته‌ای بر روی داده ذخیره شده در دریاچه داده را بر عهده دارد.

سازگاری و تجمیع محصول زیرساخت MLOps با این پلتفرم اهمیت بسیار بالایی دارد و در توسعه آن دقت به این موضوع نقش کلیدی برای کارفرما دارد.

۴-۶- محصول MLOps

محصول MLOps ابزاری است که می‌تواند در جمع‌آوری داده، ذخیره‌سازی داده، ویژگی (Feature)و مدل، بهبود داده، مهندسی نیازمندی (Requirement Engineering)، مهندسی ویژگی (Feature Engineering)، مهندسی داده (DataEngineering)، مهندسی مدل (Model Engineering)، تست و اعتبارسنجی مدل، نصب و راه‌اندازی مدل، جریان CI/CD مدل، داده و کد، نظارت و هشداردهی (Monitoring & Triggering) و ارائه بستر پردازشی و ذخیره‌سازی متمرکز ایفای نقش کند و فرایند توسعه موتورهای تحلیلی را به‌مراتب ساده‌تر و بهینه‌تر کند. شکل زیر معماری کلی زیرساخت MLOps حاوی مؤلفه‌های اصلی و ارتباطات بین آن‌ها است.



مؤلفه‌های این معماری در ادامه شرح داده شده است:

- **پردازش داده (process data)** در این مؤلفه کتابخانه‌هایی در جهت استخراج داده از منابع مختلف، استخراج مشخصات آماری داده، تمیزسازی داده و ایجاد فیچرها ارائه می‌شود.
- **انباره داده آنلاین/آفلاین فیچرها (online/offline store)** با ذخیره‌کردن فیچرها، محاسبات تکراری آن‌ها در بخش‌های مختلف سازمان حذف می‌شود. انباره داده آنلاین در جهت دریافت سریع فیچرها برای استفاده در مرحله استنتاج به کار می‌رود. انباره داده آفلاین، تاریخچه مقادیر فیچرها را نگهداری می‌کند و در مرحله یادگیری مدل مورداستفاده قرار می‌گیرد. برای این مؤلفه از ابزارهای موجود در دریاچه داده استفاده می‌شود و در صورت نیاز به ابزاری که توسط این زیرساخت ارائه نمی‌شود برای انتخاب آن نیاز به هماهنگی با تیم کارفرما وجود دارد.
- **توسعه مدل (train/tune/evaluate)** این مؤلفه بستری برای توسعه مدل از طریق آزمایش‌ها مختلف را فراهم می‌کند. این مؤلفه شامل نوت‌بوک‌ها و کتابخانه‌های مرسوم برای توسعه و ارزیابی مدل و کتابخانه‌هایی برای تنظیم پارامترها می‌باشد.
- **رجیستری مدل (model registry)** رجیستری مدل یک مخزن برای ذخیره مدل‌های یادگیری ماشین و فراداده‌های مرتبط است. در این رجیستری، نسخه‌های مختلف داده و مدل قرار می‌گیرد.
- **مانیتورینگ (monitoring)** این مؤلفه وظیفه نظارت بر مدل و تشخیص مشکلات را بر عهده دارد. مشکلات می‌تواند مرتبط باکیفیت داده، کیفیت مدل و دریفت باشد.
- **استقرار (deploy)** این مؤلفه وظیفه استقرار مدل در محیط عملیاتی و سرو آن را بر عهده دارد. این مؤلفه عملیاتی مانند مقیاس‌پذیری متناسب با بار ورودی و استقرار قناری را پشتیبانی می‌کند.

- **زمان‌بند (scheduler)** این مؤلفه می‌تواند فرایند یادگیری مجدد را در بازه‌های زمانی مشخص اجرا کند.
- **ردیاب جامع (lineage tracker)** این مؤلفه بستری برای ثبت و ضبط اطلاعات تمام منابع در هر نقطه از زمان را ارائه می‌کند. این اطلاعات می‌تواند شامل نسخه کد، داده، فیچرها، مدل و نتایج مربوطه باشد.

۵-۶- نیازمندی‌ها فنی در توسعه و استقرار محصول

۱. پروژه در دو محیط آزمایشگاه و عملیاتی مستقر خواهد شد. به‌منظور اینکه تیم پیمانکار بتواند استقلال مورد نظر خود را داشته باشد، راه‌اندازی آزمایشگاه که شامل مولفه‌های زیرساختی است و نصب و راه‌اندازی پروژه در محیط آزمایشگاه کاملاً بر عهده تیم پیمانکار می‌باشد. در محیط عملیاتی، زیرساخت ابری پروژه شامل کوبرنیتیز، پایگاه‌داده‌ها و ابزارهای زیرساختی مانند ابزار مانیتورینگ و مدیریت لاگ توسط تیم کارفرما نصب و راه‌اندازی خواهد شد؛ بنابراین فرایند نصب و اعمال تمامی مولفه‌های محصول MLOps، پلاگین‌ها، تنظیمات باید قابل‌تکرار در محیط‌های مختلف بوده و بستری برای نصب مجدد پروژه در محیط دلخواه ارائه شود تا بتوان پروژه را روی محیط عملیاتی نیز راه‌اندازی نمود.
۲. تمامی کدهای پروژه باید از ابتدا در یک codebase که توسط تیم کارفرما نیز قابل‌دسترسی است، قرار گیرند. تمامی مولفه‌های توسعه داده شده، تنظیمات صورت‌گرفته، مولفه‌های و پلاگین‌های توسعه داده شده، پایپ لاین‌های داده، و اسکریپت‌های نصب، راه‌اندازی، و نگهداری باید در codebase قرار گیرند.
۳. تمامی مولفه‌های توسعه داده شده باید به‌صورت Docker قابل‌اجرا باشند .Dockerfileهای مربوطه و اسکریپت‌های Build باید در codebase قرار گیرد.
٤. ابزارهای 3rd Party مورداستفاده باید با هماهنگی با تیم کارفرما انتخاب شوند. به‌عنوان‌مثال مدنظر است برای orchestration از ابزار Kubernetes و برای صف از ابزار Kafka استفاده شود.
۵. مولفه‌های توسعه داده شده و تنظیمات صورت‌گرفته باید به‌نوعی باشند که بر روی زیرساخت ابری نیز قابل‌اعمال بوده و سازگار باشد.

۶. لیست تغییرات بین هر دو نسخه اصلی ارائه شده و درصورتی‌که تغییر از یک نسخه به نسخه دیگر نیازمند migration می‌باشد راهنما و اسکریپت لازم برای این تغییر ارائه شود.
۷. تیم پیمانکار موظف به ارائه مستندات و آموزش کافی در مورد جزئیات مؤلفه‌ها و همچنین نصب، پیکربندی، راه‌اندازی، و نگهداری محصول می‌باشد.
۸. در صورت نیاز، نماینده فنی کارفرما باید بتواند جلسات هفتگی با تیم پیمانکار داشته باشد تا در جریان نحوه و کیفیت پیشبرد اهداف قرار گیرد و از طرفی آموزش‌های موردنیاز را دریافت کند.
۹. در صورت رخداد مشکلی در نصب و راه‌اندازی و یا Bug در زمان اجرا در محیط عملیاتی، تیم پیمانکار موظف به حل مشکل می‌باشد.
۱۰. اصول کیفیت کد شامل Clean Code، Unit Testing و Documentation باید در codebase رعایت شود.
۱۱. کدها، مؤلفه‌ها، آرتیفکت‌ها، تنظیمات و اسکریپت‌های نصب و راه‌اندازی باید طی جلساتی به نماینده کارفرما تحویل داده شده و انتقال دانش مورد نظر نیز انجام پذیرد. موارد تحویلی باید از جهت دارابودن سطح کیفیت کافی ذکر شده در بندهای بالا، مورد تأیید کارفرما قرار گیرد.
۱۲. تمامی مؤلفه‌ها باید به‌صورت کوبرنتیزی (از طریق هلم) انتشار یابند. سرویس‌های غیر کوبرنتیزی موردنیاز از طریق مذاکره و تخصیص زمان قابل فراهم‌سازی است.
۱۳. پادها امکان دسترسی به دیسک‌های لوکال و یا اجرا با privilege بالا را ندارند و برای بحث ذخیره‌کردن state باید از api سرویس‌های گرانیتی مانند پایگاه‌های داده استفاده کنند.
۱۴. تمامی objectهای کوبرنتیزی پروژه باید در namespace مشخص شده قرار گیرند. استفاده از taint و affinity با مذاکره قابل‌انجام است.
۱۵. به‌منظور اتصال به سرویس‌های موجود در گرانیت، آدرس و credentials سرویس‌ها به‌صورت متنی در اختیار سرویس‌گیرنده قرار می‌گیرد.
۱۶. برای مدیریت پیش‌نیازهای مؤلفه‌ها بر روی بستر گرانیت مانند ایجاد یا تغییر پایگاه‌های داده و جداول، می‌توان از CRDها و init-container های گرانیتی استفاده نمود.

۱۷. مؤلفه‌ها باید از بستر مانیتورینگ (Grafana/Prometheus) و مدیریت لاگ (EFK)گرانیت استفاده کرده ،و Observability موردنیاز برای نگهداشت سامانه با SLA مشخص شامل آلرت‌های disaster و high و داشبوردهای مناسب را ایجاد کنند.

۱۸. SPOF در سطح فرایند یا داده نباید وجود داشته باشد؛ لذا تمامی مؤلفه‌ها باید HA باشند و داده‌ها بر روی بسترهای با replication ذخیره‌سازی شوند. همچنین درصورتی‌که نیاز به فرایندهای مشخصی برای Disaster Recovery همچون پشتیبان‌گیری از داده‌های حیاتی وجود دارد، این فرایندها باید مستندسازی یا خودکار شوند.