aws

Menggunakan Amazon Comprehend Medical dan untuk perawatan kesehatan LLMs dan ilmu kehidupan

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Menggunakan Amazon Comprehend Medical dan untuk perawatan kesehatan LLMs dan ilmu kehidupan

Table of Contents

Menggunakan Amazon Comprehend Medical dan untuk perawatan kesehatan LLMs dan ilmu kehidupan

Amazon Web Services (???kontributor)

Desember 2025 (riwayat dokumen)

Ikhtisar

Volume data medis yang terus meningkat dan kebutuhan akan pemrosesan yang efisien dan akurat telah mendorong adopsi pemrosesan bahasa alami (NLP) dengan teknologi kecerdasan buatan dan pembelajaran mesin (AI/ML). Model pengklasifikasi terlatih dan model bahasa besar (LLMs) telah muncul sebagai alat yang ampuh untuk berbagai tugas NLP medis, termasuk menjawab pertanyaan klinis, ringkasan laporan, dan pembuatan wawasan. Namun, domain perawatan kesehatan dan ilmu kehidupan menghadirkan tantangan unik karena kompleksitas terminologi medis, pengetahuan khusus domain, dan persyaratan peraturan. Secara efektif menggunakan pengklasifikasi terlatih atau LLMs dalam domain ini memerlukan pendekatan yang dirancang dengan baik yang menggabungkan kekuatan model ini dengan sumber daya dan teknik khusus domain.

Praktik industri dalam perawatan kesehatan dan ilmu kehidupan secara tradisional mengandalkan sistem berbasis aturan, pengkodean manual, dan proses tinjauan ahli. Sistem dan proses ini memakan waktu dan rawan kesalahan. Integrasi teknologi AI dan NLP, seperti Amazon Comprehend Medical dan model foundation di Amazon Bedrock, menawarkan solusi yang efisien dan terukur untuk memproses data medis sekaligus meningkatkan akurasi dan konsistensi.

Panduan ini mengeksplorasi penggunaan Amazon Comprehend Medical dan untuk otomatisasi cerdas di LLMs industri perawatan kesehatan. Ini menguraikan praktik terbaik, tantangan, dan pendekatan praktis untuk merampingkan pengkodean medis, ekstraksi informasi pasien, dan mencatat proses ringkasan. Dengan menggunakan kemampuan Amazon Comprehend Medical LLMs dan, organisasi perawatan kesehatan dapat membuka tingkat efisiensi operasional baru, mengurangi biaya, dan berpotensi meningkatkan perawatan pasien.

Panduan ini merinci pertimbangan unik dari domain perawatan kesehatan, seperti memahami terminologi medis, menggunakan domain khusus LLMs, dan mengatasi keterbatasan sistem. AI/ML Ini memberikan jalur keputusan yang komprehensif untuk manajer TI perawatan kesehatan, arsitek, dan petunjuk teknis untuk menilai kesiapan organisasi, mengevaluasi opsi implementasi, dan menggunakan alat yang sesuai Layanan AWS dan alat untuk otomatisasi yang sukses.

Dengan mengikuti pedoman dan praktik terbaik yang diuraikan dalam panduan ini, organisasi perawatan kesehatan dapat memanfaatkan kekuatan AI/ML teknologi sambil menavigasi kompleksitas domain medis. Pendekatan ini mendukung kepatuhan terhadap pedoman etika dan peraturan dan mempromosikan penggunaan sistem AI yang bertanggung jawab dalam perawatan kesehatan. Ini dirancang untuk menghasilkan wawasan yang akurat dan pribadi.

Audiens yang dituju

Panduan ini ditujukan untuk pemangku kepentingan teknologi, arsitek, pemimpin teknis, dan pengambil keputusan yang ingin menerapkan solusi pemrosesan bahasa alami bertenaga AI untuk analisis dan otomatisasi data medis.

Tujuan

Organisasi kesehatan dan ilmu hayati dapat memenuhi beberapa tujuan bisnis dengan menggunakan Amazon Comprehend Medical dan. LLMs Hasil ini biasanya mencakup peningkatan efisiensi operasional, mengurangi biaya, dan meningkatkan perawatan pasien. Bagian ini menguraikan tujuan bisnis utama dan manfaat terkait dari penerapan strategi dan praktik terbaik yang diuraikan dalam panduan ini.

Berikut ini adalah beberapa tujuan yang dapat dicapai organisasi dengan menerapkan pedoman dan praktik terbaik dalam panduan ini:

- Mengurangi waktu pengembangan — Tujuan akhir panduan ini adalah untuk mengurangi waktu pengembangan dengan biaya, mengurangi utang teknis, dan mengurangi potensi kegagalan proyek dari POC. Dengan memahami AI/ML layanan utama, seperti Amazon Comprehend Medical, dan keuntungan dan keterbatasan penggunaan LLM untuk tugas-tugas perawatan kesehatan, bisnis dapat mencapai waktu yang lebih cepat untuk memasarkan dan meningkatkan kecepatan mereka dalam memenuhi tujuan bisnis.
- Ekstrak informasi untuk mengotomatiskan tugas pengkodean medis - Setelah kunjungan pasien, spesialis dan penyedia pengkodean dapat mengekstrak wawasan dari teks medis, seperti catatan subjektif, objektif, penilaian, dan rencana (SOAP). Hal ini dapat mengurangi upaya dokumentasi manual dan membantu penyedia fokus pada kebutuhan pasien. Dengan menggabungkan kemampuan pengenalan entitas Amazon Comprehend Medical LLMs dengan, organisasi dapat mengekstrak informasi medis yang relevan dari catatan pasien, catatan klinis, dan sumber data perawatan kesehatan lainnya. Ini dapat meminimalkan kesalahan manusia dan mempromosikan praktik yang konsisten.

- Meringkas catatan pasien dan dokumentasi klinis - Ringkasan otomatis riwayat pasien, rencana perawatan, dan hasil medis dapat menghemat waktu yang berharga bagi penyedia layanan kesehatan. LLMs dapat membantu menghasilkan dokumentasi klinis yang komprehensif dan terstruktur. Anda bisa mendapatkan konteks tambahan dengan Amazon Comprehend Medical, menggunakan LLM domain medis, atau menyempurnakan LLM dengan data medis. Pendekatan ini dapat membantu memberikan ringkasan yang akurat dan memastikan bahwa dokumentasi mematuhi persyaratan dan standar kepatuhan.
- Mendukung keputusan klinis dan perawatan pasien — Dengan menggunakan tautan ontologi di Amazon Comprehend Medical LLMs dan dengan menggunakan, penyedia dapat menjawab pertanyaan medis atau mencari rekomendasi untuk menangani perawatan pasien. Ini memberdayakan profesional kesehatan untuk membuat keputusan berdasarkan informasi yang meningkatkan hasil pasien dan mengurangi risiko kesalahan medis.

Pendekatan AI dan NLP generatif untuk perawatan kesehatan dan ilmu kehidupan

Natural language processing (NLP) adalah teknologi pembelajaran mesin yang memberi komputer kemampuan untuk menafsirkan, memanipulasi, dan memahami bahasa manusia. Organisasi kesehatan dan ilmu hayati memiliki volume data yang besar dari catatan pasien. Mereka dapat menggunakan perangkat lunak NLP untuk memproses data ini secara otomatis. Misalnya, mereka dapat menggabungkan NLP dengan AI generatif untuk merampingkan pengkodean medis, mengekstrak informasi pasien, dan meringkas catatan.

Bergantung pada tugas NLP yang ingin Anda lakukan, arsitektur yang berbeda mungkin paling cocok untuk kasus penggunaan Anda. Panduan ini membahas opsi AI dan NLP generatif berikut untuk aplikasi perawatan kesehatan dan ilmu hayati di: AWS

- Menggunakan Amazon Comprehend Medical— Pelajari cara menggunakan Amazon Comprehend Medical secara mandiri, tanpa mengintegrasikannya dengan model bahasa besar (LLM).
- Menggabungkan Amazon Comprehend Medical dengan model bahasa besar— Pelajari tentang cara menggabungkan Amazon Comprehend Medical dengan LLM dalam arsitektur Retrieval Augment Generation (RAG).
- Menggunakan model bahasa besar untuk perawatan kesehatan dan kasus penggunaan ilmu hayati- Pelajari tentang cara menggunakan LLM untuk aplikasi perawatan kesehatan dan ilmu hayati, baik dengan menggunakan LLM yang disetel dengan baik atau arsitektur RAG.

Menggunakan Amazon Comprehend Medical

Amazon Comprehend Medical adalah Layanan AWS yang mendeteksi dan mengembalikan informasi berguna dalam teks klinis yang tidak terstruktur seperti catatan dokter, ringkasan pelepasan, hasil tes, dan catatan kasus. Ini menggunakan model pemrosesan bahasa alami (NLP) untuk mendeteksi entitas. Entitas adalah referensi tekstual untuk informasi medis, seperti kondisi medis, obat-obatan, atau informasi kesehatan yang dilindungi (PHI).

> **Important**
>
> Amazon Comprehend Medical bukan pengganti saran medis profesional, diagnosis, atau perawatan. Amazon Comprehend Medical memberikan skor kepercayaan yang menunjukkan

tingkat kepercayaan pada keakuratan entitas yang terdeteksi. Identifikasi ambang kepercayaan yang tepat untuk kasus penggunaan Anda, dan gunakan ambang kepercayaan tinggi dalam situasi yang membutuhkan akurasi tinggi. Dalam kasus penggunaan tertentu, hasil harus ditinjau dan diverifikasi oleh pengulas manusia yang terlatih dengan tepat. Misalnya, Amazon Comprehend Medical hanya boleh digunakan dalam skenario perawatan pasien setelah ditinjau untuk akurasi dan penilaian medis yang baik oleh profesional medis terlatih.

Anda dapat mengakses Amazon Comprehend Medical melalui Konsol Manajemen AWS,AWS CLI(), AWS Command Line Interface atau melalui. AWS SDKs AWS SDKs Tersedia untuk berbagai bahasa pemrograman dan platform, seperti Java, Python, Ruby, .NET, iOS, dan Android. Anda dapat menggunakan SDKs untuk mengakses Amazon Comprehend Medical secara terprogram dari aplikasi klien Anda.

Bagian ini mengulas kemampuan utama Amazon Comprehend Medical. Ini juga membahas keuntungan menggunakan layanan ini dibandingkan dengan model bahasa besar (LLM).

Amazon Comprehend Medical kemampuan

Amazon Comprehend Medical APIs menawarkan untuk inferensi hampir real-time dan batch. Ini APIs dapat menelan teks medis dan memberikan hasil untuk tugas NLP medis dengan menggunakan pengenalan entitas medis dan mengidentifikasi hubungan entitas. Anda dapat melakukan analisis baik pada file tunggal atau sebagai analisis batch pada beberapa file yang disimpan dalam bucket Amazon Simple Storage Service (Amazon S3). Amazon Comprehend Medical menawarkan operasi API analisis teks berikut untuk deteksi entitas sinkron:

- Mendeteksi entitas - Mendeteksi kategori medis umum seperti anatomi, kondisi medis, kategori PHI, prosedur, dan ekspresi waktu.
- Deteksi PHI — Mendeteksi entitas tertentu seperti usia, tanggal, nama, dan informasi pribadi serupa.

Amazon Comprehend Medical juga menyertakan beberapa operasi API yang dapat Anda gunakan untuk melakukan analisis teks batch pada dokumen klinis. Untuk mempelajari lebih lanjut tentang cara menggunakan operasi API ini, lihat Kumpulan analisis teks APIs.

Gunakan Amazon Comprehend Medical untuk mendeteksi entitas dalam teks klinis dan menghubungkan entitas tersebut dengan konsep dalam ontologi medis standar, termasuk RxNorm basis pengetahuan, ICD-10-CM, dan SNOMED CT. Anda dapat melakukan analisis baik pada file tunggal atau sebagai analisis batch pada dokumen besar atau beberapa file yang disimpan dalam bucket Amazon S3. Amazon Comprehend Medical menawarkan operasi API penautan ontologi berikut:

- Menyimpulkan ICD10 CM — Operasi Index ICD10 CM mendeteksi potensi kondisi medis dan menghubungkannya dengan kode dari Klasifikasi Penyakit Internasional, Revisi ke-10, Modifikasi Klinis (ICD-10-CM) versi 2019. Untuk setiap kondisi medis potensial yang terdeteksi, Amazon Comprehend Medical mencantumkan kode dan deskripsi ICD-10-CM yang cocok. Kondisi medis yang tercantum dalam hasil termasuk skor kepercayaan, yang menunjukkan keyakinan yang dimiliki Amazon Comprehend Medical dalam keakuratan entitas terhadap konsep yang cocok dalam hasil.
- InferRxNormInferRxNormOperasi mengidentifikasi obat-obatan yang tercantum dalam catatan pasien sebagai entitas. Ini menghubungkan entitas ke pengidentifikasi konsep (RxCui) dari RxNorm database dari Perpustakaan Kedokteran Nasional. Setiap RxCui unik untuk kekuatan dan bentuk dosis yang berbeda. Obat-obatan yang terdaftar dalam hasil termasuk skor kepercayaan, yang menunjukkan keyakinan yang dimiliki Amazon Comprehend Medical dalam keakuratan entitas yang cocok dengan konsep dari basis pengetahuan. RxNorm Amazon Comprehend Medical mencantumkan CUIs Rx teratas yang berpotensi cocok untuk setiap obat yang dideteksi dalam urutan menurun berdasarkan skor kepercayaan.
- InfersNomeDCT - Operasi InfersNomeDCT mengidentifikasi konsep medis yang mungkin sebagai entitas dan menghubungkannya ke kode dari versi 2021-03 dari Nomenklatur Kedokteran Sistematisasi, Istilah Klinis (SNOMED CT). SNOMED CT menyediakan kosakata komprehensif konsep medis, termasuk kondisi medis dan anatomi, serta tes medis, perawatan, dan prosedur. Untuk setiap ID konsep yang cocok, Amazon Comprehend Medical mengembalikan lima konsep medis teratas, masing-masing dengan skor kepercayaan diri dan informasi kontekstual seperti sifat dan atribut. Konsep CT SNOMED kemudian IDs dapat digunakan untuk menyusun data klinis pasien untuk pengkodean medis, pelaporan, atau analitik klinis bila digunakan dengan polihierarki CT SNOMED.

Untuk informasi selengkapnya, lihat Analisis teks APIs dan Penautan Ontologi APIs di dokumentasi Amazon Comprehend Medical.

Kasus penggunaan untuk Amazon Comprehend Medical

Sebagai layanan mandiri, Amazon Comprehend Medical dapat menangani kasus penggunaan organisasi Anda. Amazon Comprehend Medical dapat melakukan tugas-tugas seperti berikut:

- Bantuan dengan pengkodean medis dalam catatan pasien
- Mendeteksi data informasi kesehatan yang dilindungi (PHI)
- Memvalidasi obat, termasuk atribut seperti dosis, frekuensi, dan bentuk

Hasil Amazon Comprehend Medical dapat dicerna untuk sebagian besar praktik medis. Namun, Anda mungkin perlu mempertimbangkan alternatif jika Anda memiliki batasan seperti berikut:

- Definisi entitas yang berbeda — Misalnya, definisi Anda `FREQUENCY` tentang entitas obat mungkin berbeda. Untuk frekuensi, Amazon Comprehend Medical memprediksi sesuai kebutuhan, tetapi organisasi Anda mungkin menggunakan istilah pro re nata (PRN).
- Jumlah hasil yang luar biasa — Misalnya, catatan pasien sering berisi beberapa gejala dan kata kunci yang dipetakan ke beberapa kode ICD-10-CM. Namun, beberapa kata kunci tidak berlaku untuk diagnosis. Dalam hal ini, penyedia harus mengevaluasi banyak entitas ICD-10-CM dan skor kepercayaan mereka, yang memerlukan waktu pemrosesan manual.
- Entitas khusus atau tugas NLP — Misalnya, penyedia mungkin ingin mengekstrak bukti PRN, seperti mengambil sesuai kebutuhan untuk rasa sakit. Karena ini tidak tersedia melalui Amazon Comprehend Medical, model yang berbeda diperlukan. AI/ML AI/ML Solusi yang berbeda diperlukan jika tugas NLP berada di luar pengakuan entitas, seperti meringkas, menjawab pertanyaan, dan analisis sentimen.

Menggabungkan Amazon Comprehend Medical dengan model bahasa besar

Sebuah studi 2024 oleh NEJM AI menunjukkan bahwa menggunakan LLM, dengan bidikan nol, untuk tugas pengkodean medis umumnya mengarah pada kinerja yang buruk. Menggunakan Amazon Comprehend Medical dengan LLM dapat membantu mengurangi masalah kinerja ini. Hasil Amazon Comprehend Medical adalah konteks yang berguna untuk LLM yang melakukan tugas NLP. Misalnya, memberikan konteks dari Amazon Comprehend Medical ke model bahasa besar dapat membantu Anda:

- Tingkatkan akurasi pemilihan entitas dengan menggunakan hasil awal dari Amazon Comprehend Medical sebagai konteks untuk LLM
- Menerapkan pengenalan entitas kustom, meringkas, menjawab pertanyaan, dan kasus penggunaan tambahan

Bagian ini menjelaskan bagaimana Anda dapat menggabungkan Amazon Comprehend Medical dengan LLM dengan menggunakan pendekatan Retrieval Augmented Generation (RAG). Retrieval Augmented Generation (RAG) adalah teknologi AI generatif di mana LLM mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Untuk informasi lebih lanjut, lihat Apa itu RAG.

Untuk mengilustrasikan pendekatan ini, bagian ini menggunakan contoh pengkodean medis (diagnosis) yang terkait dengan ICD-10-CM. Ini mencakup contoh arsitektur dan templat teknik yang cepat untuk membantu mempercepat inovasi Anda. Ini juga mencakup praktik terbaik untuk menggunakan Amazon Comprehend Medical dalam alur kerja RAG.

Arsitektur berbasis RAG dengan Amazon Comprehend Medical

Diagram berikut menggambarkan pendekatan RAG untuk mengidentifikasi kode diagnosis ICD-10-CM dari catatan pasien. Ini menggunakan Amazon Comprehend Medical sebagai sumber pengetahuan. Dalam pendekatan RAG, metode pengambilan biasanya mengambil informasi dari database vektor yang berisi pengetahuan yang berlaku. Alih-alih database vektor, arsitektur ini menggunakan Amazon Comprehend Medical untuk tugas pengambilan. Orkestrator mengirimkan informasi catatan pasien ke Amazon Comprehend Medical dan mengambil informasi kode ICD-10-CM. Orkestrator mengirimkan konteks ini ke model pondasi hilir (LLM), melalui Amazon Bedrock. LLM menghasilkan respons dengan menggunakan informasi kode ICD-10-CM, dan respons itu dikirim kembali ke aplikasi klien.

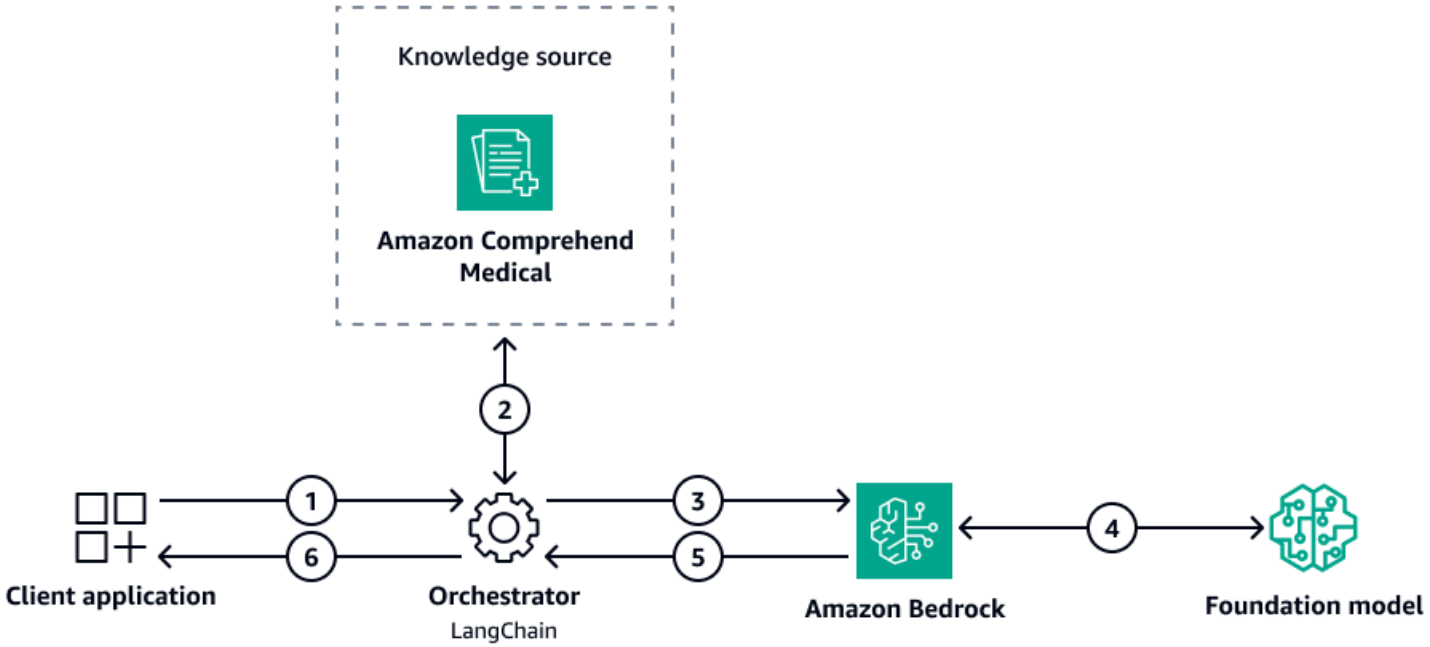


Diagram menunjukkan alur kerja RAG berikut:

1. Aplikasi klien mengirimkan catatan pasien sebagai kueri ke orkestrator. Contoh dari catatan pasien ini mungkin “Pasien adalah pasien wanita berusia 71 tahun dari Dr. X. Pasien yang dipresentasikan ke ruang gawat darurat tadi malam dengan riwayat sakit perut sekitar 7 hari hingga 8 hari, yang telah persisten. Dia tidak memiliki demam atau kedinginan yang pasti dan tidak ada riwayat penyakit kuning. Pasien menyangkal adanya penurunan berat badan yang signifikan baru-baru ini.”
2. Orkestrator menggunakan Amazon Comprehend Medical untuk mengambil kode ICD-10-CM yang relevan dengan informasi medis dalam kueri. Ini menggunakan Infer ICD10 CM API untuk mengekstrak dan menyimpulkan kode ICD-10-CM dari catatan pasien.
3. Orkestrator membuat prompt yang menyertakan templat prompt, kueri asli, dan kode ICD-10-CM yang diambil dari Amazon Comprehend Medical. Ini mengirimkan konteks yang ditingkatkan ini ke Amazon Bedrock.
4. Amazon Bedrock memproses input dan menggunakan model dasar untuk menghasilkan respons yang mencakup kode ICD-10-CM dan bukti yang sesuai dari kueri. Respons yang dihasilkan mencakup kode ICD-10-CM yang diidentifikasi dan bukti dari catatan pasien yang mendukung setiap kode. Berikut ini adalah contoh respons:

```
<response>
<icd10>
```

```
<code>R10.9</code>
<evidence>history of abdominal pain</evidence>
</icd10>
<icd10>
<code>R10.30</code>
<evidence>history of abdominal pain</evidence>
</icd10>
</response>
```

5. Amazon Bedrock mengirimkan respons yang dihasilkan ke orkestrator.
6. Orchestrator mengirimkan respon kembali ke aplikasi klien, di mana pengguna dapat meninjau respon.

Kasus penggunaan untuk menggunakan Amazon Comprehend Medical dalam alur kerja RAG

Amazon Comprehend Medical dapat melakukan tugas NLP tertentu. Untuk informasi selengkapnya, lihat Kasus penggunaan untuk Amazon Comprehend Medical.

Anda mungkin ingin mengintegrasikan Amazon Comprehend Medical ke dalam alur kerja RAG untuk kasus penggunaan lanjutan, seperti berikut ini:

- Hasilkan ringkasan klinis terperinci dengan menggabungkan entitas medis yang diekstraksi dengan informasi kontekstual dari catatan pasien
- Otomatiskan pengkodean medis untuk kasus kompleks dengan menggunakan entitas yang diekstraksi dengan informasi terkait ontologi untuk penetapan kode
- Otomatiskan pembuatan catatan klinis terstruktur dari teks tidak terstruktur dengan menggunakan entitas medis yang diekstraksi
- Menganalisis efek samping obat berdasarkan nama dan atribut obat yang diekstraksi
- Mengembangkan sistem pendukung klinis cerdas yang menggabungkan informasi medis yang diekstraksi dengan up-to-date penelitian dan pedoman

Praktik terbaik untuk menggunakan Amazon Comprehend Medical dalam alur kerja RAG

Saat mengintegrasikan hasil Amazon Comprehend Medical ke dalam prompt untuk LLM, penting untuk mengikuti praktik terbaik. Ini dapat meningkatkan kinerja dan akurasi. Berikut ini adalah rekomendasi utama:

- Memahami skor kepercayaan Amazon Comprehend Medical — Amazon Comprehend Medical memberikan skor kepercayaan untuk setiap entitas yang terdeteksi dan tautan ontologi. Sangat penting untuk memahami arti dari skor ini dan menetapkan ambang batas yang sesuai untuk kasus penggunaan spesifik Anda. Skor kepercayaan membantu menyaring entitas dengan kepercayaan rendah, mengurangi kebisingan dan meningkatkan kualitas input LLM.
- Gunakan skor kepercayaan diri dalam rekayasa cepat — Saat menyusun petunjuk untuk LLM, pertimbangkan untuk memasukkan skor kepercayaan Amazon Comprehend Medical sebagai konteks tambahan. Ini membantu LLM memprioritaskan atau menimbang entitas berdasarkan tingkat kepercayaan mereka, berpotensi meningkatkan kualitas output.
- Evaluasi hasil Amazon Comprehend Medical dengan data kebenaran dasar — Data kebenaran dasar adalah informasi yang diketahui benar. Ini dapat digunakan untuk memvalidasi bahwa AI/ML aplikasi menghasilkan hasil yang akurat. Sebelum mengintegrasikan hasil Amazon Comprehend Medical ke dalam alur kerja LLM Anda, evaluasi kinerja layanan pada sampel data Anda yang representatif. Bandingkan hasilnya dengan anotasi kebenaran dasar untuk mengidentifikasi potensi perbedaan atau area untuk perbaikan. Evaluasi ini membantu Anda memahami kekuatan dan keterbatasan Amazon Comprehend Medical untuk kasus penggunaan Anda.
- Pilih informasi yang relevan secara strategis — Amazon Comprehend Medical dapat memberikan sejumlah besar informasi, tetapi tidak semuanya mungkin relevan dengan tugas Anda. Pilih dengan hati-hati entitas, atribut, dan metadata yang paling relevan dengan kasus penggunaan Anda. Memberikan terlalu banyak informasi yang tidak relevan ke LLM dapat menimbulkan kebisingan dan berpotensi menurunkan kinerja.
- Sejajarkan definisi entitas — Pastikan bahwa definisi entitas dan atribut yang digunakan oleh Amazon Comprehend Medical selaras dengan interpretasi Anda. Jika ada perbedaan, pertimbangkan untuk memberikan konteks atau klarifikasi tambahan ke LLM untuk menjembatani kesenjangan antara hasil Amazon Comprehend Medical dan kebutuhan Anda. Jika entitas Amazon Comprehend Medical tidak memenuhi harapan Anda, Anda dapat menerapkan deteksi entitas kustom dengan menyertakan instruksi tambahan (dan contoh yang mungkin) dalam prompt.

- Berikan pengetahuan khusus domain — Meskipun Amazon Comprehend Medical memberikan informasi medis yang berharga, mungkin tidak menangkap semua nuansa domain spesifik Anda. Pertimbangkan untuk melengkapi hasil Amazon Comprehend Medical dengan sumber pengetahuan khusus domain tambahan, seperti ontologi, terminologi, atau kumpulan data yang dikuratori oleh ahli. Ini memberikan konteks yang lebih komprehensif untuk LLM.
- Patuhi pedoman etika dan peraturan — Saat berhadapan dengan data medis, penting untuk mematuhi prinsip-prinsip etika dan pedoman peraturan, seperti yang terkait dengan privasi data, keamanan, dan penggunaan sistem AI yang bertanggung jawab dalam perawatan kesehatan. Pastikan implementasi Anda mematuhi hukum dan praktik terbaik industri yang relevan.

Dengan mengikuti praktik terbaik ini, AI/ML praktisi dapat secara efektif menggunakan kekuatan Amazon Comprehend Medical dan. LLMs Untuk tugas NLP medis, praktik terbaik ini membantu mengurangi potensi risiko dan dapat meningkatkan kinerja.

Rekayasa cepat untuk konteks Amazon Comprehend Medical

Rekayasa cepat adalah proses merancang dan menyempurnakan petunjuk untuk memandu solusi AI generatif untuk menghasilkan output yang diinginkan. Anda memilih format, frasa, kata, dan simbol yang paling tepat yang memandu AI untuk berinteraksi dengan pengguna Anda secara lebih bermakna.

Bergantung pada operasi API yang Anda lakukan, Amazon Comprehend Medical mengembalikan entitas yang terdeteksi, kode ontologi dan deskripsi, dan skor kepercayaan. Hasil ini menjadi konteks dalam prompt ketika solusi Anda memanggil LLM target. Anda harus merekayasa prompt untuk menyajikan konteks dalam template prompt.

> **Note**
>
> Contoh petunjuk di bagian ini mengikuti panduan Antropik. Jika Anda menggunakan penyedia LLM yang berbeda, ikuti rekomendasi dari penyedia itu.

Secara umum, Anda memasukkan teks medis asli dan hasil Amazon Comprehend Medical ke dalam prompt. Berikut ini adalah struktur prompt umum:

```
<medical_text>
medical text
```

```
</medical_text>

<comprehend_medical_text_results>
comprehend medical text results
</comprehend_medical_text_results>

<prompt_instructions>
prompt instructions
</prompt_instructions>
```

Bagian ini menyediakan strategi untuk memasukkan hasil Amazon Comprehend Medical sebagai konteks cepat untuk tugas NLP medis umum berikut:

- Filter hasil Amazon Comprehend Medical
- Perpanjang tugas NLP medis dengan Amazon Comprehend Medical
- Terapkan pagar pembatas dengan Amazon Comprehend Medical

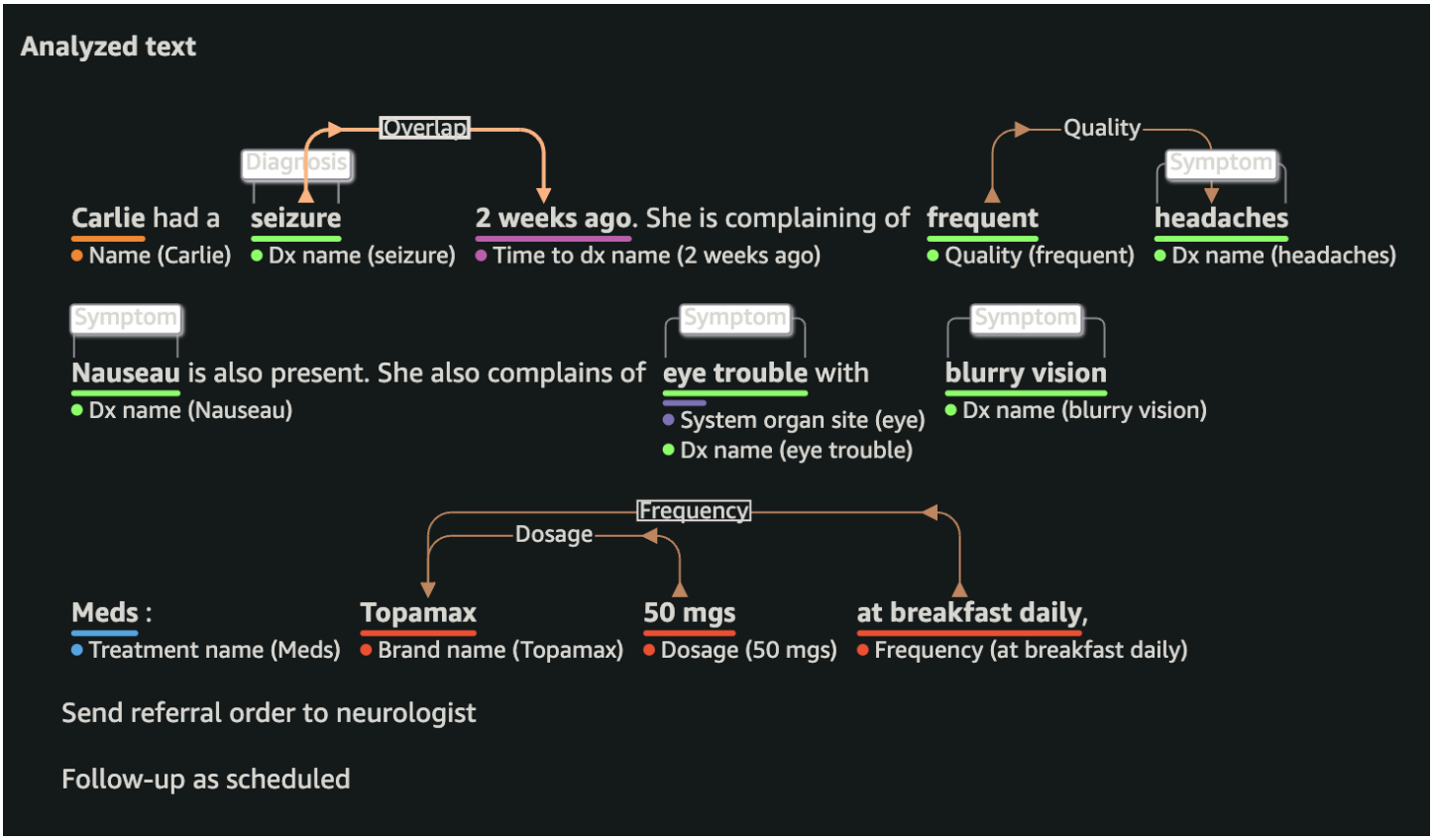
Filter hasil Amazon Comprehend Medical

Amazon Comprehend Medical biasanya menyediakan sejumlah besar informasi. Anda mungkin ingin mengurangi jumlah hasil yang harus ditinjau oleh profesional medis. Dalam hal ini, Anda dapat menggunakan LLM untuk memfilter hasil ini. Entitas Amazon Comprehend Medical menyertakan skor kepercayaan yang dapat Anda gunakan sebagai mekanisme penyaringan saat merancang prompt.

Berikut ini adalah contoh catatan pasien:

```
Carlie had a seizure 2 weeks ago. She is complaining of frequent headaches
Nausea is also present. She also complains of eye trouble with blurry vision
Meds : Topamax 50 mgs at breakfast daily,
Send referral order to neurologist
Follow-up as scheduled
```

Dalam catatan pasien ini, Amazon Comprehend Medical mendeteksi entitas berikut.



Entitas menautkan ke kode ICD-10-CM berikut untuk kejang dan sakit kepala.

Kategori	Kode ICD-10-CM	Deskripsi ICD-10-CM	Skor kepercayaan
Kejang	R56.9	Kejang yang tidak ditentukan	0.8348
Kejang	G40.909	Epilepsi, tidak spesifik, tidak sulit diobati, tanpa status epileptikus	0,5424
Kejang	R56.00	Kejang demam sederhana	0,4937
Kejang	G40.09	Kejang lainnya	0.4397
Kejang	G40.409	Epilepsi umum dan sindrom epilepsi	0.4138

		lainnya, tidak sulit diobati, tanpa status epileptikus	
Sakit kepala	R51	Sakit kepala	0,4067
Sakit kepala	R51.9	Sakit kepala, tidak ditentukan	0.3844
Sakit kepala	G44.52	Sakit kepala persisten harian baru (NDPH)	0,3005
Sakit kepala	G44	Sindrom sakit kepala lainnya	0,2670
Sakit kepala	G44.8	Sindrom sakit kepala tertentu lainnya	0,2542

Anda dapat meneruskan kode ICD-10-CM ke prompt untuk meningkatkan presisi LLM. Untuk mengurangi kebisingan, Anda dapat memfilter kode ICD-10-CM dengan menggunakan skor kepercayaan yang termasuk dalam hasil Amazon Comprehend Medical. Berikut ini adalah contoh prompt yang hanya menyertakan kode ICD-10-CM yang memiliki skor kepercayaan lebih tinggi dari 0,4:

```
<patient_note>
Carlie had a seizure 2 weeks ago. She is complaining of frequent headaches
Nausea is also present. She also complains of eye trouble with blurry vision
Meds : Topamax 50 mgs at breakfast daily,
Send referral order to neurologist
Follow-up as scheduled
</patient_note>

<comprehend_medical_results>
<icd-10>
  <entity>
    <text>seizure</text>
    <code>
      <description>Unspecified convulsions</description>
      <code_value>R56.9</code_value>
      <score>0.8347607851028442</score>
```

```
    </code>
    <code>
      <description>Epilepsy, unspecified, not intractable, without status epilepticus</
description>
      <code_value>G40.909</code_value>
      <score>0.542376697063446</score>
    </code>
    <code>
      <description>Other seizures</description>
      <code_value>G40.89</code_value>
      <score>0.4396627545356750 5</score>
    </code>
    <code>
      <description>Other generalized epilepsy and epileptic syndromes, not intractable,
 without status epilepticus</description>
      <code_value>G40.409</code_value>
      <score>0.41382506489753723</score>
    </code>
  </entity>
  <entity>
    <text>headaches</text>
    <code>
      <description>Headache</description>
      <code_value>R51</code_value>
      <score>0.4066613018512726</score>
    </code>
  </entity>
  <entity>
    <text>Nausea</text>
    <code>
      <description>Nausea</description>
      <code_value>R11.0</code_value>
      <score>0.6460834741592407</score>
    </code>
  </entity>
  <entity>
    <text>eye trouble</text>
    <code>
      <description>Unspecified disorder of eye and adnexa</description>
      <code_value>H57.9</code_value>
      <score>0.6780954599380493</score>
    </code>
    <code>
```

```
      <description>Unspecified visual disturbance</description>
      <code_value>H53.9</code_value>
      <score>0.5871203541755676</score>
    </code>
    <code>
      <description>Unspecified disorder of binocular vision</description>
      <code_value>H53.30</code_value>
      <score>0.5539672374725342</score>
    </code>
  </entity>
  <entity>
    <text>blurry vision</text>
    <code>
      <description>Other visual disturbances</description>
      <code_value>H53.8</code_value>
      <score>0.9001834392547607</score>
    </code>
  </entity>
</icd-10>
</comprehend_medical_results>

<prompt>
Given the patient note and Amazon Comprehend Medical ICD-10-CM code results above,
 please select the most relevant ICD-10-CM diagnosis codes for the patient.
For each selected code, provide a brief explanation of why it is relevant based on the
 information in the patient note.
</prompt>
```

Perpanjang tugas NLP medis dengan Amazon Comprehend Medical

Saat memproses teks medis, konteks dari Amazon Comprehend Medical dapat membantu LLM memilih token yang lebih baik. Dalam contoh ini, Anda ingin mencocokkan gejala diagnosis dengan obat-obatan. Anda juga ingin menemukan teks yang berhubungan dengan tes medis, seperti istilah yang berhubungan dengan tes panel darah. Anda dapat menggunakan Amazon Comprehend Medical untuk mendeteksi entitas dan nama obat. Dalam hal ini, Anda akan menggunakan DetectEntitiesV2 dan InferRxNorm APIs untuk Amazon Comprehend Medical.

Berikut ini adalah contoh catatan pasien:

```
Carlie had a seizure 2 weeks ago. She is complaining of increased frequent headaches
Given lyme disease symptoms such as muscle ache and stiff neck will order prescription.
Meds : Topamax 50 mgs at breakfast daily. Amoxicillan 25 mg by mouth twice a day
```

```
Place MRI radiology order at RadNet
```

Untuk fokus pada kode diagnosis, hanya entitas yang terkait `MEDICAL_CONDITION` dengan tipe with `DX_NAME` yang digunakan dalam prompt. Metadata lain dikecualikan karena tidak relevan. Untuk entitas pengobatan, nama obat bersama dengan atribut yang diekstraksi disertakan. Metadata entitas obat lain dari Amazon Comprehend Medical dikecualikan karena tidak relevan. Berikut ini adalah contoh prompt yang menggunakan hasil Amazon Comprehend Medical yang difilter. Prompt berfokus pada `MEDICAL_CONDITION` entitas yang memiliki `DX_NAME` tipe. Prompt ini dirancang untuk lebih tepat menghubungkan kode diagnosis dengan obat-obatan dan lebih tepat mengekstrak tes pesanan medis:

```
<patient_note>
Carlie had a seizure 2 weeks ago. She is complaining of increased freqeunt headaches
Given lyme disease symptoms such as muscle ache and stiff neck will order
 prescription.
Meds : Topamax 50 mgs at breakfast daily. Amoxicillan 25 mg by mouth twice a day
Place MRI radiology order at RadNet
</patient_note>

<detect_entity_results>
<entity>
    <text>seizure</text>
    <category>MEDICAL_CONDITION</category>
    <type>DX_NAME</type>
</entity>
<entity>
    <text>headaches</text>
    <category>MEDICAL_CONDITION</category>
    <type>DX_NAME</type>
</entity>
<entity>
    <text>lyme disease</text>
    <category>MEDICAL_CONDITION</category>
    <type>DX_NAME</type>
</entity>
<entity>
    <text>muscle ache</text>
    <category>MEDICAL_CONDITION</category>
    <type>DX_NAME</type>
</entity>
<entity>
```

```
    <text>stiff neck</text>
    <category>MEDICAL_CONDITION</category>
    <type>DX_NAME</type>
</entity>
</detect_entity_results>

<rx_results>
<entity>
    <text>Topamax</text>
    <category>MEDICATION</category>
    <type>BRAND_NAME</type>
    <attributes>
        <attribute>
            <type>FREQUENCY</type>
            <text>at breakfast daily</text>
        </attribute>
        <attribute>
            <type>DOSAGE</type>
            <text>50 mgs</text>
        </attribute>
        <attribute>
            <type>ROUTE_OR_MODE</type>
            <text>by mouth</text>
        </attribute>
    </attributes>
</entity>
<entity>
    <text>Amoxicillan</text>
    <category>MEDICATION</category>
    <type>GENERIC_NAME</type>
    <attributes>
        <attribute>
            <type>ROUTE_OR_MODE</type>
            <text>by mouth</text>
        </attribute>
        <attribute>
            <type>DOSAGE</type>
            <text>25 mg</text>
        </attribute>
        <attribute>
            <type>FREQUENCY</type>
            <text>twice a day</text>
        </attribute>
```

```
    </attributes>
</entity>
</rx_results>

<prompt>
Based on the patient note and the detected entities, can you please:
1. Link the diagnosis symptoms with the medications prescribed.
Provide your reasoning for the linkages.
2. Extract any entities related to medical order tests mentioned in the note.
</prompt>
```

Terapkan pagar pembatas dengan Amazon Comprehend Medical

Anda dapat menggunakan LLM dan Amazon Comprehend Medical untuk membuat pagar pembatas sebelum respons yang dihasilkan digunakan. Anda dapat menjalankan alur kerja ini pada teks medis yang tidak dimodifikasi atau pasca-diproses. Kasus penggunaan termasuk menangani informasi kesehatan yang dilindungi (PHI), mendeteksi halusinasi, atau menerapkan kebijakan khusus untuk mempublikasikan hasil. Misalnya, Anda dapat menggunakan konteks dari Amazon Comprehend Medical untuk mengidentifikasi data PHI dan kemudian menggunakan LLM untuk menghapus data PHI tersebut.

Berikut ini adalah contoh informasi dari catatan pasien yang mencakup PHI:

```
Patient name: John Doe
Patient SSN: 123-34-5678
Patient DOB: 01/01/2024
Patient address: 123 Main St, Anytown USA
Exam details: good health. Pulse is 60 bpm. needs to work on diet with BMI of 190
```

Berikut ini adalah contoh prompt yang menyertakan hasil Amazon Comprehend Medical sebagai konteks:

```
<original_text>
Patient name: John Doe
Patient SSN: 123-34-5678 Patient DOB: 01/01/2024
Patient address: 123 Main St, Anytown USA
Exam details: good health. Pulse is 60 bpm. needs to work on diet with BMI of 190
</original_text>

<comprehend_medical_phi_entities>
<entity>
```

```
  <text>John Doe</text>
  <category>PROTECTED_HEALTH_INFORMATION</category>
  <score>0.9967944025993347</score>
  <type>NAME</type>
</entity>
<entity>
  <text>123-34-5678</text>
  <category>PROTECTED_HEALTH_INFORMATION</category>
  <score>0.9998034834861755</score>
  <type>ID</type>
</entity>
<entity>
  <text>01/01/2000</text>
  <category>PROTECTED_HEALTH_INFORMATION</category>
  <score>0.9964448809623718</score>
  <type>DATE</type>
</entity>
</comprehend_medical_phi_entities>

<instructions>
Using the provided original text and the Amazon Comprehend Medical PHI entities
 detected, please analyze the text to determine if it contains any additional protected
 health information (PHI) beyond the entities already identified. If additional PHI is
 found, please list and categorize it. If no additional PHI is found, please state that
 explicitly.
In addition if PHI is found, generate updated text with the PHI removed.
</instructions>
```

Menggunakan model bahasa besar untuk perawatan kesehatan dan kasus penggunaan ilmu hayati

Ini menjelaskan bagaimana Anda dapat menggunakan model bahasa besar (LLMs) untuk aplikasi perawatan kesehatan dan ilmu hayati. Beberapa kasus penggunaan memerlukan penggunaan model bahasa besar untuk kemampuan AI generatif. Ada kelebihan dan batasan bahkan untuk sebagian besar state-of-the-art LLMs, dan rekomendasi di bagian ini dirancang untuk membantu Anda mencapai hasil target Anda.

Anda dapat menggunakan jalur keputusan untuk menentukan solusi LLM yang sesuai untuk kasus penggunaan Anda, dengan mempertimbangkan faktor-faktor seperti pengetahuan domain dan data pelatihan yang tersedia. Selain itu, bagian ini membahas praktik medis LLMs dan terbaik yang telah

dilatih sebelumnya untuk pemilihan dan penggunaannya. Ini juga membahas trade-off antara solusi yang kompleks dan berkinerja tinggi dan pendekatan yang lebih sederhana dan berbiaya rendah.

Gunakan kasus untuk LLM

Amazon Comprehend Medical dapat melakukan tugas NLP tertentu. Untuk informasi selengkapnya, lihat Kasus penggunaan untuk Amazon Comprehend Medical.

Kemampuan AI logis dan generatif dari LLM mungkin diperlukan untuk kasus penggunaan perawatan kesehatan dan ilmu hayati tingkat lanjut, seperti berikut ini:

- Mengklasifikasikan entitas medis khusus atau kategori teks
- Menjawab pertanyaan klinis
- Meringkas laporan medis
- Menghasilkan dan mendeteksi wawasan dari informasi medis

Pendekatan kustomisasi

Sangat penting untuk memahami bagaimana LLMs diimplementasikan. LLMs biasanya dilatih dengan miliaran parameter, termasuk data pelatihan dari banyak domain. Pelatihan ini memungkinkan LLM untuk menangani sebagian besar tugas umum. Namun, tantangan sering muncul ketika pengetahuan khusus domain diperlukan. Contoh pengetahuan domain dalam perawatan kesehatan dan ilmu kehidupan adalah kode klinik, terminologi medis, dan informasi kesehatan yang diperlukan untuk menghasilkan jawaban yang akurat. Oleh karena itu, menggunakan LLM apa adanya (bidikan nol tanpa menambah pengetahuan domain) untuk kasus penggunaan ini kemungkinan menghasilkan hasil yang tidak akurat. Ada beberapa pendekatan populer yang dapat Anda gunakan untuk mengatasi tantangan ini: teknik cepat, Retrieval Augmented Generation (RAG), dan fine-tuning.

Rekayasa yang cepat

Rekayasa cepat adalah proses di mana Anda memandu solusi AI generatif untuk membuat output yang diinginkan dengan menyesuaikan input ke LLM. Dengan menyusun petunjuk yang tepat dengan konteks yang relevan, dimungkinkan untuk memandu model menuju penyelesaian tugas perawatan kesehatan khusus yang memerlukan penalaran. Rekayasa cepat yang efektif dapat secara signifikan meningkatkan kinerja model untuk kasus penggunaan perawatan kesehatan tanpa memerlukan modifikasi model. Untuk informasi selengkapnya tentang teknik cepat, lihat Menerapkan teknik

prompt lanjutan dengan Amazon Bedrock (posting AWS blog). Few-shot prompt dan chain-of-thought prompt adalah teknik yang dapat Anda gunakan dalam rekayasa yang cepat.

Beberapa bidikan yang diminta

Few-shot prompt adalah teknik di mana Anda memberikan LLM dengan beberapa contoh input-output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Dalam konteks perawatan kesehatan, pendekatan ini sangat berharga untuk tugas-tugas khusus, seperti pengenalan entitas medis atau ringkasan catatan klinis. Dengan memasukkan 3-5 contoh berkualitas tinggi dalam prompt Anda, Anda dapat secara signifikan meningkatkan pemahaman model tentang terminologi medis dan pola spesifik domain. Untuk contoh petunjuk beberapa bidikan, lihat Beberapa rekayasa cepat dan penyetelan halus untuk Amazon Bedrock (posting blog). LLMs AWS

Misalnya, ketika Anda mengekstrak dosis obat dari catatan klinis, Anda dapat memberikan contoh gaya notasi berbeda yang membantu model mengenali variasi dalam cara profesional kesehatan mendokumentasikan resep. Pendekatan ini sangat efektif ketika bekerja dengan format dokumentasi standar atau ketika pola yang konsisten ada dalam data.

Chain-of-thought mendorong

Chain-of-thought (CoT) mendorong LLM melalui proses penalaran. step-by-step Ini membuatnya berharga untuk dukungan keputusan medis yang kompleks dan tugas penalaran diagnostik. Dengan secara eksplisit menginstruksikan model untuk “berpikir langkah demi langkah” saat menganalisis skenario klinis, Anda dapat meningkatkan kemampuannya untuk mengikuti protokol penalaran medis dan mengurangi kesalahan diagnostik.

Teknik ini unggul ketika penalaran klinis memerlukan beberapa langkah logis, seperti diagnosis banding atau perencanaan perawatan. Namun, pendekatan ini memiliki keterbatasan ketika berhadapan dengan pengetahuan medis yang sangat khusus di luar data pelatihan model atau ketika presisi absolut diperlukan untuk keputusan perawatan kritis.

Dalam kasus ini, menggabungkan CoT dengan pendekatan lain dapat menghasilkan hasil yang lebih baik. Salah satu opsi adalah menggabungkan CoT dengan dorongan konsistensi diri. Untuk informasi selengkapnya, lihat Meningkatkan kinerja model bahasa generatif dengan petunjuk konsistensi diri di Amazon Bedrock (AWS posting blog). Pilihan lain adalah menggabungkan kerangka penalaran, seperti ReAct prompt, dengan RAG. Untuk informasi selengkapnya, lihat Mengembangkan asisten berbasis obrolan AI generatif tingkat lanjut dengan menggunakan RAG dan ReAct prompt (Panduan Preskriptif).AWS

Pengambilan Generasi Augmented

Retrieval Augmented Generation (RAG) adalah teknologi AI generatif di mana LLM mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Sistem RAG dapat mengambil informasi ontologi medis (seperti klasifikasi penyakit internasional, file obat nasional, dan judul subjek medis) dari sumber pengetahuan. Ini memberikan konteks tambahan untuk LLM untuk mendukung tugas NLP medis.

Seperti yang dibahas di Menggabungkan Amazon Comprehend Medical dengan model bahasa besar bagian ini, Anda dapat menggunakan pendekatan RAG untuk mengambil konteks dari Amazon Comprehend Medical. Sumber pengetahuan umum lainnya termasuk data domain medis yang disimpan dalam layanan database, seperti Amazon OpenSearch Service, Amazon Kendra, atau Amazon Aurora. Mengekstrak informasi dari sumber pengetahuan ini dapat mempengaruhi kinerja pengambilan, terutama dengan kueri semantik yang menggunakan database vektor.

Opsi lain untuk menyimpan dan mengambil pengetahuan khusus domain adalah dengan menggunakan Amazon Q Business dalam alur kerja RAG Anda. Amazon Q Business dapat mengindeks repositori dokumen internal atau situs web yang menghadap publik (seperti CMS.gov untuk data ICD-10). Amazon Q Business kemudian dapat mengekstrak informasi yang relevan dari sumber-sumber ini sebelum meneruskan kueri Anda ke LLM.

Ada beberapa cara untuk membangun alur kerja RAG kustom. Misalnya, ada banyak cara untuk mengambil data dari sumber pengetahuan. Untuk mempermudah, kami merekomendasikan pendekatan pengambilan umum menggunakan database vektor, seperti Amazon OpenSearch Service, untuk menyimpan pengetahuan sebagai embeddings. Ini mengharuskan Anda menggunakan model embedding, seperti transformator kalimat, untuk menghasilkan embeddings untuk kueri dan untuk pengetahuan yang disimpan dalam database vektor.

Untuk informasi selengkapnya tentang pendekatan RAG yang dikelola sepenuhnya dan kustom, lihat opsi dan arsitektur Retrieval Augmented Generation di. AWS

Penyetelan halus

Menyesuaikan model yang ada melibatkan pengambilan LLM, seperti model Amazon Titan, Mistral, atau Llama, dan kemudian mengadaptasi model ke data kustom Anda. Ada berbagai teknik untuk fine-tuning, yang sebagian besar melibatkan memodifikasi hanya beberapa parameter alih-alih memodifikasi semua parameter dalam model. Ini disebut parameter-efficient fine-tuning (PEFT). Untuk informasi lebih lanjut, lihat Hugging Face GitHub PEFT di.

Berikut ini adalah dua kasus penggunaan umum ketika Anda mungkin memilih untuk menyempurnakan LLM untuk tugas NLP medis:

- Tugas generatif - Model berbasis decoder melakukan tugas AI generatif. AI/ML praktisi menggunakan data kebenaran dasar untuk menyempurnakan LLM yang ada. Misalnya, Anda dapat melatih LLM dengan menggunakan MedQuAD, kumpulan data penjawab pertanyaan medis publik. Saat Anda memanggil kueri ke LLM yang disetel dengan baik, Anda tidak memerlukan pendekatan RAG untuk memberikan konteks tambahan ke LLM.
- Embeddings — Model berbasis encoder menghasilkan embeddings dengan mengubah teks menjadi vektor numerik. Model berbasis encoder ini biasanya disebut model embedding. Model transformator kalimat adalah jenis spesifik dari model embedding yang dioptimalkan untuk kalimat. Tujuannya adalah untuk menghasilkan embeddings dari teks input. Embeddings kemudian digunakan untuk analisis semantik atau dalam tugas pengambilan. Untuk menyempurnakan model penyematan, Anda harus memiliki kumpulan pengetahuan medis, seperti dokumen, yang dapat Anda gunakan sebagai data pelatihan. Ini dicapai dengan pasangan teks berdasarkan kesamaan atau sentimen untuk menyempurnakan model transformator kalimat. Untuk informasi lebih lanjut, lihat Melatih dan Menyematkan Model Penyematan dengan Transformer Kalimat v3 di Hugging Face.

Anda dapat menggunakan Amazon SageMaker Ground Truth untuk membuat kumpulan data pelatihan berlabel berkualitas tinggi. Anda dapat menggunakan output dataset berlabel dari Ground Truth untuk melatih model Anda sendiri. Anda juga dapat menggunakan output sebagai kumpulan data pelatihan untuk model SageMaker AI Amazon. Untuk informasi selengkapnya tentang pengenalan entitas bernama, klasifikasi teks label tunggal, dan klasifikasi teks multi-label, lihat Pelabelan teks dengan Ground Truth dalam dokumentasi Amazon SageMaker AI.

Untuk informasi lebih lanjut tentang fine-tuning, lihat Menyesuaikan model bahasa besar dalam perawatan kesehatan di panduan ini.

Memilih LLM

Amazon Bedrock adalah titik awal yang direkomendasikan untuk mengevaluasi kinerja tinggi LLMs. Untuk informasi selengkapnya, lihat Model foundation yang didukung di Amazon Bedrock. Anda dapat menggunakan pekerjaan evaluasi model di Amazon Bedrock untuk membandingkan output dari beberapa output dan kemudian memilih model yang paling cocok untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Memilih model berkinerja terbaik menggunakan evaluasi Amazon Bedrock dalam dokumentasi Amazon Bedrock.

Beberapa LLMs memiliki pelatihan terbatas pada data domain medis. Jika kasus penggunaan Anda memerlukan fine-tuning LLM atau LLM yang tidak didukung Amazon Bedrock, pertimbangkan untuk menggunakan Amazon AI. SageMaker Di SageMaker AI, Anda dapat menggunakan LLM yang disetel dengan baik atau memilih LLM khusus yang telah dilatih tentang data domain medis.

Tabel berikut mencantumkan populer LLMs yang telah dilatih tentang data domain medis.

LLM	Tugas	Pengetahuan	Arsitektur
BioBert	Pengambilan informasi, klasifikasi teks, dan pengenalan entitas bernama	Abstrak dari PubMed, artikel teks lengkap dari PubMedCentral, dan pengetahuan domain umum	Encoder
Clinicalbert	Pengambilan informasi, klasifikasi teks, dan pengenalan entitas bernama	Dataset multi-pusat yang besar bersama dengan lebih dari 3.000.000 catatan pasien dari sistem catatan kesehatan elektronik (EHR)	Encoder
ClinicalGPT	Meringkas, menjawab pertanyaan, dan pembuatan teks	Kumpulan data medis yang luas dan beragam, termasuk catatan medis, pengetahu an khusus domain, dan konsultasi dialog multi-putaran	Dekoder
GatorTron-OG	Meringkas, menjawab pertanyaan, pembuatan teks, dan pengambilan informasi	Catatan klinis dan literatur biomedis	Encoder

Med-bert	Pengambilan informasi, klasifikasi teks, dan pengenalan entitas bernama	Kumpulan data besar teks medis, catatan klinis, makalah penelitian, dan dokumen terkait perawatan kesehatan	Encoder
Med-telapak tangan	Menjawab pertanyaan untuk tujuan medis	Kumpulan data teks medis dan biomedis	Dekoder
MedalPaca	Tugas menjawab pertanyaan dan dialog medis	Berbagai teks medis, yang mencakup sumber daya seperti kartu flash medis, wiki, dan kumpulan data dialog	Dekoder
BioMedbert	Pengambilan informasi, klasifikasi teks, dan pengenalan entitas bernama	Eksklusif abstrak dari PubMed dan artikel teks lengkap dari PubMedCentral	Encoder
BioMedLM	Meringkas, menjawab pertanyaan, dan pembuatan teks	Literatur biomedis dari sumber pengetahuan PubMed	Dekoder

Berikut ini adalah praktik terbaik untuk menggunakan medis LLMs terlatih:

- Pahami data pelatihan dan relevansinya dengan tugas NLP medis Anda.
- Identifikasi arsitektur LLM dan tujuannya. Encoder sesuai untuk penyematan dan tugas NLP. Decoder adalah untuk tugas pembuatan.
- Mengevaluasi infrastruktur, kinerja, dan persyaratan biaya untuk menjadi tuan rumah LLM medis yang telah dilatih sebelumnya.
- Jika fine-tuning diperlukan, pastikan kebenaran atau pengetahuan dasar yang akurat untuk data pelatihan. Pastikan Anda menutupi atau menyunting informasi identitas pribadi (PII) atau informasi kesehatan yang dilindungi (PHI).

Tugas NLP medis dunia nyata mungkin berbeda dari yang telah dilatih sebelumnya LLMs dalam hal pengetahuan atau kasus penggunaan yang dimaksudkan. Jika LLM khusus domain tidak memenuhi tolok ukur evaluasi Anda, Anda dapat menyempurnakan LLM dengan kumpulan data Anda sendiri atau Anda dapat melatih model fondasi baru. Melatih model pondasi baru adalah usaha yang ambisius, dan seringkali mahal. Untuk sebagian besar kasus penggunaan, kami merekomendasikan untuk menyempurnakan model yang ada.

Saat Anda menggunakan atau menyempurnakan LLM medis yang telah dilatih sebelumnya, penting untuk mengatasi infrastruktur, keamanan, dan pagar pembatas.

Infrastruktur

Dibandingkan dengan menggunakan Amazon Bedrock untuk inferensi sesuai permintaan atau batch, hosting LLM medis terlatih (umumnya dari Hugging Face) membutuhkan sumber daya yang signifikan. Untuk meng-host LLM medis terlatih sebelumnya, biasanya menggunakan image Amazon SageMaker AI yang berjalan pada instans Amazon Elastic Compute Cloud (Amazon EC2) dengan satu atau GPUs lebih, seperti instans ml.g5 untuk komputasi yang dipercepat atau instans ml.inf2 untuk. AWS Inferentia Ini karena LLMs mengkonsumsi sejumlah besar memori dan ruang disk.

Keamanan dan pagar pembatas

Bergantung pada persyaratan kepatuhan bisnis Anda, pertimbangkan untuk menggunakan Amazon Comprehend dan Amazon Comprehend Medical untuk menutupi atau menyunting informasi identitas pribadi (PII) dan informasi kesehatan yang dilindungi (PHI) dari data pelatihan. Ini membantu mencegah LLM menggunakan data rahasia saat menghasilkan respons.

Kami menyarankan Anda mempertimbangkan dan mengevaluasi bias, keadilan, dan halusinasi dalam aplikasi AI generatif Anda. Apakah Anda menggunakan LLM yang sudah ada sebelumnya atau fine-tuning, terapkan pagar pembatas untuk mencegah respons berbahaya. Guardrails adalah perlindungan yang Anda sesuaikan dengan persyaratan aplikasi AI generatif dan kebijakan AI yang bertanggung jawab. Misalnya, Anda dapat menggunakan Amazon Bedrock Guardrails.

Menyesuaikan model bahasa besar dalam perawatan kesehatan

Pendekatan fine-tuning yang dijelaskan dalam bagian ini mendukung kepatuhan terhadap pedoman etika dan peraturan dan mempromosikan penggunaan sistem AI yang bertanggung jawab dalam perawatan kesehatan. Ini dirancang untuk menghasilkan wawasan yang akurat dan pribadi. AI

generatif merevolusi pemberian layanan kesehatan, tetapi off-the-shelf model sering gagal dalam lingkungan klinis di mana akurasi sangat penting dan kepatuhan tidak dapat dinegosiasikan. Model fondasi fine-tuning dengan data spesifik domain menjembatani kesenjangan ini. Ini membantu Anda membuat sistem AI yang berbicara bahasa kedokteran sambil mematuhi standar peraturan yang ketat. Namun, jalan menuju fine-tuning yang sukses membutuhkan navigasi yang cermat terhadap tantangan unik perawatan kesehatan: melindungi data sensitif, membenarkan investasi AI dengan hasil yang terukur, dan mempertahankan relevansi klinis dalam lanskap medis yang berkembang cepat.

Ketika pendekatan yang lebih ringan mencapai batasnya, fine-tuning menjadi investasi strategis. Harapannya adalah bahwa keuntungan dalam akurasi, latensi, atau efisiensi operasional akan mengimbangi biaya komputasi dan rekayasa yang signifikan yang diperlukan. Penting untuk diingat bahwa laju kemajuan dalam model pondasi cepat, sehingga keunggulan model yang disetel dengan baik mungkin hanya bertahan hingga rilis model utama berikutnya.

Bagian ini menambatkan diskusi dalam dua kasus penggunaan berdampak tinggi berikut dari pelanggan AWS layanan kesehatan:

- Sistem pendukung keputusan klinis — Meningkatkan akurasi diagnostik melalui model yang memahami riwayat pasien yang kompleks dan pedoman yang berkembang. Penyetelan halus dapat membantu model sangat memahami riwayat pasien yang kompleks dan mengintegrasikan pedoman khusus Ini berpotensi mengurangi kesalahan prediksi model. Namun, Anda perlu mempertimbangkan keuntungan ini terhadap biaya pelatihan pada kumpulan data yang besar dan sensitif serta infrastruktur yang diperlukan untuk aplikasi klinis berisiko tinggi. Akankah peningkatan akurasi dan kesadaran konteks membenarkan investasi, terutama ketika model-model baru sering dirilis?
- Analisis dokumen medis — Mengotomatiskan pemrosesan catatan klinis, laporan pencitraan, dan dokumen asuransi sambil mempertahankan kepatuhan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA). Di sini, fine-tuning memungkinkan model untuk menangani format unik, singkatan khusus, dan persyaratan peraturan secara lebih efektif. Imbalannya sering terlihat dalam pengurangan waktu peninjauan manual dan peningkatan kepatuhan. Namun, penting untuk menilai apakah perbaikan ini cukup besar untuk menjamin sumber daya fine-tuning. Tentukan apakah rekayasa yang cepat dan orkestrasi alur kerja dapat memenuhi kebutuhan Anda.

Skenario dunia nyata ini menggambarkan perjalanan fine-tuning, dari eksperimen awal hingga penerapan model, sambil menangani persyaratan unik perawatan kesehatan di setiap tahap.

Memperkirakan biaya dan laba atas investasi

Berikut ini adalah faktor biaya yang harus Anda pertimbangkan saat menyempurnakan LLM:

- Ukuran model - Model yang lebih besar harganya lebih mahal untuk disempurnakan
- Dataset size — Biaya komputasi dan waktu meningkat dengan ukuran dataset untuk fine-tuning
- Strategi fine-tuning — Metode hemat parameter dapat mengurangi biaya dibandingkan dengan pembaruan parameter lengkap

Saat menghitung laba atas investasi (ROI), pertimbangkan peningkatan metrik yang Anda pilih (seperti akurasi) dikalikan dengan volume permintaan (seberapa sering model akan digunakan) dan durasi yang diharapkan sebelum model dilampaui oleh versi yang lebih baru.

Juga, pertimbangkan umur LLM dasar Anda. Model dasar baru muncul setiap 6-12 bulan. Jika detektor penyakit langka Anda membutuhkan waktu 8 bulan untuk menyempurnakan dan memvalidasi, Anda mungkin hanya mendapatkan 4 bulan kinerja superior sebelum model yang lebih baru menutup celah.

Dengan menghitung biaya, ROI, dan potensi umur untuk kasus penggunaan Anda, Anda dapat membuat keputusan berdasarkan data. Misalnya, jika menyempurnakan model dukungan keputusan klinis Anda mengarah pada pengurangan kesalahan diagnostik yang terukur di ribuan kasus per tahun, investasi mungkin akan cepat terbayar. Sebaliknya, jika rekayasa cepat saja membawa alur kerja analisis dokumen Anda mendekati akurasi target Anda, mungkin bijaksana untuk menunda fine-tuning sampai model generasi berikutnya tiba.

Fine-tuning tidak. one-size-fits-all Jika Anda memutuskan untuk menyempurnakan, pendekatan yang tepat tergantung pada kasus penggunaan, data, dan sumber daya Anda.

Memilih strategi fine-tuning

Setelah Anda menentukan bahwa fine-tuning adalah pendekatan yang tepat untuk kasus penggunaan perawatan kesehatan Anda, langkah selanjutnya adalah memilih strategi fine-tuning yang paling tepat. Ada beberapa pendekatan yang tersedia. Masing-masing memiliki keunggulan dan trade-off yang berbeda untuk aplikasi perawatan kesehatan. Pilihan antara metode ini tergantung pada tujuan spesifik Anda, data yang tersedia, dan kendala sumber daya.

Tujuan pelatihan

Domain-adaptive pre-training (DAPT) adalah metode tanpa pengawasan yang melibatkan pra-pelatihan model pada sejumlah besar teks khusus domain dan tidak berlabel (seperti jutaan dokumen medis). Pendekatan ini sangat cocok untuk meningkatkan kemampuan model untuk memahami singkatan khusus medis dan terminologi yang digunakan oleh ahli radiologi, ahli saraf, dan penyedia khusus lainnya. Namun, DAPT membutuhkan sejumlah besar data dan tidak membahas output tugas tertentu.

Supervised fine-tuning (SFT) mengajarkan model untuk mengikuti instruksi eksplisit dengan menggunakan contoh input-output terstruktur. Pendekatan ini unggul untuk alur kerja analisis dokumen medis, seperti ringkasan dokumen atau pengkodean klinis. Penyetelan instruksi adalah bentuk umum SFT di mana model dilatih pada contoh yang mencakup instruksi eksplisit yang dipasangkan dengan output yang diinginkan. Ini meningkatkan kemampuan model untuk memahami dan mengikuti beragam petunjuk pengguna. Teknik ini sangat berharga dalam pengaturan perawatan kesehatan karena melatih model dengan contoh klinis tertentu. Kelemahan utama adalah bahwa hal itu membutuhkan contoh yang diberi label dengan hati-hati. Selain itu, model yang disetel dengan baik mungkin berjuang dengan kasus tepi di mana tidak ada contoh. Untuk petunjuk tentang fine-tuning dengan Amazon SageMaker Jumpstart, lihat Instruksi fine-tuning untuk FLAN T5 XL dengan Amazon Jumpstart (posting blog). SageMaker AWS

Pembelajaran penguatan dari umpan balik manusia (RLHF) mengoptimalkan perilaku model berdasarkan umpan balik dan preferensi ahli. Gunakan model hadiah yang dilatih tentang preferensi dan metode manusia, seperti optimasi kebijakan proksimal (PPO) atau optimasi preferensi langsung (DPO), untuk mengoptimalkan model sambil mencegah pembaruan yang merusak. RLHF sangat ideal untuk menyelaraskan output dengan pedoman klinis dan memastikan bahwa rekomendasi tetap dalam protokol yang disetujui. Pendekatan ini membutuhkan waktu dokter yang signifikan untuk umpan balik dan melibatkan jalur pelatihan yang kompleks. Namun, RLHF sangat berharga dalam perawatan kesehatan karena membantu para ahli medis membentuk bagaimana sistem AI berkomunikasi dan membuat rekomendasi. Misalnya, dokter dapat memberikan umpan balik untuk memastikan bahwa model mempertahankan cara samping tempat tidur yang tepat, tahu kapan harus mengekspresikan ketidakpastian, dan tetap dalam pedoman klinis. Teknik seperti PPO secara iteratif mengoptimalkan perilaku model berdasarkan umpan balik ahli sambil membatasi pembaruan parameter untuk melestarikan pengetahuan medis inti. Hal ini memungkinkan model untuk menyampaikan diagnosis kompleks dalam bahasa yang ramah pasien sambil tetap menandai kondisi serius untuk perhatian medis segera. Ini sangat penting untuk perawatan kesehatan di mana akurasi dan gaya komunikasi penting. Untuk informasi lebih lanjut tentang RLHF, lihat Sempurnakan

model bahasa besar dengan pembelajaran penguatan dari umpan balik manusia atau AI (posting blog).AWS

Metode implementasi

Pembaruan parameter lengkap melibatkan pembaruan semua parameter model selama pelatihan. Pendekatan ini bekerja paling baik untuk sistem pendukung keputusan klinis yang memerlukan integrasi mendalam dari riwayat pasien, hasil laboratorium, dan pedoman yang berkembang. Kekurangannya termasuk biaya komputasi yang tinggi dan risiko overfitting jika kumpulan data Anda tidak besar dan beragam.

Metode fine-tuning (PEFT) yang efisien parameter hanya memperbarui sebagian parameter untuk mencegah overfitting atau hilangnya kemampuan bahasa yang dahsyat. Jenis termasuk adaptasi peringkat rendah (LoRa), adaptor, dan penyetelan awalan. Metode PEFT menawarkan biaya komputasi yang lebih rendah, pelatihan yang lebih cepat, dan bagus untuk eksperimen seperti mengadaptasi model pendukung keputusan klinis dengan protokol atau terminologi rumah sakit baru. Keterbatasan utama berpotensi mengurangi kinerja dibandingkan dengan pembaruan parameter penuh.

Untuk informasi selengkapnya tentang metode fine-tuning, lihat Metode fine-tuning lanjutan di SageMaker Amazon AI (posting blog).AWS

Membangun kumpulan data fine-tuning

Kualitas dan keragaman kumpulan data fine-tuning sangat penting untuk memodelkan kinerja, keamanan, dan pencegahan bias. Berikut ini adalah tiga area penting yang perlu dipertimbangkan saat membangun kumpulan data ini:

- Volume berdasarkan pendekatan fine-tuning
- Anotasi data dari pakar domain
- Keragaman kumpulan data

Seperti yang ditunjukkan pada tabel berikut, persyaratan ukuran kumpulan data untuk fine-tuning bervariasi berdasarkan jenis fine-tuning yang dilakukan.

Strategi fine-tuning	Ukuran dataset
Domain diadaptasi pra-pelatihan	100.000+ teks domain

Penyetelan halus yang diawasi	10.000+ pasangan berlabel
Pembelajaran penguatan dari umpan balik manusia	1.000+ pasangan preferensi ahli

Anda dapat menggunakan AWS Glue, Amazon EMR, dan Amazon SageMaker Data Wrangler untuk mengotomatiskan proses ekstraksi dan transformasi data guna mengkurasi kumpulan data yang Anda miliki. Jika Anda tidak dapat mengkurasi kumpulan data yang cukup besar, Anda dapat menemukan dan mengunduh kumpulan data langsung ke perangkat Anda. Akun AWS AWS Data Exchange Konsultasikan dengan penasihat hukum Anda sebelum menggunakan kumpulan data pihak ketiga.

Ahli anotator dengan pengetahuan domain, seperti dokter medis, ahli biologi, dan ahli kimia, harus menjadi bagian dari proses kurasi data untuk memasukkan nuansa data medis dan biologis ke dalam output model. Amazon SageMaker Ground Truth menyediakan antarmuka pengguna kode rendah bagi para ahli untuk membuat anotasi kumpulan data.

Kumpulan data yang mewakili populasi manusia sangat penting untuk perawatan kesehatan dan ilmu hayati menyempurnakan kasus penggunaan untuk mencegah bias dan mencerminkan hasil dunia nyata. AWS Glue sesi interaktif atau instans SageMaker notebook Amazon menawarkan cara yang ampuh untuk menjelajahi kumpulan data secara berulang dan menyempurnakan transformasi dengan menggunakan notebook yang kompatibel dengan Jupiter. Sesi interaktif memungkinkan Anda untuk bekerja dengan pilihan lingkungan pengembangan terintegrasi populer (IDEs) di lingkungan lokal Anda. Atau, Anda dapat bekerja dengan AWS Glue atau notebook Amazon SageMaker Studio melalui. Konsol Manajemen AWS

Menyetel model

AWS menyediakan layanan seperti Amazon SageMaker AI dan Amazon Bedrock yang sangat penting untuk fine-tuning yang sukses.

SageMaker AI adalah layanan pembelajaran mesin yang dikelola sepenuhnya yang membantu pengembang dan ilmuwan data untuk membangun, melatih, dan menerapkan model ML dengan cepat. Tiga fitur berguna SageMaker AI untuk fine-tuning meliputi:

- SageMakerPelatihan - Fitur ML yang dikelola sepenuhnya yang membantu Anda melatih berbagai model secara efisien dalam skala

- SageMaker JumpStart— Kemampuan yang dibangun di atas pekerjaan SageMaker Pelatihan untuk menyediakan model yang telah dilatih sebelumnya, algoritme bawaan, dan templat solusi untuk tugas ML
- SageMaker HyperPod— Solusi infrastruktur yang dibangun khusus untuk pelatihan terdistribusi model pondasi dan LLMs

Amazon Bedrock adalah layanan terkelola penuh yang menyediakan akses ke model foundation berkinerja tinggi melalui API, dengan fitur keamanan, privasi, dan skalabilitas bawaan. Layanan ini menyediakan kemampuan untuk menyempurnakan beberapa model dasar yang tersedia. Untuk informasi selengkapnya, lihat Model dan Wilayah yang Didukung untuk menyempurnakan dan melanjutkan pra-pelatihan dalam dokumentasi Amazon Bedrock.

Saat mendekati proses fine-tuning dengan salah satu layanan, pertimbangkan model dasar, strategi fine-tuning, dan infrastruktur.

Pilihan model dasar

Model sumber tertutup, seperti Anthropic Claude, Meta Llama, dan Amazon Nova, memberikan kinerja yang out-of-the-box kuat dengan kepatuhan terkelola tetapi membatasi fleksibilitas fine-tuning ke opsi yang didukung penyedia seperti dikelola seperti Amazon Bedrock. APIs Ini membatasi penyesuaian, terutama untuk kasus penggunaan perawatan kesehatan yang diatur. Sebaliknya, model sumber terbuka, seperti Meta Llama, memberikan kontrol dan fleksibilitas penuh di seluruh layanan Amazon SageMaker AI, menjadikannya ideal saat Anda perlu menyesuaikan, mengaudit, atau menyesuaikan model secara mendalam dengan data spesifik atau persyaratan alur kerja Anda.

Strategi fine-tuning

Penyetelan instruksi sederhana dapat ditangani oleh kustomisasi model Amazon Bedrock atau Amazon. SageMaker JumpStart Pendekatan PEFT yang kompleks, seperti LoRa atau adaptor, memerlukan pekerjaan SageMaker Pelatihan atau fitur fine-tuning khusus di Amazon Bedrock. Pelatihan terdistribusi untuk model yang sangat besar didukung oleh SageMaker HyperPod.

Skala dan kontrol infrastruktur

Layanan yang dikelola sepenuhnya, seperti Amazon Bedrock, meminimalkan manajemen infrastruktur dan ideal untuk organisasi yang memprioritaskan kemudahan penggunaan dan kepatuhan. Opsi semi-terkelola, seperti SageMaker JumpStart, menawarkan beberapa fleksibilitas dengan kompleksitas yang lebih sedikit. Opsi ini cocok untuk pembuatan prototipe cepat atau saat

menggunakan alur kerja pra-bangun. Kontrol dan kustomisasi penuh datang dengan pekerjaan SageMaker Pelatihan dan HyperPod, meskipun ini membutuhkan lebih banyak keahlian dan yang terbaik ketika Anda perlu meningkatkan untuk kumpulan data besar atau memerlukan saluran pipa khusus.

Memantau model yang disetel dengan baik

Dalam perawatan kesehatan dan ilmu kehidupan, pemantauan LLM fine-tuning memerlukan pelacakan beberapa indikator kinerja utama. Akurasi memberikan pengukuran dasar, tetapi ini harus diimbangi dengan presisi dan penarikan, terutama dalam aplikasi di mana kesalahan klasifikasi membawa konsekuensi yang signifikan. Skor F1 membantu mengatasi masalah ketidakseimbangan kelas yang umum terjadi pada kumpulan data medis. Untuk informasi selengkapnya, lihat Mengevaluasi aplikasi LLMs perawatan kesehatan dan ilmu hayati dalam panduan ini.

Metrik kalibrasi membantu Anda memastikan bahwa tingkat kepercayaan model sesuai dengan probabilitas dunia nyata. Metrik keadilan dapat membantu Anda mendeteksi potensi bias di berbagai demografi pasien.

MLflowadalah solusi open source yang dapat membantu Anda melacak eksperimen fine-tuning. MLflow didukung secara native dalam Amazon SageMaker AI, yang membantu Anda membandingkan metrik secara visual dari latihan. Untuk pekerjaan fine-tuning di Amazon Bedrock, metrik dialirkan ke Amazon CloudWatch sehingga Anda dapat memvisualisasikan metrik di konsol. CloudWatch

Memilih pendekatan NLP untuk perawatan kesehatan dan ilmu kehidupan

Pendekatan AI dan NLP generatif untuk perawatan kesehatan dan ilmu kehidupanBagian ini menjelaskan pendekatan berikut untuk menangani tugas pemrosesan bahasa alami (NLP) untuk aplikasi perawatan kesehatan dan ilmu hayati:

- Menggunakan Amazon Comprehend Medical
- Menggabungkan Amazon Comprehend Medical dengan LLM dalam alur kerja Retrieval Augment Generation (RAG)
- Menggunakan LLM yang disetel dengan baik
- Menggunakan alur kerja RAG

Dengan mengevaluasi batasan yang diketahui LLMs untuk tugas domain medis dan kasus penggunaan Anda, Anda dapat memilih pendekatan mana yang paling sesuai untuk tugas Anda. Pohon keputusan berikut dapat membantu Anda memilih pendekatan LLM untuk tugas NLP medis Anda:

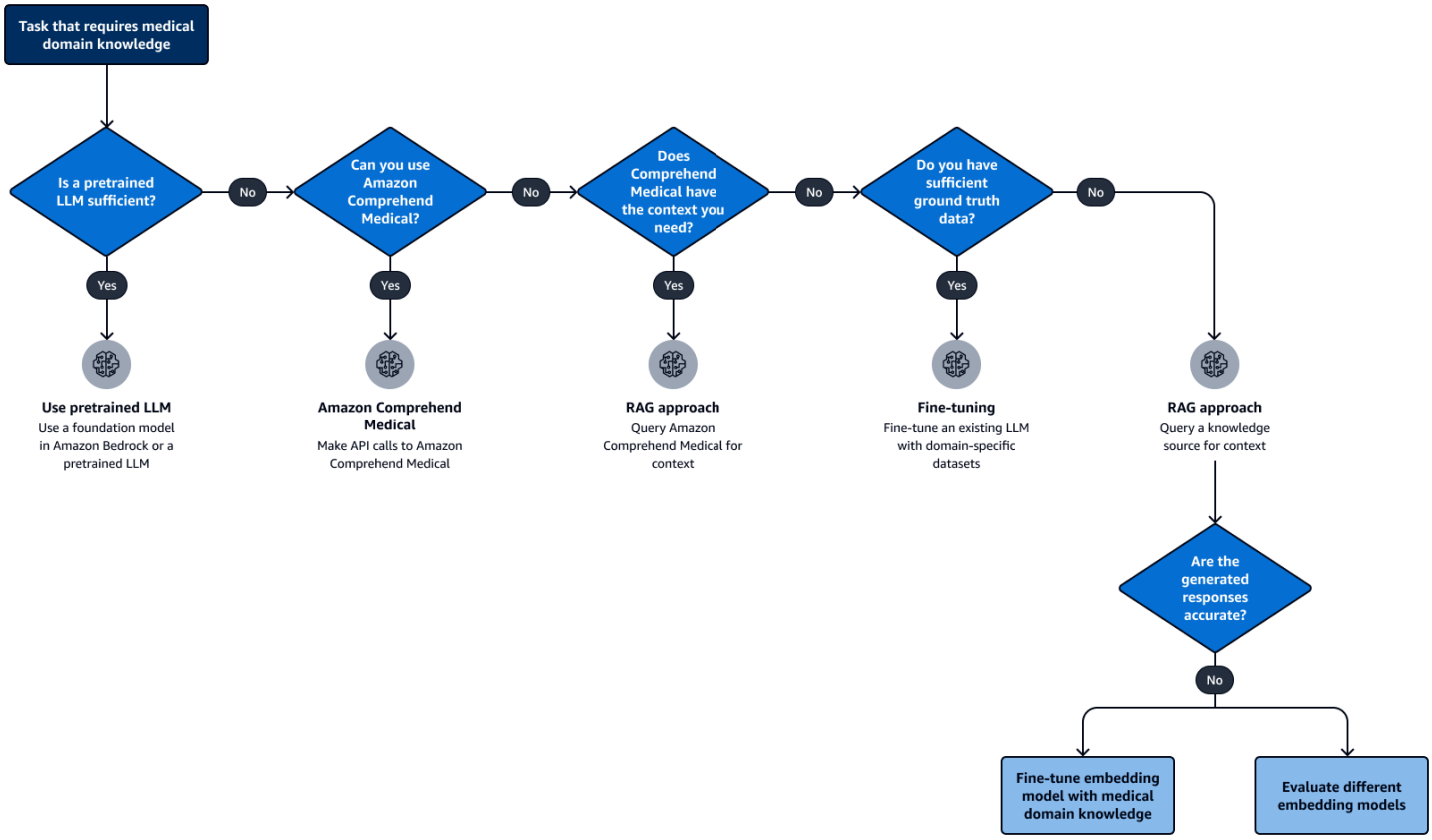


Diagram menunjukkan alur kerja berikut:

1. Untuk kasus penggunaan perawatan kesehatan dan ilmu hayati, identifikasi apakah tugas NLP memerlukan pengetahuan domain tertentu. Sesuai kebutuhan, berkoordinasi dengan ahli materi pelajaran (SMEs).
2. Jika Anda dapat menggunakan LLM umum atau model yang telah dilatih pada dataset medis, maka gunakan model fondasi yang tersedia di Amazon Bedrock atau LLM yang telah dilatih sebelumnya. Untuk informasi selengkapnya, lihat Memilih LLM dalam panduan ini.
3. Jika kemampuan deteksi entitas dan penautan ontologi Amazon Comprehend Medical menangani kasus penggunaan Anda, maka gunakan Amazon Comprehend Medical. APIs Untuk informasi selengkapnya, lihat Menggunakan Amazon Comprehend Medical dalam panduan ini.
4. Terkadang, Amazon Comprehend Medical memiliki konteks yang diperlukan tetapi tidak mendukung kasus penggunaan Anda. Misalnya, Anda mungkin memerlukan definisi entitas yang berbeda, menerima hasil yang sangat banyak, memerlukan entitas khusus, atau memerlukan tugas NLP khusus. Jika ini masalahnya, gunakan pendekatan RAG untuk menanyakan Amazon Comprehend Medical untuk konteks. Untuk informasi selengkapnya, lihat Menggabungkan Amazon Comprehend Medical dengan model bahasa besar dalam panduan ini.

5. Jika Anda memiliki jumlah data kebenaran dasar yang cukup, sempurnakan LLM yang ada. Untuk informasi selengkapnya, lihat Pendekatan kustomisasi dalam panduan ini.
6. Jika pendekatan lain tidak memenuhi tujuan tugas NLP medis Anda, terapkan solusi RAG. Untuk informasi selengkapnya, lihat Pendekatan kustomisasi dalam panduan ini.
7. Setelah menerapkan solusi RAG, evaluasi apakah respons yang dihasilkan akurat. Untuk informasi selengkapnya, lihat Mengevaluasi aplikasi LLMs perawatan kesehatan dan ilmu hayati dalam panduan ini. Adalah umum untuk memulai dengan model Amazon Titan Text Embeddings atau model transformator kalimat umum, seperti All-minilm-L6-v2. Namun, karena kurangnya konteks domain, model ini mungkin tidak menangkap terminologi medis teks. Jika perlu, pertimbangkan penyesuaian berikut:
   a. Evaluasi model penyematan lainnya
   b. Sempurnakan model penyematan dengan kumpulan data khusus domain

Pertimbangan kematangan bisnis

Kematangan bisnis sangat penting ketika mengadaptasi solusi LLM untuk aplikasi perawatan kesehatan dan ilmu hayati. Organisasi-organisasi ini menghadapi berbagai tingkat kompleksitas saat menerapkan LLMs, tergantung pada kriteria penerimaan mereka. Seringkali, organisasi yang kekurangan AI/ML sumber daya berinvestasi dalam dukungan kontraktor untuk membangun solusi LLM. Dalam situasi ini, penting untuk memahami trade-off berikut:

- Kinerja tinggi untuk biaya dan pemeliharaan tinggi — Anda mungkin memerlukan solusi kompleks yang melibatkan penyesuaian atau penyesuaian khusus LLMs untuk memenuhi standar kinerja yang ketat. Namun, ini datang dengan biaya dan persyaratan pemeliharaan yang lebih tinggi. Anda mungkin perlu menyewa sumber daya khusus atau bermitra dengan kontraktor untuk mempertahankan solusi canggih ini. Ini berpotensi memperlambat perkembangan.
- Kinerja yang baik untuk biaya rendah dan pemeliharaan — Atau, Anda mungkin menemukan bahwa layanan seperti Amazon Bedrock atau Amazon Comprehend Medical memberikan kinerja yang dapat diterima. Meskipun ini LLMs atau pendekatan mungkin memberikan hasil yang sempurna, solusi ini sering dapat memberikan hasil yang konsisten dan berkualitas tinggi. Solusi ini adalah biaya yang lebih rendah dan mengurangi beban pemeliharaan. Hal ini dapat mempercepat pembangunan.

Jika pendekatan yang lebih sederhana dan berbiaya lebih rendah secara konsisten memberikan hasil berkualitas tinggi yang memenuhi kriteria penerimaan Anda, pertimbangkan apakah peningkatan

kinerja sepadan dengan biaya, pemeliharaan, dan pengorbanan waktu. Namun, jika solusi yang lebih sederhana jauh dari kinerja target, dan jika organisasi Anda tidak memiliki kapasitas investasi untuk solusi kompleks dan persyaratan pemeliharaannya, pertimbangkan untuk menunda AI/ML pengembangan sampai lebih banyak sumber daya atau solusi alternatif tersedia.

Selain itu, untuk solusi NLP medis apa pun yang bergantung pada LLM, kami menyarankan Anda melakukan pemantauan dan evaluasi berkelanjutan. Menilai umpan balik dari pengguna dari waktu ke waktu, dan menerapkan penilaian berkala untuk memastikan bahwa solusi terus memenuhi tujuan bisnis Anda.

Mengevaluasi aplikasi LLMs perawatan kesehatan dan ilmu hayati

Bagian ini memberikan gambaran komprehensif tentang persyaratan dan pertimbangan untuk mengevaluasi model bahasa besar (LLMs) dalam kasus penggunaan perawatan kesehatan dan ilmu hayati.

Penting untuk menggunakan data kebenaran dasar dan umpan balik UKM untuk mengurangi bias dan memvalidasi keakuratan respons yang dihasilkan LLM. Bagian ini menjelaskan praktik terbaik untuk mengumpulkan dan mengkurasi data pelatihan dan pengujian. Ini juga membantu Anda menerapkan pagar pembatas dan mengukur bias dan keadilan data. Ini juga membahas tugas-tugas pemrosesan bahasa alami medis umum (NLP), seperti klasifikasi teks, pengenalan entitas bernama, dan pembuatan teks, dan metrik evaluasi terkait.

Ini juga menyajikan alur kerja untuk melakukan evaluasi LLM selama fase eksperimen pelatihan dan fase pasca-produksi. Pemantauan model dan operasi LLM adalah elemen penting dari proses evaluasi ini.

Data pelatihan dan pengujian untuk tugas NLP medis

Tugas NLP medis biasanya menggunakan corpora medis (seperti PubMed) atau informasi pasien (seperti catatan kunjungan pasien klinik) untuk mengklasifikasikan, meringkas, dan menghasilkan wawasan. Tenaga medis, dokter, administrator perawatan kesehatan, atau teknisi, bervariasi dalam keahlian dan sudut pandang. Karena subjektivitas antara tenaga medis ini, kumpulan data pelatihan dan pengujian yang lebih kecil menimbulkan risiko bias. Untuk mengurangi risiko ini, kami merekomendasikan praktik terbaik berikut:

- Saat menggunakan solusi LLM yang telah dilatih sebelumnya, pastikan Anda memiliki jumlah data pengujian yang memadai. Data tes harus sangat mirip dengan data medis yang sebenarnya. Tergantung pada tugasnya, ini dapat berkisar dari 20 hingga lebih dari 100 catatan.
- Saat menyempurnakan LLM, kumpulkan cukup banyak catatan berlabel (kebenaran dasar) dari berbagai domain medis yang SMEs ditargetkan. Titik awal umum setidaknya 100 catatan berkualitas tinggi. Namun, mengingat kompleksitas tugas dan kriteria penerimaan akurasi Anda, lebih banyak catatan mungkin diperlukan.
- Jika diperlukan untuk kasus penggunaan medis Anda, terapkan pagar pembatas dan ukur bias dan keadilan data. Misalnya, pastikan bahwa LLM mencegah kesalahan diagnosis karena profil ras

pasien. Untuk informasi lebih lanjut, lihat Keamanan dan pagar pembatas bagian dalam panduan ini.

Banyak perusahaan penelitian dan pengembangan AI, seperti Anthropic, telah menerapkan pagar pembatas dalam model fondasi mereka untuk menghindari toksisitas. Anda dapat menggunakan deteksi toksisitas untuk memeriksa petunjuk input dan respons keluaran dari. LLMs Untuk informasi selengkapnya, lihat Deteksi toksisitas di dokumentasi Amazon Comprehend dan lihat Guardrails di dokumentasi Amazon Bedrock.

Dalam tugas AI generatif apa pun, ada risiko halusinasi. Anda dapat mengurangi risiko ini dengan melakukan tugas NLP, seperti klasifikasi. Anda juga dapat menggunakan teknik yang lebih canggih, seperti metrik kesamaan teks. BertScoreadalah metrik kesamaan teks yang umum diadopsi. Untuk informasi lebih lanjut tentang teknik yang dapat Anda gunakan untuk mengurangi halusinasi, lihat Survei Komprehensif Teknik Mitigasi Halusinasi dalam Model Bahasa Besar.

Metrik untuk tugas NLP medis

Anda dapat membuat metrik yang dapat diukur setelah Anda membuat data kebenaran dasar dan label yang disediakan SME untuk pelatihan dan pengujian. Memeriksa kualitas melalui proses kualitatif, seperti stress testing dan meninjau hasil LLM, sangat membantu untuk pengembangan cepat. Namun, metrik bertindak sebagai tolok ukur kuantitatif yang mendukung operasi LLM masa depan dan bertindak sebagai tolok ukur kinerja untuk setiap rilis produksi.

Memahami tugas medis sangat penting. Metrik biasanya dipetakan ke salah satu tugas NLP umum berikut:

- Klasifikasi teks - LLM mengkategorikan teks ke dalam satu atau lebih kategori yang telah ditentukan, berdasarkan prompt input dan konteks yang disediakan. Contohnya adalah mengklasifikasikan kategori nyeri dengan menggunakan skala nyeri. Contoh metrik klasifikasi teks meliputi:
  - Akurasi
  - Presisi, juga dikenal sebagai presisi makro
  - Ingat, juga dikenal sebagai recall makro
  - Skor F1, juga dikenal sebagai skor F1 makro
  - Kehilangan Hamming

- Pengenalan entitas bernama (NER) - Juga dikenal sebagai ekstraksi teks, pengenalan entitas bernama adalah proses menemukan dan mengklasifikasikan entitas bernama yang disebutkan dalam teks tidak terstruktur ke dalam kategori yang telah ditentukan. Contohnya adalah mengekstraksi nama-nama obat dari catatan pasien. Contoh metrik NER meliputi:
  - Akurasi
  - presisi
  - Ingat
  - Skor F1
  - Kehilangan Hamming
- Generasi - LLM menghasilkan teks baru dengan memproses konteks prompt dan disediakan. Generasi mencakup tugas meringkas atau tugas menjawab pertanyaan. Contoh metrik generasi meliputi:
  - Pengganti Berorientasi Recall untuk Evaluasi Gisting (ROUGE)
  - Metrik untuk Evaluasi Terjemahan dengan Eksplisit ORdering (METEOR)
  - Pengganti evaluasi bilingual (BLEU) (untuk terjemahan)
  - Jarak string, juga dikenal sebagai kesamaan kosinus

FAQ tentang perawatan kesehatan dan kasus penggunaan ilmu hayati

Berikut ini adalah pertanyaan yang sering diajukan terkait dengan penggunaan Amazon Comprehend Medical atau untuk tugas-tugas NLP medis LLMs .

Bagaimana cara memilih antara Amazon Comprehend Medical dan LLM?

Jika tugas Anda adalah mendeteksi entitas medis dalam teks medis Anda, tinjau dokumentasi Amazon Comprehend Medical untuk memahami entitas medis mana yang dapat diekstraksi dan jika ada ontologi yang menangani kasus penggunaan Anda. Jika tidak, pertimbangkan untuk menggunakan LLM. Untuk informasi lebih lanjut, lihat Kasus penggunaan untuk Amazon Comprehend Medical dan Gunakan kasus untuk LLM di panduan ini.

Bagaimana saya bisa memberikan hasil Amazon Comprehend Medical ke LLM?

Anda dapat memasukkan hasil Amazon Comprehend Medical sebagai konteks dalam permintaan LLM Anda. Ini memberikan pengetahuan medis tambahan dan terminologi untuk LLM. Konteks yang disediakan dapat meningkatkan kinerja LLM pada tugas-tugas seperti pengakuan entitas, ringkasan, atau menjawab pertanyaan. Panduan ini memberikan beberapa contoh cara menyusun petunjuk dengan hasil Amazon Comprehend Medical. Untuk informasi selengkapnya, lihat Menggabungkan Amazon Comprehend Medical dengan model bahasa besar dalam panduan ini.

Apa sajakah praktik terbaik saat menggunakan Amazon Comprehend Medical dengan? LLMs

Sebaiknya gunakan skor kepercayaan Amazon Comprehend Medical untuk memfilter atau memprioritaskan entitas dalam permintaan Anda. Penting juga untuk mengevaluasi kinerjanya pada data spesifik Anda dan memvalidasi bahwa definisi entitas sesuai dengan kebutuhan Anda. Menggabungkan Amazon Comprehend Medical dengan sumber pengetahuan khusus domain dapat lebih meningkatkan kinerja LLM. Untuk informasi selengkapnya, lihat Praktik terbaik untuk menggunakan Amazon Comprehend Medical dalam alur kerja RAG dalam panduan ini.

Haruskah saya menggunakan LLM medis terlatih atau menyempurnakan LLM umum untuk kasus penggunaan perawatan kesehatan saya?

Keputusan tergantung pada persyaratan spesifik Anda dan ketersediaan data pelatihan berkualitas tinggi. Medis pra-terlatih LLMs dapat memberikan titik awal yang baik. Namun, Anda mungkin masih perlu menyempurnakannya dengan data spesifik domain Anda. Jika Anda memiliki data berlabel yang cukup, menyempurnakan LLM umum dapat menjadi pilihan yang layak. Untuk informasi lebih lanjut, lihat Memilih LLM dan Memilih pendekatan NLP untuk perawatan kesehatan dan ilmu kehidupan di panduan ini.

Bagaimana cara mengevaluasi kinerja LLMs untuk tugas-tugas NLP medis?

Kami merekomendasikan penggunaan metrik kuantitatif, seperti akurasi, presisi, penarikan, dan skor F1 untuk klasifikasi teks dan tugas pengenalan entitas bernama. Anda dapat menggunakan ROUGE dan METEOR untuk tugas pembuatan teks. Penting untuk memiliki data kebenaran dasar yang andal yang diberi label oleh pakar materi pelajaran dan untuk menerapkan proses untuk memantau kinerja model dari waktu ke waktu. Untuk informasi selengkapnya, lihat Mengevaluasi aplikasi LLMs perawatan kesehatan dan ilmu hayati dalam panduan ini.

Apa trade-off antara solusi LLM kompleksitas tinggi dan kompleksitas rendah?

Menyesuaikan LLM atau membangun LLM khusus adalah solusi yang sangat kompleks. Pendekatan ini dapat meningkatkan kinerja tetapi datang dengan biaya dan persyaratan pemeliharaan yang lebih tinggi. Solusi yang lebih sederhana, seperti menggunakan Amazon Comprehend Medical yang telah dilatih LLMs sebelumnya, dapat memberikan kinerja yang dapat diterima dengan biaya yang lebih rendah dan siklus pengembangan yang lebih cepat. Namun, pendekatan ini mungkin tidak memenuhi persyaratan akurasi yang ketat untuk beberapa kasus penggunaan. Untuk informasi selengkapnya, lihat Pertimbangan kematangan bisnis dalam panduan ini.

Langkah dan sumber daya selanjutnya

Panduan ini membantu Anda menggunakannya Layanan AWS untuk mengotomatiskan NLP medis dan tugas AI generatif untuk aplikasi dunia nyata di lingkungan produksi. Ini menjelaskan bagaimana Anda dapat menggunakan Amazon Comprehend Medical, didukung LLMs di Amazon Bedrock, LLMs medis terlatih, atau LLMs disesuaikan untuk mencapai tujuan bisnis perawatan kesehatan dan ilmu hayati Anda. Panduan ini menjelaskan keuntungan dan batasan untuk pendekatan berikut:

- Menggunakan Amazon Comprehend Medical secara mandiri
- Memberikan hasil Amazon Comprehend Medical ke LLM
- Menggunakan LLM umum yang telah dilatih sebelumnya atau LLM medis dalam pendekatan Retrieval Augmented Generation (RAG)
- Menyesuaikan LLM umum atau LLM medis

Gunakan pohon keputusan dan pertimbangan kematangan bisnis dalam panduan ini untuk memilih di antara pendekatan-pendekatan ini berdasarkan tingkat AI/ML kematangan organisasi Anda. Meskipun Amazon Comprehend Medical dan LLMs Amazon Bedrock menawarkan kemampuan yang kuat, mereka hanya berhasil jika Anda menerapkan dan mengevaluasinya dengan benar. Gunakan informasi evaluasi dan metrik yang dijelaskan dalam panduan ini untuk memvalidasi kinerja solusi Anda.

Untuk langkah selanjutnya, kami merekomendasikan agar manajer TI perawatan kesehatan, arsitek, dan pemimpin teknis bekerja dengan AI/ML praktisi untuk mengidentifikasi tugas medis NLP mereka. Gunakan panduan ini untuk memilih jalur pengembangan, lalu gunakan yang sesuai Layanan AWS dan fitur untuk berhasil menerapkan solusi otomatis AWS.

AWS sumber daya

- Dokumentasi Amazon Comprehend Medical:
  - Panduan Pengembang
  - Referensi API
- Dokumentasi Amazon Bedrock
  - Evaluasi model Amazon Bedrock
  - Penyetelan halus di Amazon Bedrock

- Sempurnakan model di Amazon AI SageMaker
- Amazon SageMaker Ground Truth
- Amazon Comprehend deteksi toksisitas
- AWS Mitra Kompetensi Kesehatan

Sumber daya lainnya

- Buka Papan Peringkat Medis-LLM
- Survei Model Bahasa Besar untuk Perawatan Kesehatan: dari Data, Teknologi, dan Aplikasi hingga Akuntabilitas dan Etika
- Model Bahasa Besar Adalah Coders Medis yang Buruk — Benchmarking Query Kode Medis
- Dari Pemula hingga Pakar: Memodelkan Pengetahuan Medis menjadi Umum LLMs

Kontributor

Mengotorisasi

- Joe King, Ilmuwan Data AWS Senior
- Ankith Ede, Arsitek Solusi AWS
- Clement Perrot, Ahli Strategi AI AWS Generatif Senior
- Jillian Forde, Arsitek Solusi Senior AWS
- Rajesh Sitaraman, Konsultan Pengiriman Senior AWS
- Ross Claytor, Ilmuwan Terapan Utama AWS
- Shivesh Ummat, Arsitek Solusi AWS

Meninjau

- Dilshad Raihan Akkam Veettil, Ilmuwan Data Senior AWS
- Joseph Cottingham, Arsitek Pembelajaran AWS Mendalam

Penulisan teknis

- Lilly AbouHarb, Penulis Teknis AWS Senior

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Bagian baru	Kami menambahkan model bahasa besar Fine-tuning di bagian perawatan kesehatan dan bagian teknik Prompt.	Desember 5, 2025
Publikasi awal	—	Desember 16, 2024

AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

Nomor

7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di. AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

Metode migrasi database di mana database sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi aktif-pasif.

migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi `SUM` dan`MAX`.

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk penemuan portofolio dan proses analisis dan membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat Apa itu Kecerdasan Buatan?

operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat panduan integrasi operasi.

enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat ABAC untuk AWS dokumentasi AWS Identity and Access Management (IAM).

sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan pemrosesan atau modifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat situs web AWS CAF dan whitepaper AWS CAF.

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool ()AWS SCT. Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau menyebabkan kerugian bagi individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat Data dalam grafik perilaku di dokumentasi Detektif.

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga endianness.

klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

deployment biru/hijau

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

botnet

Jaringan bot yang terinfeksi oleh malware dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat Tentang cabang (GitHub dokumentasi).

akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator Implementasikan prosedur break-glass dalam panduan Well-Architected AWS .

strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian Terorganisir di sekitar kemampuan bisnis dari Menjalankan layanan mikro kontainer di whitepaper. AWS

perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

rekayasa kekacauan

Sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan AWS Fault Injection Service (AWS FIS) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat posting CCo E di Blog Strategi AWS Cloud Perusahaan.

komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi edge computing.

model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat Membangun Model Operasi Cloud Anda.

tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCo E, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog The Journey Toward Cloud-First & the Stages of Adoption di blog Strategi Perusahaan. AWS Cloud Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat panduan kesiapan migrasi.

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat penyimpanan atau kelas yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Wilayah, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat Paket kesesuaian dalam dokumentasi. AWS Config

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

D

data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisan dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat Klasifikasi data.

penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat Membangun perimeter data pada AWS.

prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

DDL

Lihat bahasa definisi database.

ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

defense-in-depth

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, defense-in-depth pendekatan mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat Layanan yang berfungsi dengan AWS Organizations AWS Organizations dokumentasi.

deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

tabel dimensi

Dalam skema bintang, tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh bencana. Untuk informasi selengkapnya, lihat Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk mendeteksi penyimpangan dalam sumber daya sistem, atau Anda dapat menggunakannya AWS Control Tower untuk mendeteksi perubahan di landing zone yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan komputasi awan, komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat Apa itu Pertukaran Data Elektronik.

enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat layanan titik akhir di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, MES, dan manajemen proyek) untuk suatu perusahaan.

enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat Enkripsi amplop dalam dokumentasi AWS Key Management Service (AWS KMS).

lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.

- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat panduan implementasi program.

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

F

tabel fakta

Tabel tengah dalam skema bintang. Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat Batas Isolasi AWS Kesalahan.

cabang fitur

Lihat cabang.

fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal “2021-05-27 00:15:37" menjadi “2021", “Mei”, “Kamis”, dan “15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

beberapa tembakan mendorong

Menyediakan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga zero-shot prompting.

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui pengambilan data perubahan untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih

menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat model pondasi.

model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FMs mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat Apa itu Model Foundation.

G

AI generatif

Subset model AI yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat Apa itu AI Generatif.

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat Membatasi distribusi geografis konten Anda dalam dokumentasi. CloudFront

Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan alur kerja berbasis batang adalah pendekatan modern yang lebih disukai.

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur,

gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi ()OUs. Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. AWS menyediakan AWS SCT yang membantu dengan konversi skema.

ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran mesin. Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah. Untuk informasi selengkapnya, lihat praktik terbaik Deploy using immutable infrastructure di AWS Well-Architected Framework.

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

Industri 4.0

Sebuah istilah yang diperkenalkan oleh Klaus Schwab pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML.

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAc dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi lebih lanjut, lihat Membangun strategi transformasi digital Internet of Things (IIoT) industri.

inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPCs (dalam yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat Apa itu IoT?

interpretasi

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan integrasi operasi.

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan.

model bahasa besar (LLM)

Model AI pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke dalam bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat Apa itu LLMs.

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga endianness.

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

PETA

Lihat Program Percepatan Migrasi.

mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi lebih lanjut, lihat Membangun mekanisme di AWS Well-Architected Framework.

akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

Protokol komunikasi ringan machine-to-machine (M2M), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi dengan jelas APIs dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan ringan. APIs Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat Menerapkan layanan mikro di AWS.

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik dan pelajaran terbaik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari strategi AWS migrasi.

pabrik migrasi

Tim lintas fungsi yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat diskusi tentang pabrik migrasi dan panduan Pabrik Migrasi Cloud di kumpulan konten ini.

metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 AWS dengan Layanan Migrasi Aplikasi.

Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat panduan kesiapan migrasi. MRA adalah tahap pertama dari strategi AWS migrasi.

strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke file. AWS Cloud Untuk informasi lebih lanjut, lihat entri 7 Rs di glosarium ini dan lihat Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar.

ML

Lihat pembelajaran mesin.

modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat Strategi untuk memodernisasi aplikasi di. AWS Cloud

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat Mengevaluasi kesiapan modernisasi untuk aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat Mengurai monolit menjadi layanan mikro.

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya “Apakah produk ini buku, mobil, atau telepon?” atau “Kategori produk mana yang paling menarik bagi pelanggan ini?”

infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan infrastruktur yang tidak dapat diubah sebagai praktik terbaik.

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi lebih lanjut, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja Well-Architected AWS .

teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi Industri 4.0.

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat panduan integrasi operasi.

jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat Membuat jejak untuk organisasi dalam CloudTrail dokumentasi.

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis dan permintaan ke bucket S3. `PUT` `DELETE`

identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga OAC, yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

P

batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat Batas izin dalam dokumentasi IAM.

Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakanberbasis sumber daya), atau menentukan izin maksimum untuk semua akun dalam organisasi di (lihat kebijakan kontrol layanan). AWS Organizations

persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

Kondisi kueri yang mengembalikan `true` atau`false`, biasanya terletak di `WHERE` klausa.

predikat pushdown

Teknik optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat Kontrol pencegahan dalam Menerapkan kontrol keamanan pada. AWS

principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam istilah dan konsep Peran dalam dokumentasi IAM.

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau lebih. VPCs Untuk informasi selengkapnya, lihat Bekerja dengan zona yang dihosting pribadi di dokumentasi Route 53.

kontrol proaktif

Kontrol keamanan yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat panduan referensi Kontrol dalam AWS Control Tower dokumentasi dan lihat Kontrol proaktif dalam Menerapkan kontrol keamanan pada AWS.

manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

rantai cepat

Menggunakan output dari satu prompt LLM sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

pseudonimisasi

Proses penggantian pengidentifikasi pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

publish/subscribe (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam MES berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

Q

rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

R

Matriks RACI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

LAP

Lihat Retrieval Augmented Generation.

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

refactor

Lihat 7 Rs.

Region

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah “Berapa harga rumah ini akan dijual?” Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. Ketersediaan tinggi dan pemulihan bencana adalah pertimbangan umum ketika merencanakan ketahanan di. AWS Cloud Untuk informasi lebih lanjut, lihat AWS Cloud Ketahanan.

kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang

didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Jenis dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat Kontrol responsif dalam Menerapkan kontrol keamanan pada AWS.

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

Teknologi AI generatif di mana LLM mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat Apa itu RAG.

rotasi

Proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensil.

kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke Konsol Manajemen AWS atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

PENIPUAN

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat Apa yang ada di rahasia Secrets Manager? dalam dokumentasi Secrets Manager.

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.

pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCPs menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCPs daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti waktu kerja dan kinerja layanan.

indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator tingkat layanan.

model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama.

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan

mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat Pendekatan bertahap untuk memodernisasi aplikasi di. AWS Cloud

SPOF

Lihat satu titik kegagalan.

skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam gudang data atau untuk tujuan intelijen bisnis.

pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini diperkenalkan oleh Martin Fowler sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan Amazon CloudWatch Synthetics untuk membuat tes ini.

sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke LLM untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai sumber daya AWS.

variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat Apa itu gateway transit dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Organizations dengan AWS layanan lain dalam AWS Organizations dokumentasi.

penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat lingkungan.

V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPCs yang memungkinkan Anda untuk merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Apa itu peering VPC di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

Cache buffer yang berisi data terkini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap tidak dapat diubah.

Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan zero-day.

kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

bisikan zero-shot

Memberikan LLM dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk

menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa bidikan yang diminta.

aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.