MySQL – zarządzania relacyjnymi bazami danych, magazynowanie danych, integracja z różnymi językami

PostgreSQL – rozszerza SQL, obsługa JSON, XML, analiza danych, obsługa złożonych zapytań

Microsoft SQL Server – magazynowanie danych, analiza, raportowanie i integracja danych, integracja z usługami Microsoft

MariaDB - Lepsza wydajność, bezpieczeństwo i skalowalność niż MySQL,

Apache Cassandra - Rozproszona baza danych NoSQL, duża ilość danych, skalowalność, duża tolerancja błędów.

MongoDB - analizy w czasie rzeczywistym, elastyczna, skalowalna,

Amazon Aurora – magazyny danych, zgodna z MySQL i PostgreSQL, zaprojektowana dla chumury

Google BigQuery - zapytania SQL, magazyn danych, integralność z usługami Google Cloud

Apache HBase - Analiza w czasie rzeczywistym, wysoka przepustowość zapisu, skalowalność i integracja z Hadoop.

IBM Db2 – zarządzanie relacyjną bazą danych, chmura, magazynowanie danych

Apache Hive – zarządzanie dużymi zestawami danych, może się łączyć z Hadoop (HDFS)

Google BigQuery - magazyn danych, integruje się z usługami Google AI

Snowflake – magazynowanie danych w chmurze

**Monitoring**

Logowanie, przetwarzanie, wizualizacja wyników – zapis w postaci logów i analiza w czasie rzeczywistym za pomocą Elasticsearch i Kiban

Metryki modelu – monitoring za pomocą Prometheus – dane mogą być agregowane i wysyłane do kolejki w formacie json (monitoring: dokładność, precyzja, czułość )

Monitorowanie wydajności różnych wersji modelu AI - Optimizely, Google Analytics za ich pomocą można wysyłać wyniki testów w formacie JSON do kolejki

System alertów – anomalie w wynikach, Grafana z wtyczką Alertmanager, może być przekazywane do kolejki w formacie json.

Model AI jest zintegrowany z zewnętrznymi usługami (np. API), dane zwrotne z tych usług również mogą być zbierane i wysyłane do kolejki.

Po przetworzeniu danych przez model AI, wyniki oraz metryki wydajności mogą być przesyłane do wybranego systemu monitorowania. Dane te mogą być agregowane, przetwarzane, a następnie w formacie JSON wysyłane do kolejki. RabbitMQ - może przesyłać te dane do dalszej analizy lub przechowywania.

Monitorowanie z istniejącym AI runtime, aby móc zbierać dane zwrotne i metryki.

Dane formatowane jako JSON i przesyłane do kolejki, z której będą mogły być dalej przetwarzane lub analizowane.

To podejście zapewni ciągłe monitorowanie i ocenę wydajności modelu AI, umożliwiając reagowanie na wszelkie nieprawidłowości i optymalizację modelu w czasie rzeczywistym.

**Inne**

Apache Hadoop - zestaw narzędzi do przetwarzania dużych zbiorów danych w rozproszonym środowisku. Składa się z Hadoop Distributed File System (HDFS) do przechowywania danych oraz MapReduce do przetwarzania danych.HDFS umożliwia przechowywanie ogromnych ilości danych w rozproszony sposób, a MapReduce pozwala na przetwarzanie tych danych lokalnie. Hadoop można integrować z innymi narzędziami, takimi jak Apache Hive, aby wykonywać zapytania SQL na danych. Idealne do przetwarzania dużych, złożonych zbiorów danych lokalnie.

Hive jest narzędziem, które umożliwia wykonywanie zapytań SQL na danych przechowywanych w HDFS. Jest to narzędzie stworzone na bazie Hadoop i jest często używane do analizy dużych zbiorów danych. Idealne do analizowania dużych zbiorów danych przy użyciu SQL.

Spark - narzędzie do szybkiego przetwarzania danych, które może działać na Hadoop HDFS, ale także na własnym lokalnym systemie plików. oferuje wsparcie dla operacji na dużych zbiorach danych, w tym analizy w pamięci, strumieniowania, oraz machine learning. Szybkie przetwarzanie i analiza dużych zbiorów danych, trenowanie modeli AI na danych tabelarycznych.

Elasticsearch - rozproszony silnik wyszukiwania i analizy, używany do indeksowania i wyszukiwania dużych zbiorów danych. Elasticsearch przechowuje dane w formie dokumentów JSON, ale może być używany do zarządzania danymi tabelarycznymi poprzez odpowiednie mapowania. Wyszukiwanie, analiza danych, monitorowanie danych w czasie rzeczywistym.

Druid - rozproszona baza danych zaprojektowana do bardzo szybkiego przetwarzania zapytań analitycznych na dużych ilościach danych. Druid łączy cechy baz danych kolumnowych oraz analityki w pamięci. pozwala na zarządzanie danymi w postaci tabel i oferuje bardzo szybkie odpowiedzi na zapytania analityczne. Analiza dużych ilości danych w czasie rzeczywistym, BI (business intelligence).

ClickHouse to rozproszona baza danych kolumnowych zaprojektowana do analizy dużych ilości danych w czasie rzeczywistym. Jest optymalizowana pod kątem zapytań analitycznych. umożliwia przechowywanie dużych zbiorów danych tabelarycznych, co pozwala na bardzo szybkie przetwarzanie zapytań analitycznych. Analiza danych w czasie rzeczywistym, BI, big data.

Presto - rozproszony silnik zapytań SQL typu open source, zapytania na dużych zestawach danych.

może wykonywać zapytania na danych w miejscu ich przechowywania, w HDFS, S3, Cassandra, relacyjnych bazach danych, bez przenoszenia danych.

Apache Flink - możliwość obsługi dużych ilości danych w czasie rzeczywistym, co może być niezbędne do szkolenia i wnioskowania modeli AI.

analiza w czasie rzeczywistym i uczenia maszynowego na danych przesyłanych strumieniowo, lokalne zarządzanie danymi w aplikacjach AI.

Monitorinnng - Aby zaprojektować architekturę monitoringu dla systemu AI, który zasilany jest danymi z Oracle, przetwarza dane za pomocą kolejki RabbitMQ, a informacje są wysyłane do modelu AI w formie plików JSON.

Metryki do monitorowania - reklamacje rzeczywiście przepływają przez syste? od bazy, przez RabbitMQ, do modelu A?

- Liczba reklamacji, które wpłynęły do systemu w określonym czasie (z Oracle).

- Liczba reklamacji, które przeszły przez RabbitMQ (czy wiadomości nie zostały utracone).

- Liczba reklamacji przetworzonych przez model (czy każda reklamacja została przetworzona).

Alerting - Prometheus Alertmanager - Skonfigurowanie alertów w przypadku nietypowych zdarzeń, takich jak brak napływających danych. Zabbix - Monitorowanie stanu serwerów i aplikacji oraz konfiguracja powiadomień.

**Architektura monitoringu**

* Monitoring Oracle (z OEM lub custom SQL scripts)
* RabbitMQ Monitoring (RabbitMQ Management Plugin, Prometheus/Grafana)
* AI Runtime Monitoring (Log Analysis, ELK, Prometheus)
* JSON Integrity Checks (Validation Scripts, Monitoring)
* Model Monitoring (Accuracy, Time to Response)
* System Alerts (Alertmanager, Zabbix).

**6. Monitoring i ewaluacja modeli AI**

* **Prometheus + Grafana:** Prometheus może zbierać metryki z różnych systemów (w tym RabbitMQ i modeli AI), a Grafana służy do wizualizacji danych, co pozwala monitorować wydajność modeli w czasie rzeczywistym.
* **MLflow:** Narzędzie open-source do zarządzania procesem trenowania modeli, monitorowania metryk i zarządzania eksperymentami. Oferuje również rejestrowanie modeli i zarządzanie wersjami.
* **TensorBoard:** Narzędzie do monitorowania i wizualizacji trenowania modeli w TensorFlow. Pozwala na śledzenie metryk, takich jak dokładność, straty, i inne wskaźniki wydajności.
* **Neptune.ai:** Platforma do zarządzania eksperymentami w uczeniu maszynowym i AI. Pozwala na śledzenie i porównywanie różnych treningów modeli oraz wyników.
* **Weights & Biases:** Popularne narzędzie do zarządzania eksperymentami AI. Pozwala na śledzenie procesów trenowania modeli, zarządzanie hiperparametrami oraz monitorowanie wyników.

**Gdzie zapisywać wyniki z modelu?**

**Tworzenie nowej bazy danych na wyniki**

* Baza danych zapewnia trwałe przechowywanie wyników, co jest szczególnie przydatne, jeśli wyniki muszą być analizowane lub przetwarzane w przyszłości.
* Przy dużej ilości danych lepszym rozwiązaniem będzie przechowywanie wyników w dedykowanej bazie danych, ponieważ logi na dysku mogą szybko stać się nieefektywne. W zależności od skali, możesz zdecydować się na relacyjną bazę danych (RDBMS) lub rozwiązanie NoSQL.
* Użycie bazy danych umożliwia łatwe tworzenie zapytań SQL lub innych typów zapytań, aby analizować wyniki, monitorować trendy, sprawdzać poprawność oraz wyciągać wnioski na temat jakości modelu.
* Każdy wiersz może reprezentować jeden wynik, a kolumny mogą zawierać kluczowe dane, takie jak:

- Identyfikator przetworzonej reklamacji.

- Dane wejściowe.

- Wynik modelu.

- Czas przetwarzania.

- Ewentualne metryki modelu (np. pewność predykcji).

- Status reklamacji (czy została zaakceptowana, odrzucona, itp.).

* Możliwość łatwego wyszukiwania, filtrowania i analizy wyników.
* Trwałe przechowywanie wyników z dostępem do historycznych danych.
* Skalowalność – bazy danych lepiej radzą sobie z dużymi ilościami danych niż system plików.

**Logi zapisywane na dysku**

* Szybkie i łatwe w implementacji rozwiązanie.
* Dobrze nadaje się do systemów, które nie muszą przechowywać wyników w dłuższym okresie.
* Możliwość późniejszej agregacji i przetwarzania logów przy użyciu narzędzi takich jak ELK Stack (Elasticsearch, Logstash, Kibana) do analizy danych.
* Skalowalność może być problemem, jeśli dane zaczną się szybko gromadzić.
* Trudniejszy dostęp do historycznych danych i analiza w porównaniu do bazy danych.
* Potencjalne problemy z zarządzaniem dużymi plikami.

**Przechowywanie wyników w systemach Message Queue RabbitMQ**

* Jeśli wyniki muszą być przetwarzane asynchronicznie lub przez różne systemy, przechowywanie wyników w systemie kolejki, takim jak RabbitMQ czy Kafka, pozwala na ich dalszą dystrybucję.
* Przetwarzanie w czasie rzeczywistym - Kolejki wiadomości pozwalają na rozproszone przetwarzanie wyników i mogą być używane do przesyłania wyników do innych systemów w celu dalszej analizy.
* Można wysyłać wyniki z modelu do nowych kolejek, które będą odbierane przez inne komponenty systemu (np. do analizy lub dalszego przetwarzania).
* Umożliwia asynchroniczne przetwarzanie wyników przez wiele systemów.
* Dobre rozwiązanie dla systemów czasu rzeczywistego.
* Centralizacja logów i analiza (np. ELK Stack)
* Centralizacja logów z wynikami modelu AI w ELK Stack (Elasticsearch, Logstash, Kibana). To rozwiązanie umożliwia analizę wyników, tworzenie raportów i monitorowanie wyników modelu w czasie rzeczywistym.
* Logstash - agreguje logi z różnych systemów, w tym z wyników modelu AI.
* Elasticsearch - indeksuje logi i umożliwia szybkie wyszukiwanie wyników.
* Kibana - umożliwia wizualizację wyników oraz tworzenie raportów i dashboardów.
* monitorowanie w czasie rzeczywistym, łatwe przeszukiwanie i analiza wyników
* Jeśli wyniki są generowane w dużych ilościach i muszą być przechowywane przez dłuższy czas -baza danych najlepszym rozwiązaniem.
* skomplikowane analizy wyników - użycie bazy danych lub narzędzi do analizy logów (np. ELK Stack).
* Jeśli wyniki muszą być przetwarzane przez inne systemy (np. BI, analityka) – kolejki wiadomości lub bazy danych.
* Dostęp do historycznych danych - Baza danych pozwala łatwiej przeszukiwać i analizować dane historyczne niż logi na dysku (trudne do zarządzania, gdy rosną w objętości).

Kiedy potrzebna baza danych:

- Trwałe przechowywanie wyników - RabbitMQ działa jako system kolejek wiadomości i nie jest przeznaczony do długoterminowego przechowywania danych. Jeśli potrzebujesz przechowywać wyniki modelu AI na stałe, np. do audytu, analizy lub raportowania, warto rozważyć zapisanie tych wyników do bazy danych (np. PostgreSQL, MongoDB).

Analiza wyników - baza danych umożliwi szybkie wyszukiwanie, filtrowanie i przetwarzanie tych danych historycznych.

Baza danych może służyć jako bezpieczny magazyn dla wyników na wypadek problemów z RabbitMQ (np. awarie, utrata danych).

Kiedy nie potrzebna baza danych

Krótkoterminowe przetwarzanie - Jeśli wyniki są przetwarzane niemal natychmiast i nie potrzeba przechowywać ich przez dłuższy czas

Przetwarzanie danych w czasie rzeczywistym, nie wymaga długoterminowego przechowywania

implementacja dysków:

przechowywanie danych przez pewien czas (monitoring, analiza),

źle skalowalne przy dużych zbiorach danych

krótkoterminowe zapisywanie wyników – zapis w plikach json lub csv i analiza w partiach

przeszukiwanie i filtrowanie na poziomie plików może być uciążliwe

JSON - najpopularniejszy format do przesyłania wyników, RabbitMQ dobrze obsługuje dane tekstowe, a JSON jest łatwy do odczytania i parsowania przez różne komponenty. Możesz zapisywać wyniki modelu AI w formacie JSON (zawierającym kluczowe dane, takie jak identyfikator, wynik predykcji, czas przetwarzania itp.) i wysyłać je do RabbitMQ jako wiadomości.

Monitoring i alerty - monitorowanie RabbitMQ - RabbitMQ Management Plugin lub Prometheus + Grafana

Logi na dyskach - Mogą być stosowane jako dodatkowe rozwiązanie do krótkoterminowego przechowywania wyników, ale baza danych będzie bardziej efektywna w przypadku dużych ilości danych i konieczności ich przechowywania, analizy, raportowania.

Użycie formatu JSON i kolejek RabbitMQ, z monitorowaniem i alertami, aby zapewnić prawidłowy przepływ wyników modelu AI

+---------------------------------------+

| Oracle Database (dane) |

+---------------------------------------+

|

v

+---------------------------------------+

| Kolejka RabbitMQ | <-- Monitoring wiadomości w kolejce

+---------------------------------------+

|

v

+---------------------------------------+

| AI Runtime/Model | <-- Monitorowanie wyników modelu

+---------------------------------------+

|

v

+---------------------------------------+

| Kolejka RabbitMQ (wyniki) | <-- Monitoring wyników modelu

+---------------------------------------+

|

v

+---------------------------------------+

| Konsumenci wyników | <-- Ewaluacja wyników

+---------------------------------------+

|

v

+---------------------------------------+

| Baza danych na wyniki (opcjonalna) | <-- Przechowywanie wyników do analizy

+---------------------------------------+

**Narzędzia do monitorowania modelu AI:**

TensorBoard może służyć do monitorowania metryk treningu i predykcji modelu. Można również używać do monitorowania wydajności modelu w środowisku produkcyjnym.

Prometheus + Grafana - Wydajność modelu AI (czas odpowiedzi, liczba przetworzonych wiadomości, błędy) może być monitorowana w czasie rzeczywistym za pomocą metryk zbieranych przez Prometheus i wizualizowanych w Grafana.

MLflow / Neptune.ai - Te narzędzia pozwalają monitorować metryki trenowania i wdrażania modeli AI, a także śledzić ewaluację i automatyzację eksperymentów.

**Przykład metryk dla modelu AI:**

* Czas odpowiedzi modelu
* Dokładność
* Ilość przetworzonych wiadomości

**Monitoring wyników z RabbitMQ (po stronie konsumenta)**

Wyniki generowane przez model AI są wysyłane z powrotem do RabbitMQ, gdzie są odbierane przez konsumentów (np. do bazy danych lub innych systemów). Monitoring tego etapu jest istotny, aby upewnić się, że wyniki są prawidłowo przetwarzane i dostarczane.

Każdy wynik modelu AI powinien być walidowany, aby upewnić się, że został poprawnie wygenerowany (np. walidacja struktury JSON, sprawdzanie kluczowych pól).

Liczba przetworzonych wyników - RabbitMQ powinni być monitorowani, aby sprawdzić, ile wyników zostało prawidłowo przetworzonych, ile wiadomości nie zostało przetworzonych z powodu błędów, Mierzenie czasu, jaki zajmuje konsumentom przetwarzanie wyników modelu.

Narzędzia do monitorowania wyników:

* Elasticsearch + Kibana - wysyłanie logów wyników do Elasticsearch, aby łatwo analizować i przeszukiwać je za pomocą Kibana. To pozwala na łatwe monitorowanie trendów i błędów w przetwarzaniu wyników.
* Prometheus + Grafana - Zbieranie i wizualizacja metryk wyników w czasie rzeczywistym, np. liczba przetworzonych wiadomości, liczba błędnych wyników.
* Logi RabbitMQ - Sprawdzanie logów RabbitMQ w celu monitorowania stanu wiadomości i sprawdzania, czy wyniki są poprawnie dostarczane.

Ewaluacja modelu AI obejmuje ocenę jego skuteczności na podstawie różnych metryk, które są monitorowane w środowisku produkcyjnym. Ewaluacja pozwala na ciągłą optymalizację modelu i zapobieganie degradacji jego wydajności.

Jakie metryki ewaluować?

* Dokładność predykcji - Dokładność modelu powinna być regularnie oceniana na rzeczywistych danych produkcyjnych.
* Drift danych - Monitorowanie zmiany rozkładu danych wejściowych w czasie

**MLflow vs Neptune.ai**

MLflow - Jest bardziej kompleksowym narzędziem do monitorowania i zarządzania eksperymentami, a także wdrażania modeli AI w różnych środowiskach. Jest open-source i łatwy do rozszerzenia, ale wymaga dodatkowej konfiguracji (np. backendu do metryk)

Neptune.ai - Jest bardziej zaawansowanym narzędziem do zarządzania eksperymentami, z naciskiem na wizualizację wyników i zarządzanie modelami. Jego siłą jest łatwość użycia i gotowe rozwiązania do monitorowania modeli w czasie rzeczywistym.

**Rodzaje monitorowania danych:**

* Monitorowanie jakości danych (Data Quality Monitoring)
* Brakujące dane (Missing Data)- Sprawdzanie, czy dane wejściowe nie mają brakujących wartości, które mogą wpłynąć na wydajność modelu.
* Błędy w danych (Data Errors) - Monitorowanie wystąpień nieprawidłowych wartości (np. ujemne wartości dla zmiennych, które powinny być dodatnie).
* Outliery - Wykrywanie wartości odstających, które mogą zniekształcać wyniki modelu.
* Monitorowanie spójności danych (Data Consistency Monitoring)
* Sprawdzanie, czy dane wejściowe zachowują spójność z danymi treningowymi. Należy monitorować struktury danych, ich typy i zakresy wartości.
* Monitorowanie rozkładu danych (Data Distribution Monitoring)
* Śledzenie rozkładu cech danych w czasie, aby upewnić się, że dane wejściowe nie zmieniają się w sposób, który wpłynie na model. To kluczowe dla wykrywania dryfu danych.
* Monitorowanie danych strumieniowych (Streaming Data Monitoring)
* Jeśli dane napływają do systemu w czasie rzeczywistym (np. przez RabbitMQ), należy monitorować ich przepływ, aby sprawdzić, czy model otrzymuje odpowiednią ilość danych, a także wykrywać opóźnienia i brakujące dane.
* Walidacja danych (Data Validation)
* Sprawdzanie, czy dane są zgodne ze schematami, które model wymaga. Może to obejmować walidację typów danych, ich zakresów wartości oraz struktury.

**Narzędzia do monitorowania dancyh:**

* Great Expectations - Narzędzie do walidacji i monitorowania jakości danych w systemach produkcyjnych.
* Deequ (AWS) - Narzędzie do monitorowania jakości danych i wykrywania problemów w dużych zbiorach danych.
* DataRobot MLOps - Narzędzie do monitorowania i zarządzania jakością danych oraz monitorowania modeli.

Śledzenie jakości wyników predykcji na podstawie danych rzeczywistych. Jeśli jest możliwość porównania wyników modelu z rzeczywistymi wynikami (np. w systemach, gdzie prawdziwe wyniki są znane po czasie), można monitorować, jak dokładnie model przewiduje wartości.

Niektóre modele AI mogą przewidywać wyniki z różnymi poziomami pewności. Monitorowanie niepewności wyników pomaga ocenić, kiedy model przewiduje wyniki z wysoką niepewnością i może wymagać dodatkowej analizy.

Detekcja anomalii wyników

Monitorowanie nietypowych wyników, które mogą wskazywać na problemy z modelem lub danymi. Na przykład, jeśli model AI nagle zaczyna przewidywać wyniki zbyt jednostronne (np. wszystkie pozytywne klasy), może to wskazywać na problem.

Monitorowanie decyzji biznesowych opartych na wynikach modelu

wyniki modelu są wykorzystywane do podejmowania decyzji biznesowych, monitorowanie wpływu tych decyzji (np. sprzedaż, zadowolenie klientów, wskaźniki konwersji) pomaga ocenić skuteczność modeli AI w rzeczywistym środowisku.

Monitorowanie modeli AI - obejmuje śledzenie metryk wydajności, opóźnień, dryfu modelu oraz zasobów systemowych. Jest to kluczowe dla zapewnienia, że model działa wydajnie i nie występują problemy po wdrożeniu.

Monitorowanie danych - koncentruje się na jakości, spójności i rozkładzie danych wejściowych, co jest kluczowe dla zapewnienia, że model przetwarza odpowiednie dane.

Monitorowanie wyników - pozwala na ocenę, czy model dostarcza poprawne i trafne predykcje, a także jak wyniki te wpływają na decyzje biznesowe lub inne procesy.

**MLflow nie posiada wbudowanej funkcjonalności alertów** w sposób podobny do narzędzi takich jak **Prometheus** czy **Grafana**. Jednak możesz ustawić alerty na podstawie wyników monitorowanych przez MLflow, korzystając z integracji z zewnętrznymi systemami, które mają możliwości alertowania.

**Grafana vs tableau**

Tableau – Zalety

* Zaawansowane raportowanie
* Interaktywne dashboardy
* Bogata integracja

Tableau – Wady

* Brak natywnych alertów - nie oferuje wbudowanej obsługi alertów w czasie rzeczywistym

Grafana – Zalety

* Real-time monitoring
* Natywne alerty

Grafana – Wady

* Ograniczone raportowanie - Grafana nie jest zaprojektowana jako narzędzie do raportowania biznesowego

Tableau - może być z powodzeniem użyte do monitorowania wyników modelu AI, zwłaszcza w zakresie tworzenia interaktywnych raportów i analizowania danych historycznych. Jednak

Grafana - lepiej sprawdzi się do monitorowania w czasie rzeczywistym i ustawiania alertów.

Narzędzia podobne do Tableau:

Power BI - Idealny do interaktywnych dashboardów i automatycznych raportów z dobrym stosunkiem jakości do ceny.

Qlik Sense - Zaawansowane narzędzie analityczne z elastyczną integracją i możliwością tworzenia zaawansowanych raportów.

Looker - Narzędzie Google Cloud o wysokiej elastyczności w integracji z danymi chmurowymi.

Sisense - Skupione na analizie dużych zbiorów danych i automatyzacji raportów.

Zoho Analytics - Bardziej dostępne cenowo narzędzie z dobrymi funkcjami do raportowania i wizualizacji.

Domo – Narzędzie oferujące generowanie raportów w czasie rzeczywistym z silną integracją z bazami danych.

Metabase – Open-source'owe narzędzie do prostych wizualizacji i raportów.

Redash – Open-source'owe narzędzie skoncentrowane na prostocie i szybkim tworzeniu raportów z baz danych.

Grafana może być z powodzeniem stosowana do ciągłego monitorowania wyników modeli AI oraz generowania raportów na podstawie danych historycznych.

Grafana - może być zintegrowana z bazą danych Hive (często używaną w ekosystemach Hadoop do zarządzania dużymi zbiorami danych) oraz Oracle do monitorowania wyników modelu AI i generowania długoterminowych raportów.

Grafana + Presto/ Trino - Grafana może łączyć się z Presto, aby pobierać dane z Hive i wizualizować je na dashboardach.

Oracle to relacyjna baza danych o szerokim zastosowaniu w biznesie, a Grafana wspiera bezpośrednią integrację z bazami Oracle.

Grafana posiada wtyczkę do połączeń z bazami danych Oracle. Można skonfigurować Grafanę do bezpośredniego połączenia z bazą Oracle, a następnie tworzyć zapytania SQL do pobierania i wizualizacji danych historycznych.

Aby monitorować wyniki modelu AI w perspektywie długoterminowej, Grafana może łączyć się z bazą danych (Hive lub Oracle) i wykonywać zapytania, które pobierają dane historyczne.

Tworzenie dashboardów - które będą wyświetlały dane z długiego okresu (np. półroczne, roczne).

Generowanie raporty - z danych zapisanych w bazie, które można eksportować i dystrybuować jako raporty kwartalne, półroczne czy roczne.

Ustawiać alerty - na podstawie danych historycznych – np. jeśli wskaźniki jakości modelu spadają poniżej określonego progu w dłuższym okresie.

Dashboard z danymi historycznymi - wyniki modelu AI za ostatni rok (np. dokładność, precyzję, liczbę TP/FP) na podstawie danych przechowywanych w Hive lub Oracle.

Automatyczne raporty - W Grafana można skonfigurować automatyczne generowanie raportów w formie PDF na podstawie danych z Hive lub Oracle, które będą wysyłane w określonym harmonogramie (np. co miesiąc, pół roku, rok).

Grafana może zostać zintegrowana zarówno z Hive (za pośrednictwem Presto/Trino), jak i z Oracle co pozwala na monitorowanie wyników modelu AI oraz generowanie długoterminowych raportów, takich jak półroczne czy roczne analizy wyników.

W przypadku Hive integracja z Presto/Trino umożliwia wykonywanie zapytań SQL, które pobierają dane z rozproszonych systemów, a Grafana może te dane wizualizować i analizować.

W przypadku Oracle Grafana bezpośrednio łączy się z bazą, co umożliwia tworzenie raportów na podstawie danych przechowywanych w długim okresie.