EBOOK

Il grande libro dell'ingegneria dei dati

Una raccolta di blog tecnici, con esempi di codice e notebook

databricks

Sommario

databricks

CAPITOLO

01

Introduzione all’ingegneria dei dati su Databricks

databricks

Le organizzazioni capiscono l'importanza dei dati come risorsa strategica per realizzare diversi obiettivi, ad esempio aumentare i ricavi, migliorare l'esperienza del cliente, operare in modo efficiente, migliorare un prodotto o servizio. Tuttavia, l'accesso e la gestione dei dati per queste iniziative sono diventati sempre più complessi. Tale complessità è dovuta in gran parte all'esplosione del volume e delle tipologie di dati: le organizzazioni accumulano ormai una quota di dati non strutturati e semi-strutturati stimata attorno all'80%. A fronte di una raccolta di dati in continuo aumento, il 73% dei dati resta inutilizzato ai fini dell'analisi e delle decisioni. Per provare a diminuire queste percentuali e utilizzare più dati, i team di ingegneria dei dati sono incaricati di costruire pipeline di dati per rendere disponibili i dati in modo efficiente e affidabile. Il processo di costruzione di queste pipeline di dati complesse comporta però numerose difficoltà:

- per trasferire i dati in un data lake, gli ingegneri dei dati devono dedicare moltissimo tempo alla scrittura manuale di codice per attività ripetitive di acquisizione (ingestione) dei dati;
- poiché le piattaforme di gestione dei dati cambiano continuamente, gli ingegneri devono dedicare molto tempo alla costruzione e alla manutenzione, e poi al rifacimento, di un'infrastruttura scalabile complessa;
- poiché è sempre più importante avere dati in tempo reale, servono pipeline di dati a bassa latenza, che sono ancora più difficili da costruire e mantenere;
- infine, una volta scritte tutte le pipeline, gli ingegneri dei dati si devono focalizzare costantemente sulle prestazioni, ottimizzando le pipeline e le architetture per rispettare gli accordi di servizio (SLA).

Quale aiuto offre Databricks?

Con Databricks Lakehouse Platform, gli ingegneri dei dati hanno accesso a una soluzione di ingegneria dei dati a 360 gradi per acquisire, trasformare, elaborare, schedulare e fornire dati. La piattaforma lakehouse automatizza la complessità dei processi di costruzione e manutenzione delle pipeline e di esecuzione dei carichi di lavoro ETL direttamente su un data lake, in modo che gli ingegneri possano concentrarsi sulla qualità e sull'affidabilità per fornire informazioni approfondite e dettagliate.

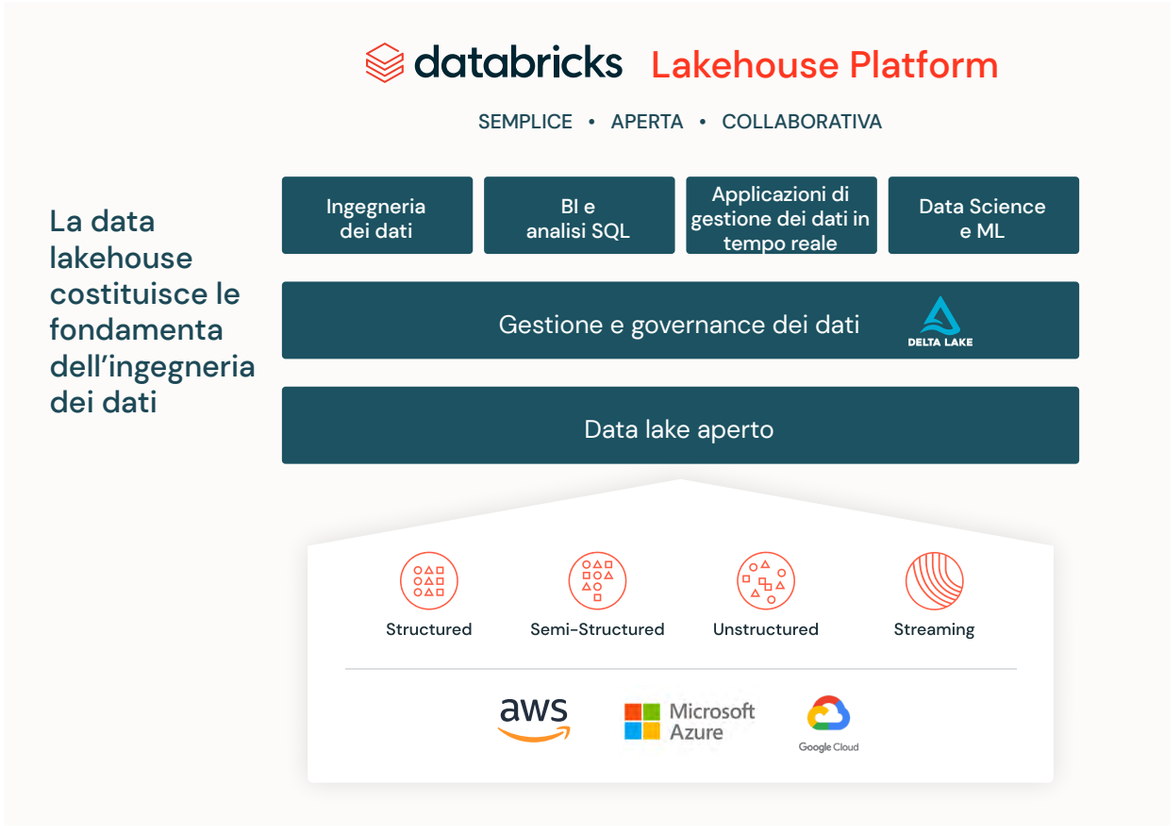

Figura 1
Databricks Lakehouse Platform unifica dati, analisi e IA su un'unica piattaforma per tutti i casi di utilizzo dei dati

Elementi distintivi dell'ingegneria dei dati con Databricks

Semplificando l'attività con un'architettura lakehouse, gli ingegneri dei dati hanno bisogno di un approccio "professionale" alla costruzione di pipeline di dati. Per operare con successo, un team di ingegneria dei dati deve soddisfare otto requisiti fondamentali.

Acquisizione di dati continua o programmata

Avendo la capacità di "ingerire" petabyte di dati con schemi auto-evolventi, gli ingegneri dei dati possono fornire dati affidabili in modo veloce, scalabile e automatico, a scopo di analisi, data science o machine learning. In particolare devono:

- elaborare dati in modo progressivo ed efficiente man mano che arrivano da file o sorgenti di streaming come Kafka, DBMS e NoSQL;
- inferire automaticamente schemi e rilevare cambiamenti di colonne per formati di dati strutturati e non strutturati;
- tracciare dati in modo automatico ed efficiente man mano che arrivano, senza interventi manuali;
- prevenire la perdita di dati salvando le colonne di dati.

Pipeline ETL dichiarative

Gli ingegneri dei dati possono ridurre lavoro e tempi di sviluppo per focalizzarsi sull'implementazione di controlli della logica operativa e della qualità dei dati all'interno della pipeline, utilizzando SQL o Python. Questo obiettivo si raggiunge:

- utilizzando uno sviluppo dichiarativo basato sugli intenti per semplificare "come" e definire "cosa" risolvere;
- creando automaticamente schemi di provenienza (lineage) dei dati di alta qualità e gestendo le interdipendenze fra le tabelle su tutta la pipeline dei dati;
- verificando automaticamente eventuali dipendenze mancanti o errori di sintassi, e gestendo il ripristino delle pipeline di dati.

Convalida e monitoraggio della qualità dei dati

Si può migliorare l'affidabilità dei dati su tutta la data lakehouse, in modo che i team di gestione dei dati possano avere fiducia nelle informazioni che utilizzeranno nelle attività a valle, in particolare nei seguenti modi:

- definendo controlli di qualità e integrità dei dati all'interno della pipeline con aspettative ben definite;
- gestendo gli errori di qualità dei dati con politiche predefinite (fail, drop, allerta, quarantena);
- sfruttando le metriche di qualità dei dati acquisite, tracciate e analizzate nei report per l'intera pipeline dei dati.

Ripristino automatico con tolleranza agli errori

È possibile gestire errori transitori e recuperare dalle condizioni di errore più comuni che si verificano durante il funzionamento di una pipeline con un ripristino veloce, scalabile e automatico che comprenda:

- meccanismi di tolleranza agli errori per ripristinare lo stato dei dati in modo coerente;
- capacità di tracciare automaticamente lo stato di avanzamento dalla sorgente con punti di controllo (checkpoint);
- capacità di recuperare e ripristinare automaticamente lo stato della pipeline di dati.

Osservabilità della pipeline di dati

Lo stato complessivo della pipeline di dati può essere monitorato attraverso un dashboard grafico del flusso di dati e si può tracciare visivamente la salute dell'intera pipeline per verificare prestazioni, qualità e latenza. Le funzionalità di osservazione della pipeline di dati comprendono:

- un diagramma della provenienza dei dati di alta qualità e fedeltà, che offra visibilità su come i dati affluiscono per un'analisi ad alto impatto;
- registrazione granulare con prestazioni e stato della pipeline dei dati a livello di riga;
- monitoraggio continuo dei lavori sulla pipeline dei dati per garantire la continuità operativa.

Elaborazione dei dati in batch e in streaming

Gli ingegneri dei dati possono perfezionare la latenza dei dati con controllo dei costi senza bisogno di conoscere processi complessi di elaborazione dello streaming o implementare logiche di recupero.

- I carichi di lavoro della pipeline di dati possono essere eseguiti su cluster elastici basati su Apache Spark™ per svolgere attività su larga scala con prestazioni elevate
- Si possono utilizzare cluster di ottimizzazione delle prestazioni che parallelizzano i lavori e riducono al minimo il movimento dei dati

Implementazioni e attività operative automatiche

La fornitura affidabile e prevedibile di dati per casi di utilizzo di analisi e machine learning può essere garantita mediante implementazioni semplici e automatiche di pipeline di dati per ridurre al minimo i tempi morti. Vantaggi:

- implementazione completa, parametrizzata e automatizzata per la fornitura continua di dati;
- orchestrazione, collaudo e monitoraggio completi dell'implementazione della pipeline di dati su tutti i principali fornitori di servizi cloud.

Pipeline e flussi di lavoro programmati

Orchestrazione semplice, chiara e affidabile delle attività di elaborazione dei dati per pipeline di dati e machine learning, con la capacità di eseguire molteplici compiti non interattivi, come un grafo aciclico diretto (DAG) su un cluster di calcolo Databricks.

- Le attività possono essere facilmente orchestrate in un DAG utilizzando l'interfaccia utente e l'API di Databricks.
- Si possono creare e gestire molteplici attività nei lavori tramite UI o API e feature, ad esempio avvisi via mail per il monitoraggio.
- È possibile orchestrare qualsiasi attività che abbia un'API al di fuori di Databricks e su tutti i cloud.

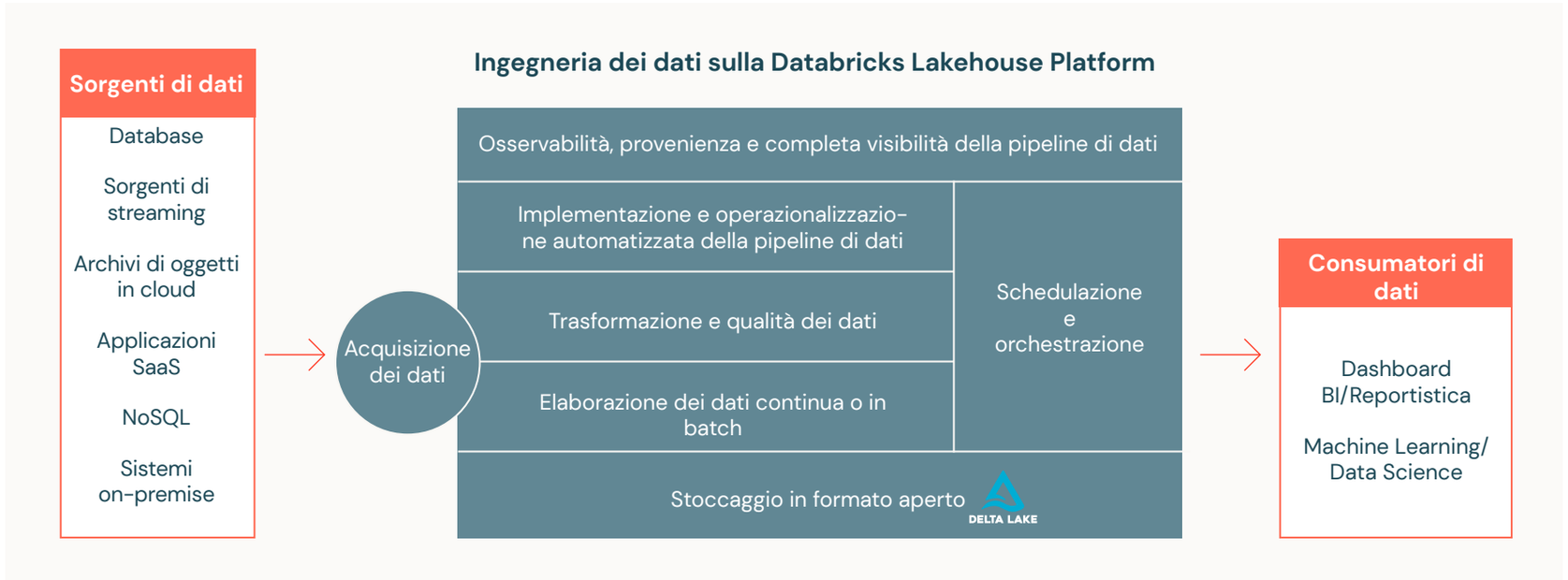

Figura 2
Ingegneria dei dati su architettura di riferimento Databricks

Conclusioni

Con un numero crescente di organizzazioni che vogliono essere guidate dai dati, l'ingegneria dei dati diventa una chiave del successo. Per fornire dati affidabili, gli ingegneri dei dati non devono essere costretti a sprecare tempo per sviluppare e mantenere manualmente un ciclo di vita ETL completo. I team di ingegneria dei dati hanno bisogno di un metodo efficiente e scalabile per semplificare lo sviluppo ETL, migliorare l'affidabilità dei dati e gestire le attività operative.

Gli otto fattori distintivi sopra descritti semplificano la gestione del ciclo di vita ETL automatizzando e mantenendo tutte le dipendenze dei dati, sfruttando controlli di qualità integrati con il monitoraggio e offrendo visibilità approfondita sulle operazioni della pipeline con ripristino automatico. I team di ingegneria dei dati possono così focalizzarsi su attività semplici e veloci per costruire pipeline di dati affidabili e complete, pronte per andare in produzione, utilizzando solo SQL o Python per elaborazioni in batch e streaming che forniscano dati ad alto valore per analisi, data science e machine learning.

Casi di utilizzo

Nel prossimo capitolo descriveremo le best practice per casi di utilizzo dell'ingegneria dei dati end-to-end ispirati a esempi reali. Dall'acquisizione ed elaborazione dei dati, all'analisi e al machine learning, spiegheremo come trasformare i dati grezzi in informazioni fruibili. Metteremo a disposizione set di

CAPITOLO

02

Casi di utilizzo reali sulla Databricks Lakehouse Platform

Analisi del punto vendita in tempo reale con la data lakehouse

Costruire una lakehouse di sicurezza informatica per gli eventi su CrowdStrike Falcon

Sfruttare la potenza dei dati sanitari con una moderna data lakehouse

Puntualità e affidabilità nella trasmissione di report normativi

Soluzioni AML su larga scala con la Databricks Lakehouse Platform

Costruire un modello IA in tempo reale per rilevare comportamenti tossici nei video-game

Guidare la trasformazione in Northwestern Mutual (Insights Platform) con il passaggio a un'architettura lakehouse scalabile aperta

Come il team dei dati di Databricks ha costruito una lakehouse su tre cloud e oltre 50 regioni

PAR. 2.1

Analisi del punto vendita in tempo reale con la data lakehouse

di **BRYAN SMITH** e **ROB SAKER**

9 settembre 2021

I problemi della catena di fornitura (dalla riduzione delle forniture di prodotti alla diminuzione delle capacità di magazzino), sommati alla rapida evoluzione delle aspettative dei consumatori che chiedono esperienze omnicanale pienamente integrate, stanno spingendo le aziende del commercio al dettaglio a ripensare alle loro modalità di utilizzo dei dati per gestire la loro attività. Prima della pandemia, il 71% delle aziende indicava la mancanza di visibilità in tempo reale sull'inventario come uno degli ostacoli principali alla realizzazione dell'omnicanalità. La pandemia ha semplicemente aumentato la domanda di esperienza online e in negozio integrate, aumentando ulteriormente la pressione sui commercianti chiamati a indicare l'esatta disponibilità dei prodotti e gestire modifiche agli ordini in tempo reale. Un migliore accesso a informazioni in tempo reale è la chiave per rispondere alle richieste dei consumatori nella nuova normalità.

In questo blog analizzeremo la necessità di dati in tempo reale nel commercio al dettaglio e spiegheremo come muovere flussi di dati del punto vendita in tempo reale in grande quantità con una data lakehouse.

Il sistema del punto vendita

Il sistema di gestione del punto vendita (POS) è da tempo il fulcro dell'infrastruttura del negozio, dove si registrano lo scambio di merci e servizi fra commerciante e cliente. Per gestire questo scambio, il POS in genere traccia gli inventari dei prodotti e agevola il ripristino delle quantità quando le scorte scendono sotto il livello critico. L'importanza del POS per le attività in negozio non è mai sottolineata a sufficienza; trattandosi del sistema che registra le attività di vendita e inventario, l'accesso ai suoi dati è fondamentale per gli analisti.

Tradizionalmente, a causa della connettività limitata fra i singoli negozi e gli uffici centrali, il sistema POS (non solo i terminali) risiedeva fisicamente all'interno del negozio. Nelle ore di minore attività, questi sistemi si collegavano con il sistema centrale per trasmettere i dati riepilogativi che, una volta consolidati in un data warehouse, fornivano una vista relativa al giorno precedente dei risultati operativi, che invecchiavano ulteriormente fino al successivo invio dei dati alla sera seguente.

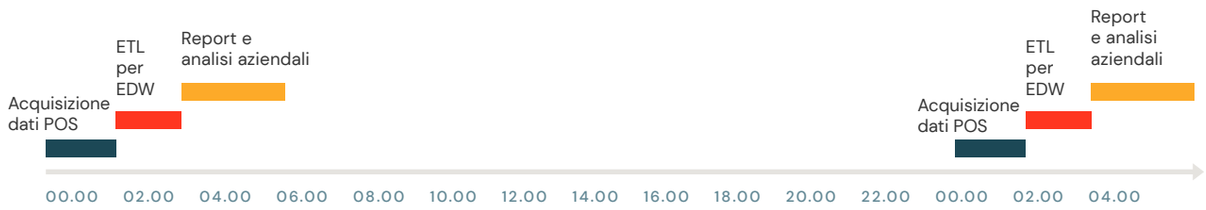

Figura 1
Disponibilità di inventario con schemi ETL tradizionali in batch

Con il miglioramento della connettività, molti commercianti sono passati a un sistema POS centralizzato in cloud, mentre altri stanno sviluppando integrazioni in tempo quasi reale fra sistemi in negozio e back-office centrale. Grazie alla disponibilità di informazioni in tempo quasi reale, i commercianti possono aggiornare continuamente le stime della disponibilità degli articoli. Così l'azienda non gestisce più l'attività sulla base di inventari vecchi di un giorno, ma al contrario agisce conoscendo lo stato attuale delle scorte.

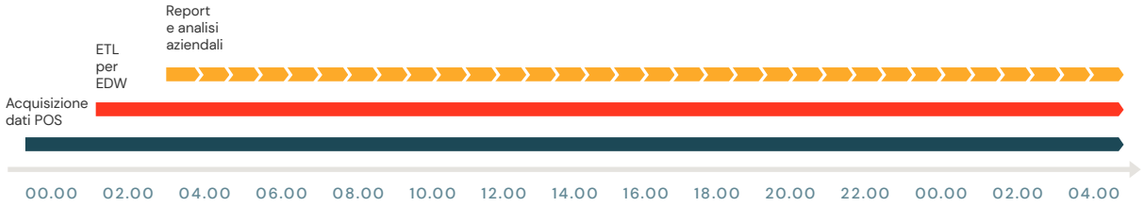

Figura 2
Disponibilità di inventario con schemi ETL tradizionali in streaming

Informazioni approfondite in tempo quasi reale

Data l'importanza delle informazioni in tempo quasi reale per l'attività di un negozio, il passaggio da processi notturni a un flusso continuo di informazioni comporta sfide rilevanti, non solo per l'ingegnere dei dati che deve progettare un diverso tipo di flusso di lavoro per l'elaborazione dei dati, ma anche per il consumatore delle informazioni. In questo post condividiamo l'esperienza di alcuni clienti che hanno appena intrapreso questo percorso ed esaminiamo come modalità e funzionalità offerte dalla lakehouse possano portare al successo.

LEZIONE 1

Valutare attentamente l'ambito

I sistemi POS spesso non si limitano alla gestione di vendite e inventario. Possono fornire un'ampia gamma di funzionalità, quali pagamenti, gestione del credito, fatturazione, ordini, gestione di programmi fedeltà, programmazione dei turni e degli orari di lavoro, fino alle buste paga, proponendosi come veri e propri "coltellini svizzeri" super-accessoriati.

Di conseguenza, i dati contenuti nel POS sono solitamente sparsi in una struttura di database ampia e complessa. Nei casi più fortunati, la soluzione POS offre un livello di accesso ai dati che rende i dati stessi accessibili attraverso strutture di più facile interpretazione. Altrimenti l'ingegnere dei dati deve districarsi in un oscuro labirinto di tabelle per capire che cosa è utile e che cosa non lo è.

Indipendentemente dal modo in cui i dati vengono esposti, vale il consiglio classico: individuare una valida giustificazione per la soluzione e utilizzarla per limitare le informazioni consumate inizialmente. Tale giustificazione viene spesso fornita da un forte sostenitore interno all'azienda, incaricato di risolvere un problema specifico, che considera la disponibilità di informazioni più tempestive fondamentale per avere successo.

Per chiarire questo concetto, pensiamo a una problematica tipica per molte aziende retail: la realizzazione di soluzioni omnicanale. Queste soluzioni, che consentono di effettuare acquisti online con ritiro in negozio e transazioni fra diversi negozi, dipendono da una ragionevole accuratezza dei dati di inventario dei negozi. Se dovessimo limitare lo scopo iniziale a quest'unica necessità, il fabbisogno di informazioni per il nostro sistema di monitoraggio e analisi sarebbe drasticamente ridotto. Una volta fornita una soluzione di inventario in tempo reale il cui valore è riconosciuto dall'azienda, possiamo ampliare il nostro raggio d'azione ad altre esigenze, ad esempio il monitoraggio delle promozioni e il rilevamento delle frodi, espandendo lo spettro di risorse informative sfruttate a ogni iterazione.

LEZIONE 2

Allineare la trasmissione con schemi di generazione dei dati e sensibilità temporali

Diversi processi generano dati in modo diverso all'interno del POS. Le transazioni di vendita probabilmente lasceranno una scia di nuovi record collegati alle tabelle corrispondenti. I resi possono seguire percorsi diversi, innescando aggiornamenti di registrazioni di vendita passate, l'inserimento di nuovi record di vendite inverse e/o l'inserimento di nuove informazioni in strutture specifiche per i resi. Documentazione del fornitore, know-how e persino qualche indagine indipendente potrebbero essere necessari per scoprire esattamente come e dove finiscono all'interno del POS le informazioni su eventi specifici.

Capire questi schemi può aiutare a costruire una strategia di trasmissione dei dati per specifici tipi di informazioni. Modalità con frequenza e granularità maggiore potrebbero risultare ideali per lo streaming continuo. Eventi meno frequenti su più larga scala potrebbero essere più in linea con trasmissioni di dati in batch in grande quantità. Queste due modalità di trasmissione dei dati rappresentano gli estremi, ma la maggior parte degli eventi acquisiti dal POS potrebbe ricadere in qualche punto a metà strada.

L'aspetto interessante dell'approccio della data lakehouse all'architettura dei dati è che si possono impiegare in parallelo diverse modalità di trasmissione dei dati. Per i dati compatibili con la trasmissione continua, si può utilizzare lo streaming. Per i dati più idonei a trasmissioni in massa, si possono utilizzare processi batch. E per i dati a metà strada, si può valutare la tempestività dei dati necessari per i processi decisionali e, da lì, decidere quale strada seguire. Tutte queste modalità possono essere gestite con un approccio coerente all'implementazione ETL, un problema che in passato ha fatto naufragare molte implementazioni di quelle che spesso vengono definite architetture lambda.

LEZIONE 3

Gestire i dati in diverse fasi

I dati arrivano dai sistemi POS in negozio con frequenze, formati e aspettative di disponibilità temporale diverse. Sfruttando il metodo di progettazione Bronze, Silver & Gold molto diffuso nelle lakehouse, si può separare la fase iniziale di pulizia, riformattazione e persistenza dei dati dalle trasformazioni più complesse richieste per la produzione di informazioni e materiali specifici.

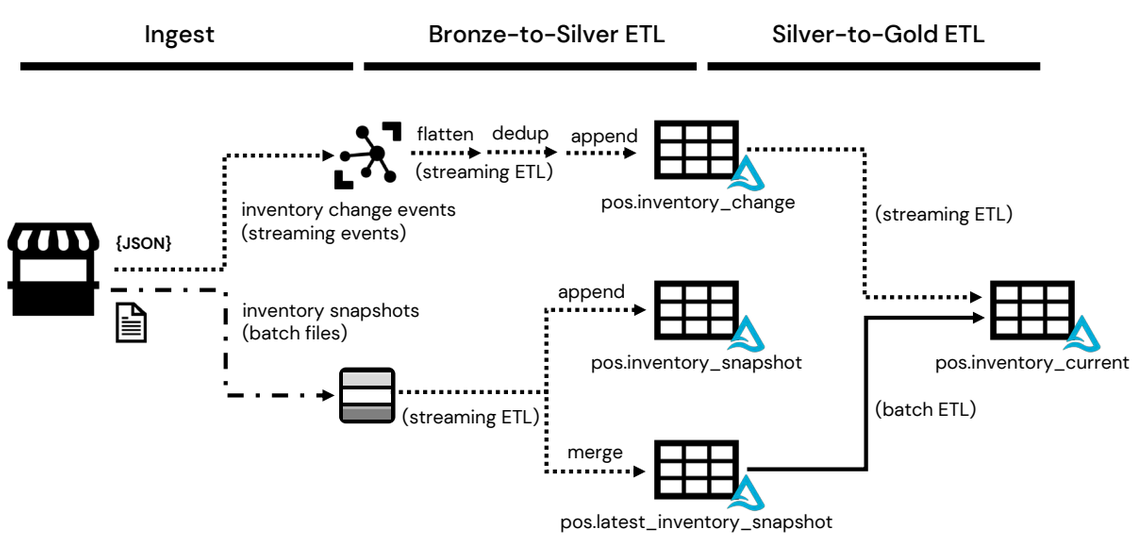

Figura 3
Architettura di data lakehouse per il calcolo dell'inventario corrente utilizzando lo schema di persistenza dei dati Bronze, Silver & Gold

LEZIONE 4

Gestire le aspettative

Il passaggio all'analisi in tempo quasi reale richiede un cambio organizzativo. Gartner descrive questa svolta nel modello di Streaming Analytics Maturity, nel quale l'analisi dei dati in streaming viene integrata nella trama delle operazioni giornaliere. Questo passaggio non può avvenire in un giorno.

Gli ingegneri dei dati hanno infatti bisogno di tempo per capire le problematiche inerenti alla fornitura di dati in streaming da negozi fisici a un back-office centralizzato in cloud. I miglioramenti della connettività e dell'affidabilità dei sistemi, unitamente a flussi di lavoro ETL sempre più robusti, consentono di trasferire dati in modo più puntuale, affidabile e coerente. Questo comporta spesso la necessità di consolidare i rapporti con gli ingegneri sistemi e gli sviluppatori di applicazioni, per ottenere un livello di integrazione che non esisteva ai tempi dei flussi di lavoro ETL solo in batch.

Gli analisti aziendali devono acquisire familiarità con la "rumorosità" intrinseca dei dati che vengono aggiornati continuamente. Dovranno imparare nuovamente come eseguire il lavoro di diagnostica e convalida su un set di dati, ad esempio quando una query eseguita pochi secondi prima restituisce un risultato leggermente diverso. Devono essere più consapevoli dei problemi presenti nei dati, che spesso restano nascosti quando vengono presentati i dati aggregati giornalieri. Tutto questo richiederà aggiustamenti sia all'analisi, sia alla risposta ai segnali rilevati nei risultati.

E tutto questo avviene nelle primissime fasi della maturazione. Nelle fasi successive, la capacità dell'organizzazione di rilevare segnali significativi all'interno del flusso può portare a una maggiore automazione delle capacità di rilevamento e risposta. È qui che viene sfruttato il massimo potenziale dei flussi di dati in termini di valore. Monitoraggio e governance devono però essere attuati e verificati prima che l'azienda affidi le proprie attività operative a queste tecnologie.

Implementare lo streaming del POS

Per mostrare come si può applicare un'architettura lakehouse a dati POS, abbiamo sviluppato un flusso di lavoro dimostrativo nel quale calcoliamo un inventario in tempo quasi reale. Abbiamo immaginato due sistemi POS che trasmettono informazioni di inventario associate a dati di vendita, rifornimento e restringimento, unitamente a transazioni con acquisto online e ritiro in negozio (quindi iniziate in un sistema e finalizzate in un altro), nell'ambito di un flusso in streaming di variazione dell'inventario. Il POS effettua conteggi periodici (istantanee) delle unità di prodotto a scaffale e trasmette i dati in massa. Questi dati vengono simulati per un periodo di un mese e riprodotti a velocità decuplicata per avere maggiore visibilità sulle variazioni di inventario.

I processi ETL (come mostrato in Figura 3) rappresentano un insieme di tecniche di streaming e batch. Un approccio in due fasi con una minima trasformazione dei dati acquisiti in tabelle Delta che rappresentano il nostro livello Silver distingue l'approccio ETL iniziale, più tecnico, dall'approccio più orientato al business richiesto per i calcoli di inventario attuali. La seconda fase è stata implementata utilizzando funzionalità di streaming strutturato tradizionali, che potrebbero essere riviste con la nuova funzionalità Delta Live Tables ma mano che diventa disponibile a tutti.

La demo utilizza Azure IOT Hubs e Azure Storage per l'acquisizione dei dati, ma funzionerebbe allo stesso modo sui cloud AWS e GCP con tecnologie sostitutive appropriate.

Comincia a sperimentare con questi notebook Databricks gratuiti

- **POS 01: Configurazione ambiente**
- **POS 02: Generazione dati**
- **POS 03: ETL di acquisizione**
- **POS 04: Inventario corrente**

PAR. 2.2 Costruire una lakehouse di sicurezza informatica per gli eventi su CrowdStrike Falcon

di AEMRO AMARE, ARUN PAMULAPATI, YONG SHENG HUANG e JASON POHL

20 maggio 2021

I team di sicurezza hanno bisogno dei dati degli endpoint per rilevare e respingere le minacce, investigare gli incidenti e soddisfare i requisiti di conformità. I volumi di dati possono essere nell'ordine dei terabyte al giorno o petabyte all'anno. La maggior parte delle organizzazioni fatica a raccogliere, conservare e analizzare i dati degli endpoint a causa dei costi e delle complessità legati a volumi di dati così ingenti. Ma non è detto che debba essere per forza così.

In questa serie di blog in due parti, spiegheremo come operazionalizzare petabyte di dati di endpoint con Databricks, per migliorare la sicurezza attraverso l'analisi avanzata in modo economico. La Parte 1 (questo blog) riguarda l'architettura di raccolta dei dati e l'integrazione con un SIEM (Splunk). Al termine di questo blog, avendo a disposizione i notebook, si potranno utilizzare i dati per l'analisi. La Parte 2 tratterà specifici casi di utilizzo e spiegherà come creare modelli ML e arricchimenti e analisi automatizzati. Al termine della Parte 2 si potranno implementare i notebook per rilevare ed esaminare minacce utilizzando i dati degli endpoint.

Utilizzeremo i registri di CrowdStrike Falcon per il nostro esempio. Per accedere ai registri di Falcon, si può usare Falcon Data Replicator (FDR) per trasferire dati di eventi grezzi dalla piattaforma di CrowdStrike a storage in cloud come Amazon S3. Questi dati possono essere acquisiti, trasformati, analizzati e conservati con la Databricks Lakehouse Platform insieme al resto della telemetria di sicurezza.

I clienti possono acquisire dati di CrowdStrike Falcon, applicare rilevamenti in tempo reale basati su Python, effettuare ricerche sui dati storici con Databricks SQL e interrogazioni con strumenti SIEM come Splunk con Databricks Add-on for Splunk.

La sfida dell'operazionalizzazione dei dati di CrowdStrike

Nonostante i dati di CrowdStrike Falcon offrano dettagli completi sulla registrazione degli eventi, è comunque arduo acquisire, elaborare e operazionalizzare volumi grandi e complessi di dati sulla sicurezza informatica in tempo quasi reale e in modo economico. Ecco alcune delle problematiche note:

- **Acquisizione di dati in tempo reale su larga scala:** È difficile tenere traccia di file di dati grezzi elaborati e non elaborati, scritti da FDR su storage in cloud in tempo quasi reale.
- **Trasformazioni complesse:** Il formato dei dati è semi-strutturato. Ogni linea di ogni file di registro contiene centinaia di tipi di payload diversi e la struttura dei dati dell'evento può cambiare nel tempo.
- **Governance dei dati:** Questo tipo di dati può essere sensibile e l'accesso deve essere limitato solo agli utenti che ne hanno bisogno.

databricks

- **Analisi di sicurezza semplificata a 360 gradi:** Sono necessari strumenti scalabili per svolgere attività di ingegneria dei dati, ML e analisi su questi set di dati caratterizzati da grandi volumi e alta velocità di movimento.
- **Collaborazione:** Una collaborazione efficace consente di sfruttare le competenze di ingegneri dei dati, analisti di sicurezza informatica e ingegneri ML. Una piattaforma collaborativa aumenta quindi l'efficienza dei carichi di lavoro di analisi e risposta nell'ambito della sicurezza informatica.

Di conseguenza, gli esperti di sicurezza in tutte le imprese si trovano in una situazione difficile, faticando a gestire l'efficienza economica e operativa. Devono scegliere se vincolarsi a sistemi proprietari molto costosi oppure compiere un lavoro enorme per costruire propri strumenti di sicurezza degli endpoint, affrontando problemi di scalabilità e prestazioni.

Lakehouse per la sicurezza informatica su Databricks

Databricks offre a team di sicurezza e data scientist una nuova prospettiva per l'esecuzione di attività in modo efficiente ed efficace, oltre a un set di strumenti per affrontare le problematiche crescenti di Big Data e minacce evolute.

Lakehouse, un'architettura aperta che combina i migliori elementi di data lake e data warehouse, semplifica la costruzione di una pipeline multi-hop di ingegneria dei dati che struttura progressivamente i dati stessi. Il vantaggio di un'architettura multi-hop è che gli ingegneri dei dati possono costruire una pipeline che parte dai dati grezzi come un'unica fonte di dati da cui deriva tutto il resto. I dati grezzi semi-strutturati di CrowdStrike possono essere conservati per anni e le varie trasformazioni e aggregazioni possono essere effettuate in modalità di streaming per raffinare i dati e applicare una struttura contestualizzata, per analizzare e rilevare rischi di sicurezza in diversi scenari.

- **Acquisizione dei dati:** Auto Loader (AWS | Azure | GCP) aiuta a leggere immediatamente i dati non appena CrowdStrike FDR scrive un nuovo file nello storage dei dati grezzi. Questo sistema sfrutta servizi di notifica in cloud per elaborare progressivamente nuovi file man mano che arrivano sul cloud. Auto Loader, inoltre, configura e "ascolta" il servizio di notifica per ricevere nuovi file e può essere dimensionato per gestire milioni di file al secondo.
- **Elaborazione unificata in streaming e batch:** Delta Lake è un approccio aperto per portare la gestione e la governance dei dati nei data lake, sfruttando la potenza di calcolo distribuita di Apache Spark™ per volumi enormi di dati e metadati. Databricks Delta Engine è un motore altamente ottimizzato in grado di elaborare milioni di record al secondo.
- **Governance dei dati:** Con Databricks Table Access Control (AWS | Azure | GCP), gli amministratori possono offrire diversi livelli di accesso alle tabelle Delta tables in base alla funzione di ciascun utente.

- **Strumenti di analisi di sicurezza:** Databricks SQL aiuta a creare un dashboard interattivo con avvisi automatici quando vengono rilevati schemi (pattern) insoliti.
  Allo stesso modo si può facilmente integrare con strumenti BI molto diffusi come Tableau, Microsoft Power BI e Looker.
- **Collaborazione su notebook di Databricks:** I notebook collaborativi di Databricks consentono ai team di sicurezza di collaborare in tempo reale. Più utenti possono eseguire query (interrogazioni) in diversi linguaggi, condividere visualizzazioni e fare commenti nello stesso spazio di lavoro, per portare avanti le investigazioni senza interruzioni.

Architettura Lakehouse per dati di CrowdStrike Falcon

Raccomandiamo la seguente architettura lakehouse per i carichi di lavoro di sicurezza informatica, come i dati di CrowdStrike Falcon. Auto Loader and Delta Lake semplificano il processo di lettura dei dati grezzi dallo storage in cloud e scrittura in una tabella Delta con costi ridotti e un minimo lavoro di DevOps.

In questa architettura, i dati semi-strutturati di CrowdStrike vengono caricati sullo storage in cloud del cliente nella zona di "atterraggio". Successivamente Auto Loader utilizza i servizi di notifica in cloud per attivare automaticamente l'elaborazione e l'acquisizione di nuovi file nelle tabelle Bronze del cliente, che fungeranno da unica fonte di dati per tutti i lavori a valle. Auto Loader traccerà i file elaborati e non utilizzando checkpoint per evitare elaborazioni di dati duplicate.

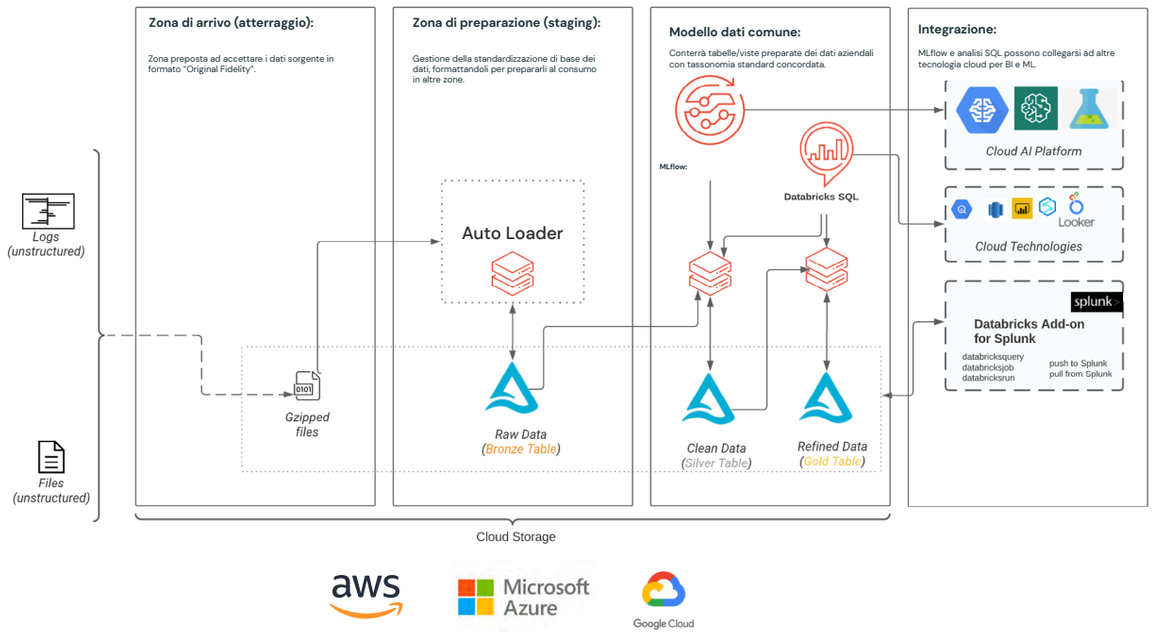

Figura 1
Architettura lakehouse per dati di CrowdStrike Falcon

Passando dallo stadio Bronze al Silver, verrà aggiunto uno schema per dare struttura ai dati. Leggendo i dati da un'unica fonte, siamo in grado di processare tutti i diversi tipi di eventi e applicare lo schema corretto quando i dati vengono scritti nelle rispettive tabelle. La possibilità di applicare schemi al livello Silver offre una solida base per costruire carichi di lavoro di ML e analisi.

La fase Gold, che aggrega i dati per velocizzare le query e le prestazioni in dashboard e strumenti BI, è facoltativa, a seconda del caso di utilizzo e dei volumi di dati. Si possono impostare avvisi che scattano nel momento in cui vengono osservate tendenze impreviste.

Un'altra funzione opzionale è il Databricks Add-on for Splunk, che consente ai team di sicurezza di sfruttare il modello economico di Databricks e la potenza dell'IA senza rinunciare alle comodità di Splunk. I clienti possono eseguire query mirate su Databricks da un dashboard oppure dalla barra di ricerca di Splunk con il modulo add-on. Gli utenti possono anche lanciare notebook o lavori in Databricks attraverso un dashboard di Splunk o in risposta a una ricerca di Splunk. L'integrazione con Databricks è bidirezionale, consentendo quindi di riassumere i dati rumorosi o effettuare verifiche in Databricks che vengono visualizzate in Splunk Enterprise Security. I clienti possono anche effettuare ricerche Splunk da un notebook di Databricks per evitare la duplicazione dei dati.

L'integrazione fra Splunk e Databricks consente ai clienti di ridurre i costi, aumentare il numero di sorgenti di dati analizzate e ottenere i risultati di un motore di analisi più robusto; tutto questo senza cambiare gli strumenti di lavoro quotidiano dei loro addetti.

Analisi del codice

Poiché Auto Loader astrae la parte più complessa dell'acquisizione di dati da file, è possibile creare una pipeline di acquisizione dai dati grezzi a Bronze con poche linee di codice. Riportiamo di seguito un esempio di codice Scala per una pipeline di acquisizione Delta. I record di eventi CrowdStrike Falcon hanno un solo nome di campo comune: "event_simpleName".

```
val crowdstrikeStream = spark.readStream
  .format("cloudFiles")
  .option("cloudFiles.format", "text")    // text file doesn't need schema
  .option("cloudFiles.region", "us-west-2")
  .option("cloudFiles.useNotifications", "true")
  .load(rawDataSource)
  .withColumn("load_timestamp", current_timestamp())
  .withColumn("load_date", to_date($"load_timestamp"))
  .withColumn("eventType", from_json($"value", "struct", Map.empty[String, String]))
.selectExpr("eventType.event_simpleName","load_date","load_timestamp", "value" )
  .writeStream
  .format("delta")
  .option("checkpointLocation", checkPointLocation)
  .table("demo_bronze.crowdstrike")
```

Al livello raw-to-Bronze viene estratto dai dati grezzi solo il nome dell'evento. Aggiungendo un timbro temporale di caricamento e colonne con le date, gli utenti memorizzano i dati grezzi nella tabella Bronze. La tabella Bronze è suddivisa per nome dell'evento e data di caricamento, in modo da migliorare le prestazioni dei lavori Bronze-to-Silver, soprattutto quando l'interesse è

circoscritto solo ad alcune date degli eventi. Successivamente, un'attività di streaming Bronze-to-Silver legge gli eventi da una tabella Bronze, applica uno schema e scrive in centinaia di tabelle di eventi in base al nome dell'evento. Riportiamo di seguito un esempio di codice Scala:

```
spark
  .readStream
  .option("ignoreChanges", "true")
  .option("maxBytesPerTrigger", "2g")
  .option("maxFilesPerTrigger", "64")
  .format("delta")
  .load(bronzeTableLocation)
  .filter($"event_simpleName" === "event_name")
  .withColumn("event", from_json($"value", schema_of_json(sampleJson)) )
  .select($"event.*", $"load_timestamp", $"load_date")
  .withColumn("silver_timestamp", current_timestamp())
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("mergeSchema", "true")
  .option("checkpointLocation", checkPoint)
  .option("path", tableLocation)
  .start()
```

Ogni schema di evento può essere memorizzato in un registro di schemi o in una tabella Delta nel caso in cui debba essere condiviso fra più servizi data-driven. Da notare che il codice precedente utilizza una stringa campione JSON letta dalla tabella Bronze e che lo schema viene dedotto da JSON utilizzando schema_of_json(). Poi la stringa JSON viene convertita in una struttura utilizzando from_json(). La struttura viene poi appiattita, aggiungendo una colonna con il timbro temporale. Questi passaggi aggiungono a un DataFrame tutte le colonne necessarie affinché possa essere abbinato a una tabella di eventi. Infine, scriviamo questi dati strutturati in una tabella di eventi con la modalità "append".

È possibile anche diffondere gli eventi su più tabelle con un unico stream con foreachBatch, definendo una funzione che gestirà microbatch. Con foreachBatch() si possono riutilizzare sorgenti di dati in batch esistenti per filtraggio e scrittura in diverse tabelle. Tuttavia, foreachBatch() offre solo garanzie di scrittura del tipo "almeno una volta". Serve quindi un intervento manuale per applicare una semantica del tipo "esattamente una volta".

In questa fase i dati strutturati possono essere interrogati con qualsiasi linguaggio supportato dai notebook e dai lavori di Databricks: Python, R, Scala e SQL. I dati nella fase Silver sono comodi da utilizzare per ML e analisi di attacchi informatici.

La pipeline di streaming successiva è Silver-to-Gold. In questa fase è possibile aggregare di dati per dashboard e avvisi. Nella seconda parte di questa serie di blog approfondiremo come costruire dashboard utilizzando Databricks SQL.

Prossimamente

Sul blog sono in arrivo nuovi post che valorizzano ulteriormente questo caso di utilizzo applicando ML e utilizzando Databricks SQL.

Abbiamo predisposto alcuni notebook che possono essere utilizzati nell'implementazione di Databricks. Ogni sezione dei notebook è corredata di commenti. Siamo a disposizione per qualsiasi informazione all'indirizzo cybersecurity@databricks.com. Attendiamo domande e suggerimenti per rendere il notebook ancora più facile da capire e utilizzare.

Comincia a sperimentare con questi notebook Databricks gratuiti.

databricks

PAR. 2.3

Sfruttare la potenza dei dati sanitari con una moderna data lakehouse

di MICHAEL ORTEGA, MICHAEL SANKY e AMIR KERMANY

19 luglio 2021

Come risolvere le problematiche di data warehouse e data lake nei settori di sanità e bioscienze

Ogni paziente genera circa 80 megabyte di dati medici ogni anno. Moltiplicando questo dato per migliaia di pazienti e per gli anni di vita, si arriva a petabyte di dati che contengono informazioni preziose sui pazienti. Accedere a queste informazioni può aiutare a ottimizzare le attività cliniche, accelerare la ricerca e lo sviluppo di farmaci e migliorare la salute dei pazienti. Ma, come prima cosa, i dati devono essere preparati per l'analisi e l'IA a valle. Purtroppo, la maggior parte delle organizzazioni nel mondo della sanità e delle bioscienze dedica una quantità di tempo spropositata a raccogliere, pulire e strutturare i dati.

Ogni paziente genera oltre 80 megabyte di dati medici ogni anno.

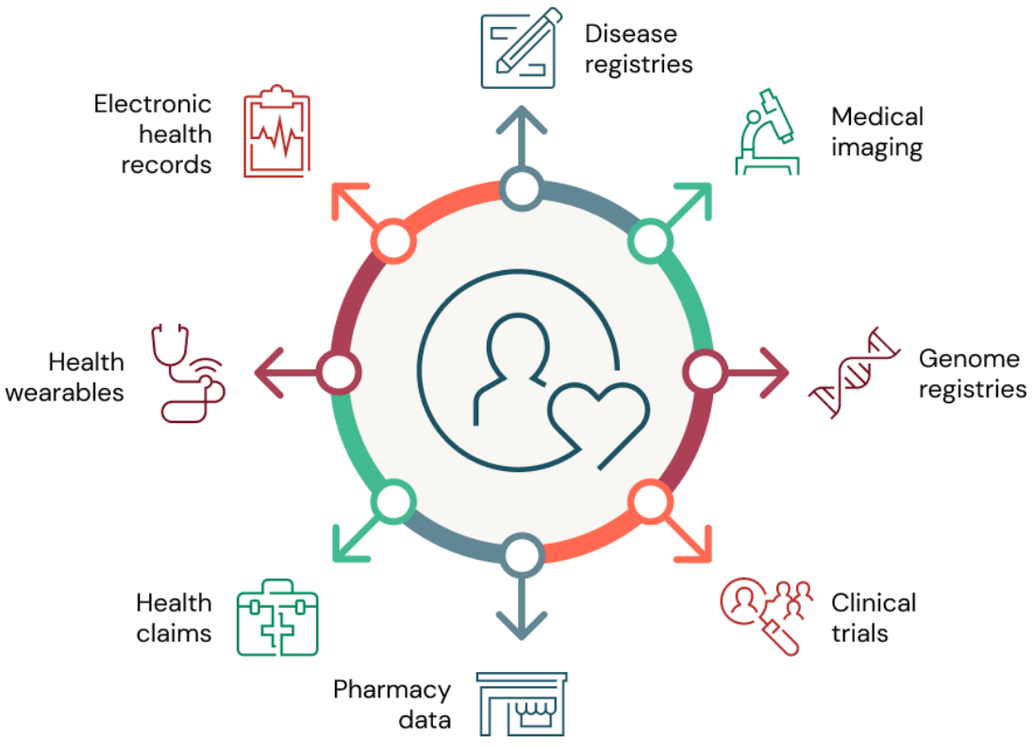

Figura 1
I dati sanitari crescono in modo esponenziale: ogni paziente genera oltre 80 megabyte di dati all'anno

Problematiche di analisi dei dati nella sanità e nelle bioscienze

Esistono molte ragioni per cui la preparazione, l'analisi e l'IA dei dati rappresentano una sfida per chi opera nella sanità, ma molte sono legate agli investimenti già fatti nelle architetture di dati esistenti costruite su data warehouse. Nel settore vediamo attualmente quattro sfide principali:

SFIDA 1: VOLUME

Dimensionare il sistema per gestire dati sanitari in rapida espansione

La genomica è forse l'esempio più eclatante di crescita esplosiva del volume di dati nella sanità. Il sequenziamento del primo genoma è costato oltre un miliardo di dollari. Dati i costi proibitivi, gli sforzi iniziali (e molti ancora oggi) sono stati focalizzati sulla genotipizzazione, un processo che cerca varianti specifiche in una frazione piccolissima del genoma di una persona, tipicamente dello 0,1%. L'evoluzione di questo processo è il sequenziamento dell'intero esoma (WES), che coinvolge le parti del genoma che codificano le proteine, ma rappresenta comunque meno del 2% dell'intero genoma. Ora le aziende offrono direttamente ai consumatori test per il sequenziamento dell'intero genoma (WGS) a meno di 300 dollari per 30 WGS. Parlando di intere popolazioni, la Biobank del Regno Unito ha rilasciato quest'anno oltre 200.000 genomi interi a scopo di ricerca. E non c'è solo la genomica. TAC, risonanze, ecografie, radiografie, dispositivi indossabili e cartelle cliniche elettroniche crescono a ritmi altrettanto sostenuti.

Scalabilità è la parola del momento per progetti come l'analisi della salute della popolazione e la scoperta di farmaci. Purtroppo, molte architetture esistenti sono costruite on-premise e progettate in base alla capacità massima richiesta. Questo approccio si traduce in potenza di calcolo inutilizzata (e quindi spreco di denaro) nei periodi di minore utilizzo, senza avere la possibilità di crescere quando serve maggiore capacità.

SFIDA 2: VARIETÀ

Analizzare dati sanitari variegati

Le organizzazioni nel settore della sanità e delle bioscienze hanno a che fare con una grandissima varietà di dati, ciascuno dei quali ha le proprie sfumature. Tutti sanno che oltre l'80% dei dati non è strutturato, ma molti continuano a puntare su data warehouse progettati per dati strutturati e analisi SQL tradizionale. I dati non strutturati comprendono le immagini, fondamentali per diagnosticare e misurare l'avanzamento di patologie in ambito oncologico, immunologico e neurologico (che sono le aree di costo con i tassi di crescita maggiori) e i testi descrittivi nelle note cliniche, essenziali per capire l'anamnesi sanitaria e sociale del paziente. Ignorare questi tipi di dati, o accantonarli, non è possibile.

A complicare ulteriormente le cose, l'ecosistema della sanità diventa sempre più interconnesso, costringendo tutti i soggetti a gestire nuovi tipi di dati. Ad esempio, i fornitori di servizi hanno bisogno dei dati delle denunce per gestire e assegnare contratti di condivisione del rischio, mentre i pagatori hanno bisogno di dati clinici a supporto di processi quali autorizzazioni preliminari o per effettuare valutazioni della qualità. Queste organizzazioni sono spesso sprovviste di architetture e piattaforme idonee per gestire questi nuovi tipi di dati.

Alcune hanno investito nei data lake per supportare dati non strutturati e analisi avanzata, ma questa scelta pone nuovi problemi. In questo contesto, i team devono gestire due sistemi (data warehouse e data lake) nei quali i dati vengono copiati mediante strumenti isolati, con inevitabili problemi di qualità e gestione dei dati.

SFIDA 3: VELOCITÀ

Elaborare dati in streaming per ricavare informazioni approfondite sui pazienti in tempo reale

In molte situazioni, la sanità è questione di vita o di morte. Le condizioni possono cambiare velocemente e l'elaborazione di dati in batch (anche se effettuata quotidianamente) spesso non è sufficiente. L'accesso a informazioni aggiornate all'ultimo secondo è fondamentale per il buon esito di cure e interventi. Per salvare vite, ospedali e sistemi sanitari nazionali usano dati in streaming per qualsiasi attività, dalla prevenzione della sepsi alle proiezioni in tempo reale sul fabbisogno di posti in terapia intensiva.

Inoltre, la velocità dei dati è un elemento chiave nella rivoluzione digitale della sanità. Ogni individuo ha accesso a una quantità di informazioni senza precedenti e può intervenire sulle cure in tempo reale. Ad esempio, i dispositivi indossabili, come i misuratori di glicemia di Livongo, trasmettono dati in tempo reale ad app mobili che forniscono raccomandazioni comportamentali personalizzate.

Nonostante questi primi successi, la maggior parte delle organizzazioni non ha progettato l'architettura dei dati per gestire la velocità dei dati in streaming. L'innovazione è ostacolata da problemi e difficoltà di integrazione dei dati in tempo reale con i dati storici.

SFIDA 4: VERIDICITÀ

Creare fiducia nei dati e nell'IA per il settore sanitario

L'ultimo aspetto, non meno importante, è che gli standard clinici e normativi richiedono la massima precisione dei dati in ambito sanitario. Le organizzazioni del settore devono infatti rispettare requisiti elevati di conformità in materia di sanità pubblica. La democratizzazione dei dati all'interno delle organizzazioni deve essere governata.

Inoltre, le organizzazioni hanno bisogno di una buona governance dei modelli quando portano l'intelligenza artificiale (IA) e il machine learning (ML) in un contesto clinico. Purtroppo, la maggior parte delle organizzazioni utilizza piattaforme separate per i flussi di data science scollegati dai data warehouse. Questo genera seri problemi in termini di affidabilità e riproducibilità delle applicazioni basate su IA.

Attingere ai dati sanitari con una lakehouse

L'architettura lakehouse aiuta le organizzazioni di sanità e bioscienze a vincere queste sfide con una moderna architettura dei dati che unisce i costi ridotti, la scalabilità e la flessibilità del data lake in cloud con le prestazioni e la governance di un data warehouse. Con una lakehouse, le organizzazioni possono conservare tutti i tipi di dati e alimentare tutti i tipi di analisi e ML in un ambiente aperto.

Costruzione di una lakehouse per sanità e bioscienze

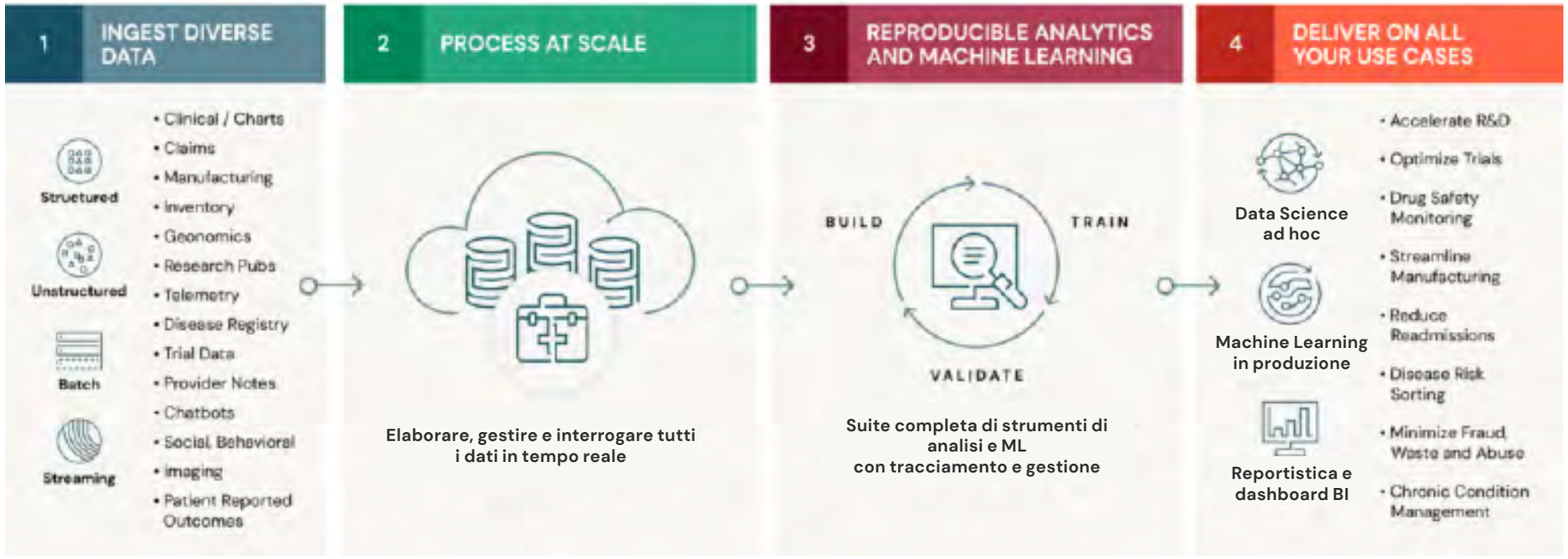

Figura 2
Coprire tutti i casi di utilizzo dell'analisi dei dati nella sanità e nelle bioscienze con una moderna architettura lakehouse

Nello specifico, la lakehouse offre i seguenti vantaggi a chi opera nei settori della sanità e delle bioscienze:

- **Organizzare tutti i dati sanitari su larga scala.** Il cuore della Databricks Lakehouse Platform è Delta Lake, un livello di gestione dei dati open-source che porta flessibilità e prestazioni al data lake. Diversamente da un tradizionale data warehouse, Delta Lake supporta tutti i tipi di dati strutturati e no; inoltre, per semplificare l'acquisizione di dati sanitari, Databricks ha sviluppato connettori per tipi di dati specifici come le cartelle cliniche e la genomica. Questi connettori vengono forniti in pacchetti con modelli di dati standard, sotto forma di Solution Accelerator pronti all'uso. Inoltre, Delta Lake offre ottimizzazioni integrate per la memorizzazione in cache e l'indicizzazione dei dati, per aumentare sensibilmente la velocità di elaborazione dei dati. Grazie a queste funzionalità, i team possono far arrivare tutti i dati grezzi in un unico punto e poi trattarli per creare una vista completa della salute del paziente.
- **Alimentare tutte le analisi e l'IA del paziente.** Centralizzando tutti i dati in una lakehouse, i team possono costruire analisi dei pazienti e modelli predittivi avanzati lavorando direttamente sui dati. Per sfruttare queste capacità, Databricks mette a disposizione spazi di lavoro collaborativi con una suite completa di strumenti di analisi e IA e un ampio ventaglio di linguaggi di programmazione, come SQL, R, Python e Scala. In questo modo, diverse categorie di utenti, fra cui data scientist, ingegneri dei dati e informatici clinici, possono lavorare insieme per analizzare, modellare e visualizzare tutti i dati sanitari.

databricks

- **Informazioni approfondite sui pazienti in tempo reale.** La lakehouse mette a disposizione un'architettura unificata per dati in streaming e in batch. Non è più necessario supportare due diverse architetture, né gestire problemi di affidabilità. Inoltre, implementando l'architettura lakehouse su Databricks, le organizzazioni hanno accesso a una piattaforma nativa per il cloud che si autodimensiona in base al carico di lavoro. Diventa così facile acquisire dati in streaming e fonderli con petabyte di dati storici per ottenere informazioni dettagliate e approfondite in tempo quasi reale per tutta la popolazione.
- **Garantire la qualità e la conformità dei dati.** Per garantire la veridicità dei dati, la lakehouse offre funzionalità assenti nei tradizionali data lake, come applicazione di schemi, revisioni (audit), gestione delle versioni e controllo granulare degli accessi. Un vantaggio importante della lakehouse è la capacità di effettuare sia analisi sia ML sulla stessa sorgente di dati affidabili. Inoltre, Databricks offre funzionalità di tracciamento e gestione dei modelli ML per agevolare i team nel riprodurre i risultati su diversi ambienti e rispettare gli standard di conformità. Tutte queste funzionalità vengono rese disponibili in un ambiente di analisi conforme a HIPAA.

Questa lakehouse è l'architettura migliore per gestire i dati di sanità e bioscienze. Abbinando questa architettura alle funzionalità di Databricks, le organizzazioni possono gestire un'ampia gamma di casi di utilizzo ad alto impatto, dalla scoperta di farmaci ai programmi di gestione delle malattie croniche.

Cominciare a costruire una lakehouse per sanità e bioscienze

Come accennato in precedenza, siamo felici di proporre una serie di Solution Accelerator per aiutare le organizzazioni nei settori di sanità e bioscienze ad avviare la costruzione di una lakehouse per le loro specifiche esigenze. I nostri Solution Accelerator comprendono dati campione, codice precompilato e istruzioni passo-passo all'interno di un notebook di Databricks.

Nuovo Solution Accelerator: Lakehouse for Real-World Evidence. I dati raccolti nel mondo reale forniscono alle aziende farmaceutiche nuove informazioni sulla salute dei pazienti e sull'efficacia dei farmaci al di fuori dei test clinici. Questo acceleratore consente di costruire una Lakehouse for Real-World Evidence su Databricks. Spiegheremo come acquisire dati sanitari elettronici (EHR) per una popolazione di pazienti, strutturarli utilizzando il modello di dati comune OMOP e, infine, eseguire analisi su larga scala per attività quali, ad esempio, esaminare le modalità di prescrizione di farmaci.

Comincia a sperimentare con questi notebook Databricks gratuiti.

Maggiori informazioni su tutte le nostre soluzioni per sanità e bioscienze .

databricks

PAR. 2.4

Puntualità e affidabilità nella trasmissione di report normativi

di **ANTOINE AMEND** e **FAHMID KABIR**

17 settembre 2021

Gestire il rischio e la conformità normativa è un'attività sempre più complessa e costosa. Le modifiche normative sono aumentate del 500% dopo la crisi finanziaria del 2008, facendo salire alle stelle i costi della conformità. Per evitare le sanzioni dovute a non conformità o violazioni degli SLA (le banche hanno raggiunto un nuovo record di 10 miliardi di multe nel 2019 legate all'antiriciclaggio), l'elaborazione dei report deve proseguire anche se i dati sono incompleti. Per altro, anche le registrazioni di dati di qualità scadente vengono sanzionate per "controlli insufficienti". Di conseguenza, molti istituti di servizi finanziari si trovano spesso a combattere con dati di qualità scadente e SLA severi, cercando il giusto equilibrio fra affidabilità dei dati e tempestività di raccolta ed elaborazione.

In questo Solution Accelerator per i report normativi, dimostreremo come le Delta Live Tables possono garantire l'acquisizione e l'elaborazione di dati normativi in tempo reale, per rispettare gli SLA. Combinando Delta Sharing e Delta Live Tables, gli analisti acquisiscono fiducia nella qualità dei dati normativi trasmessi in tempo reale. Questo post illustra i vantaggi dell'architettura lakehouse nel combinare i modelli di dati del settore dei servizi finanziari con la flessibilità del cloud computing, per implementare standard elevati di governance con bassi costi di sviluppo. Spiegheremo ora che cos'è un modello di dati FIRE e come si possono integrare le Delta Live Tables per costruire pipeline di dati robuste.

Modello di dati FIRE

Lo standard di dati Financial Regulatory (FIRE) definisce una specifica comune per la trasmissione di dati granulari fra diversi sistemi normativi nel settore finanziario. Per dati normativi si intendono dati soggetti a disposizioni, requisiti e calcoli normativi, utilizzati a scopo di politiche, monitoraggio e supervisione. Lo standard di dati FIRE è supportato dalla Commissione Europea, dall'Open Data Institute e dall'Open Data Incubator FIRE data standard for Europe attraverso il programma di finanziamenti Horizon 2020. Nell'ambito di questa soluzione abbiamo fornito un modulo PySpark che può interpretare i modelli di dati FIRE nelle pipeline operative di Apache Spark™.

databricks

Delta Live Tables

Databricks ha annunciato recentemente un nuovo prodotto per l'orchestrazione delle pipeline di dati, Delta Live Tables, che semplifica la costruzione e la gestione di pipeline di dati affidabili su scala aziendale. Grazie alla possibilità di valutare molteplici aspettative, scartare o monitorare i record non validi in tempo reale, i vantaggi derivanti dall'integrazione del modello di dati FIRE su Delta Live Tables sono evidenti. Come mostrato nell'architettura raffigurata sotto, Delta Live Tables **acquisisce** i dati normativi granulari che arrivano nello storage in cloud, **schematizza** i contenuti e **convalida** la conformità dei record alla specifica di dati FIRE. Nei prossimi paragrafi dimostreremo l'uso di Delta Sharing per scambiare informazioni granulari fra sistemi normativi in modo sicuro, scalabile e trasparente.

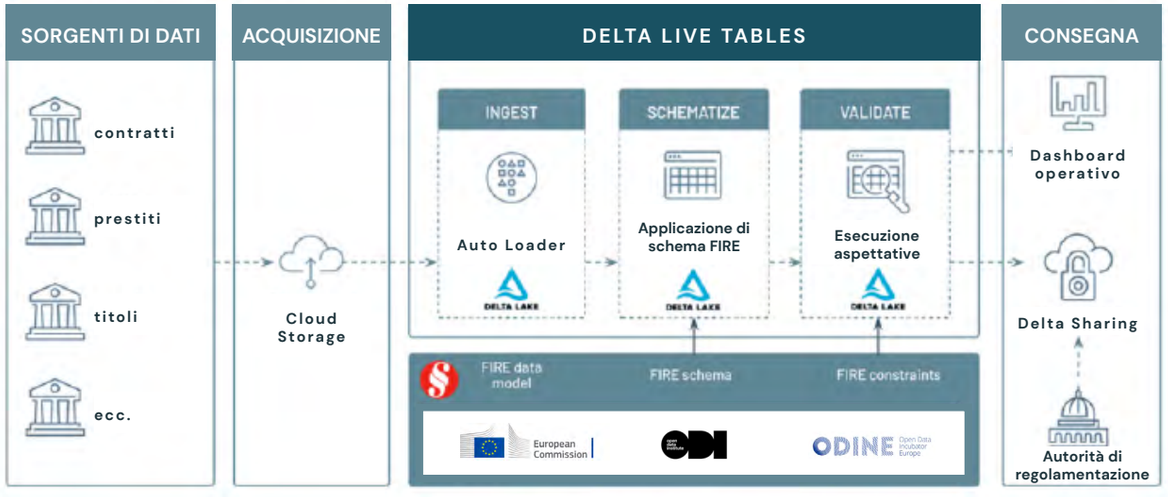

Figura 1

Applicazione di schemi

Nonostante alcuni formati di dati possano "sembrare" strutturati (ad es. i file JSON), applicare uno schema non è solo una buona pratica ingegneristica. In contesti aziendali, soprattutto nell'ambito della conformità normativa, l'applicazione di schemi garantisce la verifica dei campi attesi, lo scarto dei campi inattesi e la piena valutazione delle tipologie di dati (ad es. un dato deve essere trattato come oggetto e non come stringa). Inoltre, l'applicazione di uno schema consente di verificare eventuali "derive" dei dati nel sistema. Utilizzando il modulo FIRE PySpark, richiamiamo programmaticamente lo schema Spark richiesto per elaborare una determinata entità FIRE (entità collaterale nell'esempio proposto) che applicheremo su un flusso di record grezzi.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_schema = fire_model.schema
```

Nell'esempio riportato sotto, applichiamo lo schema a file CSV in ingresso. Arricchendo questo processo con l'annotazione @dlt, definiamo il punto di entrata nella nostra Delta Live Table, leggendo i file CSV grezzi da una directory montata e scrivendo record schematizzati in un livello Bronze.

```
@dlt.create_table()
def collateral_bronze():
  return (
    spark
      .readStream
      .option("maxFilesPerTrigger", "1")
      .option("badRecordsPath", "/path/to/invalid/collateral")
      .format("csv")
      .schema(fire_schema)
      .load("/path/to/raw/collateral")
```

Valutazione delle aspettative

Applicare uno schema è una cosa, imporre i relativi vincoli è un'altra. Data la definizione dello schema di un'entità FIRE (vedi l'esempio della definizione dello schema collaterale), possiamo capire se un campo è richiesto o meno. Dato un oggetto di enumerazione, verifichiamo che i suoi valori siano coerenti (ad es. codice valuta). Oltre ai vincoli tecnici imposti dallo schema, il modello FIRE riporta anche le aspettative dell'azienda, come minimo, massimo, monetarie e maxItems. Tutti questi vincoli tecnici e operativi verranno recuperati programmaticamente dal modello di dati FIRE e interpretati come una serie di espressioni Spark SQL.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_constraints = fire_model.constraints
```

Con Delta Live Tables, gli utenti possono valutare molteplici aspettative contemporaneamente, con la possibilità di scartare record non validi, monitorare semplicemente la qualità dei dati o abortire un'intera pipeline. Nel nostro scenario specifico, vogliamo scartare i record che non soddisfano una qualsiasi delle nostre aspettative e che metteremo poi in una tabella di quarantena, come indicato nei notebook forniti in questo blog.

```
@dlt.create_table()
@dlt.expect_all_or_drop(fire_constraints)
def collateral_silver():
  return dlt.read_stream("collateral_bronze")
```

Con poche linee di codice ci siamo assicurati che la nostra tabella Silver sia corretta sia sintatticamente (schema valido) sia semanticamente (aspettative valide). Come mostrato di seguito, i responsabili della conformità hanno piena visibilità sul numero di record elaborati in tempo reale. In questo esempio specifico, abbiamo garantito che la nostra entità collaterale sia esattamente completa al 92,2% (la quarantena gestisce il restante 7,8%).

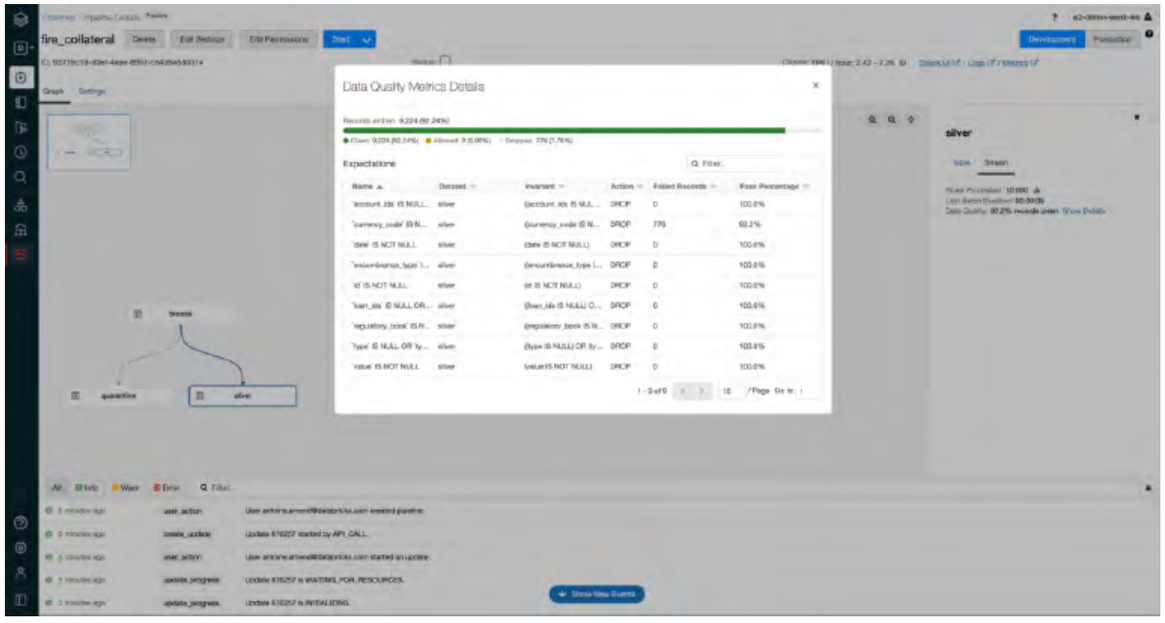

Figura 2

Stoccaggio di dati operativi

Oltre ai dati memorizzati come file Delta, Delta Live Tables conserva anche le metriche operative come formato "delta" sotto sistema/eventi. Seguiamo un modello standard di architettura lakehouse "abbonandoci" a nuove metriche operative con Auto Loader, elaborando gli eventi di sistema man mano che le nuove metriche si presentano, in batch o in tempo reale. Grazie al registro delle transazioni di Delta Lake che tiene traccia di ogni aggiornamento dei dati, le organizzazioni possono accedere a nuove metriche senza dover costruire e mantenere processi di checkpoint propri.

```
input_stream = spark \
    .readStream \
    .format("delta") \
    .load("/path/to/pipeline/system/events")

output_stream = extract_metrics(input_stream)

output_stream \
    .writeStream \
    .format("delta") \
    .option("checkpointLocation", "/path/to/checkpoint") \
    .table(metrics_table)
```

Avendo tutte le metriche disponibili centralmente in un archivio operativo, gli analisti possono utilizzare Databricks SQL per creare semplici funzionalità di dashboarding e meccanismi di avviso più complessi per individuare problemi di qualità dei dati in tempo reale.

L'immutabilità del formato Delta Lake, abbinata alla trasparenza sulla qualità dei dati offerta da Delta Live Tables, consente agli istituti finanziari di "viaggiare nel tempo" fra diverse versioni dei dati che rispettano i volumi e la qualità richiesti dalla conformità normativa. Nel nostro esempio specifico, riproporre il 7,8% dei record non validi memorizzati nella quarantena risulterà in una versione Delta differente allegata alla nostra tabella Silver, versione che può essere condivisa fra gli enti normativi.

```
DESCRIBE HISTORY fire.collateral_silver
```

	entity	expectation_name	expectation_value
1	adjustment	[id] is mandatory	`id` IS NOT NULL
2	adjustment	[date] is mandatory	`date` IS NOT NULL
3	adjustment	[col] is mandatory	`col` IS NOT NULL
4	adjustment	[contribution_amount] is mandatory	`contribution_amount` IS NOT NULL
5	adjustment	[currency_code] is mandatory	`currency_code` IS NOT NULL
6	adjustment	[currency_code] not allowed value	(`currency_code` IS NULL) OR (`currency_code` IN ('AED', 'AFN', 'ALL', 'AMD', 'ANG', 'AOA', 'ARS', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BIF', 'BMD', 'BND', 'BOB', 'BOV', 'BRL', 'BSD', 'BTN', 'BWP', 'BYN', 'BZD', 'CAD', 'CDF', 'CHE', 'CHF', 'CHW', 'CLF', 'CLP', 'CNY', 'COP', 'COU', 'CRC', 'CUC', 'CUP', 'CVE', 'CZK', 'DJF', 'DKK', 'DOP', 'DZD', 'EGP', 'ERN', 'ETB', 'EUR', 'FJD', 'FKP', 'GBP', 'GEL', 'GHS', 'GIP', 'GMD', 'GNF', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HTG', 'HUF', 'I...

Figura 3

Trasmissione di dati normativi

Avendo piena fiducia nella qualità e nel volume dei dati, gli istituti finanziari possono scambiare informazioni in sicurezza fra diversi sistemi normativi con Delta Sharing, un protocollo aperto per lo scambio di dati fra imprese. Non vincolando gli utilizzatori finali alla stessa piattaforma, né facendo affidamento su pipeline ETL complesse per consumare i dati (accedendo ai file di dati attraverso un server SFTP, ad esempio), la natura open-source di Delta Lake consente ai consumatori di accedere a dati schematizzati nativamente da Python, Spark o direttamente attraverso dashboard MI/BI (come Tableau o Power BI).

Anche se potremmo condividere la nostra tabella Silver così com'è, potrebbero esistere regole aziendali che consentono la condivisione di dati normativi solo quando viene rispettata una soglia predefinita di qualità dei dati. In questo esempio, cloniamo la nostra tabella Silver a una versione differente e in una posizione specifica isolata dalle nostre reti interne e accessibile agli utenti finali (detta "zona demilitarizzata", o DMZ).

```
from delta.tables import *

deltaTable = DeltaTable.forName(spark, "fire.collateral_silver")
deltaTable.cloneAtVersion(
  approved_version,
  dmz_path,
  isShallow=False,
  replace=True
)

spark.sql(
  "CREATE TABLE fire.colleral_gold USING DELTA LOCATION '{}'"
    .format(dmz_path)
)
```

Nonostante la soluzione open-source Delta Sharing si appoggi su un server di condivisione per gestire i permessi, Databricks sfrutta Unity Catalog per centralizzare e applicare politiche di controllo degli accessi, fornire agli utenti funzionalità complete di registro delle revisioni e semplificare la gestione degli accessi attraverso la sua interfaccia SQL. Nell'esempio riportato di seguito, creiamo una condivisione (SHARE) che include le nostre tabelle normative e un destinatario (RECIPIENT) con cui condividere i dati.

```
-- DEFINE OUR SHARING STRATEGY
CREATE SHARE regulatory_reports;

ALTER SHARE regulatory_reports ADD TABLE fire.collateral_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.loan_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.security_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.derivative_gold;

-- CREATE RECIPIENTS AND GRANT SELECT ACCESS
CREATE RECIPIENT regulatory_body;

GRANT SELECT ON SHARE regulatory_reports TO RECIPIENT regulatory_body;
```

Qualsiasi autorità di regolamentazione o utente in possesso dei permessi può accedere ai nostri dati sottostanti utilizzando un token di accesso personale scambiato attraverso il processo. Per maggiori informazioni su Delta Sharing, raccomandiamo di visitare la pagina del prodotto e contattare il proprio referente Databricks.

Verificare la conformità

Attraverso questa serie di notebook e lavori Delta Live Tables, abbiamo dimostrato i vantaggi dell'architettura lakehouse per l'acquisizione, l'elaborazione, la convalida e la trasmissione di dati normativi. Nello specifico, abbiamo esaminato l'esigenza delle organizzazioni di garantire omogeneità, integrità e puntualità delle pipeline normative che potrebbero essere facilmente realizzate utilizzando un modello di dati comune (FIRE) abbinato a un motore di orchestrazione flessibile (Delta Live Tables). Con le funzionalità di Delta Sharing, abbiamo dimostrato come gli istituti di servizi finanziari possono portare piena trasparenza e avere fiducia nei dati normativi scambiati fra diversi sistemi, al tempo stesso soddisfacendo le esigenze di reportistica, riducendo i costi operativi e adattandosi a nuovi standard.

Vi invitiamo a familiarizzare con le pipeline di dati FIRE utilizzando i notebook allegati e a visitare il nostro Solution Accelerators Hub per restare aggiornati sulle nostre soluzioni più recenti per i servizi finanziari.

Comincia a sperimentare con questi notebook Databricks gratuiti.

databricks

PAR. 2.5

Soluzioni AML su larga scala con la Databricks Lakehouse Platform

di **SRI GHATTAMANENI, RICARDO PORTILLA** e **ANINDITA MAHAPATRA**

16 luglio 2021

Risolvere le problematiche principali della costruzione di una soluzione contro i crimini finanziari

La conformità all'antiriciclaggio (AML) è indubbiamente uno dei temi in cima all'agenda delle autorità di regolamentazione che sorvegliano gli istituti finanziari in tutto il mondo. L'evoluzione dell'antiriciclaggio, grazie allo sviluppo di tecniche sempre più sofisticate, negli scorsi decenni è andata di pari passo con l'evoluzione dei requisiti normativi atti a contrastare i metodi moderni di riciclaggio del denaro e finanziamenti a organizzazioni terroristiche. Il Bank Secrecy Act del 1970 forniva indicazioni e un quadro normativo agli istituti finanziari per adottare controlli idonei a monitorare le transazioni finanziarie e riferire attività fiscali sospette alle autorità competenti. La legge ha definito un contesto per spiegare agli istituti finanziari come contrastare il riciclaggio di denaro e il terrorismo finanziario.

Perché l'antiriciclaggio è così complesso

Le attività di antiriciclaggio attuali sono molto diverse rispetto al passato. La transizione al banking digitale, con istituzioni finanziarie che elaborano miliardi di transazioni ogni giorno, ha portato a una crescente diffusione del riciclaggio di denaro, nonostante sistemi più rigorosi per il monitoraggio delle transazioni e robuste soluzioni "Know Your Customer" (KYC). In questo blog, condividiamo le nostre esperienze al fianco di clienti nel mondo dei servizi finanziari per costruire soluzioni di antiriciclaggio per aziende sulla piattaforma lakehouse che consentono una supervisione affidabile e forniscono soluzioni innovative scalabili per adattarsi alla realtà delle moderne minacce di riciclaggio online.

Costruire una soluzione AML con la lakehouse

Il fardello operativo di elaborare miliardi di transazioni al giorno deriva dalla necessità di conservare dati provenienti da molte sorgenti e di alimentare soluzioni AML di nuova generazione. Queste soluzioni offrono capacità avanzate di analisi del rischio e reportistica, supportando al tempo stesso l'utilizzo di modelli di machine learning avanzati per ridurre i falsi positivi e migliorare l'efficienza di investigazione a valle. Gli istituti finanziari hanno già adottato misure per risolvere i problemi di infrastruttura e scalabilità, passando da strutture on-premise a sistemi in cloud per migliorare la sicurezza, l'agilità e le economie di scala necessarie per immagazzinare enormi quantità di dati.

databricks

Esiste poi il problema di come dare un senso agli ingenti volumi di dati strutturati e non, raccolti e conservati su sistemi economici di storage a oggetti. Se i fornitori di servizi in cloud offrono un modo economico di stoccare i dati, per interpretare quegli stessi dati ai fini delle attività a valle per la gestione del rischio e la conformità alle norme antiriciclaggio bisogna partire dallo stoccaggio dei dati in formati di alta qualità e alte prestazioni, pensati per il consumo a valle. Databricks Lakehouse Platform fa esattamente questo. Combinando il basso costo di stoccaggio dei data lake con le robuste funzionalità transazionali dei data warehouse, gli istituti finanziari possono veramente costruire una moderna piattaforma AML.

Oltre alle problematiche di stoccaggio dei dati sopra descritte, gli analisti AML devono affrontare alcune sfide specifiche del settore:

- migliorare il time-to-value nell'analisi di dati non strutturati come immagini, testi e link di rete;
- ridurre il carico DevOps per supportare funzionalità ML critiche come risoluzione di entità, visione computerizzata e analisi dei grafi su metadati di entità;
- abbattere i compartimenti stagni introducendo l'ingegneria analitica e un livello di dashboarding su transazioni AML e tabelle arricchite.

Fortunatamente, Databricks aiuta a risolvere questi problemi sfruttando Delta Lake per conservare e combinare dati strutturati e non strutturati, per costruire relazioni fra le entità; inoltre, Databricks Delta Engine offre un accesso efficiente utilizzando il nuovo sistema di calcolo Photon per velocizzare le query BI sulle tabelle. In aggiunta a queste funzionalità, il Machine Learning occupa un ruolo di primo piano nella lakehouse; questo significa che analisti e data scientist non sprecano tempo a sottocampionare o spostare dati per condividere dashboard e restare un passo avanti rispetto ai "cattivi maestri".

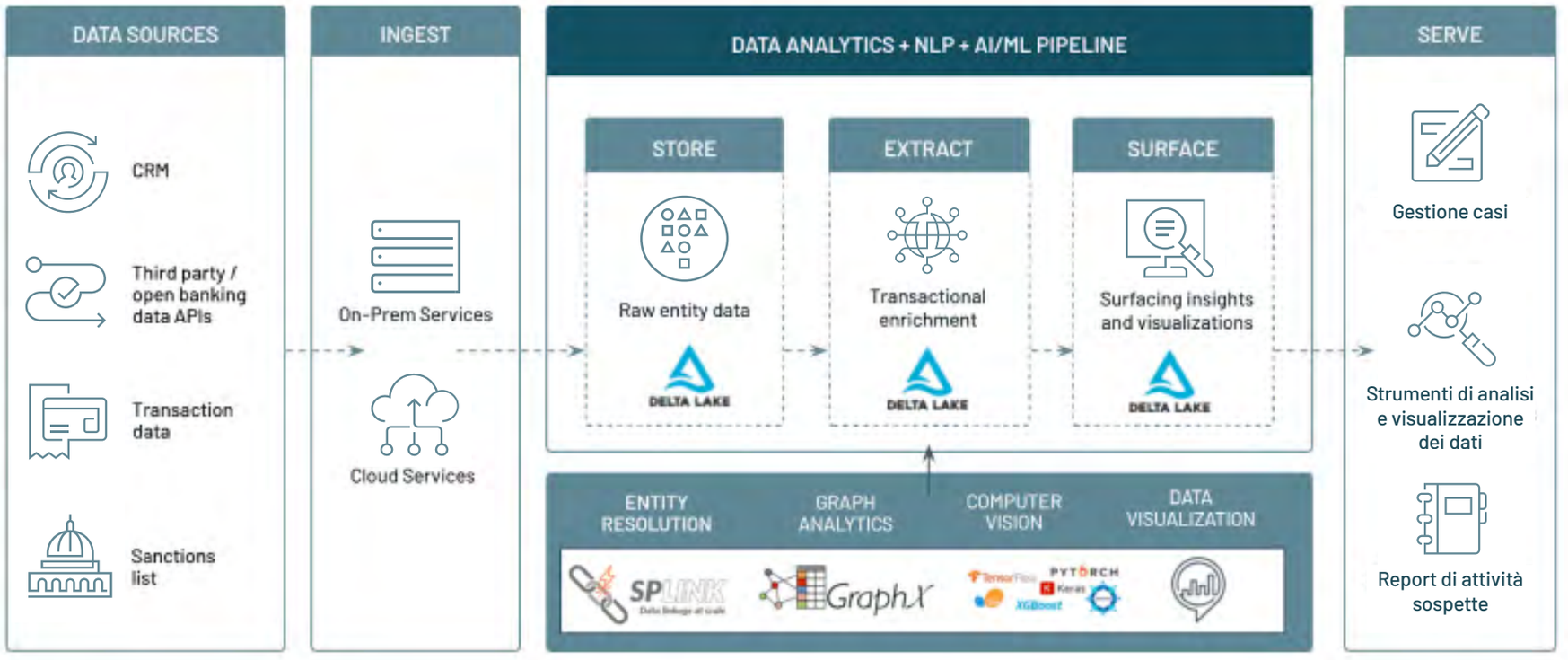

Figura 1

Rilevare attività di antiriciclaggio con funzionalità di grafi

Una delle sorgenti di dati principali utilizzata dagli analisti AML nell'ambito di un caso sono i *dati delle transazioni*. Anche se questi dati sono tabulari e facilmente accessibili con SQL, diventa macchinoso tracciare catene di transazioni che sono profonde tre o più livelli utilizzando query SQL. Per questo motivo è importante avere una suite flessibile di linguaggi e API per esprimere concetti semplici come una rete connessa di individui sospetti che fanno transazioni illegali fra loro. Fortunatamente, questo compito è semplice con GraphFrames, un'API per grafi preinstallata in Databricks Runtime for Machine Learning.

In questa sezione, mostreremo come utilizzare l'analisi dei grafi per rilevare schemi AML come identità sintetica e livelli/strutture. Utilizzeremo un set di dati costituito da transazioni, oltre a entità derivate da transazioni, per rilevare la presenza di questi schemi (pattern) con Apache Spark™, GraphFrames and Delta Lake. I pattern persistenti vengono salvati in Delta Lake in modo che Databricks SQL possa essere applicato alle versioni di questi rilevamenti aggregate nel livello Gold, consentendo agli utenti finali di sfruttare la potenza dell'analisi dei grafi.

Scenario 1 — identità sintetiche

Come detto sopra, l'esistenza di identità sintetiche può essere causa di allarme. Utilizzando l'analisi dei grafi, tutte le entità delle transazioni possono essere analizzate in massa per rilevare un livello di rischio. Nella nostra analisi, questo avviene in tre fasi:

- in base ai dati delle transazioni, estrarre le entità;
- creare connessioni fra le entità in base a indirizzo, numero di telefono o e-mail;
- utilizzare componenti connessi a GraphFrames per stabilire se più entità (identificate da un ID e altri attributi di cui sopra) sono collegate fra loro in uno o più modi.

In base al numero di connessioni (cioè attributi comuni) fra le entità, possiamo assegnare un punteggio di rischio maggiore o minore e creare un avviso per i gruppi con i punteggi più alti. La figura mostra una rappresentazione semplificata di questo concetto.

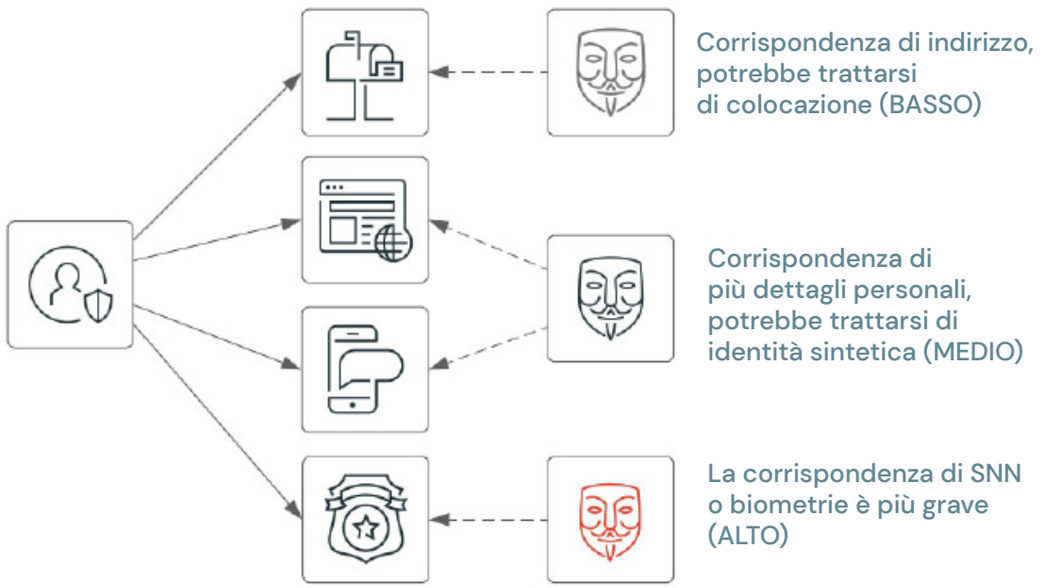

Figura 2

Innanzitutto, creiamo un grafo di identità utilizzando un indirizzo fisico, un indirizzo e-mail e un numero di telefono per collegare fra loro diversi individui se vi sono corrispondenze fra questi attributi.

```
e_identity_sql = '''
select entity_id as src, address as dst from aml.aml_entities_synth  where address is not
null
UNION
select entity_id as src, email as dst from aml.aml_entities_synth  where email_addr is not
null
UNION
select entity_id as src, phone as dst from aml.aml_entities_synth  where phone_number is not
null
'''

from graphframes import *
from pyspark.sql.functions import *
aml_identity_g = GraphFrame(identity_vertices, identity_edges)
result = aml_identity_g.connectedComponents()

result \
 .select("id", "component", 'type') \
 .createOrReplaceTempView("components")
```

Successivamente, eseguiremo query per individuare quando due identità presentano sovrapposizioni nell'identificativo personale e nei punteggi. In base ai risultati di questi componenti del grafo di interrogazione, ci aspetteremmo una coorte composta da un unico attributo coincidente (ad esempio l'indirizzo), che non desta grande preoccupazione. Tuttavia, aumentando il numero di attributi coincidenti, è lecito attendersi di essere avvisati. Come mostrato sotto, possiamo segnalare i casi in cui gli attributi coincidono, consentendo agli analisti SQL di ricavare risultati ogni giorno dalle analisi dei grafi eseguite su tutte le entità.

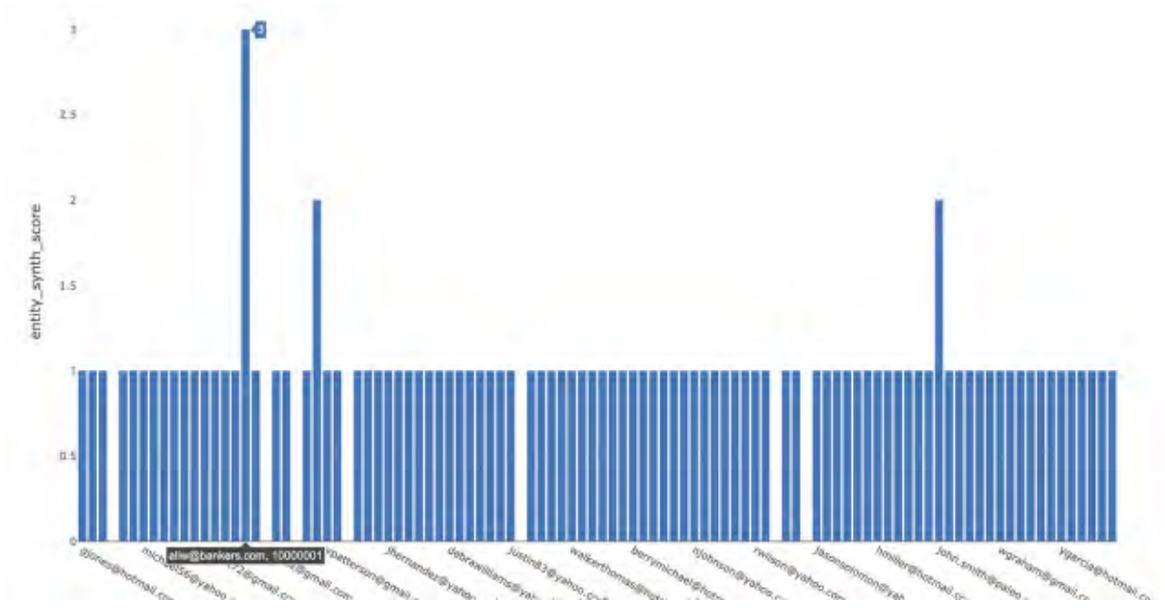

Figura 3

Scenario 2 — strutturazione

Un'altra situazione frequente è la cosiddetta *strutturazione*, che si verifica quando più entità "cospirano" e inviano piccoli pagamenti "sotto soglia" a una serie di banche, che poi girano importi maggiori a un istituto finale (a destra nella figura successiva). In questo scenario, tutte le parti sono rimaste sotto il limite soglia dei 10.000 dollari, che farebbe scattare la segnalazione alle autorità. Non solo questo rilevamento viene effettuato facilmente con l'analisi dei grafi, ma la *tecnica di individuazione di schemi* può essere automatizzata allo scopo di estenderla ad altre permutazioni di reti e individuare altre transazioni sospette con lo stesso metodo.

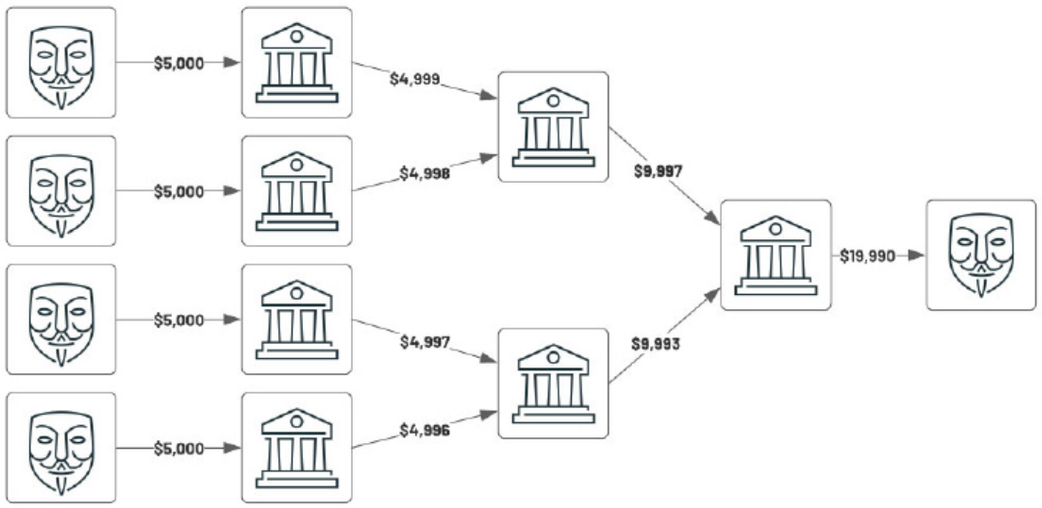

Figura 4

Ora scriveremo il codice base di rilevamento degli schemi per individuare lo scenario sopra descritto con le funzionalità dei grafi. Da notare che, in questo caso, il risultato è un JSON semistrutturato; tutti i tipi di dati, inclusi i tipi non strutturati, sono facilmente accessibili nella lakehouse; salveremo questi risultati particolari per i report SQL.

```
motif = "(a)-[e1]->(b); (b)-[e2]->(c); (c)-[e3]->(d); (e)-[e4]->(f); (f)-[e5]-
>(c); (c)-[e6]->(g)"
struct_scn_1 = aml_entity_g.find(motif)

joined_graphs = struct_scn_1.alias("a") \
 .join(struct_scn_1.alias("b"), col("a.g.id") == col("b.g.id")) \
 .filter(col("a.e6.txn_amount") + col("b.e6.txn_amount") > 10000)
```

Utilizzando la ricerca di schemi, abbiamo estratto pattern interessanti nei quali il denaro passa attraverso quattro diverse entità, restando sotto la soglia dei 10.000 dollari. Ricolleghiamo i metadati del grafo ai set di dati strutturati per generare informazioni approfondite utili a un analista AML per ulteriori indagini.

	top_entity_id	first_entity	second_entity	third_entity	fourth_entity
1	1	Brenda Thomas	Teresa Gibson	Mary Strong	Robert Wilkinson
2	3	Lindsey Barber	Joshua Harris	Mary Strong	Robert Wilkinson
3	5	Bruce White	Kathleen Elliott	Victor Arias	Robert Wilkinson
4	7	Jeffrey Lara	Amy Campbell	Victor Arias	Robert Wilkinson

Figura 5

Scenario 3 — propagazione del punteggio di rischio

Le entità ad alto rischio individuate avranno un impatto (effetto rete) sulla loro cerchia. Pertanto, il punteggio di rischio di tutte le entità con cui interagiscono verrà adeguato per riflettere l'area di influenza. Utilizzando un approccio interattivo, possiamo seguire il flusso delle transazioni a qualsiasi profondità e adeguare i punteggi di rischio di altre entità coinvolte nella rete. Come accennato in precedenza, l'analisi del grafo evita procedure SQL multiple e logiche complesse, che potrebbero incidere negativamente sulle prestazioni per limiti di memoria. L'analisi dei grafi e l'API Pregel sono state concepite esattamente per questo scopo. Sviluppato originariamente da Google, Pregel consente di "propagare" in modo ricorsivo messaggi da qualsiasi vertice ai relativi "vicini", aggiornando lo stato del vertice (nel nostro esempio, il punteggio di rischio) a ogni passaggio. Possiamo rappresentare il nostro approccio dinamico al rischio utilizzando l'API Pregel nel modo seguente.

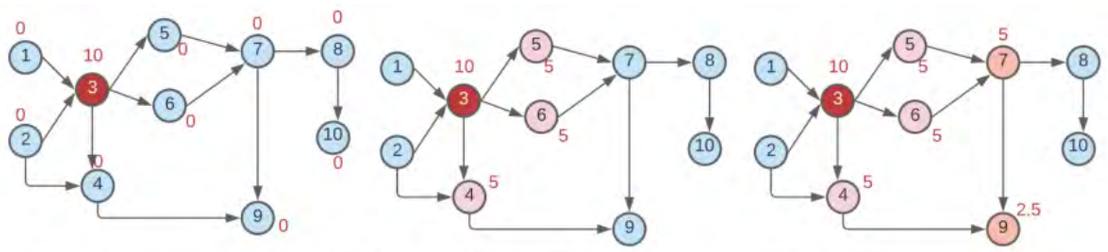

Figura 6

Il diagramma in basso a sinistra mostra lo stato di partenza della rete e due iterazioni successive. Supponiamo di essere partiti con un unico "malfattore", detto "bad actor", (Nodo 3) con un punteggio di rischio pari a 10. Vogliamo penalizzare tutte le persone che effettuano transazioni con quel nodo (Nodi 4, 5 e 6) e ricevono fondi, trasferendo, ad esempio, metà del punteggio di rischio del bad actor, che va a sommarsi al loro punteggio base. Nell'iterazione successiva, tutti i nodi a valle dei Nodi 4, 5 e 6 vedranno un adeguamento del proprio punteggio.

N° nodo	Iterazione n° 0	Iterazione n° 1	Iterazione n° 2
1	0	0	0
2	0	0	0
3	10	10	10
4	0	5	5
5	0	5	5
6	0	5	5
7	0	0	5
8	0	0	0
9	0	0	2,5
10	0	0	0

Utilizzando l'API Pregel di GraphFrame, possiamo fare questo calcolo ed estendere i punteggi modificati ad altre applicazioni a valle.

```
from graphframes.lib import Pregel

ranks = aml_entity_g.pregel \
    .setMaxIter(3) \
    .withVertexColumn(
       "risk_score",
       col("risk"),
       coalesce(Pregel.msg()+ col("risk"),
       col("risk_score"))
    ) \
    .sendMsgToDst(Pregel.src("risk_score")/2 )  \
    .aggMsgs(sum(Pregel.msg())) \
    .run()
```

Indirizzo coincidente

Un pattern sul quale ci vogliamo soffermare brevemente è la coincidenza di un indirizzo fra il testo e le immagini di Street View. Spesso l'analista AML ha bisogno di verificare la legittimità degli indirizzi collegati a entità censite. L'indirizzo corrisponde a un edificio commerciale, un'area residenziale o una semplice casella postale? Analizzare le immagini comporta però spesso un processo manuale, lungo e noioso, di reperimento, pulizia e convalida. Un'architettura lakehouse consente di automatizzare gran parte di questa attività utilizzando runtime Python e ML con PyTorch e modelli open-source pre-addestrati. Riportiamo di seguito un esempio di indirizzo valido all'occhio umano. Per automatizzare la convalida, utilizzeremo un modello VGG pre-addestrato per il quale esistono centinaia di oggetti validi che possiamo utilizzare per individuare una residenza.

Figura 7

Utilizzando il codice sottostante, che può essere automatizzato per essere eseguito giornalmente, potremo applicare un'etichetta su tutte le nostre immagini: abbiamo inoltre caricato tutti i riferimenti e le etichette delle immagini in una tabella SQL per semplificare le query. Osserviamo come sia facile, nel codice riportato sotto, interrogare una serie di immagini rispetto agli oggetti che contengono. La capacità di interrogare dati non strutturati con Delta Lake fa risparmiare molto tempo agli analisti e riduce il processo di convalida a pochi minuti, invece di giorni o settimane.

```
from PIL import Image
from matplotlib import cm

img = Image.fromarray(img)
...

vgg = models.vgg16(pretrained=True)
prediction = vgg(img)
prediction = prediction.data.numpy().argmax()
img_and_labels[i] = labels[prediction]
```

Cominciando a riassumere, notiamo che emergono alcune categorie interessanti. Come mostra il grafico riportato sotto, esistono alcune etichette ovvie come *patio*, *casa mobile* e *motoscooter* che è lecito attendersi che vengano rilevate in un indirizzo residenziale. D'altro canto, il modello CV ha etichettato in un'immagine una parabola solare da oggetti circostanti (*Nota: poiché siamo limitati a un modello open-source non addestrato su un set personalizzato di immagini, l'etichetta della parabola solare non è precisa*). Analizzando ulteriormente l'immagine, andiamo nel dettaglio e vediamo subito che i) non c'è veramente una parabola solare, e soprattutto ii) questo indirizzo non è una residenza (immagini affiancate nella Figura 7). Il formato Delta Lake consente di memorizzare un riferimento ai nostri dati non strutturati insieme a un'etichetta, per effettuare facilmente query nella nostra classificazione riportata nel grafico a torta.

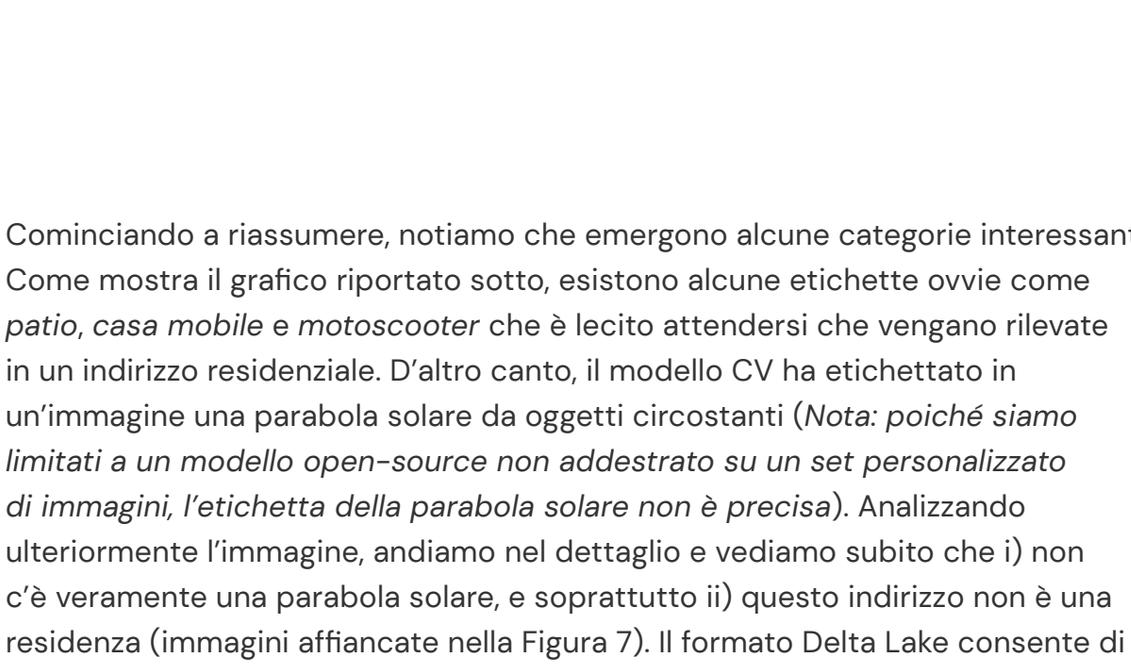

Figura 8

Image Name	Rendered Image	Main Object	Risk Level
img_0.jpg		Patio	Low
img_1.jpg		Solar Dish	High

Figura 9

Risoluzione di entità

L'ultima categoria di problematiche di antiriciclaggio sulla quale ci soffermeremo è la risoluzione di entità. Molte librerie open-source risolvono questo problema; pertanto, per alcune corrispondenze casuali di entità, abbiamo scelto di utilizzare Splink, che effettua la correlazione su larga scala e offre configurazioni per specificare colonne di corrispondenza e regole di blocco.

Nel contesto delle entità derivate dalle nostre transazioni, è semplice inserire le nostre transazioni Delta Lake nel contesto di Splink.

```
settings = {
  "link_type": "dedupe_only",
  "blocking_rules": [
      "l.txn_amount = r.txn_amount",
  ],
  "comparison_columns": [
      {
          "col_name": "rptd_originator_address",
      },
      {
          "col_name": "rptd_originator_name",
      }
  ]
}

from splink import Splink
linker = Splink(settings, df2, spark)
df2_e = linker.get_scored_comparisons()
```

Splink assegna una probabilità di corrispondenza che può essere usata per identificare transazioni nelle quali gli attributi delle entità sono estremamente simili, sollevando un potenziale rischio rispetto a un indirizzo, un nome di entità o un importo di transazione segnalato. Poiché la risoluzione delle entità può richiedere un processo essenzialmente manuale per abbinare le informazioni degli account, la disponibilità di librerie open-source che automatizzano questo compito e salvano le informazioni in Delta Lake può aumentare notevolmente la produttività degli investigatori nella risoluzione dei casi. Esistono molte opzioni per verificare la corrispondenza fra entità, ma raccomandiamo di utilizzare Locality-Sensitive Hashing (LSH) per individuare l'algoritmo giusto per ogni lavoro. Maggiori informazioni su LSH e sui suoi vantaggi sono fornite in questo post del blog.

Come riferito sopra, abbiamo individuato velocemente alcune incongruenze per l'indirizzo della banca NY Mellon, con "Canada Square, Canary Wharf, London, United Kingdom" simile a "Canada Square, Canary Wharf, London, UK". Possiamo memorizzare i nostri record de-deduplicati in una tabella Delta che può essere utilizzata per indagini di antiriciclaggio.

unique_id_l	unique_id_r	rptd_originator_address_l	rptd_originator_address_r
223254	223256	Canada Square, Canary Wharf, London, United Kingdom	Canada Square, Canary Wharf, London, UK

Figura 10

Dashboard della lakehouse AML

Databricks SQL sulla lakehouse sta colmando il divario rispetto ai data warehouse tradizionali in termini di gestione semplificata dei dati, prestazioni con nuovi motori di query Photon e simultaneità degli utenti. Questo aspetto è importante perché molte organizzazioni non hanno budget sufficiente per acquistare costosi software AML proprietari per gestire una miriade di casi di utilizzo, ad esempio combattere il finanziamento ad associazioni terroristiche, contribuendo a contrastare i crimini finanziari. Sul mercato esistono soluzioni dedicate per l'analisi di grafi, per attività di Business Intelligence in un data warehouse e per Machine Learning. Il progetto della lakehouse AML unifica i tre aspetti. I team che usano la piattaforma di dati AML possono sfruttare Delta Lake beneficiando dei minori costi di storage in cloud, integrando facilmente al tempo stesso tecnologie open-source per produrre report curati basati sulla tecnologia dei grafi, visione computerizzata e analisi SQL. La Figura 11 mostra una rappresentazione concreta della reportistica per antiriciclaggio.

I notebook allegati hanno prodotto un oggetto di transazioni, un oggetto di entità, oltre a riepiloghi come prospetti di strutturazione, livelli di identità sintetiche e classificazioni di indirizzi, utilizzando modelli pre-addestrati. Nella visualizzazione di Databricks SQL riportata sotto, abbiamo usato il nostro motore Photon SQL per eseguire riepiloghi su questi elementi e visualizzazioni integrate per approntare un dashboard di reportistica in pochi minuti. Ci sono ACL completi in entrambe le tabelle, oltre al dashboard stesso, per consentire agli utenti di condividere le informazioni con dirigenti e team di gestione dei dati; è stato inserito anche uno schedulatore per eseguire i report periodicamente. Il dashboard è la "punta dell'iceberg" del sistema di IA, BI e analisi integrato nella soluzione AML.

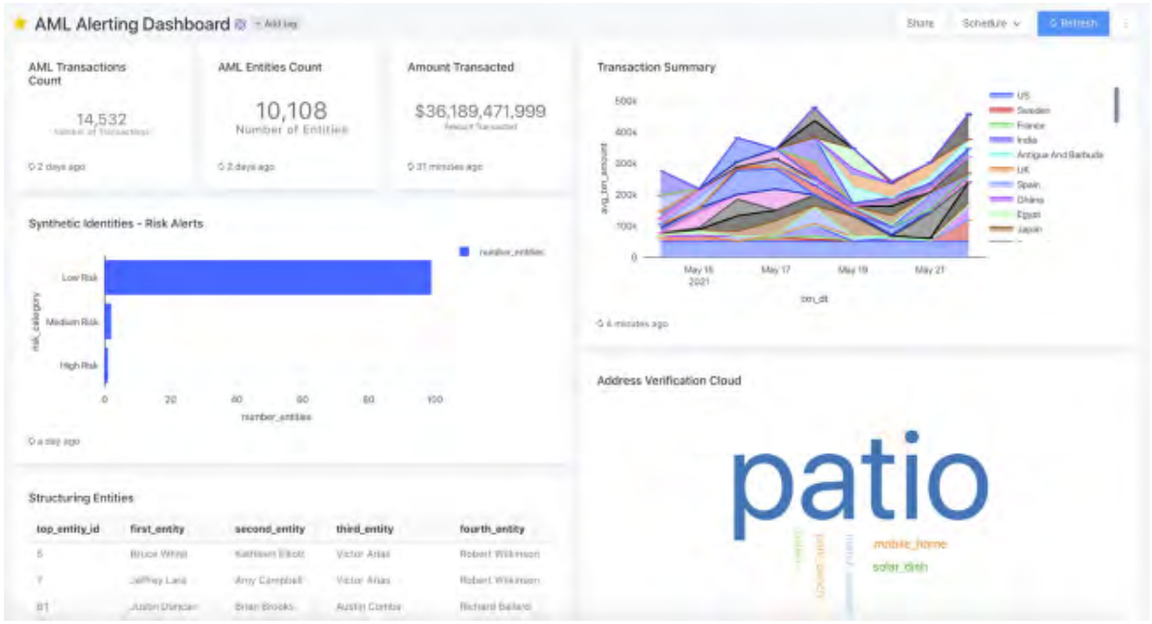

Figura 11

La rivoluzione della banca aperta

Il fenomeno dell'open banking consente agli istituti finanziari di offrire ai clienti un'esperienza migliore grazie alla condivisione di dati fra consumatori, istituti stessi e fornitori di servizi, attraverso apposite API. Un esempio è la Payment Services Directive (PSD2), che ha rivoluzionato i servizi finanziari nell'Unione Europea nell'ambito del regolamento Open Banking Europe. Gli istituti finanziari hanno accesso a più dati da più banche e fornitori di servizi, compresi i dati dei conti corrente e delle transazioni dei clienti. Questa evoluzione si è estesa fino al mondo delle frodi e dei crimini finanziari con le ultime disposizioni del FinCEN (Financial Crimes Enforcement Network degli Stati Uniti) riportate nella sezione 314(b) del USA Patriot Act; gli istituti finanziari interessati possono condividere informazioni con altri istituti e con le filiali nazionali ed estere, relativamente a individui, entità, organizzazioni e altri soggetti sospettati di coinvolgimenti in attività di riciclaggio.

Le disposizioni sulla condivisione di informazioni favoriscono la trasparenza e proteggono i sistemi finanziari degli Stati Uniti da attività di riciclaggio e finanziamento al terrorismo, ma lo scambio di informazioni deve essere effettuato con protocolli che offrono adeguata protezione dei dati e della sicurezza. Per risolvere il problema di proteggere la condivisione delle informazioni, Databricks ha recentemente annunciato Delta Sharing, un protocollo aperto e sicuro per la condivisione dei dati. Utilizzando API open-source molto diffuse, come pandas e Spark, produttori e consumatori di dati possono ora condividere dati utilizzando protocolli sicuri e aperti, oltre a mantenere pieno controllo su tutte le transazioni di dati per garantire il rispetto dei regolamenti FinCEN.

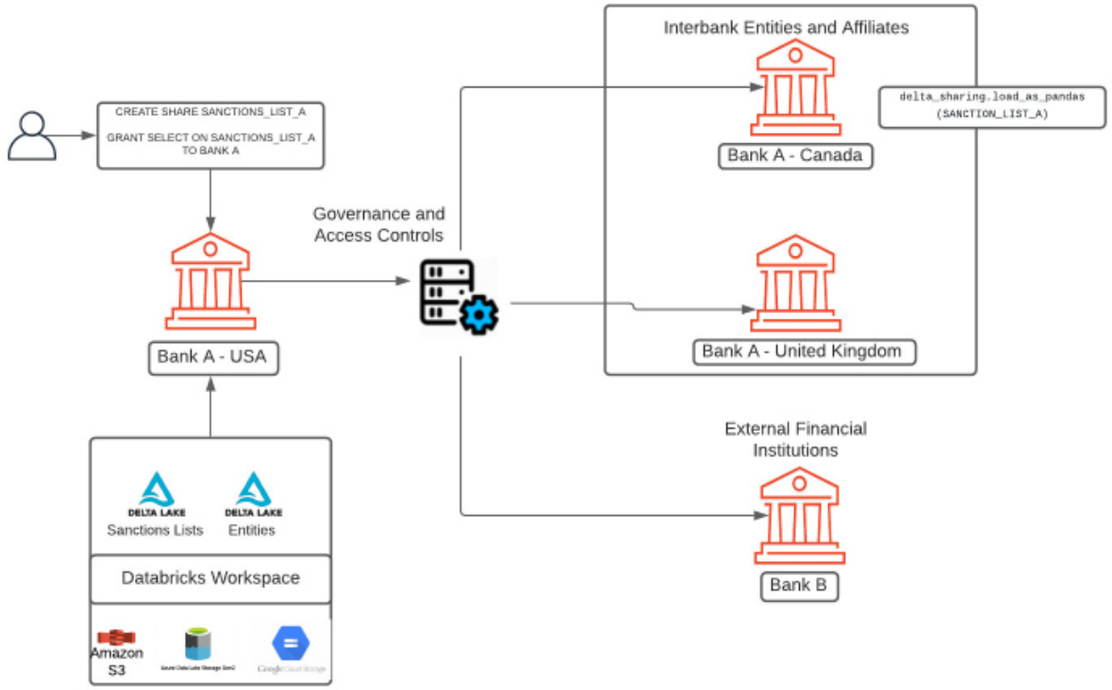

Figura 12

Conclusioni

L'architettura lakehouse è la piattaforma più scalabile e versatile per supportare gli analisti nelle loro attività di analisi AML. La lakehouse supporta casi di utilizzo che vanno dalle corrispondenze casuali all'analisi di immagini, fino alla BI con dashboard integrati; tutte queste funzionalità consentono alle organizzazioni di ridurre i costi di gestione rispetto a soluzioni AML proprietarie. Il team dei servizi finanziari di Databricks sta lavorando a svariati problemi nel settore dei servizi finanziari per consentire ai professionisti dell'ingegneria dei dati e della data science di cominciare il viaggio con Databricks adottando Solution Accelerator come AML.

Comincia a sperimentare con questi notebook Databricks gratuiti

- Introduzione alla teoria dei grafi per AML
- Introduzione alla visione computerizzata per AML
- Introduzione alla risoluzione di entità per AML

PAR. 2.6

Costruire un modello IA in tempo reale per rilevare comportamenti tossici nei videogame

di **DAN MORRIS** e **DUNCAN DAVIS**
16 giugno 2021

Nei giochi multiplayer online (sia MMO, sia MOBA) e in altre forme di videogame online, i giocatori interagiscono continuamente in tempo reale per collaborare o competere verso un obiettivo comune: la vittoria. Questa interattività è parte integrante delle dinamiche di gioco, ma al tempo stesso espone i giocatori al rischio di comportamenti tossici, un fenomeno sempre più diffuso nel mondo dei videogame online.

Il comportamento tossico o deviante si manifesta in varie forme, ad esempio diversi gradi di afflizione, cyber-bullismo e molestie sessuali, come mostrato nella matrice a destra realizzata da Behaviour Interactive, che elenca le tipologie di interazioni rilevate fra partecipanti al gioco multiplayer "Dead by Night."

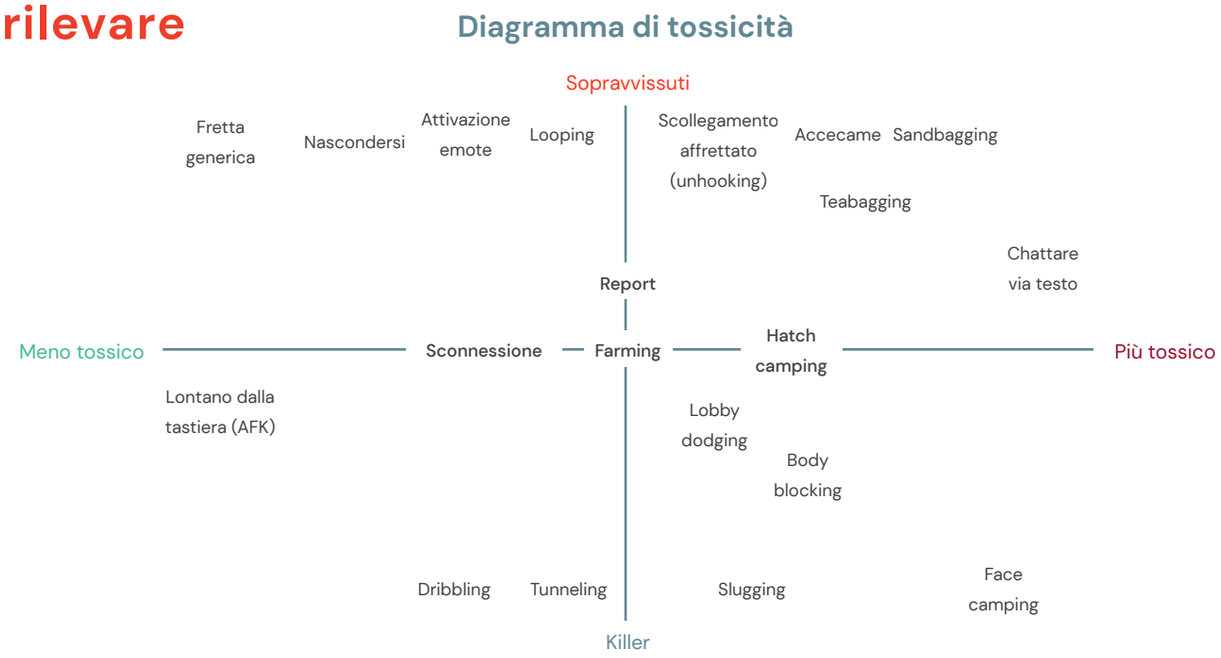

Figura 1
Matrice di interazioni tossiche vissute dai giocatori

Oltre ai problemi personali che i comportamenti tossici possono causare ai giocatori e alla comunità (un aspetto da non sottovalutare), questo fenomeno è dannoso anche per i risultati economici di molti produttori di videogiochi. Ad esempio, uno studio della Michigan State University rivela che l'80% dei giocatori ha mostrato sintomi di tossicità in tempi recenti e, di questi, il 20% ha abbandonato il gioco a causa di queste interazioni problematiche. Un altro studio della Tilburg University ha dimostrato che un incontro destabilizzante o tossico nella prima sessione di gioco triplica la possibilità che un giocatore abbandoni il gioco senza più tornare. Poiché la fidelizzazione dei giocatori è una priorità per i produttori di videogame, soprattutto con la transizione dalla distribuzione di giochi "fisici" alla fornitura di servizi a lungo termine, è chiaro che la tossicità deve essere combattuta.

Questo fenomeno, unito al tasso di abbandono fisiologico, spinge le aziende ad

affrontare il problema della tossicità fin dalle prime fasi dello sviluppo, ancor prima del lancio. Ad esempio, Crucible di Amazon è stato rilasciato in test senza chat testuale o vocale in parte a causa della mancanza di un sistema per monitorare o gestire le interazioni o i giocatori tossici. Questo dimostra che la portata degli spazi di gioco ha ampiamente superato la capacità dei team di gestire questi comportamenti con report o intervenendo sulle interazioni destabilizzanti. Con questi dati, è essenziale che gli studi integrino l'analisi nei giochi dall'inizio del ciclo di sviluppo dei prodotti, per poi progettare la gestione continua delle interazioni tossiche.

La tossicità nei videogame è chiaramente un problema con molte sfaccettature, che è diventato parte della cultura dei videogame e non può essere affrontato con un unico metodo. Detto questo, gestire la tossicità nelle chat dei giochi può rivelarsi molto efficace, data la frequenza dei comportamenti tossici e la possibilità di automatizzare il rilevamento con strumenti di elaborazione del linguaggio naturale (NLP).

Rilevamento della tossicità in Gaming Solution Accelerator di Databricks

Utilizzando dati di commenti tossici provenienti da Jigsaw e dati di partite di Dota 2, questo Solution Accelerator esegue i passaggi necessari per individuare commenti tossici in tempo reale mediante strumenti NLP e la lakehouse esistente. Per la parte NLP, questo Solution Accelerator usa Spark NLP di John Snow Labs, una soluzione open-source per aziende, costruita nativamente su Apache Spark.™

I passaggi effettuati con questo Solution Accelerator sono:

- caricare i dati di Jigsaw e Dota 2 in tabelle con Delta Lake;
- classificare i commenti tossici utilizzando la classificazione multi-etichetta (Spark NLP);
- tracciare esperimenti e modelli di registro con MLflow;
- applicare inferenze su dati in batch e in streaming;
- esaminare l'impatto della tossicità sui dati delle partite.

Rilevare la tossicità nelle chat dei giochi in produzione

Con questo Solution Accelerator diventa più facile integrare il rilevamento della tossicità nei videogame. Ad esempio, l'architettura di riferimento raffigurata sotto mostra come prelevare i dati della chat e del gioco da diverse sorgenti, come flussi, file, database vocali od operativi, e utilizzare Databricks per acquisire, memorizzare e curare i dati in tabelle di feature per pipeline di machine learning (ML), ML all'interno del gioco, tabelle BI per analisi e persino interazione diretta con strumenti utilizzati per moderare la community.

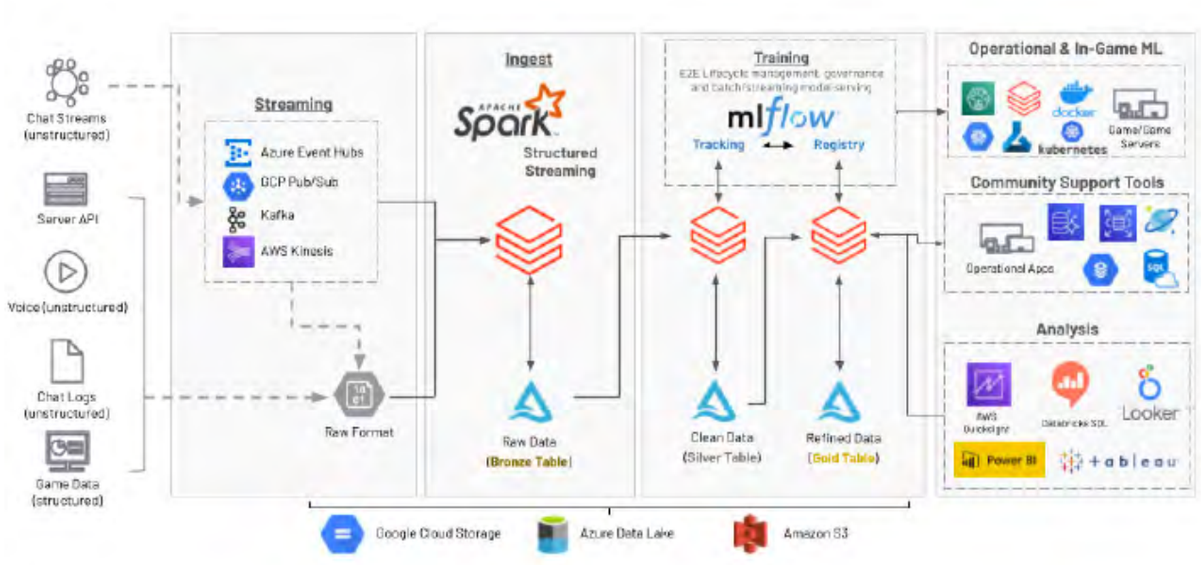

Figura 2
Architettura di riferimento per il rilevamento della tossicità

Disponendo di un'architettura scalabile in tempo reale per rilevare la tossicità nella community, si possono semplificare i flussi di lavoro per chi gestisce le relazioni nella community e si ha la capacità di filtrare milioni di interazioni in carichi di lavoro gestibili. Allo stesso modo, la possibilità di segnalare eventi gravemente tossici in tempo reale, o persino di automatizzare una risposta come silenziare alcuni giocatori o avvisare velocemente dell'incidente un responsabile delle relazioni con la clientela, può incidere direttamente sulla fidelizzazione dei giocatori. Oppure, avere una piattaforma in grado di elaborare grandi set di dati, provenienti da sorgenti disparate, può consentire di monitorare la percezione del brand attraverso report e dashboard.

Come cominciare

L'obiettivo di questo Solution Accelerator è supportare la gestione continua delle interazioni tossiche nei giochi online, rilevando in tempo reale commenti tossici nelle chat dei giochi. Comincia subito importando questo Solution Accelerator direttamente nello spazio di lavoro Databricks.

Una volta importato, saranno disponibili notebook con due pipeline pronte per andare in produzione.

- Una pipeline ML che utilizza classificazione con etichette multiple e addestramento su set di dati reali in inglese provenienti da Google Jigsaw. Questo modello classificherà ed etichetterà le forme di tossicità nel testo.
- Una pipeline di inferenza dello streaming in tempo reale che sfrutta il modello di tossicità. La sorgente della pipeline può essere facilmente modificata per acquisire dati di chat da tutte le sorgenti di dati comuni.

Con queste due pipeline si può cominciare a capire e analizzare la tossicità con un impegno minimo. Questo Solution Accelerator fornisce inoltre una base per costruire, personalizzare e migliorare il modello con dati rilevanti sui meccanismi di gioco e sulle community.

Comincia a sperimentare con questi notebook Databricks gratuiti.

PAR. 2.7

Guidare la trasformazione in Northwestern Mutual (Insights Platform) con il passaggio a un'architettura lakehouse scalabile aperta

di **MADHU KOTIAN**
15 luglio 2021

La trasformazione digitale svolge un ruolo centrale nella maggior parte degli attuali progetti di Big Data, soprattutto nelle aziende con pesanti infrastrutture esistenti. Fra i componenti chiave della trasformazione digitale ci sono sicuramente i dati e i relativi sistemi di stoccaggio. Da oltre 160 anni, Northwestern Mutual aiuta famiglie e aziende a garantirsi una sicurezza finanziaria. Con oltre 31 miliardi di dollari di ricavi, più di 4,6 milioni di clienti e 9.300 professionisti della finanza, non ci sono molte aziende con questi volumi di dati distribuiti su un'ampia gamma di sorgenti.

L'acquisizione dei dati è una sfida in questa fase in cui le organizzazioni devono gestire milioni di data point in diversi formati e intervalli temporali, provenienti da diverse direzioni, in una quantità senza precedenti. Vogliamo preparare i dati per l'analisi in modo da ricavarne un significato. Sono entusiasta di condividere il nostro approccio innovativo alla trasformazione e modernizzazione del nostro processo di acquisizione dei dati, del processo di schedulazione e del percorso di creazione dei data store. Una cosa che abbiamo imparato è che un approccio efficace deve essere sfaccettato; per questo motivo, oltre agli aspetti tecnici, illustrerò anche il piano per coinvolgere il nostro team.

Sfide da affrontare

Prima di avviare la trasformazione, abbiamo lavorato con i nostri partner aziendali per individuare esattamente i vincoli tecnici e inquadrare correttamente il nostro caso specifico.

Il punto debole che abbiamo individuato era la mancanza di dati integrati, con un flusso di dati operativi e dati dei clienti provenienti da diversi team e diverse fonti, all'interno e all'esterno dell'azienda. Abbiamo capito il valore dei dati in tempo reale, ma avevamo un accesso limitato a dati di produzione/in tempo reale che ci avrebbero permesso di prendere decisioni in modo puntuale. Abbiamo imparato anche che i data store creati dal team aziendale diventavano compartimenti stagni, "silos" di dati che a loro volta causavano problemi di latenza, incremento dei costi di gestione e problemi di sicurezza.

Inoltre, c'erano problematiche tecniche relative al nostro stato corrente. A fronte di un incremento della domanda e della necessità di nuovi dati, dovevamo gestire alcune limitazioni in termini di scalabilità dell'infrastruttura, latenza dei dati, costi di gestione dei silos di dati, limiti di dimensione e volume dei dati, nonché problemi di sicurezza. Con il progressivo aggravamento di tutti questi problemi, sapevamo di dover affrontare molte sfide e dovevamo trovare i partner giusti per assisterci nel percorso di trasformazione.

Soluzione di analisi

Dovevamo diventare un'azienda guidata dai dati per essere competitivi, servire meglio i nostri clienti e ottimizzare i processi interni. Abbiamo esplorato varie opzioni ed eseguito numerosi Proof-of-Concept (POC) prima di compiere la scelta definitiva. I punti fermi della nostra strategia erano i seguenti:

- una soluzione all-inclusive per le nostre esigenze di acquisizione, gestione e analisi dei dati;
- una moderna piattaforma in grado di supportare efficacemente i nostri sviluppatori e analisti nell'esecuzione di analisi con SQL;
- un motore di gestione dei dati che supporta transazioni ACID su S3 e sicurezza basata sui ruoli;
- un sistema in grado di proteggere efficacemente i nostri dati sensibili e sanitari (PII/PHI);
- una piattaforma scalabile in base alle esigenze di elaborazione e analisi dei dati;

La nostra infrastruttura esistente era basata su MSBI Stack. Utilizzavamo SSIS per l'acquisizione, SQL Server come data store, Azure Analysis Service per il modello tabulare e Power BI per dashboard e reportistica. Anche se la piattaforma inizialmente soddisfaceva le esigenze dell'azienda, avevamo problemi di scalabilità per gestire l'incremento del volume di dati e della capacità di elaborazione richiesta, problemi che limitavano le nostre aspettative di analisi. A fronte della necessità di nuovi dati, i problemi di latenza dovuti ai ritardi nel caricamento e a un data store concepito per esigenze specifiche causavano la creazione di silos e una crescita incontrollata dei dati.

La sicurezza è diventata un problema a causa dei dati sparsi fra numerosi sistemi. Avevamo circa 300 lavori ETL che richiedevano oltre 7 ore delle nostre attività quotidiane. Il time-to-market per ogni modifica o nuovo sviluppo era all'incirca di 4-6 settimane (a seconda della complessità).

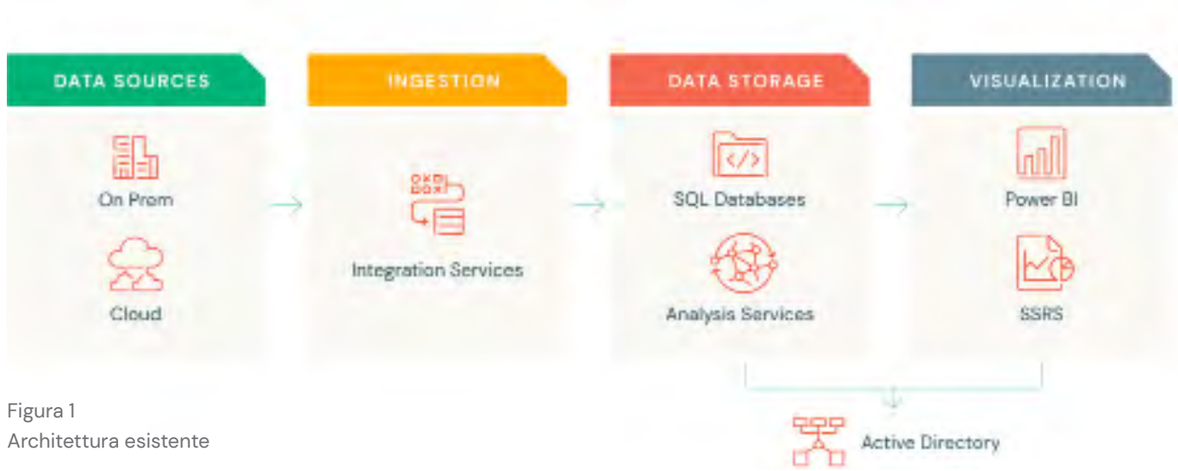

Figura 1
Architettura esistente

Dopo aver valutato molte soluzioni in commercio, abbiamo deciso di procedere con Databricks per implementare una soluzione integrata per la gestione dei dati su un'architettura lakehouse aperta.

Essendo sviluppato su Apache Spark™, Databricks ci consente di utilizzare Python per costruire il nostro framework personalizzato per l'acquisizione di dati e la gestione di metadati. Abbiamo ottenuto la flessibilità necessaria per effettuare analisi ad-hoc e altre scoperte di dati utilizzando il notebook. Databricks Delta Lake (il livello di storage costruito sul nostro data lake) offre la flessibilità per implementare svariate funzioni di gestione del database (transazioni ACID, governance di metadati, "viaggi nel tempo" ecc.), inclusa l'implementazione dei controlli di sicurezza richiesti. Databricks ha semplificato la gestione/scalabilità del cluster e ha consentito di reagire efficacemente all'impennata della domanda da parte dei nostri ingegneri e utenti aziendali.

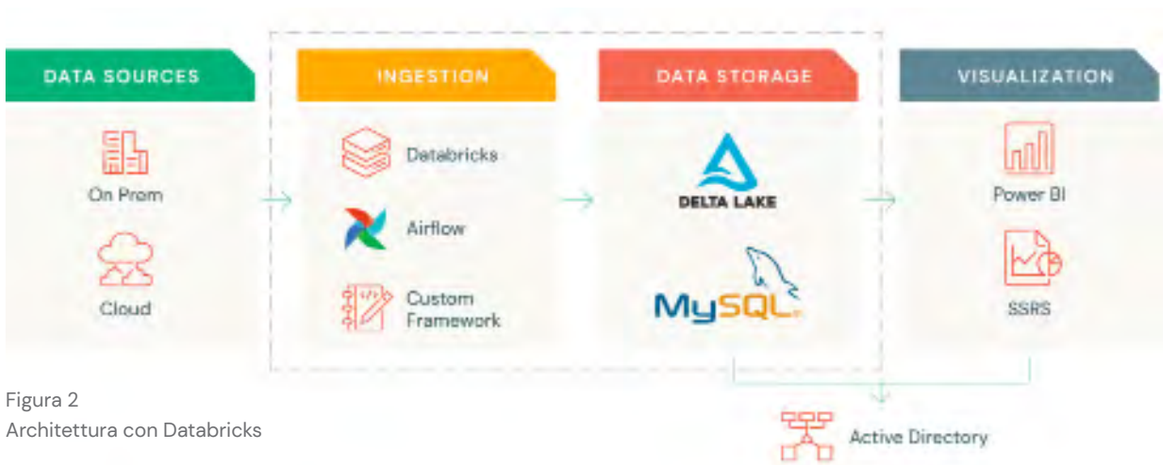

Figura 2
Architettura con Databricks

Approccio alla migrazione e onboarding delle risorse

Abbiamo cominciato con un piccolo gruppo di tecnici, assegnandoli a un team virtuale selezionato all'interno del nostro Scrum Team. Il loro obiettivo era eseguire diversi POC, lavorare sulla soluzione raccomandata, sviluppare best practice e riferire ai rispettivi team per contribuire all'onboarding. Il ricorso a membri dei team esistenti ci ha favorito, in quanto tutti possedevano già conoscenze sui sistemi esistenti, capivano le regole attuali del flusso di acquisizione dei dati e dell'azienda e avevano almeno una competenza di programmazione (conoscenze di ingegneria dei dati + ingegneria software) Il team si è prima formato su Python, ha compreso i dettagli più intricati di Spark e Delta e ha collaborato da vicino con il team di Databricks per validare la soluzione/l'approccio. Mentre il team era impegnato a definire la situazione futura, il resto dei nostri sviluppatori lavorava sulla realizzazione delle priorità dell'azienda.

Poiché la maggior parte degli sviluppatori erano ingegneri di MSBI Stack, il nostro piano d'azione era realizzare una piattaforma di gestione dei dati che funzionasse senza intoppi per i nostri sviluppatori, utenti aziendali e consulenti sul campo.

- Abbiamo costruito un framework di acquisizione che copriva tutte le nostre esigenze di caricamento e trasformazione dei dati, con controlli di sicurezza integrati che mantenevano tutti i metadati e i "segreti" dei nostri sistemi sorgente. Il processo di acquisizione accettava un file JSON che conteneva sorgente, destinazione e trasformazione richiesta. Questa procedura ha consentito trasformazioni semplici e complesse.
- Per la schedulazione abbiamo utilizzato Airflow, ma data la complessità del DAG, abbiamo costruito un nostro framework personalizzato su Airflow, che accettava un file YAML contenente le informazioni di lavoro e le relative interdipendenze.
- Per gestire le modifiche a livello di Schema con Delta, abbiamo costruito un nostro framework personalizzato per automatizzare diverse operazioni di database (DDL) senza richiedere agli sviluppatori un accesso forzato al data store. In questo modo abbiamo potuto implementare anche diversi controlli di revisione sul data store.

In parallelo, il team ha collaborato anche con il team di sicurezza per assicurarci la piena comprensione e il rispetto di tutti i criteri per la sicurezza dei dati (codifica in transito, codifica a riposo e codifica a livello di colonna per proteggere i dati sensibili).

Una volta impostati questi framework, il team ha implementato un flusso completo (da sorgente a destinazione con tutte le trasformazioni) e generato un nuovo set di report/dashboard su Power BI che puntavano a Delta Lake. L'obiettivo era testare le prestazioni dell'intero processo, validare i dati e ottenere eventuali feedback dai nostri utenti sul campo. Abbiamo migliorato progressivamente il prodotto in base ai riscontri raccolti e agli esiti dei test di prestazioni/convalida.

Contemporaneamente abbiamo realizzato guide di formazione e passo-passo per accogliere gli sviluppatori. Poco dopo abbiamo deciso di rimandare i membri del team trasversale ai rispettivi team, mantenendo solo alcuni componenti per continuare a supportare l'infrastruttura della piattaforma (DevOps). Ogni Scrum Team era responsabile di gestire e fornire all'azienda il set di funzionalità di sua competenza. Dopo che ognuno è rientrato nel proprio team, i vari membri si sono dedicati ad adeguare la velocità del team per gestire il lavoro arretrato per la migrazione. I responsabili dei team hanno ricevuto indicazioni e obiettivi appropriati per raggiungere gli obiettivi dei diversi Sprint/Program Increment. I membri dei team che avevano partecipato al team trasversale operavano ora come esperti interni al loro team, contribuendo all'onboarding sulla nuova piattaforma e mettendosi a disposizione per domande o assistenza specifiche.

Procedendo nella costruzione della nuova piattaforma, abbiamo mantenuto la vecchia a scopo di convalida e verifica.

L'inizio del successo

La trasformazione complessiva ha richiesto circa un anno e mezzo, un risultato notevole se si considera che abbiamo dovuto costruire tutti i framework, gestire le priorità dell'azienda, gestire le aspettative di sicurezza, riattrezzare il nostro team e migrare la piattaforma. Il tempo di caricamento totale è diminuito sensibilmente da 7 a solo 2 ore. Il time-to-market è di circa 1-2 settimane, nettamente inferiore alle precedenti 4-6. Questo è un miglioramento notevole che so che si estenderà a tutta l'azienda in vari modi.

Il nostro percorso non è completato. Mentre continuiamo a migliorare la nostra piattaforma, la prossima missione sarà espandere il modello della lakehouse. Stiamo lavorando alla migrazione della nostra piattaforma a E2 e implementando Databricks SQL. Stiamo lavorando sulla nostra strategia per fornire agli utenti aziendali una piattaforma self-service sulla quale effettuare analisi ad-hoc e portare tutti i loro dati con la possibilità di fare analisi con i nostri dati integrati. Abbiamo imparato che si ottengono grandi benefici utilizzando una piattaforma aperta, unificata e scalabile. Con esigenze e funzionalità in continuo aumento, sappiamo di poter contare su un partner affidabile come Databricks.

Ascolta il racconto del viaggio di Northwestern Mutual verso la lakehouse

INFORMAZIONI SU MADHU KOTIAN

Madhu Kotian è Vice President of Engineering (Investment Products Data, CRM, Apps and Reporting) di Northwestern Mutual. Vanta oltre 25 anni di esperienza nel settore informatico, con competenze di ingegneria dei dati, gestione del personale, gestione di programmi, architettura, progettazione, sviluppo e manutenzione, utilizzando pratiche agili. È anche esperto di metodologie per data warehouse e implementazione di integrazione e analisi dei dati.

PAR. 2.8

Come il team dei dati di Databricks ha costruito una lakehouse su tre cloud e oltre 50 regioni

di **JASON POH** e **SURAJ ACHARYA**
14 luglio 2021

L'infrastruttura di registro interna di Databricks si è evoluta negli anni e abbiamo imparato alcune cose su come mantenere una pipeline altamente disponibile su diversi cloud e aree geografiche. Questo blog approfondirà le modalità di raccolta e gestione di metriche in tempo reale sulla nostra piattaforma lakehouse, spiegando come utilizziamo molteplici cloud per ripristinare un sistema in caso di caduta di un cloud pubblico.

Quando venne fondata, Databricks supportava un solo cloud pubblico. Ora il servizio è cresciuto per supportare i tre principali cloud pubblici (AWS, Azure, GCP) in oltre 50 regioni in tutto il mondo. Ogni giorno, Databricks attiva milioni di macchine virtuali per conto dei nostri clienti. Il nostro team per la piattaforma di gestione dei dati, composto da meno di 10 tecnici, è responsabile della costruzione e della manutenzione dell'infrastruttura di telemetria, che elabora ogni giorno mezzo petabyte di dati. Orchestrazione, monitoraggio e utilizzo vengono catturati attraverso registri di servizio che vengono elaborati dalla nostra infrastruttura per fornire metriche puntuali e accurate. Infine, questi dati vengono conservati nel nostro Delta Lake con capacità nell'ordine dei petabyte. Il nostro team della piattaforma dati utilizza Databricks per effettuare analisi su diversi cloud, consentendoci di federare i dati laddove appropriato, mitigare il ripristino dalla caduta di un cloud regionale e ridurre al minimo l'impatto sulla nostra infrastruttura attiva.

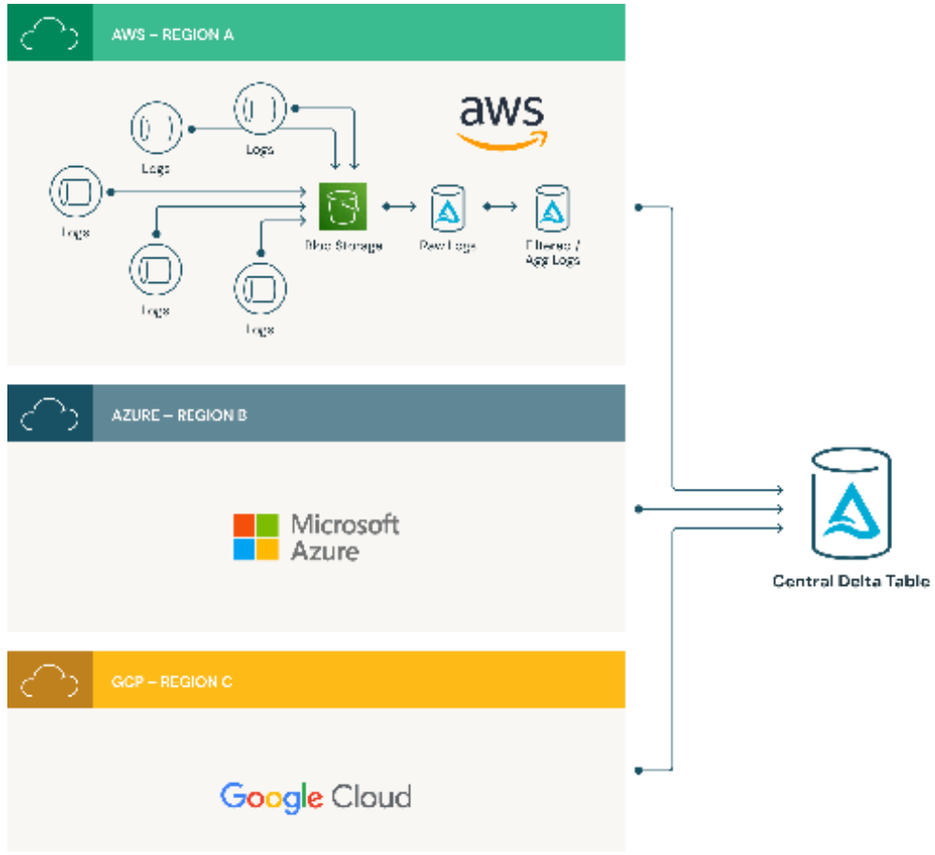

Figura 1

Architettura della pipeline

Ogni regione cloud contiene un'infrastruttura e pipeline di dati proprie per catturare, raccogliere e conservare dati di registro in un Delta Lake regionale. I dati di telemetria dei prodotti vengono acquisiti da tutte le pipeline di prodotto e le nostre pipeline con lo stesso processo replicato su ogni regione cloud. Un log daemon acquisisce i dati di telemetria e scrive poi tali registri su uno storage bucket del cloud regionale (S3, WASBS, GCS). Da lì, una pipeline schedulata acquisirà i file di log utilizzando Auto Loader (AWS | Azure | GCP) e scriverà i dati in una tabella Delta regionale. Un'altra pipeline leggerà i dati dalla tabella Delta regionale, li filtrerà e li scriverà in una tabella Delta centralizzata in una singola regione cloud.

databricks

Prima di Delta Lake

Prima di Delta Lake avremmo scritto i dati della sorgente nella loro tabella sul lake centralizzato e poi creato una vista che unificava tutte le tabelle. Questa vista doveva essere calcolata in runtime e sarebbe diventata più inefficiente con l'aggiunta di nuove regioni:

```
CREATE OR REPLACE VIEW all_logs AS
SELECT * FROM (
  SELECT * FROM region_1.log_table
  UNION ALL
  SELECT * FROM region_2.log_table
  UNION ALL
  SELECT * FROM region_3.log_table
  ...
);
```

Dopo Delta Lake

Oggi abbiamo un'unica tabella Delta che accetta dichiarazioni di scrittura simultanee da oltre 50 diverse regioni, gestendo al tempo stesso query sui dati. Eseguire una query sulla tabella centrale è molto facile:

```
SELECT * FROM central.all_logs;
```

La transazionalità viene gestita da Delta Lake. Abbiamo eliminato le singole tabelle regionali nel nostro Delta Lake centrale e ritirato la vista UNION ALL. Il codice riportato di seguito è una rappresentazione semplificata della sintassi eseguita per caricare i dati approvati per l'uscita dai Delta Lake regionali verso il Delta Lake centrale:

```
spark.readStream.format("delta")
  .load(regional_source_path)
  .where("egress_approved = true")
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("checkpointLocation", checkpoint_path)
  .start(central_target_path)
```

Disaster recovery

Uno dei vantaggi di gestire un servizio inter-cloud è che siamo ben posizionati per affrontare specifici scenari di disaster recovery. Per quanto rari, si sente spesso parlare di eventi che mettono fuori servizio una particolare regione cloud. Quando accade, lo storage in cloud è accessibile, ma la capacità di attivare nuove VM è inibita. Poiché abbiamo progettato la nostra pipeline di dati per accettare la configurazione dei percorsi di sorgente e destinazione, questo ci consente di implementare ed eseguire pipeline di dati in una regione diversa da quella in cui sono conservati i dati stessi. Il cloud per il quale il cluster è stato creato è irrilevante rispetto al cloud sul quale vengono letti o scritti i dati.

Esistono alcuni set di dati che proteggiamo da guasti del servizio di storage replicando continuamente i dati su diversi provider di servizi cloud. Questo si può realizzare facilmente sfruttando la funzionalità di clonazione Delta descritta in questo blog. Ogni volta che viene eseguito su una tabella, il comando clona aggiorna il clone solo con le modifiche incrementali apportate dall'esecuzione precedente. Si tratta di un modo efficiente per replicare i dati su diverse regioni e anche diversi cloud.

Minimizzare l'impatto sulle pipeline di dati attive

Le nostre pipeline di dati sono la linfa vitale del nostro servizio gestito e parte integrante di un'azienda multinazionale che non dorme mai. Non ci possiamo permettere di mettere in pausa le pipeline per lunghi periodi di tempo per interventi di manutenzione, aggiornamenti o caricamenti di dati. Recentemente abbiamo dovuto mettere mano alle pipeline per filtrare un sottoset dei dati che vengono normalmente scritti nella nostra tabella principale, affinché fossero scritti su un diverso cloud pubblico. Siamo riusciti a farlo senza interferire con la normale operatività.

Seguendo questi passaggi abbiamo implementato le modifiche alla nostra architettura nei nostri sistemi attivi senza alcun impatto significativo.

Abbiamo dapprima eseguito un deep clone della tabella principale in una nuova posizione
sull'altro cloud. Questa operazione copia i dati e il registro delle transazioni in modo tale da garantire la coerenza.

In secondo luogo, abbiamo rilasciato la nuova configurazione alle nostre pipeline, in modo che la maggior parte dei dati continuasse a essere scritta nella tabella principale centrale, mentre il sottoset di dati viene scritto nella nuova tabella clonata sull'altro cloud. Questa modifica può essere fatta facilmente implementando una nuova configurazione, dopodiché la tabella riceve aggiornamenti solo per le nuove modifiche.

Abbiamo poi eseguito nuovamente lo stesso comando deep clone. Delta Lake acquisirà e copierà solo le modifiche incrementali dalla tabella principale originale alla nuova tabella clonata. Questa operazione, sostanzialmente, riempie la nuova tabella con tutte le modifiche apportate ai dati fra i passaggi 1 e 2.

Infine, il sottoset di dati può essere cancellato dalla tabella principale e la maggior parte dei dati può essere cancellata dalla tabella clonata.

Ora le due tabelle rappresentano i dati che ciascuna dovrebbe contenere, con una storia completa delle transazioni, e tutto questo è stato fatto "in diretta" senza incidere sulla freschezza della pipeline.

Riepilogo

Databricks astrae i dettagli dei singoli servizi cloud, che si tratti di approntare un'infrastruttura con il nostro gestore di cluster, acquisire dati con Auto Loader o eseguire scritture transazionali sullo storage in cloud con Delta Lake. Questo ci offre il vantaggio di poter utilizzare un'unica base di codice per estendere l'elaborazione e lo storage su più cloud pubblici, a scopo sia di federazione dei dati, sia di disaster recovery. Questa funzionalità inter-cloud offre la flessibilità di spostare il calcolo e lo storage dove meglio serve a noi e ai nostri clienti.

CAPITOLO

03

Referenze dei clienti

Atlassian

ABN AMRO

J.B. Hunt

PAR. 3.1

Atlassian

Atlassian è un fornitore di software per collaborazione, sviluppo e tracciamento di problemi per gruppi di lavoro. Con oltre 150.000 clienti in tutto il mondo (fra cui 85 aziende Fortune 100), Atlassian promuove il valore della collaborazione con prodotti quali Jira, Confluence, Bitbucket, Trello e altri.

CASO APPLICATIVO

Atlassian utilizza Databricks Lakehouse Platform per democratizzare i dati in tutta l'impresa e abbassare i costi di gestione. Atlassian conta attualmente numerose applicazioni che puntano a mettere l'esperienza del cliente in primo piano.

Servizio clienti ed esperienza di assistenza
Poiché la maggior parte dei clienti opera su server (con prodotti come Jira e Confluence), Atlassian ha deciso di trasferire questi clienti in cloud per sfruttare informazioni più approfondite e dettagliate che arricchiscono l'esperienza con il servizio clienti.

Marketing personalizzato
Le stesse informazioni possono essere utilizzate per inviare e-mail di marketing personalizzate per richiamare l'attenzione
verso nuovi prodotti e funzionalità.

Lotta agli abusi e alle frodi
L'azienda riesce a prevedere abusi di licenze e comportamenti fraudolenti grazie al rilevamento delle anomalie e all'analisi predittiva.

In Atlassian dobbiamo garantire che i team possano collaborare bene trasversalmente a tutte le funzioni per raggiungere obiettivi in continua evoluzione. Un'architettura lakehouse semplificata ci consentirebbe di acquisire grandi quantità di dati degli utenti ed eseguire le analisi necessarie per prevedere meglio le esigenze e migliorare l'esperienza dei nostri clienti. Un'unica piattaforma facile da usare per l'analisi in cloud ci consente di migliorare rapidamente e costruire nuovi strumenti di collaborazione sulla base di informazioni approfondite fruibili.

Rohan Dhupelia
Data Platform Senior Manager, Atlassian

SOLUZIONI E BENEFICI

Atlassian utilizza Databricks Lakehouse Platform per democratizzare i dati su larga scala, sia internamente sia esternamente. L'azienda è passata da un paradigma di data warehouse alla standardizzazione su Databricks, diventando più data-driven in tutta l'organizzazione. Oltre 3.000 utenti interni nei reparti di risorse umane, marketing, finanza e R&D, che rappresentano più della metà dell'organizzazione, hanno accesso alle informazioni generate dalla piattaforma con frequenza mensile, tramite tecnologie aperte come Databricks SQL. Atlassian utilizza la piattaforma anche per offrire ai clienti esperienze di assistenza più personalizzate.

- Delta Lake supporta una singola lakehouse per svariati petabyte di dati accessibili a oltre 3.000 utenti di HR, marketing, finanza, vendite, assistenza e R&D
- Carichi di lavoro BI alimentati da Databricks SQL consentono di offrire dashboard di reportistica a più utenti
- MLflow snellisce MLOps per velocizzare la consegna di dati
- L'unificazione della piattaforma dati agevola la governance, mentre i cluster autogestiti favoriscono l'autonomia

Con un'architettura dimensionata per il cloud, una maggiore produttività grazie alla collaborazione fra team e la capacità di accedere a tutti i dati dei clienti per analisi e ML, la soluzione avrà un impatto enorme sull'attività di Atlassian. L'azienda ha già:

- ridotto il costo delle attività IT (in particolare i costi di calcolo) del 60% trasferendo oltre 50.000 lavori Spark da EMR e Databricks, con uno sforzo minimo e poche modifiche al codice;
- diminuito i tempi di consegna del 30% grazie a cicli di sviluppo più brevi;
- ridotto la dipendenza dei team di gestione dei dati del 70% potenziando il self-service in tutta l'organizzazione.

Maggiori informazioni

databricks

PAR. 3.2

ABN AMRO

ABN AMRO è un istituto bancario consolidato che voleva modernizzare l'attività ma era vincolata a un'infrastruttura e un data warehouse obsoleti che complicavano l'accesso ai dati da varie sorgenti e creavano inefficienza nei processi e nei flussi di lavoro con i dati. Ora, Azure Databricks consente a ABN AMRO di democratizzare i dati e l'IA per un team di oltre 500 ingegneri, scienziati e analisti che lavorano insieme al miglioramento delle attività operative e all'introduzione di nuove funzionalità di go-to-market in tutta l'azienda.

CASO APPLICATIVO

ABN AMRO utilizza Databricks Lakehouse Platform per realizzare la trasformazione dei servizi finanziari su scala globale, estendendo l'automazione e l'estrapolazione di informazioni approfondite a tutte le attività operative.

Finanza personalizzata
ABN AMRO sfrutta dati e informazioni sui clienti in tempo reale per fornire prodotti e servizi ritagliati su misura per le esigenze di ogni cliente. Ad esempio, utilizza il Machine Learning per inviare messaggi mirati nelle campagne di marketing finalizzate ad aumentare il coinvolgimento e i tassi di conversione.

Gestione del rischio
Adottando processi decisionali guidati dai dati, ABN AMRO si focalizza sulla riduzione del rischio, sia per l'azienda, sia per i suoi clienti. Ad esempio, vengono creati report e dashboard con cui le figure decisionali e i responsabili interni possono valutare meglio il rischio ed evitare che produca effetti negativi sull'attività di ABN AMRO.

Rilevamento di frodi
Per prevenire attività criminali, la banca utilizza l'analisi predittiva per intercettare le frodi prima che colpiscano i clienti. Fra le varie attività che sta cercando di contrastare ci sono il riciclaggio di denaro sporco e la contraffazione delle carte di credito.

Databricks ha cambiato il nostro modo di lavorare. Ci ha messo in una posizione migliore per portare a termine la trasformazione nell'ambito della gestione dei dati e dell'IA, fornendo ai nostri professionisti funzionalità avanzate di gestione dei dati in modo controllato e scalabile.

Stefan Groot
Head of Analytics Engineering,
ABN AMRO

SOLUZIONI E BENEFICI

Oggi, Azure Databricks consente a ABN AMRO di democratizzare i dati e l'IA per un team di oltre 500 ingegneri, scienziati e analisti che lavorano insieme al miglioramento delle attività operative e all'introduzione di nuove funzionalità di go-to-market in tutta l'azienda.

- Delta Lake offre pipeline di dati veloci e affidabili per fornire dati precisi e completi alle attività di analisi a valle
- L'integrazione con Power BI facilita l'analisi SQL e fornisce informazioni dettagliate e approfondite a oltre 500 utenti aziendali tramite report e dashboard
- MLflow velocizza l'implementazione di nuovi modelli che migliorano l'esperienza del cliente, consentendo di approntare nuovi casi d'uso in meno di due mesi

10x velocità
nel time-to market — nuovi casi d'uso realizzati in due mesi

oltre 100
casi d'uso da realizzare entro il prossimo anno

oltre 500
utenti aziendali e IT abilitati

Maggiori informazioni

databricks

PAR. 3.2

J.B. HUNT

Databricks ci ha messo a disposizione una piattaforma per il mercato del trasporto merci digitale più innovativo, consentendoci di sfruttare l'IA per offrire la migliore esperienza di trasporto possibile.

Joe Spinelle
Director, Engineering and Technology,
J.B. Hunt

Nell'ottica di costruire la rete digitale per i trasporti più efficiente del Nord America, J.B. Hunt voleva snellire la logistica del trasporto merci e offrire ai clienti la migliore esperienza in questo settore. Ma l'architettura esistente, la mancanza di funzionalità IA e l'incapacità di gestire Big Data in modo sicuro erano ostacoli insormontabili. Dopo aver implementato Databricks Lakehouse Platform e Immuta, J.B. Hunt è ora in grado di fornire soluzioni operative che vanno dal miglioramento dell'efficienza della supply chain all'aumento della produttività degli autisti, ottenendo notevoli risparmi nell'infrastruttura IT e un incremento dei ricavi.

CASO APPLICATIVO

J.B. Hunt utilizza Databricks per effettuare analisi del trasporto merci attraverso la piattaforma Carrier 360, abbattendo i costi e al tempo stesso aumentando la produttività e la sicurezza degli autisti. Le applicazioni comprendono logistica del trasporto merci, gestione dei clienti a 360 gradi, personalizzazione e altro.

SOLUZIONI E BENEFICI

J.B. Hunt usa Databricks Lakehouse Platform per costruire il mercato per la logistica dei trasporti più sicuro ed efficiente del Nord America, snellendo la logistica, ottimizzando le esperienze di trasporto e riducendo i costi.

- Delta Lake federa e democratizza i dati per ottimizzare in tempo reale i percorsi e le indicazioni agli autisti attraverso la piattaforma Carrier 360
- I notebook aumentano la produttività del team di gestione dei dati per implementare più applicazioni più velocemente
- MLflow velocizza l'adozione di nuovi modelli che migliorano l'esperienza degli autisti

Maggiori informazioni

$2,7 milioni
di risparmi nell'infrastruttura IT, aumentando la redditività

5%
di aumento dei ricavi grazie al miglioramento della logistica

99,8% più veloci
nei consigli per una migliore esperienza di trasporto

databricks

Informazioni su Databricks

Databricks è un'azienda di IA e dati. Più di 5.000 organizzazioni in tutto il mondo (fra cui Comcast, Condé Nast, H&M e oltre il 40% delle aziende Fortune 500) fanno affidamento sulla piattaforma Databricks Lakehouse per unificare dati, analisi e IA. Databricks ha la sede principale a San Francisco e uffici in tutto il mondo. Fondata dai creatori di Apache Spark™, Delta Lake e MLflow, Databricks persegue la missione di aiutare i team di gestione dei dati a risolvere i problemi più difficili del mondo. Per maggiori informazioni, segui Databricks su Twitter, LinkedIn e Facebook.

COMINCIA LA PROVA

Contattaci per una demo personalizzata
databricks.com/contact

databricks