

SNOWPARK: CREARE PIPELINE E MODELLI DI DATI MIGLIORI NEL DATA CLOUD



CHAMPION
GUIDE

EBOOK

SOMMARIO

- 3** Executive summary
- 5** Come creare pipeline e modelli di dati migliori con Snowpark
- 7** Casi d'uso di Snowpark per il data engineering
- 9** Casi d'uso di Snowpark per la data science
- 12** Casi d'uso di altri partner Snowflake
- 13** Inizia il tuo percorso con Snowpark
- 15** Informazioni su Snowflake

EXECUTIVE SUMMARY

Snowflake ha iniziato il suo percorso verso il Data Cloud riprogettando completamente il mondo dei dati e ripensando l'architettura dei sistemi di elaborazione dei dati per renderli al tempo stesso affidabili, sicuri, ad alte prestazioni e scalabili per il cloud.

Snowpark, un framework di sviluppo per Snowflake, consente a tutti gli utenti dei dati di portare il proprio lavoro nel Data Cloud di Snowflake con il supporto nativo per Python, SQL, Java e Scala, come illustrato nella figura 1. Snowpark consente a data engineer, data scientist e sviluppatori di continuare a programmare nel linguaggio che preferiscono e di eseguire pipeline, flussi di lavoro ML e app dati in modo più rapido e sicuro, in un'unica piattaforma. Con Snowpark, i vantaggi di Snowflake si estendono dagli utenti SQL a tutti i tuoi team che lavorano con i dati. Gli utenti di Python, Java e Scala ora possono approfittare delle prestazioni, dell'elasticità e della governance del motore di elaborazione di Snowflake.

Gli sviluppatori interagiscono con Snowflake attraverso tre componenti principali di Snowpark:

- **Snowpark DataFrame API.** Crea query utilizzando i familiari DataFrame dall'interfaccia utente di Snowflake (preview) o dal tuo IDE preferito; sfrutta le prestazioni e la scala del motore di elaborazione elastico di Snowflake per i processi trasferiti
- **UDF di Snowpark.** Esegui logica personalizzata scritta in codice nativo Python o Java direttamente in Snowflake utilizzando le UDF (User Defined Function)
- **Stored procedure.** Implementa e orchestra le tue pipeline e la tua logica personalizzata Python, Java o Scala direttamente all'interno di Snowflake, quindi rendila disponibile ai tuoi utenti SQL

Poiché la potenza di Python deriva dal suo ricco ecosistema di pacchetti open source, Snowpark per Python incorpora i package Anaconda piú diffusi ed è integrato con il gestore di pacchetti Conda perché gli utenti non debbano preoccuparsi di installazioni manuali o dipendenze mancanti.

Che cosa si può creare con Snowpark? In questo ebook verranno descritti alcuni esempi di casi d'uso, ma ve ne sono molti altri. Iniziamo.

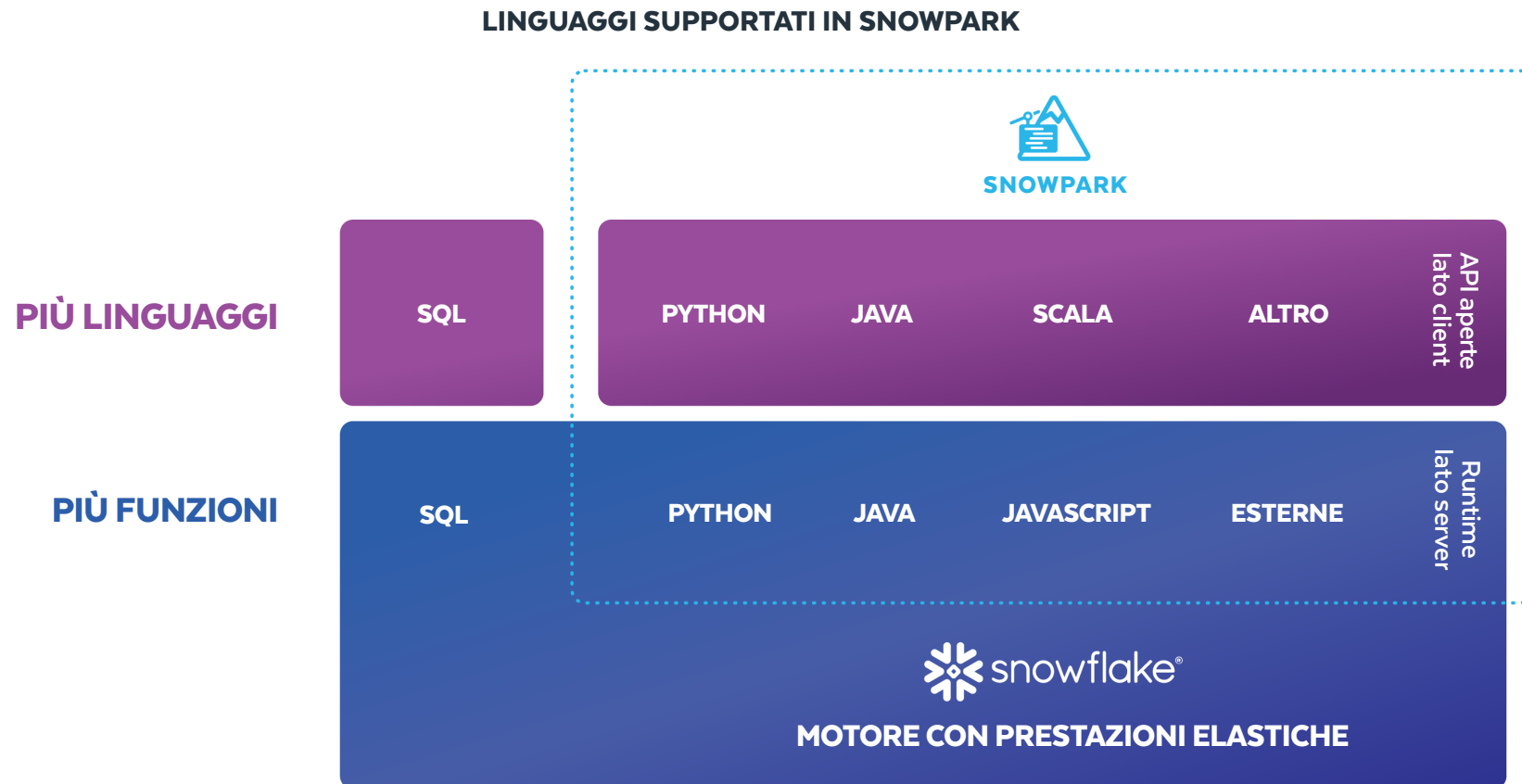


Figura 1: Snowpark consente a sviluppatori e data engineer di interagire direttamente con Snowflake senza dover spostare i dati.

COME CREARE PIPELINE E MODELLI DI DATI MIGLIORI CON SNOWPARK

Per capire come funziona Snowpark, diamo un'occhiata a un esempio generico con dati di identificazione personale (PII).

QUERY SEMPLIFICATE, CONVERSIONI DIRETTE E IMMEDIATE

Con la Snowpark API, gli sviluppatori possono creare query utilizzando DataFrame nel codice senza creare e passare stringhe SQL, ad esempio:

```
session = Session.builder.configs(connection_params).create()

sales_df = session.table('sales')
line_items_df = session.tables('sales_details')
query = sales_df \
    .join(line_items_df, sales_df.col('id') == line_items_
df.col('sid')) \
    .group_by(line_items_df.col('product_id')) \
    .count()
```

La Snowpark API fornisce un supporto di prima classe nell'ambiente di sviluppo, che include controllo del tipo, IntelliSense e segnalazione degli errori. Dietro le quinte, tutte le operazioni con DataFrame sono convertite in modo trasparente in query SQL che vengono trasferite al motore di elaborazione scalabile di Snowflake.

Inoltre Snowpark consente di scrivere logica personalizzata Python utilizzando UDF e stored procedure che vengono eseguite direttamente in Snowflake. In questo esempio viene impiegato codice personalizzato Python per mascherare i dati PII:

```
def mask_pii(pii):
    # Custom PII detection logic
```

Utilizzando la Snowpark API, è possibile classificare facilmente questo codice come **UDF (User Defined Function)** per poi utilizzarlo nelle operazioni DataFrame:

```
mask_pii_udf = udf(mask_pii)

session.table('emails') \
    .with_column('body', mask_pii_udf(col('body'))) \
    .show()
```

La Snowpark API invia la logica a Snowflake, dove viene eseguita accanto ai dati in un sandbox Python sicuro.

L'AUTOMAZIONE ELIMINA LA SCRITTURA MANUALE DI CODICE

Per scoprire come Snowpark semplifica le operazioni comuni, vediamo come consente di applicare la logica di rilevamento PII a tutte le colonne di stringhe di una tabella. Con SQL, è necessario scrivere manualmente una query per ogni tabella oppure scrivere il codice per generare la query. Con Snowpark, è facile scrivere una routine generica:

```
def mask_table(df):
    return df.select(map(lambda field : mask_pii_udf(col(field.name)) \
        if field.DataType == StringType() \
        else col(field.name), df.schema))
```

Questa routine generica può poi essere utilizzata per mascherare con facilità tutti i dati PII in qualsiasi tabella:

```
masked_emails = mask_table(session.table('emails'))
```

Con queste poche righe di codice, Snowpark genera dinamicamente una query robusta basata su schema.

CREARE UNA LOGICA COMPLESSA CON LE UDF PYTHON

Con le UDF Python, è possibile eseguire una logica complessa che espone un'interfaccia funzione semplice:

```
class Sentiment:
    def score(text):
```

Per creare queste funzioni puoi utilizzare il set di strumenti esistente, inclusi il controllo del codice sorgente, gli ambienti di sviluppo, gli strumenti di debug e tutte le librerie necessarie. Snowpark offre la possibilità di acquisire codice utile da GitHub o da altre fonti per utilizzarlo in Snowflake.

Utilizzare il tuo codice Python in SQL è semplice. Basta creare un modulo, importarlo in Snowflake e registrare una funzione.

```
create function score(text string)
    returns float
    language python
    runtime_version = '3.8'
    handler = 'Sentiment.score'
    imports = ('@my_stage/sentiment.py')
```

Ora qualsiasi utente SQL può utilizzare la logica che hai creato come qualsiasi altra funzione:

```
select id, score(body)
    from emails;
```

Puoi anche creare UDF con un approccio basato esclusivamente su Python, utilizzando Snowpark per pacchettizzare la tua funzione Python e distribuirla sul lato server con il decorator @udf e la funzione di registrazione delle UDF.

Le UDF Python consentono di utilizzare gli strumenti esistenti per i casi complessi, ma per i casi d'uso più semplici la nuova funzionalità include anche definizioni in linea:

```
create or replace function reverse(s string)
    returns string
    language python
    runtime_version = '3.8'
    handler = 'Reverse.reverse'
    target_path = '@my_stage/reverse.py'
as $$
class Reverse:
    def reverse(s):
        return s[::-1]
$$
```



CASI D'USO DI SNOWPARK PER IL DATA ENGINEERING

Il data engineering è complesso per due ragioni principali. In primo luogo, spesso i data engineer hanno bisogno di utilizzare diversi strumenti e infrastrutture per ogni linguaggio di programmazione, complicando eccessivamente l'architettura della pipeline di dati. Uno dei motivi principali per il supporto di più linguaggi è che il data engineering è un'attività svolta in collaborazione tra più team. I data analyst possono preferire strumenti basati su GUI e SQL, i data scientist preferiscono generalmente preparare i dati utilizzando notebook e Python, mentre i data engineer e gli sviluppatori hanno bisogno di strumenti aggiuntivi per affrontare complessi costrutti di codice e programmazione. I dati spesso devono viaggiare attraverso sistemi diversi per far funzionare le pipeline, determinando architetture complesse che possono mettere a rischio la sicurezza e limitare la governance dei dati.

In secondo luogo, la gestione e l'utilizzo dell'infrastruttura di elaborazione dei dati richiedono normalmente un notevole sforzo manuale e costi di manutenzione elevati. Di conseguenza, i data engineer sono spesso troppo impegnati e dedicano la maggior parte del loro tempo alla manutenzione e alla riparazione delle pipeline.

Snowpark, un framework di sviluppo moderno per Snowflake, consente ai data engineer di creare pipeline semplici, governate e veloci nel proprio linguaggio di programmazione preferito. Snowpark offre tre importanti vantaggi ai data engineer: un'unica piattaforma che supporta più linguaggi senza elaborazione esterna, una governance di livello superiore e pipeline più veloci, economiche e resilienti. L'infrastruttura intelligente elimina la complessità per i data engineer: Snowflake funziona e basta, consentendo loro di concentrarsi sui compiti veramente importanti.

I data engineer generalmente utilizzano Snowpark per i seguenti scopi.

- **ETL/ELT:** i team dei dati possono utilizzare Snowpark per trasformare i dati grezzi in formati modellati indipendentemente dal tipo, compresi i dati JSON, Parquet e XML. Tutte le trasformazioni dei dati possono quindi essere pacchettizzate come Snowpark Stored Procedures per eseguire e pianificare processi con Snowflake Tasks o altri strumenti di orchestrazione.
- **Logica personalizzata:** gli utenti possono sfruttare le Snowpark User Defined Function (UDF) per semplificare l'architettura utilizzando complesse routine di elaborazione dei dati e logica aziendale personalizzata scritte in Python o Java nella stessa piattaforma in cui vengono eseguite le trasformazioni e le query SQL. Non è necessario gestire, scalare o utilizzare cluster separati.
- **Pipeline di data science e ML:** i team dei dati possono collaborare utilizzando il repository Anaconda e il gestore di pacchetti integrati per portare in produzione le pipeline di dati ML. È inoltre possibile pacchettizzare modelli ML addestrati come UDF per eseguire l'inferenza del modello in prossimità dei dati, accelerando i percorsi dallo sviluppo del modello alla produzione.

CASO D'USO: TRASFORMAZIONE DEI DATI CON DBT

Trasformare i dati significa pulire, combinare e modellare i dati originali perché possano essere utilizzati a valle. Storicamente, il metodo più comune per trasformare i dati utilizzava il linguaggio SQL, e gli specialisti dei dati creavano pipeline di trasformazione dei dati utilizzando SQL, spesso con l'ausilio di strumenti ETL/ELT. Più di recente, tuttavia, molti hanno iniziato ad adottare a questo scopo anche l'API DataFrame in linguaggi come Python. Nella maggior parte dei casi, uno specialista dei dati può svolgere le stesse trasformazioni utilizzando entrambi gli approcci, la scelta dipende essenzialmente dalle preferenze personali e dagli specifici casi d'uso. In alcune situazioni, tuttavia, una determinata trasformazione dei dati non può essere espressa in SQL ed è necessario un approccio diverso. L'approccio più diffuso per questi casi d'uso è una combinazione di Python e di un'API DataFrame.

Oggi una delle soluzioni più frequenti per la trasformazione dei dati è dbt, che supporta un flusso di lavoro di trasformazione basato principalmente su SQL e nel 2022 ha introdotto il supporto per Python. Poiché dbt supporta sia SQL che Python, gli utenti possono scrivere trasformazioni nel linguaggio che ritengono più familiare e adatto allo scopo. Inoltre, con dbt su Snowpark è possibile eseguire analisi utilizzando strumenti disponibili nell'ecosistema open source di Python, tra cui pacchetti all'avanguardia per il data engineering e la data science, sempre all'interno del framework dbt a cui molti utenti SQL sono abituati.

CASO D'USO: FORNIRE DATI AFFIDABILI CON TALEND

La capacità di analisi di un'organizzazione dipende direttamente dalla qualità dei dati. Come misurarla? Ora è possibile eseguire un controllo dell'integrità dei dati all'interno di Snowflake utilizzando Data Trust Score™ di Talend, una funzionalità fondamentale del prodotto Talend Data Inventory. Trust Score™ aiuta a identificare e diagnosticare i principali problemi dei dati e favorisce una crescente fiducia nel processo decisionale da parte di tutti gli stakeholder chiave.

Normalmente, per profilare una grande quantità di data set con un'applicazione esterna, si estrae un campione di dati che verrà quindi analizzato in un sistema diverso da quello in cui sono memorizzati i dati. Lo spostamento dei dati al di fuori di Snowflake comporta rischi di sicurezza e privacy dei dati, costi di ingresso e uscita dei dati ed errori di elaborazione dovuti a risorse non scalabili; inoltre i risultati sono necessariamente imprecisi, perché basati su un campione di dati.

Il calcolo del punteggio Trust Score™ per i data set Snowflake mitiga molti di questi problemi, poiché i calcoli avvengono in modo nativo all'interno di Snowflake utilizzando le UDF Java, ed è estremamente preciso, perché i risultati si basano sull'intero data set, non su un campione. Le UDF Java possono essere richiamate utilizzando **Snowpark** o **SQL**. Talend ha utilizzato entrambi gli approcci: uno durante la fase di prototipazione e l'altro durante l'operationalizzazione della funzionalità in Talend Data Inventory.

CASO D'USO: GESTIRE IL CICLO DI VITA DELLO SVILUPPO CON DATAOPS.LIVE

Sempre più spesso, le funzionalità avanzate di Snowpark sono sfruttate per creare data product. Una partnership estesa tra DataOps.live e Snowflake consente di gestire il workload Snowpark insieme a tutti gli altri oggetti in Snowflake durante l'intero ciclo di sviluppo, test e produzione: da potenti operazioni di data engineering e il riaddestramento automatico periodico di modelli ML all'esecuzione di vere e proprie inferenze quando nuovi dati vengono caricati nelle pipeline. Oltre alla comune "fonte di riferimento" memorizzata nel repository Git DataOps.live, il repository soddisfa anche tutti i normali requisiti per lo sviluppo software. DataOps.live consente di ramificare, controllare le versioni, compilare, testare e distribuire il software e produrre artefatti esattamente come per qualsiasi altro progetto software.

In molti casi, un'applicazione Snowpark eseguirà la manipolazione avanzata dei dati che supera le possibilità di SQL, ma i risultati saranno sempre archiviati in Snowflake. In questi casi, è possibile utilizzare il test automatizzato dei dati all'interno del motore di modellazione e trasformazione di DataOps.live per convalidare i risultati dell'applicazione Snowpark.

CASI D'USO DI SNOWPARK PER LA DATA SCIENCE

I progressi di Snowpark in termini di programmabilità dei dati migliorano la flessibilità e l'estendibilità. I data scientist possono impiegare in Snowflake i propri linguaggi di programmazione preferiti, come Python, per accedere, visualizzare ed elaborare i dati all'interno dei propri flussi di lavoro di machine learning. Questo aiuta le organizzazioni a massimizzare il valore dei dati, inclusi i dati non strutturati e di terze parti.

Indipendentemente dal linguaggio utilizzato, l'infrastruttura intelligente di Snowflake gestisce la scalabilità e il tuning delle prestazioni, consentendo ai data scientist di dedicare più tempo alla creazione di modelli. Questo include l'addestramento su vasta scala con un unico nodo utilizzando warehouse Snowpark-optimized, che dispongono di una quantità di memoria 16 volte maggiore e una cache locale 10 volte più grande rispetto ai warehouse standard e sono ideali per le operazioni ad alta intensità di memoria, come l'addestramento e l'inferenza. Le External Functions possono fornire dati a modelli ospitati esternamente in cui i data scientist stanno già eseguendo previsioni online o a bassa latenza.

È possibile automatizzare tutti i passaggi che trasformano i dati grezzi in insight forniti da ML utilizzando Streams e Tasks di Snowflake, ma naturalmente è anche possibile integrarli con altri framework per la trasformazione e l'orchestrazione come dbt e Apache Airflow.

E Snowflake va ancora oltre, distribuendo il modello con l'opzione di creare app utilizzando esclusivamente Python con Streamlit, un'integrazione attualmente in private preview.

L'estendibilità di Snowflake consente all'utente di utilizzare il suo strumento di sviluppo preferito ben al di là degli IDE e include piattaforme leader che utilizzano Snowflake sia per la memorizzazione che per l'elaborazione dei dati. Si ottiene così una user experience ottimale attraverso altre piattaforme, come notebook Hex, piattaforme MLOps Dataiku, feature store Tecton o perfino piattaforme low-code che estendono il valore del Data Cloud di Snowflake a molti utenti dell'organizzazione.

CASO D'USO: STRUMENTI E INFRASTRUTTURA UNIFICATI PER SQL E PYTHON CON HEX

SQL e Python sono tra i linguaggi più diffusi nei moderni stack di dati per le trasformazioni, l'analisi e il ML. Se da tempo SQL è il linguaggio di database standard per le query e le trasformazioni dei dati, Python è emerso come linguaggio di programmazione preferito per data science e ML. Spesso quando si utilizzano più linguaggi è necessario combinare diversi strumenti per completare una sola analisi, una situazione che può risultare laboriosa e frustrante per i professionisti dei dati.



Figura 2: integrazione di Hex con Snowflake per data science e analisi

Hex è una piattaforma moderna per l'analisi e la data science che semplifica la connessione ai dati, l'analisi in notebook collaborativi basati su SQL e Python e la condivisione del lavoro sotto forma di dashboard e visualizzazioni dei dati. Per assicurare agli utenti una scalabilità rapida e quasi illimitata dell'elaborazione, Hex trasferisce l'elaborazione dei dati alla piattaforma Snowflake (figura 2) invece di caricare i dati in un notebook, che può essere lento e soggetto a ritardi. Utilizzando l'area di lavoro per l'analisi di Hex, bastano pochi minuti per estrarre insight dai dati in Snowflake con Python e SQL.

CASO D'USO: SCALARE LE FUNZIONI DI PREPARAZIONE DEI DATI E LA VALUTAZIONE DEL MODELLO PREDITTIVO CON DATAIKU

Dataiku fornisce al business e ai team dati una ricca interfaccia visiva per preparare i dati e creare modelli predittivi in una pipeline di dati visiva dettagliata. Indipendentemente dall'approccio preferito, senza codice, low-code o full code, i data analyst, gli esperti di dominio o i data scientist possono utilizzare Dataiku con un potente strumento AutoML per trovare il modello migliore, con supporto completo per notebook di codice, IDE, Git e strumenti CI/CD. Con Dataiku, gli utenti possono accedere ai dati senza configurazioni aggiuntive direttamente in Snowflake, senza codice o programmando nel loro linguaggio preferito, compreso Python con Snowpark. L'integrazione Snowpark di Dataiku consente agli utenti di preparare i dati, valutare e distribuire i modelli utilizzando l'interfaccia di Dataiku senza che i dati escano da Snowflake, come illustrato nella figura 3. Tutte le operazioni di calcolo su data set grandi e piccoli vengono trasferite a Snowflake per un'elaborazione sicura e controllata. L'integrazione di Dataiku con Snowflake consente ai team dei dati, di dominio e IT di collaborare in un unico ambiente per creare e distribuire progetti di data science su vasta scala.

DATAIKU E SNOWFLAKE IN AZIONE

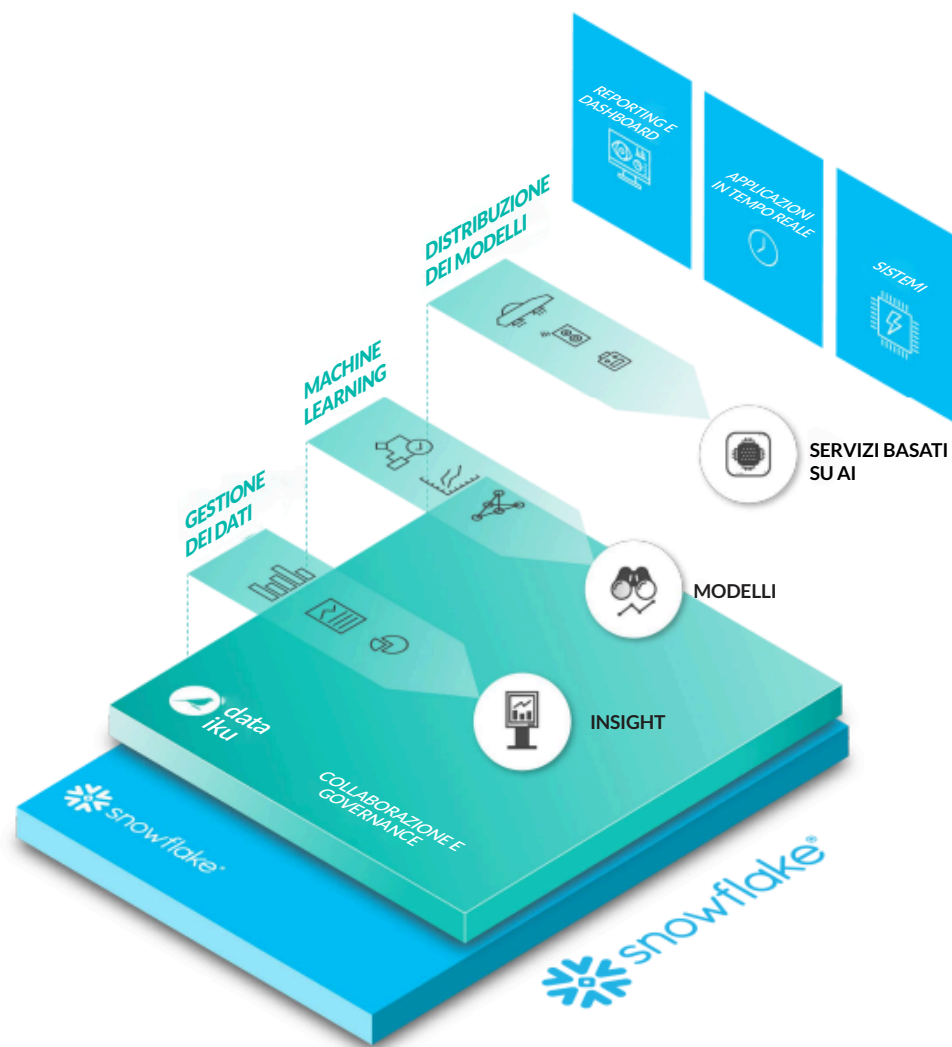


Figura 3: l'integrazione di Dataiku consente agli utenti di preparare i dati, valutare e distribuire i modelli utilizzando i dati in Snowflake

CASO D'USO: GESTIONE E DEPLOYMENT DEL ML SU LARGA SCALA CON GLI ARCHIVI DI FUNZIONALITÀ

Quando i data scientist costruiscono nuovi modelli ML, devono anche preparare i dati e creare le funzionalità. Per creare le funzionalità è necessario raccogliere i dati e preparare le colonne di dati in un formato specifico che può essere integrato nei modelli di machine learning. Dopo avere generato le funzionalità per un modello, i data scientist devono riscrivere le stesse funzionalità oppure dedicare tempo alla ricerca di funzionalità esistenti da utilizzare nel modello successivo.

Fortunatamente si è verificato un rapido aumento dell'adozione degli archivi di funzionalità, repository centrali che aiutano a migliorare la ricercabilità, la collaborazione e la scalabilità delle funzionalità di ML. Al loro interno, i data scientist possono trovare rapidamente funzioni trasformate e pronte per l'uso, accelerando in tal modo i tempi di sperimentazione e produzione. Tra gli altri vantaggi, un archivio di funzionalità permette di aumentare la collaborazione tra i team, riutilizzando il lavoro di altri data scientist, e riduce il tempo e l'impegno necessari per distribuire un modello addestrato in un ambiente di produzione, in quanto i data scientist non devono più ridefinire quella che spesso è una pipeline di dati già esistente.

Snowflake offre due approcci per la costruzione degli archivi di funzionalità, ed entrambi evitano la creazione di nuovi sistemi o silos di dati tra data scientist e data analyst.

Il primo approccio consiste nell'utilizzare una soluzione open source come Feast come interfaccia dell'archivio di funzionalità, mentre Snowflake diventa l'archivio e il motore per le funzioni. Le funzionalità rimangono sulla singola piattaforma dati, supportate da qualsiasi strumento di ingestione, ELT e catalogazione esistente. Le organizzazioni controllano sia la gestione delle pipeline di dati sia l'interfaccia utilizzata dai data scientist per scoprire dati e funzionalità e accedervi in un'unica soluzione centralizzata e scalabile.

Il secondo approccio prevede il ricorso a un archivio di funzionalità su Snowflake. I dati dei clienti rimangono nella loro forma grezza e modellata nel Data Cloud di Snowflake, la trasformazione dei dati viene eseguita in Snowflake utilizzando SQL o Snowpark per Python, mentre l'orchestrazione e la gestione della pipeline sono astratte dal fornitore dell'archivio di funzionalità. In questo ambito, Snowflake collabora tra gli altri con Tecton e Iguazio.



CASI D'USO DI ALTRI PARTNER SNOWFLAKE

Snowpark dispone di un solido ecosistema di partner che consente a data engineer, data scientist e sviluppatori che preferiscono altri linguaggi, inclusi Python, Java e Scala, di sfruttare le potenti funzionalità della piattaforma Snowflake e i vantaggi del Data Cloud di Snowflake. Queste esperienze profondamente integrate accelerano la creazione e la distribuzione di pipeline, modelli e app. Ulteriori informazioni sulle integrazioni dei partner sono disponibili alla pagina [Snowpark Accelerated](#).

INIZIA IL TUO PERCORSO CON SNOWPARK

Per iniziare a utilizzare Snowpark, consulta la [documentazione](#) e il [quickstart dettagliato](#).
Non vediamo l'ora di scoprire le cose meravigliose che riuscirai a creare.





INFORMAZIONI SU SNOWFLAKE

Snowflake permette a ogni organizzazione di mobilitare i propri dati grazie al Data Cloud. I clienti utilizzano il Data Cloud per unificare i dati contenuti nei silos, esplorare e condividere i dati in totale sicurezza, potenziare le applicazioni basate sui dati, ed eseguire diversi workload di AI/ML e analitici. Ovunque siano i dati o gli utenti, Snowflake offre un'esperienza sui dati unica che si estende a più cloud e aree geografiche. Migliaia di clienti di ogni settore, tra cui 639 della classifica 2023 Forbes Global 2000 (G2K) al 31 luglio 2023, utilizzano il Data Cloud di Snowflake per far crescere le loro aziende.

Scopri di più su [snowflake.com](https://www.snowflake.com)



© 2023 Snowflake Inc. Tutti i diritti riservati. Snowflake, il logo Snowflake e tutti gli altri nomi di prodotti, funzioni e servizi Snowflake menzionati nel presente documento sono marchi o marchi registrati di Snowflake Inc. negli Stati Uniti e in altri Paesi. Tutti gli altri nomi di marchi o loghi menzionati o usati nel presente documento sono a puro scopo identificativo e possono essere marchi registrati dei rispettivi proprietari. Snowflake non può essere associato, sponsorizzato o sostenuto da tali proprietari.