

SNOWPARK : CONSTRUIRE DE MEILLEURS PIPELINES ET MODÈLES DE DONNÉES DANS LE DATA CLOUD



EBOOK

TABLE DES MATIÈRES

- 3** Sommaire
- 5** Comment construire de meilleurs pipelines et modèles de données avec Snowpark
- 7** Cas d'usage de Snowpark pour le data engineering
- 9** Cas d'usage de Snowpark pour la data science
- 12** Cas d'usage d'autres partenaires Snowflake
- 13** Snowpark : c'est à vous de jouer
- 15** À propos de Snowflake

SOMMAIRE

Snowflake a commencé son parcours vers le Data Cloud en réorganisant complètement le monde des données et en repensant la manière dont un système de traitement des données fiable, sécurisé, performant et élastique doit être bâti pour le cloud.

Snowpark, un nouvel environnement de développement pour Snowflake, permet à tous les utilisateurs de données de centraliser leur travail dans le Data Cloud Snowflake avec une prise en charge native de Python, SQL, Java et Scala, comme le montre la figure 1. Snowpark permet aux data engineers, aux data scientists et aux développeurs d'écrire du code dans le langage de leur choix, et d'exécuter des pipelines, des opérations de ML et des applications de données plus rapidement et de manière plus sécurisée, sur une plateforme unique. Grâce à Snowpark, les avantages de Snowflake ne se limitent plus aux seuls utilisateurs de SQL, mais bénéficient à l'ensemble de vos différentes équipes Data. Les utilisateurs de Python, Java et Scala peuvent désormais exploiter les performances, l'élasticité et les capacités de gouvernance du moteur de traitement de Snowflake.

Les développeurs interagissent avec Snowflake via les trois composants principaux de Snowpark :

- **Les API DataFrame Snowpark.** Créez des requêtes à l'aide de DataFrames familiers à partir de l'interface utilisateur Snowflake (preview) ou de l'IDE de votre choix ; les processus push-down bénéficient des performances et de l'évolutivité du moteur de traitement élastique de Snowflake.
- **Les UDF Snowpark.** Exécutez votre logique personnalisée écrite en code natif Python ou Java directement dans Snowflake à l'aide de fonctions définies par l'utilisateur (UDF).
- **Les procédures stockées.** Opérationnalisez et orchestrez vos pipelines Python, Java ou Scala avec votre logique personnalisée directement dans Snowflake, puis mettez-les à la disposition de vos utilisateurs de SQL.

La puissance de Python résidant dans la richesse de son écosystème de packages open source, Snowpark for Python est préinstallé avec les packages les plus utilisés d'Anaconda et intégré au gestionnaire de packages Conda, ce qui épargne aux utilisateurs tout souci d'installation manuelle ou de dépendances manquantes.

Que pouvez-vous construire avec Snowpark ? Nous présentons quelques exemples de cas d'usage dans cet eBook, mais les possibilités sont quasi infinies. Passons à la pratique.

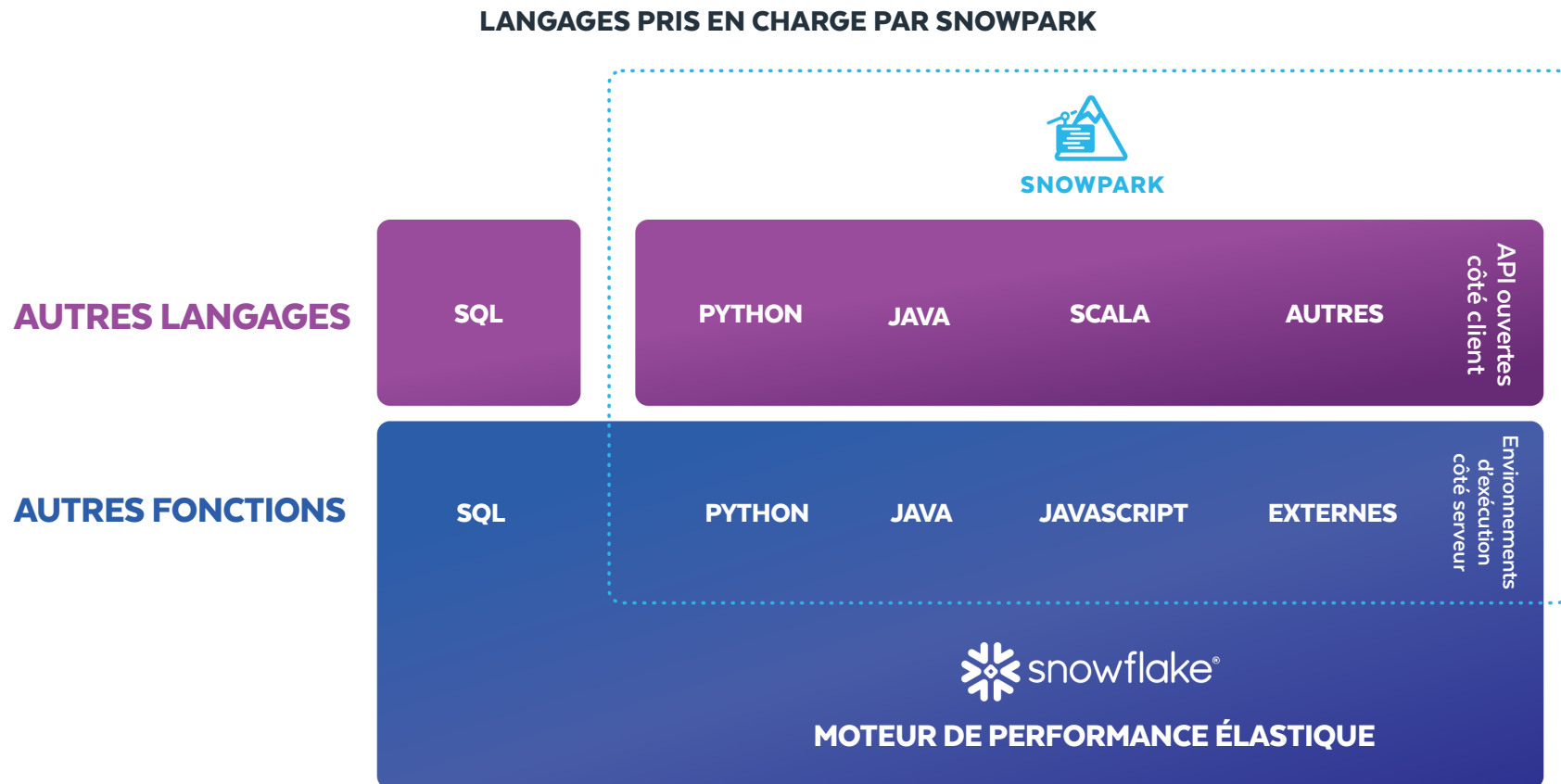


Figure 1 : Snowpark permet aux développeurs et aux ingénieurs d'interagir directement avec Snowflake sans avoir à déplacer les données.

COMMENT CONSTRUIRE DE MEILLEURS PIPELINES ET MODÈLES DE DONNÉES AVEC SNOWPARK

Pour comprendre comment Snowpark fonctionne, prenons un exemple générique avec des informations personnellement identifiables (PII).

DES REQUÊTES RATIONALISÉES, DES CONVERSIONS SANS FAILLE

Avec l'API Snowpark, les développeurs peuvent construire des requêtes en utilisant des DataFrames dans leur code sans avoir à créer ni à transmettre des chaînes SQL, par exemple :

```
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()
```

L'API Snowpark fournit un support de pointe dans l'environnement de développement, y compris la vérification de type, IntelliSense et les rapports d'erreur. En coulisses, toutes les opérations DataFrame sont converties de manière transparente en requêtes SQL, qui sont transmises au moteur de traitement évolutif Snowflake.

Snowpark permet également d'écrire une logique Python personnalisée à l'aide d'UDF et de procédures stockées, qui s'exécutent directement dans Snowflake. Voici un exemple qui utilise un code Python personnalisé pour masquer des PII :

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

En utilisant l'API Snowpark, vous pouvez facilement classer ce code comme une **fonction définie par l'utilisateur (UDF)**. Vous pouvez ensuite l'utiliser dans des opérations DataFrame :

```
mask_pii_udf = udf(mask_pii)

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

L'API Snowpark pousse votre logique vers Snowflake, afin qu'elle s'exécute à côté de vos données dans un bac à sable Python sécurisé.

L'AUTOMATISATION ÉLIMINE LE CODAGE À LA MAIN

Pour voir comment Snowpark simplifie les opérations courantes, voyons comment il vous permet d'appliquer votre logique de détection des PII dans toutes les colonnes de chaînes d'une table. Avec SQL, vous devez coder manuellement une requête pour chaque table ou écrire du code pour générer la requête. Avec Snowpark, vous pouvez facilement écrire une routine générique :

```
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))
```

Vous pouvez ensuite utiliser cette routine générique pour masquer facilement toutes les PII de n'importe quelle table :

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

Il vous suffit de ces quelques lignes de code pour que Snowpark génère dynamiquement une requête fiable et axée sur les schémas.

CONSTRUIRE UNE LOGIQUE COMPLEXE AVEC DES UDF PYTHON

Avec les UDF Python, vous pouvez exécuter une logique complexe qui expose une interface de fonction simple :

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

Pour créer ces fonctions, vous pouvez utiliser vos outils existants, notamment le contrôle des sources, les environnements de développement et les outils de débogage, ainsi que toutes les bibliothèques dont vous avez besoin. Snowpark vous donne la possibilité de récupérer du code utile de GitHub ou d'autres sources et de l'utiliser dans Snowflake.

L'utilisation de votre code Python dans SQL est simple. Créez un module, importez-le dans Snowflake et enregistrez une fonction.

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

Maintenant, n'importe quel utilisateur de SQL peut utiliser la logique que vous avez construite au même titre que toutes les autres fonctions :

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

Vous pouvez également créer des UDF en utilisant une approche purement Python pour que Snowpark regroupe votre fonction Python et la déploie côté serveur en utilisant le décorateur `@udf` et la fonction `UDF.register`.

Les UDF Python vous permettent d'utiliser vos outils existants pour les cas complexes. Mais parfois, votre cas d'usage reste classique. C'est pourquoi cette nouvelle fonctionnalité inclut également des définitions inline simples :

```
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 $$
class Reverse:
    def reverse(s):
        return s[::-1]
$$
```



CAS D'USAGE DE SNOWPARK POUR LE DATA ENGINEERING

Le data engineering est complexe pour deux raisons principales. Premièrement, les data engineers doivent souvent utiliser plusieurs outils et infrastructures pour chaque langage de programmation, ce qui rend l'architecture des pipelines de données beaucoup trop compliquée. La prise en charge de plusieurs langages découle essentiellement du fait que le data engineering est un travail de collaboration entre différentes équipes. Les data analysts optent davantage pour le SQL et des outils à interface graphique. De leur côté, les data scientists privilégient généralement Python et les notebooks pour préparer leurs données, tandis que les data engineers et les développeurs sont demandeurs d'outils supplémentaires pour traiter du code et des constructions de programmation complexes. Les données doivent souvent transiter par ces différents systèmes pour que les pipelines fonctionnent, ce qui conduit à des architectures complexes susceptibles de compromettre la sécurité et de nuire à la gouvernance des données.

Deuxièmement, gérer et exploiter une infrastructure de traitement des données implique en règle générale un nombre important d'opérations manuelles et des frais de maintenance élevés. En conséquence, les data engineers sont souvent dispersés, et passent la plupart de leur temps à maintenir et à réparer des pipelines.

Snowpark, un environnement de développement de Snowflake, permet aux data engineers de construire des pipelines simples, gouvernés et rapides dans leur langage de programmation préféré. Snowpark offre aux data engineers trois avantages majeurs : une plateforme unique qui prend en charge plusieurs langages sans calcul externe, des capacités de gouvernance supérieures et des pipelines plus rapides, moins coûteux et plus résilients. L'infrastructure intelligente de Snowflake élimine la complexité à laquelle les data engineers sont souvent confrontés. Son fonctionnement fiable leur permet de se concentrer sur l'essentiel.

Les data engineers utilisent habituellement Snowpark pour les opérations suivantes :

- **ETL/ELT** : les équipes Data peuvent utiliser Snowpark pour transformer des données brutes en formats modélisés, quel que soit leur type, y compris JSON, Parquet et XML. Toutes les transformations de données peuvent ensuite être regroupées sous forme de procédures stockées Snowpark pour exécuter et planifier des tâches gérées par Snowflake Tasks ou d'autres outils d'orchestration.
- **Logiques personnalisées** : les utilisateurs peuvent exploiter des fonctions définies par l'utilisateur (UDF) de Snowpark pour rationaliser leur architecture avec des traitements de données complexes et une logique métier personnalisée écrite en Python ou en Java dans la même plateforme que celle qui exécute les requêtes et les transformations SQL. Il n'y a pas de clusters distincts à gérer, à adapter ou à exploiter.
- **Pipelines de data science et ML** : les équipes Data peuvent utiliser le référentiel et le gestionnaire de packages Anaconda intégrés pour collaborer à la mise en production de pipelines de données de ML. Les modèles de ML entraînés peuvent également être regroupés sous forme d'UDF pour exécuter l'inférence du modèle au plus près des données, ce qui permet d'accélérer la mise en production des modèles développés.

CAS D'USAGE : TRANSFORMATION DE DONNÉES AVEC DBT

La transformation de données consiste à prendre des données sources, à les nettoyer, à les combiner et à les modéliser en vue d'une utilisation en aval.

Le langage SQL constitue traditionnellement la méthode la plus répandue pour transformer des données, et les professionnels du domaine construisent souvent des pipelines de transformation de données en SQL au moyen d'outils ETL/ELT. Pourtant, nombre d'entre eux commencent à adopter des API DataFrame dans des langages comme Python pour exécuter ce type d'opération. De façon générale, un professionnel des données peut réaliser les mêmes opérations de transformation de données, qu'il utilise l'une ou l'autre de ces approches. Le choix entre les deux relève principalement d'une question de préférence et des cas d'usage concernés. Cela dit, le langage SQL ne permet pas toujours de traiter certains cas d'usage spécifiques et une approche différente s'impose. Dans ce contexte, l'approche la plus répandue consiste à utiliser Python en association avec une API DataFrame.

Aujourd'hui, dbt représente l'une des solutions les plus prisées pour transformer des données. Cet outil intègre un flux de transformation SQL-first et, depuis 2022, prend également en charge Python. Grâce à la prise en charge de SQL et Python par dbt, les utilisateurs peuvent écrire des transformations dans le langage qui leur est le plus familier et le plus adapté. De plus, dbt sur Snowpark permet de procéder à des analyses en utilisant des outils de l'écosystème open source de Python, y compris des packages avancés pour le data engineering et la data science, le tout dans l'environnement dbt, familier à de nombreux utilisateurs de SQL.

CAS D'USAGE : FOURNIR DES DONNÉES FIABLES AVEC TALEND

Votre capacité d'analyse n'est bonne que si la qualité de vos données l'est aussi. Or, comment la mesurer ? Désormais, vous pouvez effectuer un bilan de santé de vos données au sein de Snowflake en utilisant le Data Trust Score™ de Talend en un coup d'œil, une fonctionnalité essentielle du produit Talend Data Inventory. Le Trust Score™ vous aide à identifier et à diagnostiquer les principaux problèmes liés à vos données, et aide toutes les parties prenantes clés à s'engager dans un parcours visant à renforcer la confiance dans votre prise de décision.

En général, pour établir le profil d'un grand ensemble de données par une application externe, un échantillon de vos données est extrait afin que l'analyse puisse être exécutée dans un système différent de celui où elles sont stockées. Le transfert de données en dehors de Snowflake présente ses propres défis : risques liés à la sécurité et à la confidentialité des données, frais d'entrée et de sortie des données, échec du traitement en raison de ressources non évolutives, et les résultats restent inexacts, simplement parce qu'ils sont basés sur un échantillon de données.

Le calcul du Trust Score™ pour les ensembles de données Snowflake atténue bon nombre de ces préoccupations, car les calculs de traitement du score se font nativement dans Snowflake à l'aide d'UDF Java. Ce score est très précis, car les résultats sont basés sur l'ensemble des données, et pas seulement sur un échantillon. Vous pouvez faire appel à des UDF Java en utilisant soit **Snowpark**, soit **SQL**. Talend a utilisé les deux approches : l'une pendant la phase de prototypage, et l'autre lorsque nous avons opérationnalisé la fonctionnalité dans notre produit Talend Data Inventory.

CAS D'USAGE : GESTION DU CYCLE DE VIE DU DÉVELOPPEMENT AVEC DATAOPS.LIVE

La création de produits de données fait de plus en plus appel aux capacités avancées de Snowpark. Un partenariat élargi entre DataOps.live et Snowflake permet aux workloads Snowpark d'être entièrement gérés avec tous les autres objets dans Snowflake tout au long du cycle de développement, de test et de production. Et les possibilités sont nombreuses, qu'il s'agisse d'un processus intensif de data engineering, d'un nouvel entraînement automatique et périodique de modèles de ML ou de l'exécution d'inférences réelles lors de l'introduction de nouvelles données dans les pipelines. Outre la « source de vérité » normale qui est stockée dans le référentiel Git de DataOps.live, le référentiel répond également à toutes les exigences habituelles en matière de développement de logiciels. Avec DataOps.live, vous pouvez créer des branches, des versions, compiler, tester et déployer des logiciels et produire des artefacts comme vous le faites pour tout autre projet logiciel.

Dans de nombreux cas, une application Snowpark effectuera un traitement avancé des données, allant au-delà de ce qui peut être réalisé avec SQL, mais les résultats seront toujours stockés dans Snowflake. Dans ce cas, le test automatisé des données au sein du moteur de modélisation et de transformation de DataOps.live peut être utilisé pour valider les résultats de l'application Snowpark.

CAS D'USAGE DE SNOWPARK POUR LA DATA SCIENCE

Les avancées en matière de programmabilité des données dans Snowpark offrent une plus grande flexibilité et évolutivité. Les data scientists peuvent utiliser leurs langages de programmation préférés, tels que Python, dans Snowflake pour accéder aux données, les visualiser et les traiter dans le cadre de leurs workflows de machine learning. Cette approche permet aux entreprises de maximiser la valeur des données, y compris celle des données non structurées et des données tierces.

Quel que soit le langage utilisé, l'infrastructure intelligente de Snowflake prend en charge la mise à l'échelle et le réglage des performances, ce qui permet aux data scientists de consacrer plus de temps à la construction de modèles. La plateforme offre notamment un entraînement à grande échelle sur un seul nœud via Snowpark-Optimized Warehouses, des entrepôts optimisés avec des capacités de mémoire et de cache local respectivement 16 et 10 fois supérieures à celles des entrepôts standard, une solution idéale pour les charges de travail qui ont de gros besoins en mémoire, comme l'entraînement et l'inférence. Les fonctions externes peuvent alimenter en données des modèles hébergés à l'extérieur et dans lesquels les data scientists peuvent déjà calculer des prédictions en ligne ou à faible latence.

L'automatisation de toutes les étapes qui transforment les données brutes en informations de ML peut être réalisée à l'aide de Streams et de Tasks de Snowflake. Elle peut également être intégrée à d'autres environnements de transformation et d'orchestration tels que dbt et Apache Airflow.

Snowflake va encore plus loin et permet de déployer des modèles en créant des applications uniquement via Python avec Streamlit, une intégration actuellement disponible en private preview.

Les possibilités d'extension offertes par Snowflake permettant de travailler à partir de l'outil choisi par l'utilisateur vont au-delà des IDE et s'étendent aux principales plateformes qui utilisent Snowflake pour stocker et traiter des données. Cette possibilité vous permet de bénéficier de l'expérience utilisateur exceptionnelle disponible via d'autres plateformes, telles que les notebooks Hex, les plateformes MLOps de Dataiku ou les Feature stores de Tecton, sans oublier les plateformes low-code, qui étendent les avantages du Data Cloud Snowflake à de nombreux autres utilisateurs au sein de votre entreprise.

CAS D'USAGE : OUTILS ET INFRASTRUCTURE UNIFIÉS POUR SQL ET PYTHON AVEC HEX

SQL et Python figurent parmi les langages les plus populaires dans la Modern Data Stack pour les transformations, l'analyse et le ML. Alors que SQL est depuis longtemps le langage de base de données pour l'interrogation et la transformation des données, Python s'est imposé comme le langage de programmation préféré pour la data science et le ML. Les data scientists et les data engineers doivent souvent combiner plusieurs outils pour mener à bien une seule analyse lorsqu'ils utilisent plusieurs langages, une opération qui peut s'avérer à la fois frustrante et chronophage.

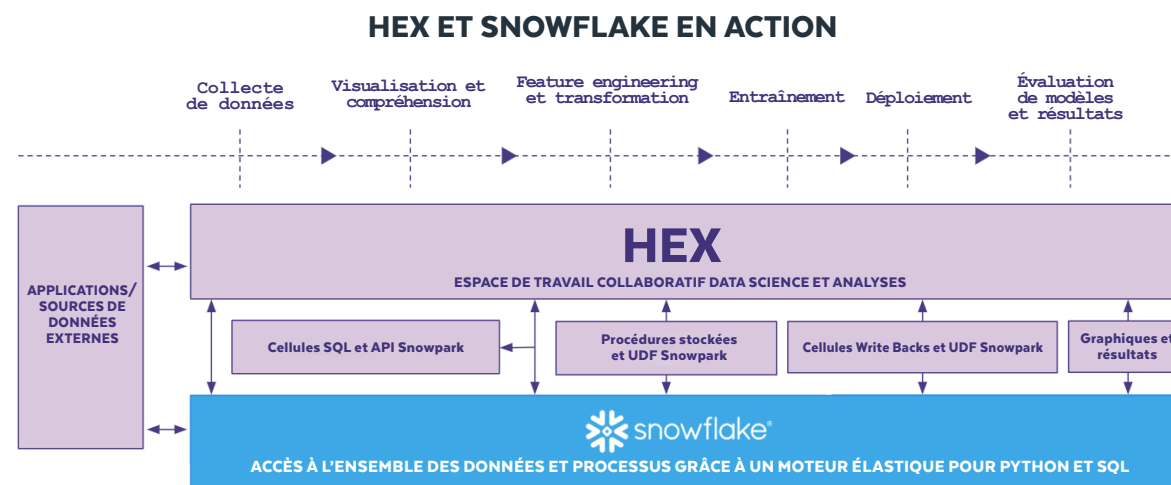


Figure 2 : Intégration de Hex avec Snowflake pour la data science et l'analyse

Hex est une plateforme moderne dédiée à l'analyse et à la data science, qui permet de se connecter facilement aux données et de les analyser dans des notebooks collaboratifs SQL et Python. Elle permet ensuite de partager le travail sous forme de cas d'usage et d'applications de données interactifs. Afin d'offrir aux utilisateurs une évolutivité de traitement rapide et quasi illimitée, Hex pousse le calcul des données vers la plateforme Snowflake (figure 2) au lieu de charger les données dans un notebook, une opération potentiellement lente et inefficace. Grâce à l'espace de travail analytique de Hex, il suffit de quelques minutes pour obtenir des informations sur les données de Snowflake à l'aide de Python et de SQL.

CAS D'USAGE : ÉTENDRE DES FONCTIONS DE PRÉPARATION DES DONNÉES ET DE NOTATION DES MODÈLES PRÉDICTIFS AVEC DATAIKU

Dataiku offre aux équipes métier et Data une interface visuelle riche pour préparer des données et construire des modèles prédictifs dans un pipeline de données visuel étape par étape. Les utilisateurs no-code, low-code et full-code, notamment les data analysts, les domain experts et les data scientists, peuvent utiliser Dataiku avec un AutoML puissant pour trouver le meilleur modèle avec prise en charge complète des notebooks de code, des IDE, de Git et des outils CI/CD. Avec Dataiku, les utilisateurs peuvent accéder aux données directement dans Snowflake sans configuration supplémentaire, soit en utilisant des recettes sans code, soit en programmant dans le langage de leur choix, y compris Python avec Snowpark. L'intégration de Snowpark par Dataiku permet aux utilisateurs de préparer des données, d'évaluer des modèles et de déployer des modèles à l'aide de l'interface de Dataiku sans déplacer de données hors de Snowflake, comme le montre la figure 3. Tous les calculs effectués à partir d'ensembles de données, petits ou grands, sont transférés vers Snowflake pour un traitement sécurisé et gouverné. L'intégration de Dataiku avec Snowflake permet aux équipes Data, IT et de domaines de collaborer dans un environnement unique pour créer et déployer des projets de data science à grande échelle.

DATAIKU ET SNOWFLAKE EN ACTION

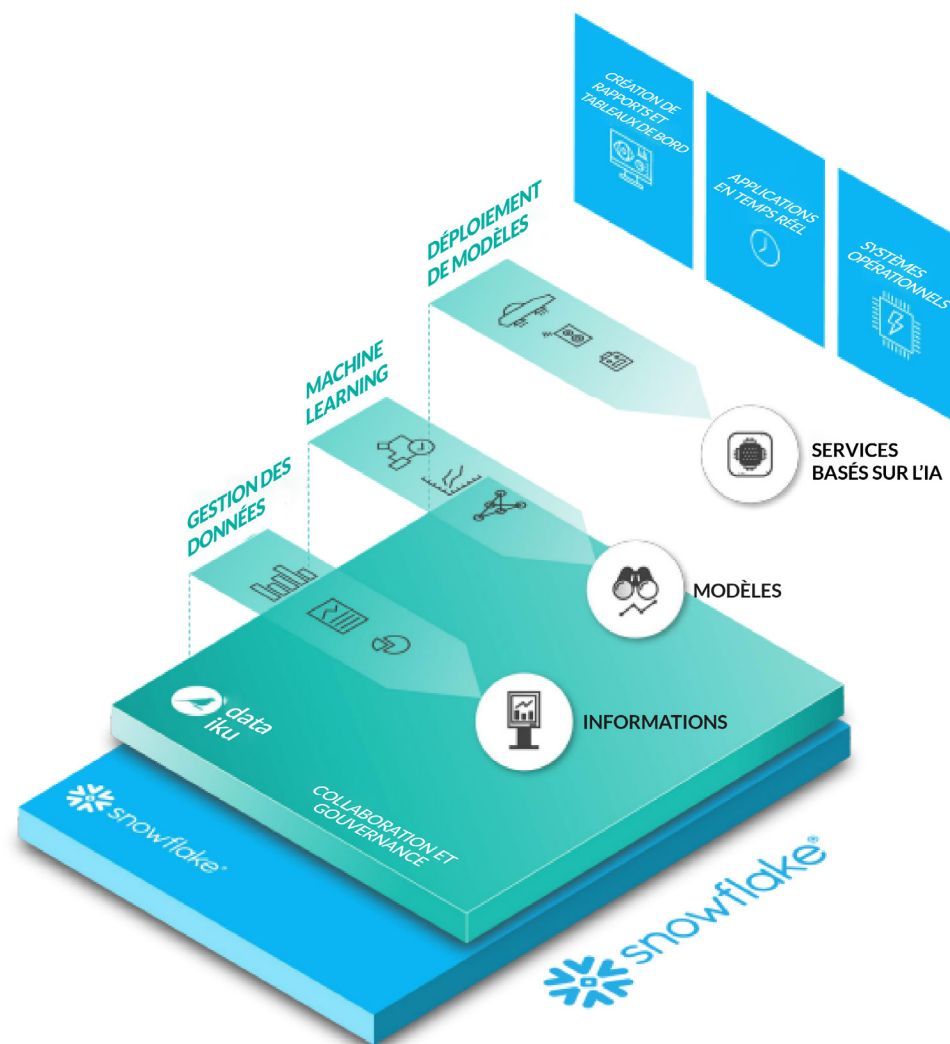


Figure 3 : L'intégration de Dataiku permet aux utilisateurs de préparer des données, d'évaluer des modèles et de les déployer en utilisant les données dans Snowflake

CAS D'USAGE : GESTION ET DÉPLOIEMENT DE FONCTIONNALITÉS DE ML À GRANDE ÉCHELLE VIA DES FEATURE STORES

Lorsque les data scientists développent de nouveaux modèles de ML, ils doivent traiter les tâches fastidieuses de préparation des données et de création des fonctionnalités. Les fonctionnalités sont créées en alimentant et en préparant des colonnes de données dans un format spécifique pouvant être injecté dans les modèles de machine learning. Une fois les fonctionnalités générées pour un modèle, les data scientists doivent s'atteler à réécrire les mêmes fonctionnalités ou passer du temps à rechercher des fonctionnalités existantes à utiliser pour le prochain modèle.

Heureusement, nous constatons une augmentation significative de l'adoption des feature stores, un référentiel central qui permet d'améliorer la recherche, la collaboration et l'évolutivité des fonctionnalités de ML. Ils permettent aux data scientists de trouver rapidement des fonctionnalités transformées et prêtes à l'emploi, accélérant ainsi les phases d'expérimentation et de mise en production. Les feature stores offrent de nombreux avantages. Ils optimisent notamment la collaboration entre les équipes par le biais de la réutilisation des travaux d'autres data scientists. Ils réduisent également le temps et les efforts requis pour déployer un modèle entraîné dans un environnement de production. En effet, les data scientists n'ont plus besoin de redéfinir ce qui est souvent un pipeline de données existant.

Snowflake fournit deux approches pour la création de feature stores, qui évitent de créer de nouveaux systèmes ou silos entre les data scientists et les data analysts.

La première approche consiste à s'appuyer sur une solution open source telle que Feast comme interface de feature store, tandis que Snowflake devient le magasin et le moteur des fonctionnalités. Les fonctionnalités sont conservées sur la même plateforme de données, avec l'aide d'outils d'ingestion, d'ELT et de catalogage existants. Avec cette solution, les entreprises maîtrisent la gestion des pipelines de données et l'interface sur laquelle les data scientists découvrent et parcourent les données et les fonctionnalités centralisées en un seul endroit évolutif.

La seconde approche consiste à utiliser une solution de feature store gérée rattachée à Snowflake. Les données client sont conservées dans leur forme brute et modélisée dans le Data Cloud de Snowflake, tandis que la transformation des données est exécutée dans Snowflake à l'aide de SQL ou Snowpark for Python, et l'orchestration et la gestion des pipelines sont abstraites par le fournisseur de feature store. Les partenaires Snowflake dans ce domaine sont Tecton et Iguazio.



CAS D'USAGE D'AUTRES PARTENAIRES SNOWFLAKE

Snowpark dispose d'un solide écosystème de partenaires qui permet aux data engineers, aux data scientists et aux développeurs qui préfèrent d'autres langages, notamment Python, Java et Scala, de profiter des capacités puissantes de la plateforme Snowflake et des avantages du Data Cloud Snowflake. Ces expériences profondément intégrées accélèrent la création et le déploiement de pipelines, de modèles et d'applications. Pour en savoir plus sur les intégrations de partenaires, consultez la [page](#) Snowpark Accelerated.

SNOWPARK : C'EST À VOUS DE JOUER

Pour commencer à utiliser Snowpark, veuillez consulter notre [documentation](#) et notre [guide de démarrage rapide](#) étape par étape. Nous sommes impatients de voir les incroyables projets que vous allez mettre en place.





À PROPOS DE SNOWFLAKE

Snowflake permet à chaque organisation de mobiliser ses données grâce au Data Cloud Snowflake. Ses clients utilisent le Data Cloud pour réunir au même endroit leurs données silotées, analyser et partager en toute sécurité les données, propulser des applications de données et exécuter diverses charges de travail analytiques et d'IA/ML. Quel que soit l'endroit où se trouvent les données ou les utilisateurs, Snowflake offre une expérience unique qui s'étend sur plusieurs clouds et régions. Au 31 juillet 2023, des milliers de clients de nombreux secteurs, dont 639 des Forbes Global 2000 (G2K) de 2023, utilisent le Data Cloud Snowflake pour dynamiser leur activité.

En savoir plus sur [snowflake.com](https://www.snowflake.com)



© 2023 Snowflake Inc. Tous droits réservés. Snowflake, le logo Snowflake et tous les autres noms de produits, de fonctionnalités et de services Snowflake mentionnés dans le présent document sont des marques déposées ou des marques commerciales de Snowflake Inc. aux États-Unis et dans d'autres pays. Tous les autres noms de marque ou logos mentionnés ou utilisés dans le présent document le sont uniquement à des fins d'identification et peuvent être des marques de commerce de leur(s) détenteur(s) respectif(s). Snowflake ne peut être associé à, ou être sponsorisé ou approuvé par, un tel détenteur.