FROM DATA WRANGLING TO FEATURE ENGINEERING

How to accelerate data access and preparation with a unified platform for analytics and data science teams

snowflake

WHITE PAPER

Introduction

Every year, insights from business analytics and machine learning (ML) have a bigger and bigger effect on how organizations solve business problems with data. However, the insights in a business dashboard or predictions from an ML model are only as valuable as the quality of the data behind them.

Building high-quality data sets is a multi-step process known as data wrangling. Also referred to as data munging, this process includes cleaning data, mapping data, and transforming data into a workable format. Data wranglers seek to eliminate faulty or incomplete data. They may also augment or enrich the data so either themselves or other business users can make better, more accurate decisions. These activities commonly involve the following:

- Merging multiple data sources into a single data set
- Identifying gaps in the data (for example, empty cells in a table) and either filling or deleting them
- Deleting data that's either unnecessary or irrelevant to the project at hand, such as removing duplicates
- Identifying extreme outliers in the data

To build an ML model that can solve a business objective through predictions or automated decisions, data scientists take data wrangling one step further through feature engineering. This is the process of generating attributes (features) from the available data in order to shape the data in a way that an ML model can understand, as well as to enhance the model's predictive performance. As described throughout this book, features are specific attributes in a data set that serve as the critical inputs for ML models.

Data wrangling and feature engineering are time-consuming processes that slow down the speed at which insights are generated. This ebook describes how analytics and data science teams can maximize efficiency by leveraging a cloud data platform to unify and govern both data wrangling and feature engineering activities. This type of data platform helps minimize the amount of time it takes to go from raw data to metrics and features, so data teams can spend more time generating new insights.

UNDERSTANDING DATA WRANGLING

Data wrangling ensures that only high-quality, complete data is used for analysis. In practice, data wrangling may involve merging data from several different sources, removing data not relevant to what's being analyzed, and identifying and addressing potential gaps in the data.

The beneficiaries of data wrangling include data application developers, data scientists, and ultimately the business users who make data-driven decisions. For example, analysts use high-quality sales data to populate dashboards and reports that help business decision-makers monitor trends in sales performance. Business analysts use data visualization tools to explore possibilities and discover insights in data. They're skilled at defining metrics, identifying key performance indicators (KPIs), and building dashboards and reports using SQL or a high-level business intelligence (BI) environment.

For small data sets, data analysts can use worksheets like Excel, write SQL queries, or use a GUI-based data preparation tool for these tasks. However, as data volumes grow, these data professionals may enlist help from data engineers, who are skilled at capturing relevant data and consolidating data at scale. For these larger data preparation jobs, they may use SQL and other programming languages such as Scala, Java, and Python. Data engineers may further prepare the data for use in downstream applications that help users manipulate the data within specific domains, such as financial reporting.

When data comes from different sources, data engineers and other skilled IT professionals may have to rationalize the differences. For example, one data source might have 100 columns, another might have 150 columns. Data wrangling involves matching up these disjointed data sets to eliminate null values, recalculate aggregations, remove duplicates, and otherwise prepare the data for smooth and accurate downstream analysis. There are six basic steps in this process:

- **Discover:** Discovery involves identifying relevant data sets, including determining which fields or features are available and defining the characteristics of those fields. For example, you need to verify that the data sets you want to join have the desired overlapping time frames in their date fields. Data wranglers investigate the raw data to gain a better understanding of the data set and figure out how it can be joined and used.
- **Structure:** Raw data is typically unorganized and may not be useful until it is transformed. Transformation is the process of converting raw data from one format to another, generally to meet the requirements of the destination platform or analytical model you're using. This is often necessary with unstructured data, such as when extracting fields from PDF documents and turning them into a tabular format for easy analysis.
- **Cleanse:** Data cleansing—also known as data cleaning or data scrubbing—fixes or removes common data errors, such as removing duplicates and outliers. This step is important in assuring the overall quality of the data.
- **Enrich:** You may decide to supplement your data by adding data from other sources, including first-party internal data, second-party data from partners, and third-party data from internet services and data marketplaces.
- **Validate:** Data validation entails checking your data for consistency, quality, and accuracy. It uses repetitive data programming sequences to uncover inconsistencies in the data and correct them.
- **Publish:** At the end of this data wrangling cycle, data is ready to be shared with others via reports, dashboards, and other business tools. Recipients include data analysts, data engineers, data scientists, and business users.

UNDERSTANDING FEATURE ENGINEERING

A feature is a unique attribute or variable in a data set that becomes an input to ML models. Feature engineering entails selecting the right variables from raw data to provide the most relevant inputs to these models. Like data wrangling, this iterative process involves selecting, transforming, extracting, combining, and manipulating data, but in this case the goal is to go one step further to establish pertinent variables for statistical analysis or predictive modeling.

Data scientists apply math, statistics, and analytics to identify useful data sets and derive meaning from those data sets. They use a combination of tools, applications, libraries, notebooks, and languages to train ML *algorithms*—advanced math functions that map inputs to outputs in order to deliver actionable results.

Feature engineering makes specific transformations with the goal of feeding those transformed or generated attributes into an ML model. It puts data into numeric formats that allow organizations to solve specific problems using the power of ML, such as forecasting revenue, predicting churn, or identifying fraud. As an example, Figure 1 below shows the data inputs for an ML model designed to predict which customers are likely to churn over the next 30 days.

In order to provide a model with a data set that fits an expected type of distribution, data scientists may need to rescale the values of the features. The end result is a two-dimensional table where each observation is a row (such as "customer") and each column is a feature (see Figure 2).

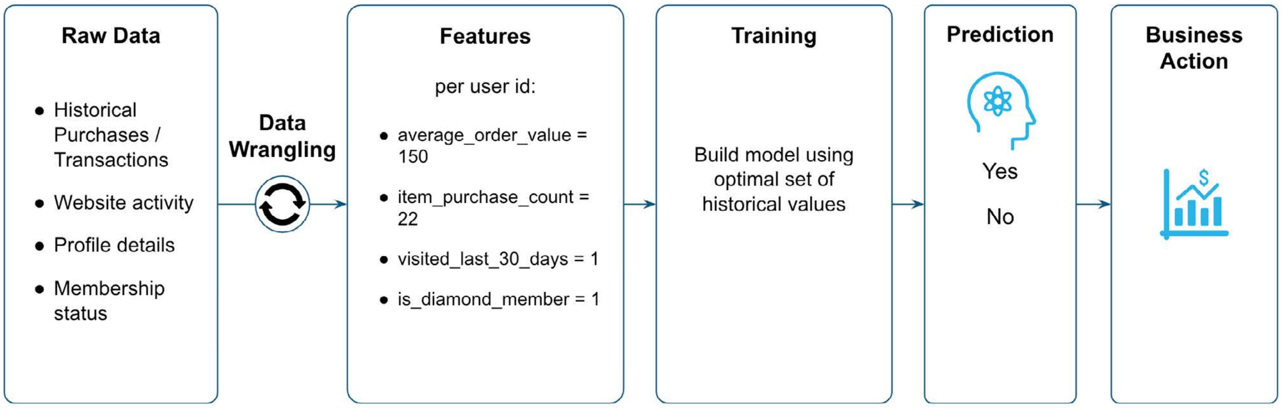


Figure 1: Feature engineering puts data into a format that allows organizations to solve specific problems using the power of machine learning, such as forecasting revenue or identifying fraud. In this case, raw data from ecommerce activities forms the basis for features that can predict customer churn.

customer id	sum_purchases_ last_7_days	count_items_bought_ last_7_days	has_visited_site_ last_30_days	is_diamond_ member
10012	150	22	1	1
10013	45	1	0	0
...				

Figure 2: Feature engineering puts the data into tables to better reflect a desired set of outcomes. All data must be presented numerically to be read by the algorithms.

There are many types of features. Some feature operations are compute-intensive and others are memory-intensive. *Derived* or *calculated* features are compute-intensive transformations, such as aggregations of purchases or features calculated from multiple table joins. *Representational* transforms are features that are calculated to turn human understandable data to machine understandable data, such as turning a variable from a text string into a numeric value.

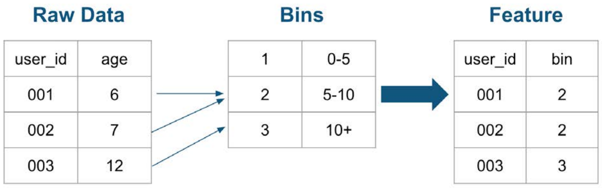
For compute-intensive features, pre-calculculating transformations (performing transformations as the data is loaded) speeds up queries since the processing engine does not have to perform these calculations at runtime, when queries are issued. For commonly used features, this method also saves compute cycles, since individual data scientists don't have to recreate their own transformation pipelines from scratch to train new models. It is most efficient to run compute-intensive operations prior to model inference, such as aggregations of which items customers have purchased. This way, the model inference step is not dependent on a lengthy set of data transformations that are needed in order to generate new predictions.

Memory-intensive operations can be calculated in the data platform's memory right before model training, such as converting text to numeric values (as in converting "diamond member" to "1," as shown in Figure 2. This minimizes storage requirements without compromising the speed at which new predictions are generated.

Pre-calculating features ensures consistency as models move from training to inference. Analysts and data scientists obtain fast access to data to experiment and build models, while optimizing the hardware and infrastructure used across different feature transformations. This reduces storage costs without compromising the time it takes to obtain results.

DATA TRANSFORMATION OPERATIONS

In addition to mentioned data-wrangling activities, feature engineering involves many additional transformations unique to ML, such as binning, log transforms, encoding, feature splitting, scaling, and grouping operations.



Raw Data

user_id	age
001	6
002	7
003	12

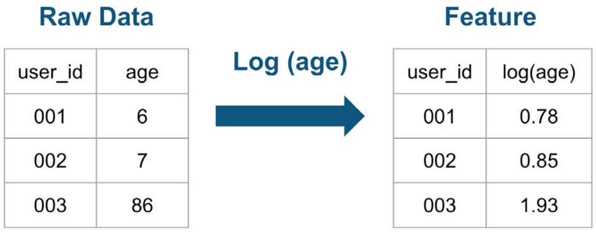
Bins

1	0-5
2	5-10
3	10+

Feature

user_id	bin
001	2
002	2
003	3

- **Binning** involves categorizing continuous features into distinct groups. This allows a data scientist to engineer categories and features that better emphasize important trends in the data. For example, you might divide numeric data referring to age into a few different buckets, such as 0 to 18, 19 to 35, 36 to 55, 56 to 75, and 75 to 100.



Raw Data

user_id	age
001	6
002	7
003	86

Log (age)

Feature

user_id	log(age)
001	0.78
002	0.85
003	1.93

- **Log transforms** are mathematical transformations that help data scientists normalize skewed data, such as to adjust the magnitude within a given range of data. For example, average income distributions tend to be heavily skewed, as some individuals make significantly more than others. Log transformations can reduce the effects of skewed distributions and normalize these magnitudes within a data set.

Raw Data

user_id	Car_ type
001	SUV
002	SUV
003	Sedan

One-Hot Encoding

Feature

user_id	SUV	Sedan
001	1	0
002	1	0
003	0	1

- **One-hot encoding** is a common encoding method that spreads the data values in a column to multiple flag columns with binary values (0 or 1) to transform one or more categorical values into numerical values.

Raw Data

user_id	$_spent
001	$9.46
001	$12.13
001	$6.15
002	$8.56

Average Order Value

Feature

user_id	AOV
001	$9.31
002	$8.56

- **Grouping operations** combine data by *instances* so that every instance is represented by only one row. In most ML algorithms, every instance is represented by a row in the training data set, and every column depicts a different feature of the instance. Common examples of numerical values include *sum*, *average*, and *max/min*. For example, to represent all the purchases of one customer, so that each row represents a customer, the average number of transactions per week can be calculated as a feature.

Raw Data

name
Alison Norman
Eugene Molina
Dominick Dawson

Feature Splitting

Feature

first name	last name
Alison	Norman
Eugene	Molina
Dominick	Dawson

- **Feature splitting** entails breaking up or extracting a single field from an existing data value (e.g., from a field that has full names, a new feature called first name can be generated).

Raw Data

user_id	value
001	10
002	23
003	52
004	95

Scale Feature based on Min and Max

Feature

user_id	value
001	0
002	0.15
003	0.49
004	1

- **Scaling** allows two or more numeric columns to be compared even if they have a different range of values. For example, the *age* and *income* columns will probably not have the same range. One way to do this is to normalize variables by subtracting the mean and dividing by the standard deviation.

Raw Data

date
2022-01-19
2022-02-25
2022-03-15

Separate Data into Columns

Feature

month	day	year
01	19	2022
02	25	2022
03	15	2022

- **Extracting date** operations allow different types of date formats to be correctly presented to ML algorithms. Extracting each part of the data into its own column (i.e., one column for month and one for day) makes it possible for ML algorithms to process them accurately.

MACHINE LEARNING: FROM EXPERIMENTATION TO PRODUCTION

The ML workflow is broadly divided into experimentation and production. During the experimentation phase, data scientists conduct feature generation on a subset of data to train the model. Once they have demonstrated that the experiment is successful, they must scale the model as they move it to the production phase.

Once the model proves to be valuable and it is ready to be used to generate predictions as part of a process or application, features need to be generated to present them to the model for model inference (generating model predictions). All the data transformations must be repeated in a scalable and automated way so that the model can generate predictions as part of a software application. This can be a challenging process that requires data scientists, or teams of data engineers and ML engineers, to rewrite code to execute the feature engineering pipelines and ML algorithms on a more scalable processing platform. This cycle is depicted in Figure 3.

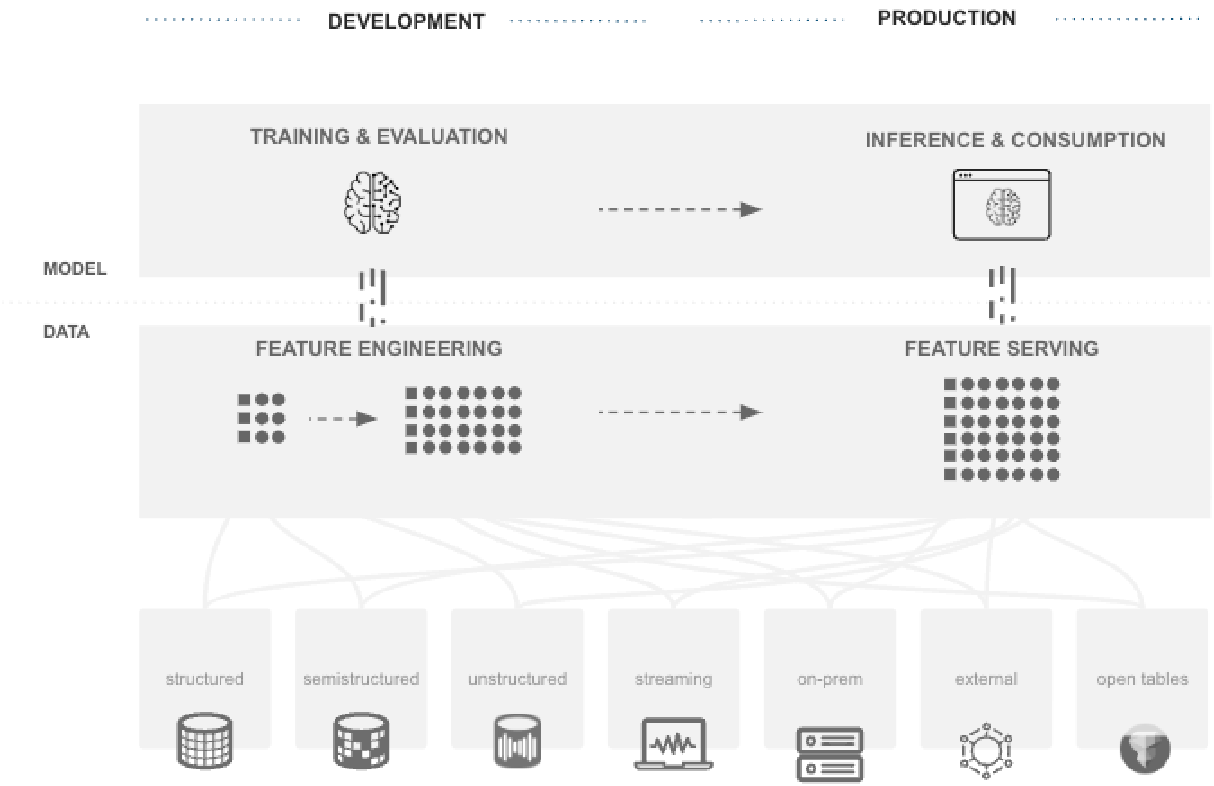


Figure 3: Scaling feature engineering from training to inference.

CASE IN POINT

The Data Science Workflow

Feature engineering is part of a larger workflow that includes multiple steps. It begins when data scientists are given a specific assignment or task, such as creating a model to forecast revenue. They may examine many different data sources, collect sample data, and experiment using tools such as an ML notebook to develop an experimental model. The experimentation process is carried to test a hypothesis, such as how they can use a particular data set to predict sales trends, with the ultimate goal of using those predictions to drive business value, such as how to optimize inventory or plan staffing schedules.

During the experimentation phase, data scientists test the hypothesis and seek to achieve accurate and relevant enough results that business teams can trust and leverage. Once a model yields valuable outcomes, and the business teams are aligned with the process in how the model will be used, they may work with data engineers and ML engineers to put the model into production.

Because of the multiple steps involved and multiple teams that must collaborate in order to put a model into production in a scalable and governed way, a complete data platform should support every aspect of the data science workflow: from collecting data to visualizing and understanding that data, engineering data features, training the model, and finally deploying it into production and monitoring the ability of the model to deliver business-ready analytics. This platform should also offer room for customization and frictionless integration to other tools that can extend a team's ability to deliver on the value of ML, including extensions to feature stores, ML monitoring tools, and governed access to a wide range of open source libraries and frameworks.

Ensuring Repeatable, Scalable Access to Metrics and Features

Whether its analysts calculating metrics through BI dashboards or data scientists generating features for a model, their work gets more challenging as the volume and complexity of data grows. As organizations scale these data processing activities, data scientists may work with data engineers to create automated, production-ready data pipelines. Data engineers must apply the same transformations that were used during model training to ensure consistency at inference time. Having a single data platform that provides data engineers, analysts, and data scientists with an elastically scalable processing engine facilitates collaboration and reduces code rewrites as experiments move into production. A data platform helps all team members to trust each others' work.

In many organizations, data scientists write data transformations using Python, a programming language with many open source libraries that accelerate code development. However, some of these libraries lack the security, scalability, and stability necessary for production applications. As a result, data engineers may have to rewrite data pipelines and feature engineering pipelines so they can be run in secure production environments.

Both data complexity and programming language issues are much easier to address if your data platform has the built-in scalability to store and process data regardless of programming language along with consistent governance controls. For example, extracting consistent features is much easier when you can store reusable code in a shared, cloud-based data repository. You can run SQL data pipelines, execute complex data transformations and ML algorithms using Python, , and effortlessly integrate applications and dashboards into the platform for consumption of model results.

In addition, storing features centrally helps your data science team to share and reuse ML artifacts, boosting overall efficiency. Data scientists can acquire the data they need to develop new applications quickly and train the associated ML models. Furthermore, being able to share and reuse artifacts within a broad data platform ecosystem makes it easier to move successful experiments into production.

And to address the challenges that come with using open source libraries, particularly, for feature transformations and ML models, the platform should provide a secure way to leverage those libraries in a production environment. For example, open source libraries such as scikit-learn can be used to streamline representational transforms (e.g., one-hot encoding) during model development but should also be used to quickly run those transformations in production for model scoring, rather than having to write custom code. Having a scalable and secure approach to onboarding open source libraries into production environments will accelerate the data science process.

CASE IN POINT

Kount Converts Billions of Interactions into Actionable Insights

To power ML models that help more than 9,000 brands protect their customer journeys, Kount, an Equifax company, collects and analyzes large amounts of data for account creation, login information, and payment transactions. For example, Kount's Identity Trust Platform analyzes signals from 32 billion interactions per year to prevent fraud and enable personalized customer experiences.

Kount's AI algorithms rely on massive amounts of near real-time and historical data to determine a customer's trust score, which is defined within 200 milliseconds of an attempted transaction.

Unfortunately, Kount's on-premises data environment could not easily scale to handle the company's data volumes and data science workloads, which inhibited data exploration and feature generation. Furthermore, Kount's enterprise customers needed a convenient way to access raw data to perform advanced analytics.

Realizing the need for modern, scalable data infrastructure, Kount's data science team turned to Snowflake to provide a governed source of truth where analysts and data scientists could process any volume of data. Built upon Snowflake, Kount's Data on Demand solution provides access to transaction data that empowers customers to perform in-depth analyses, generate personalized reporting, and generate the features needed for their machine learning models. Data scientists use the solution to efficiently explore and process large data sets for their fraud prevention solutions. Marketing teams use Data on Demand to understand buying patterns among demographic groups, identify cohorts of returning buyers, and develop customized campaigns that support revenue growth.

Snowflake's fast data processing and elastic performance engine eliminated both data access and data preparation limitations that previously slowed down data exploration and feature engineering activities. And because Snowflake can be seamlessly integrated into Kount's ML toolset, the data science team now spends less time preparing data, so they can spend more time building models.

> “ **The elasticity and near-zero maintenance of Snowflake enables our data science team to elevate our productivity by spending less time preparing data, so they can spend more time building models.”**
>
> **—MATTHEW JONES,**
> Data Science Manager, Kount

Enabling Best Practices with a Cloud Data Platform

A cloud data platform is a specialized cloud service optimized for storing, analyzing, and sharing large and diverse volumes of data for many types of workloads. As a centralized repository of structured, unstructured, and semi-structured data, it fosters collaboration between data science and related analytic endeavors. It eliminates the inconsistent results that arise when various work groups use different copies of the data, helping to combat or eliminate data silos and security risks from ungoverned copies of data. A cloud data platform also makes it easier to share data without having to copy or move that data.

Without a centralized data repository that supports all data types, tools, and programming languages that support the needs of multiple teams, data silos will proliferate. These data silos complicate data governance by making it difficult to trace the data's lineage, catalog the data, and apply security rules. As a result, obtaining the correct data and preparing it for analysis consumes 66% of data scientist's time (Anaconda, *State of Data Science 2020*)—a number that can be much larger when data is scattered across multiple systems.

Another reason data wrangling and feature engineering are often conducted separately is because analysts and data scientists work with different data sources, language, and tools. For example, in addition to working with SQL, analysts may use low-code tools such as Alteryx and Dataiku. Meanwhile, data scientists may use SQL, Python, Scala, and R in conjunction with ML libraries such as Numpy, Pandas, and scikit-learn. They may also use automated feature engineering tools such as DataRobot. Having a centralized data repository that supports all these tools and data types helps unify data wrangling and feature engineering activities.

A cloud data platform helps ensure fluidity among data science, analytics, and data engineering workloads by allowing many types of data professionals to work in concert, using their preferred programming languages, libraries, and tools. This approach is ideal for data wrangling and feature engineering since it allows business analysts, data scientists, and data engineers to all leverage a centralized repository for logic and data. A cloud data platform allows you to colocate the code and the data so you can "bring the code to the data." Data scientists can run ML procedures directly in the cloud. Analysts can pull in the result set (predictions), and surface the insights via BI apps that also leverage the cloud data platform. Figure 4 illustrates this convergence of data analytics, data science, and data engineering.

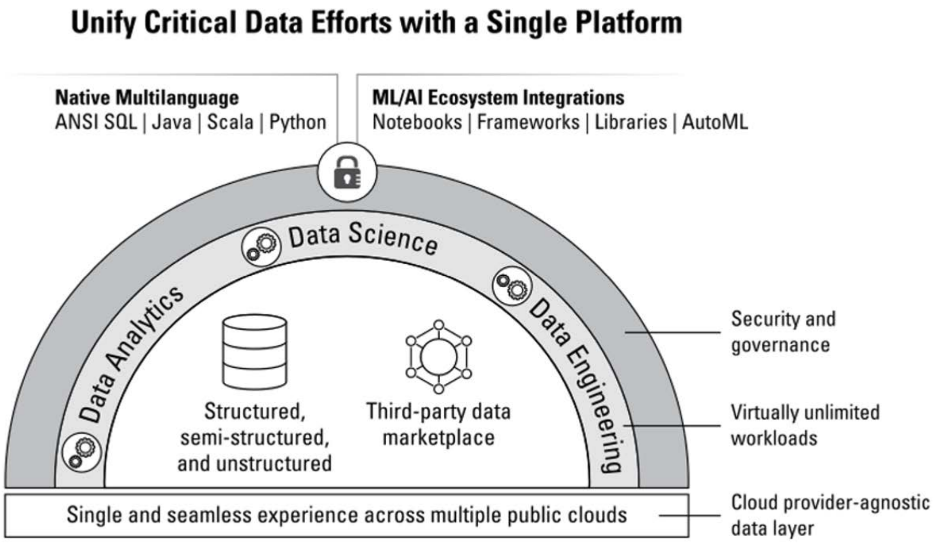

Unify Critical Data Efforts with a Single Platform
Native Multilanguage
ANSI SQL | Java | Scala | Python
ML/AI Ecosystem Integrations
Notebooks | Frameworks | Libraries | AutoML
Data Analytics
Data Science
Data Engineering
Structured, semi-structured, and unstructured
Third-party data marketplace
Security and governance
Virtually unlimited workloads
Single and seamless experience across multiple public clouds
Cloud provider-agnostic data layer

Figure 4: A cloud data platform includes a common repository that unites the work of data analysts, data scientists, and data engineers.

CASE IN POINT

From Bottlenecks to Insights

Pacific Life helps millions of individuals and families with their financial needs via a wide range of life insurance products, annuities, and mutual funds. Previously, the insurance giant used an on-premises data warehouse, but accessing 20 years of historical data wasn't easy.

Moving Pacific Life's data to Snowflake eliminated these constraints, enabling the organization's data science team to more efficiently analyze engagement patterns, equity returns, interest rate returns, and customer activity. Data scientists can now work with massive data sets, including third-party Experian data from an associated data marketplace, without manually copying or moving the data.

Instead of creating silos of information, Pacific Life's entire business community can access a unified source of governed data, including historical policyholder data, presented via BI dashboards. Data scientists and analysts across the organization can fully leverage their data, easily create workspaces to run their analytics, and then publish the resulting data sets for others to use. For example, data scientists can build models to predict policyholder customer service demand and surface results in business dashboards to optimize staffing schedules.

> **“Prior to adopting Snowflake, it was labor intensive and inefficient to access the data we need to make good business decisions. Now data scientists can apply logic to historical data to help us predict and optimize outcomes across a number of key functions.”**
>
> **—KURPAL SANDHU,**
> Director of Data and Visualization, Pacific Life

Using Snowflake as Your Cloud Data Platform

Many organizations use the Snowflake Data Cloud to execute data engineering, data science, and data application workloads. Snowflake does much more than just store your data. It is also a proven solution for *processing* that data, including running complex data engineering pipelines, performing feature engineering, and processing queries from analytic applications. Snowflake's multi-cluster, shared-data architecture offers a robust processing engine for all these needs.

Snowflake allows you to store, manage, and query structured, semi-structured, and unstructured data types using standard SQL, the ubiquitous language that powers the world's most popular analytics and data visualization tools. You can store all these data types in the Snowflake Data Cloud, including raw data such as audio files, video, and images.

Furthermore, with Snowpark, Snowflake's developer framework, you can deploy custom data wrangling workflows and apply them to data stored in the Data Cloud. Behind the scenes, Snowpark allows data engineers, data scientists, and data application developers to use familiar coding methods and languages of choice—including Python (in public preview), Java, and Scala—to execute pipelines, create ML workflows, and develop data apps efficiently and securely, all in a single platform (see Figure 5). Snowpark brings popular DataFrame-style programming concepts to the languages these data professionals like to use. Custom logic can be executed inside of Snowflake using an elastic performance engine. This logic can include your own functions or Python open source libraries available through the embedded Anaconda repository.

Finally, Snowflake's broad support for open source offerings allows you to use Python and its many open source libraries to execute built-in calculations such as one-hot encoding in a very scalable way, whether you are training a model for 30 customers or 30 million customers. You can use a variety of open source tools to create data-preparation workflows that remove errors before a data set is transformed into a format that is conducive for analysis.

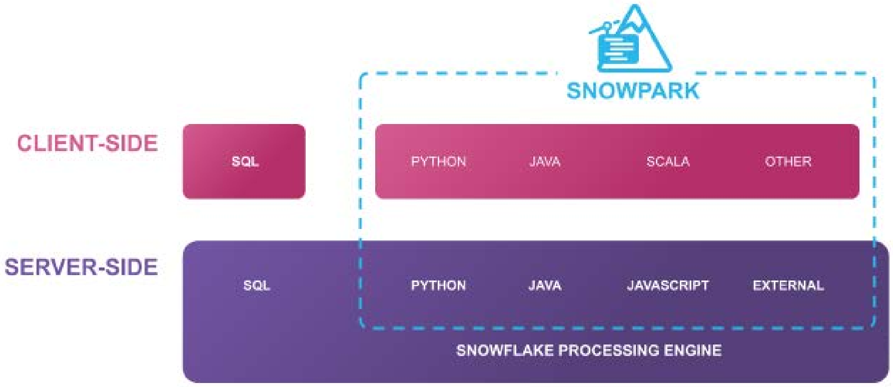


Figure 5: Extending the Snowflake processing engine to popular programming languages with Snowpark

CASE IN POINT

Bagel Brands Unlocks a Network of Data

Bagel Brands is the parent company for Einstein Bros., Bruegger's, Noah's NY, and Manhattan Bagels. Headquartered in Denver, Colorado, the company serves more than 1,100 domestic franchise and license locations supported by 16,000 employees.

Previously, Bagel Brands had weekly and monthly forecasts based on rolling averages for catering and in-store sales that determined everything from the number of staff to the amount of ingredients needed for bagels. However, capturing 11 TB of data via Secure File Transfer Protocol wasn't feasible with the company's on-premises SQL Server environment. As a result, analysts and data scientists could only access these data sets in three-day to seven-day windows.

To fuel its data analytics and data science initiatives, Bagel Brands wanted to add third-party geolocation data to enrich its daily insights, such as to predict foot traffic at stores. This is a familiar situation that many other organizations have encountered: because data to create metrics and features doesn't always exist internally, it is helpful to use a data platform that supports data sharing and includes a broad, internet-based marketplace.

With Snowflake as the central source of truth, Bagel Brands introduced new ML models that can generate daily forecasts with higher accuracy by using third-party data sources that add relevant information about changes in the economy, competitors, and more. Snowflake Secure Data Sharing and Snowflake Data Marketplace eliminated the need for Secure File Transfer Protocol processes, saving time and operational costs. For example, by receiving terabytes of geolocation information without complex manual file transfers, Bagel Brands can predict what actions will help operators improve metrics such as foot traffic.

"Instead of waiting for updates, Snowflake gave us the ability to analyze campaign performance in near real time, automate the process, and start realizing what the true ROI was for campaigns," explains Jessica Lee, Director of Data Science and Analytics at Bagel Brands. "We are now setting up ML models for predictive campaign performance and working closely with the marketing team to find the best promotional calendar simulations to see how they impact sales."

> **“Data sharing is enhancing restaurant operations by enabling data science teams to build more accurate machine learning models that give us data-driven insights around specific business metrics.”**
>
> **—JESSICA LEE,**
> Director of Data Science and Analytics, Bagel Brands

CONCLUSION

Why Business Analysts and Data Scientists Choose Snowflake

Data scientists and data analysts require massive amounts of data to generate data-driven insights and build and train ML models. They need a data platform that can make an organization's data "production ready" by putting it into a usable form. Snowflake allows these data professionals to collect, store, and analyze all their data in one place, so they can eliminate redundant and ungoverned copies. Snowflake also empowers all stakeholders to use their preferred languages and tools, including Python with its rich ecosystem of open source libraries. Finally, the Snowflake data platform is secure, governed, scalable, and integrates with best-of-breed tools, making it possible to easily take successful data science experiments from experiments to production.

The Snowflake processing engine can scale elastically to help these data professionals discover, structure, cleanse, enrich, validate, and publish data. Snowflake provides a centralized repository that allows them to manipulate data formats, scale data systems, and enforce data quality and security. These inherent capabilities of the Snowflake platform ensure that the data is clean, reliable, and properly transformed for whatever the user community requires, including running queries for frontline workers, generating reports for management, or training ML models for data scientists.

By centralizing access to data, Snowflake reduces the number of stages the data needs to move through before it becomes actionable, which eliminates the need for complex data pipeline tools. Snowflake managed storage ensures that data science teams have robust performance and security for their data in the cloud, whether it is structured, semi-structured, or unstructured. However, Snowflake also understands that some data needs to remain on premises. To facilitate these data sets, Snowflake introduced External Tables for on-premises storage devices that are compatible with the AWS S3 APIs (this feature is currently in preview). This capability also applies to accessing data stored in open table formats including Apache Iceberg (also in preview). With Snowflake, organizations are not bound to their own data—accessing shared and third-party data is just as straightforward as accessing internal data sets. There is no need to work with stale copies of data or deal with lengthy contracts to access third-party data sets. This drastically simplifies the process of supplying data to ML models for training or model inference, where scale and performance are important.

Snowflake offers tremendous flexibility in the languages data professionals choose to use, along with high-performance processing not only for data preparation and feature engineering, but also for training and inference. Snowflake can run complex algorithms internally, all while enabling users to work from external ML tools and platforms. Data scientists and data engineers can run these procedures directly in the Snowflake Data Cloud to simplify the overall process, making Snowflake ideal for both data wrangling and feature engineering.

ABOUT SNOWFLAKE

Snowflake delivers the Data Cloud—a global network where thousands of organizations mobilize data with near-unlimited scale, concurrency, and performance. Inside the Data Cloud, organizations unite their siloed data, easily discover and securely share governed data, and execute diverse analytic workloads. Wherever data or users live, Snowflake delivers a single and seamless experience across multiple public clouds. Snowflake's platform is the engine that powers and provides access to the Data Cloud, creating a solution for data warehousing, data lakes, data engineering, data science, data application development, and data sharing. Join Snowflake customers, partners, and data providers already taking their businesses to new frontiers in the Data Cloud. **snowflake.com**