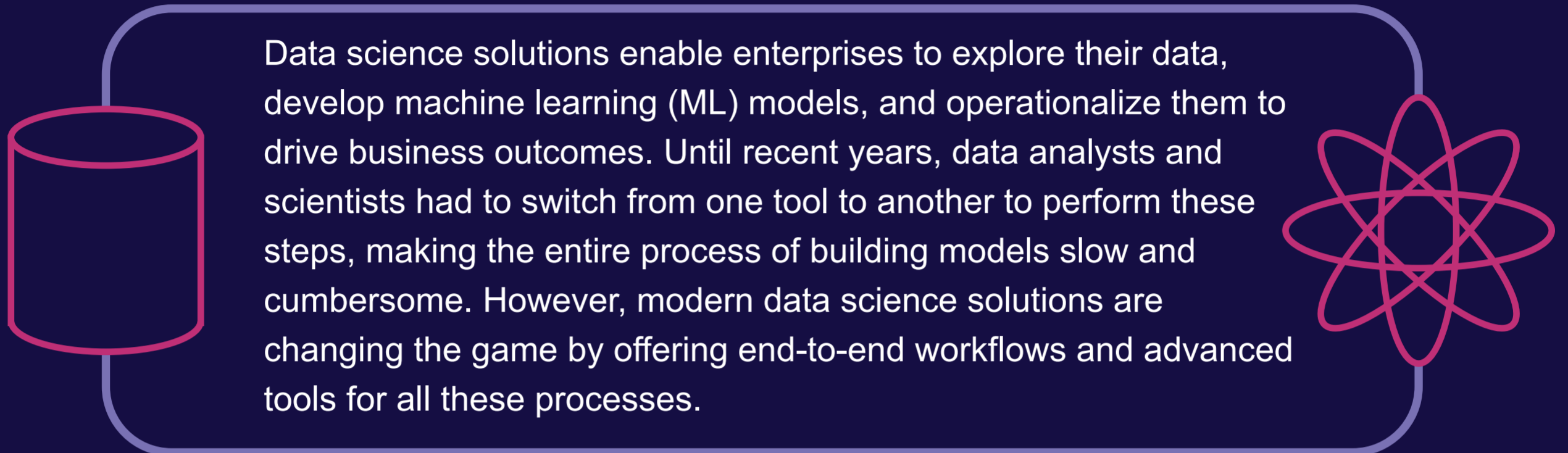


10 MUST-HAVES FOR A POWERFUL ENTERPRISE DATA SCIENCE SOLUTION



Data science solutions enable enterprises to explore their data, develop machine learning (ML) models, and operationalize them to drive business outcomes. Until recent years, data analysts and scientists had to switch from one tool to another to perform these steps, making the entire process of building models slow and cumbersome. However, modern data science solutions are changing the game by offering end-to-end workflows and advanced tools for all these processes.

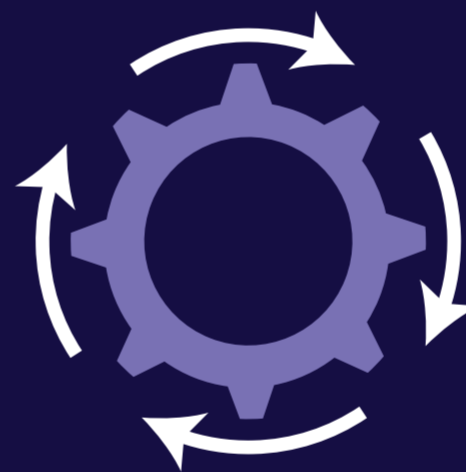
1. Ideate



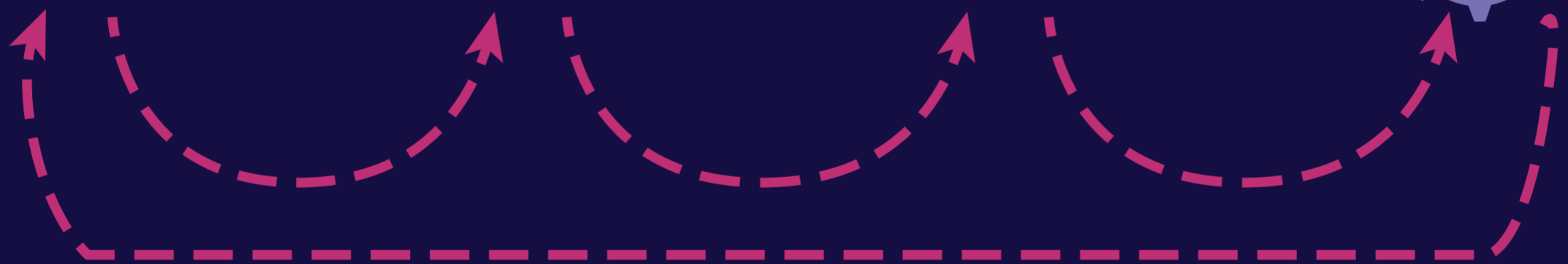
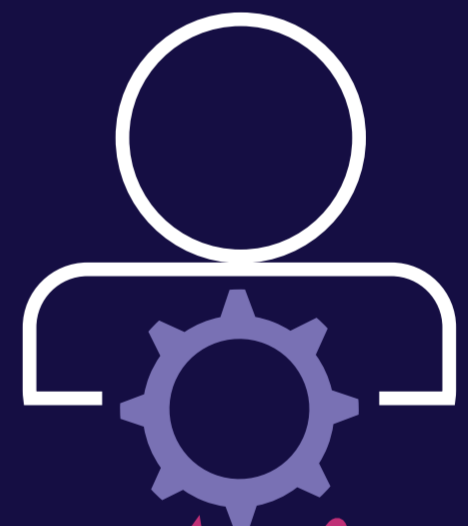
2. Experiment



3. Operationalize



4. Manage



Data science lifecycle

Data science solutions are evolving rapidly to meet the changing needs of data-driven enterprises. Here's our take on ten must-haves for a next-gen scalable enterprise data science solution.



1. Built-in data preparation tools

Data preparation processes like acquiring, cleaning, and pre-processing lay the foundation for effective model training. Poor quality data can compromise model accuracy, which in turn impacts business outcomes. Some of the common challenges faced at the data preparation stage include wrong/irrelevant records, inappropriate tags, and misspelled words. To address these, a powerful data science solution should provide integrated tools that automatically generate code to remove null values, filter features, and generate new derived features without having to switch between multiple platforms. These capabilities can help reduce coding efforts, optimize data wrangling processes, and ensure accurate model training.

2. Flexibility to experiment

Building a model is an iterative process, where each iteration is an experiment. To create an optimum model, data science teams modify model features, change algorithms, fine-tune parameters, and perform various other experiments in a sandbox. An effective data science solution should provide a separate runtime environment (along with the necessary packages and compute resources) to run these experiments. Users should also be able to easily reproduce any experiment by reproducing the environment as and when required.

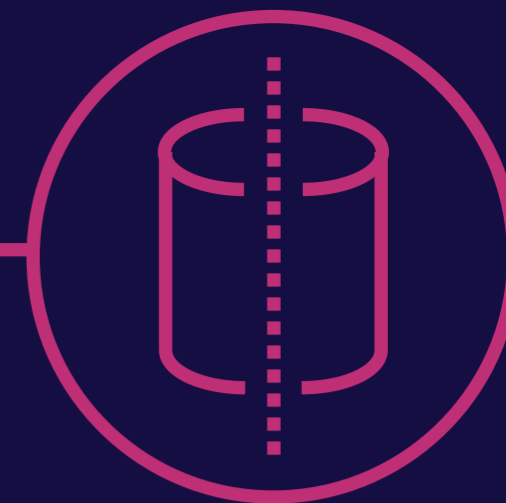


3. Support for multiple languages and packages

Today, data scientists work with many programming languages and pre-built packages. Therefore, next-gen data science solutions should support popular languages like Python, R, Julia, and Scala and give users the flexibility to create/leverage pre-defined packages as and when needed. They should also facilitate the import and export of portable models such as PMML to enable seamless cross-team collaboration. In addition, users should be able to orchestrate multi-stage analytics pipelines and integrate models at various lifecycle stages to meet business needs with ease.



4. Ability to identify the optimum model



Model training involves running various iterations or versions by changing model configurations. Data scientists often find it difficult to identify the optimum model out of these different versions. A robust data science solution should capture various evaluation metrics such as precision and F1 score to help data scientists compare multiple versions of a model. It should also provide visualization charts to help understand how the model might behave with real-time data. Such features enable data-driven decision-making and act as a single source of truth while deploying models in production.



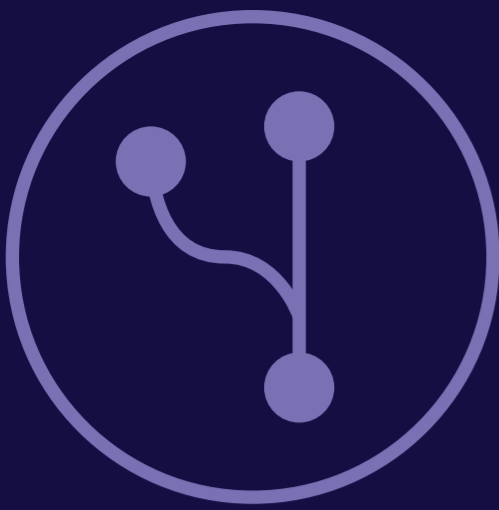
5. Ability to monitor models in production

To deliver business value, developed models need to be productionalized. They can either be published via a REST service or used in a production data pipeline. However, this process can be difficult to monitor. An efficient data science solution should enable data science teams to seamlessly monitor the productionalized models and take remedial actions in case their predictive performance degrades over time. It should also help detect model drift by performing statistical analysis on incoming data and using feedback loops to update the model configuration. Based on KPIs such as accuracy, precision-recall, and ROC curve values, the solution should generate weekly reports for improved visibility and decision-making.

6. A/B testing of models



A/B testing plays a crucial role in comparing model versions, validating their performance, and checking for decay. Often, variations in the incoming data impact model performance, which, in turn, can lead to loss of business revenue. Data science solutions that support A/B testing of models can help streamline the model version comparison process to boost conversion rates and ROI. In addition, any deviation from the expected results needs to be communicated to all relevant stakeholders via email alerts, dashboards, or by publishing reports containing key performance metrics.



7. Advanced version control management

A future-ready data science solution should integrate seamlessly with third-party version control systems like GitHub and Bitbucket. This can help users effortlessly create and manage different model versions and share versioning history with other data scientists and stakeholders. The solution should package each model along with the necessary data and configuration files so that it can easily be reproduced on a different system from scratch.

8. Centralized model repository

Data scientists spend a lot of time building models and gathering insights around different model development processes. To reduce duplication of effort, a data science solution should provide a consolidated repository of all the developed models. This enables different teams to share/reuse models as needed and achieve faster time-to-market. The solution should also allow the import/export of models and provide access to private models upon request.



9. Secure collaboration capabilities

Enabling seamless collaboration between different teams across the end-to-end data science lifecycle can help organizations save valuable time and effort. Data analysts, data scientists, MLOps engineers, and other stakeholders should be able to work on the same solution simultaneously without operational challenges. To facilitate this, the solution should provide role-based access and other stringent security features for protecting data confidentiality and integrity. A solution that supports secure cross-team collaboration can help stakeholders take preemptive decisions and achieve faster time-to-market.



10. Openness



Data science solutions should ultimately make it easier to build models and other data science deliverables. Certain solutions facilitate these processes, but only for those using a particular programming language, modeling package, or GUI tool. On the other hand, an open solution gives data scientists the freedom to use best-fit tools for each job and experiment with new packages and languages. This enables ease of operations, as they can continue using the tools they're accustomed to.

As a leading self-service data flow and ML platform, Gathr offers all the above capabilities and more. It powers data ingestion, processing, enrichment, and visualization with an intuitive drag-and-drop visual interface to build and operationalize big data applications at lightning speed.

Start using **Gathr** for free

Gathr is a next-gen, cloud-native, fully-managed, no-code data pipeline platform. It's the only all-in-one platform for all your data integration and engineering needs – batch and streaming ingestion, CDC, ETL, ELT, data preparation, machine learning, and analytics. The Spark-based platform brings unmatched speed, performance and flexibility required to handle all types of data and analytics approaches, in ways that traditional ETL tools cannot. With Gathr's visual drag-and-drop interface, native integration for all popular data sources and destinations, an exhaustive set of pre-built operators, and a rich pipeline template gallery, anyone can build and deploy data pipelines, quickly and easily.

Visit www.gathr.one or write to us at contact@gathr.one

© 2021 Impetus Technologies, Inc.

All rights reserved. Product and company names mentioned herein may be trademarks of their respective companies.

gathr