

Best Practices:

Building a Data Product With Embedded Analytics



Do you aspire to help your customers easily understand what is behind the data they use everyday?

Do you want your users to make data-driven decisions in your product?

Do you want to tell stories through the data your product generates?

Do you want to take your product to the next level by bringing data-driven decision-making tools to its core?

This e-book digs into several key aspects of an embedded analytics solution, focusing on the best practices you should employ.

Key Points to Consider

A data product is an enterprise's information assets wrapped in engaging analytics that drive significant value to its business network. The solution for engaging analytics in a software product is called embedded analytics.

Embedded analytics represents a set of analytical tools seamlessly integrated into your existing product. Subsequently, the user will see the tools as an inherent part of your product, despite the former running as a third-party analytical platform within the backend.

But why even consider a platform and then build the analytics on top of that when your teams can build it all from scratch? The answer is simple. Focusing on your key competencies is the best way to succeed on the market with your product.

Unless analytics is your primary focus, relying on a strong partner will always prove to be the better option than being burdened with development and operations that are neither your area of expertise nor your primary business case.

Here are the most important points you need to consider when building an analytical experience in your product:

Focus on your product; don't struggle with security and standards

Control and manage analytics for all your customers with out-of-the-box tools

Get a hand from professionals

Count on transparent and predictable pricing for your growth

Self-service: Prevent change-request overload

Build a solid analytical model fast — for dozens, hundreds, or thousands of customers

User interface integration, seamlessly

Integrate analytics into your interface seamlessly

Covering these points will be easiest with an embedded analytics platform, unless you already have strong expertise in your engineering and data teams. So, let's take a look at what the best practices are and what to look for in an analytics solution.

Avoid Struggles With Security and Regulatory Standards

Building a solution based on data will require that you meet various security requirements and regulatory standards. These requirements evolve: New regulatory standards for different regions or industries are introduced over time. Moreover, you might consider entering a market that requires a higher set of standards than you originally planned. Implementing such rules to an existing solution might negatively impact the roadmap of your core product features.

The ideal embedded analytics platform should cover all the problems above — with the highest level of data security and protection — in order to strengthen trust; provide full transparency; and enhance governance, risk management and compliance for companies of any size.

Let's take a look at three layers of security and their best practices.

Cloud and Physical Security

- Enterprise-class SOC 2 Type II-audited and ISO 27001-certified facilities that include 24/7/365 managed security and physical access control
- Monitoring of the platform and all of its underlying operating components for security incidents, including alert notifications generated by the platform systems, industry and open source alerts, and community alerts
- Three-level support team on duty 24/7 and trained to provide systematic and efficient responses to incidents including security and availability issues

Data Security

- Ensuring all data remains confidential, available, and backed up for governance and disaster recovery purposes
- Security zones are defined and implemented with specific enforcements, such as protocol enforcement, intrusion detection and monitoring, and the platform is subjected to regular penetration testing
- Customer data is both isolated and encrypted in transport as well as at rest to further enhance its defense

Operational Security

- Access to the production environment is under strict control
- Administrators are allowed to invoke platform-specific functions but not to directly interact with the platform's underlying components
- Access logs are monitored and regularly inspected

The measures detailed above meet the best practices for ensuring secure operation of any analytical solution. With a third-party solution, you have the freedom to choose which service-level support that your business needs.

Basic out-of-the-box security: This covers industry standard best practices for data protection in the cloud, including SOC 2 Type II report, ISO 27001:2013 compliance, GDPR compliance, and CCPA compliance.



Additional security: This supports environments with sensitive or regulated data that require additional assurance on security practices, data handling, and implementation. All platform events are tracked in an audit log and security/risk assessments, reporting, and audits are covered, too. Patch management is covered by SLA.

Regulatory Add-ons for Specific Use Cases:

- HIPAA, including BAA and additional arrangements, to ensure customers comply with the HIPAA regulatory requirements
- GDPR Article 9 add-on including DPA and additional arrangements to ensure compliance with requirements for processing of special categories of data under Article 9 EU GDPR

Control and Manage Analytics for All Your Customers With Out-of-the-Box Tools

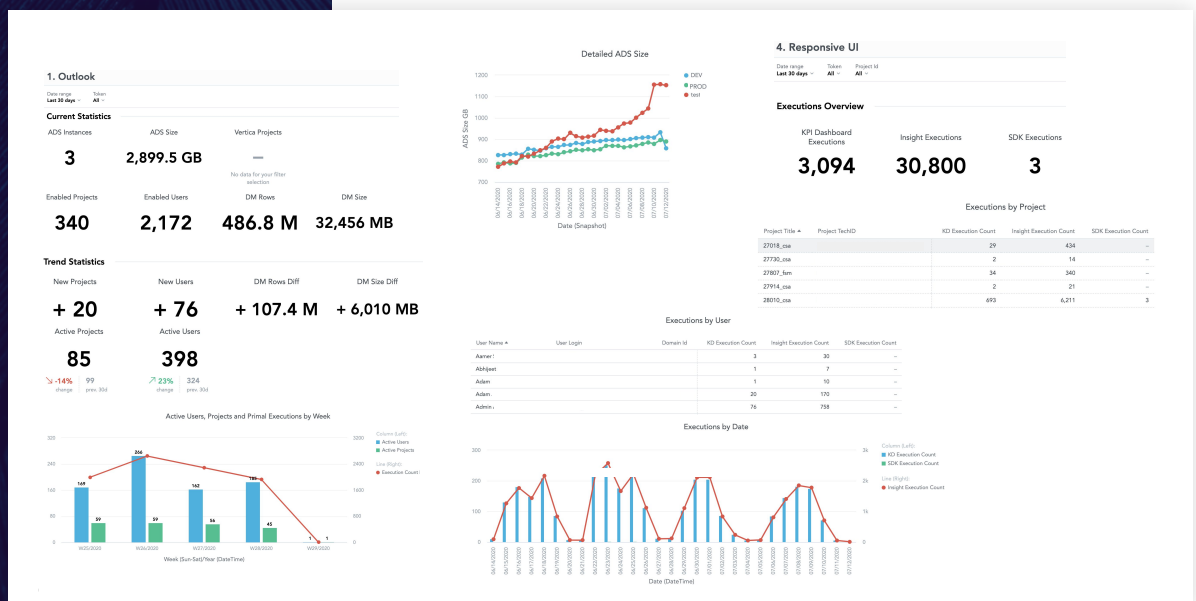
Once your analytics is up and running as an inherent part of your product, with a successful go-live launch, you will be facing the reality of everyday operations: understanding what is happening and striving to automate your operations as much as possible.

You should also be able to understand how users are consuming your analytics. The data will help you with both quantitative and qualitative research that your teams will conduct when improving the products. The final goal is clear: maximized user adoption and satisfaction, thus yielding increased usage of your product. In addition, automation needs to cover daily administrator tasks to minimize the risk of human error and to keep operating costs of analytics as low as possible — and, ultimately, maximize the solution margin. Let's take a look at how to best solve these challenges.

Analytics Over Analytics

Your analytics experience will need analytics in its own right not only to help you proactively monitor your environment, but also to answer key questions, such as:

- I've released a new dashboard. What is the rate of adoption?
- I've been in analytics for a while already. I have dozens of insights, dashboards ... I would like to understand: Which ones are actually being used? Which should I focus on? What can I delete (impact analysis)?
- Some of my users have ad-hoc capability. Which parts of my model do they use? How often? What data are my users interested in? Is the adoption/usage actually better compared to canned analytics only?
- Are there specific areas of analytics with high utilisation? Should I focus on/optimize for them?
- I would like to see my overall contract utilisation — all environments, all instances (including personal and development ones). What is the trend?
- I'm refreshing the workspaces frequently. Is it necessary for all of them? Which are frequently used? (I would like to optimize the costs, prioritize the ETL, etc.)



With properly set up analytics, you will get the answers and understand how your data product works among your customers and users. You will see the patterns that might lead you to a change in your solution. You will also be able to prototype a new analytical function and study how it's being used or find product champions, among your customers, who can help you with your user research.

Operational Automation

Product teams need to plan for how to change analytics at the same pace as their application — as well as how they will make analytics instantly available, and how they will comply with security and regional regulations, support localization, and disaster recovery. Your embedded analytics platform should address these issues by providing an end-to-end managed service, including runtime, operation, security, and management of the platform. This lets organizations focus on their core business instead of managing the components and costs needed to support analytical solutions.

Once a business implements an analytical solution, they will need to continuously address several ongoing tasks. These responsibilities include change management and the management of updates across many different versions and customers, handling local customizations, and onboarding new customers while ensuring that they have the correct authorizations.

Let's take a look at a common flow:

You will create the first version of your analytics	EASY
You will create multiple variants for different customer tiers: basic, premium ...	EASY
Then you will allow your customers to make their own reports and customizations	EASY
Then you will push new versions of the data product to the right customers without breaking their own reports and customizations	DIFFICULT
And most importantly, you will need to do this for each and every one of your 6,000 clients	VERY DIFFICULT

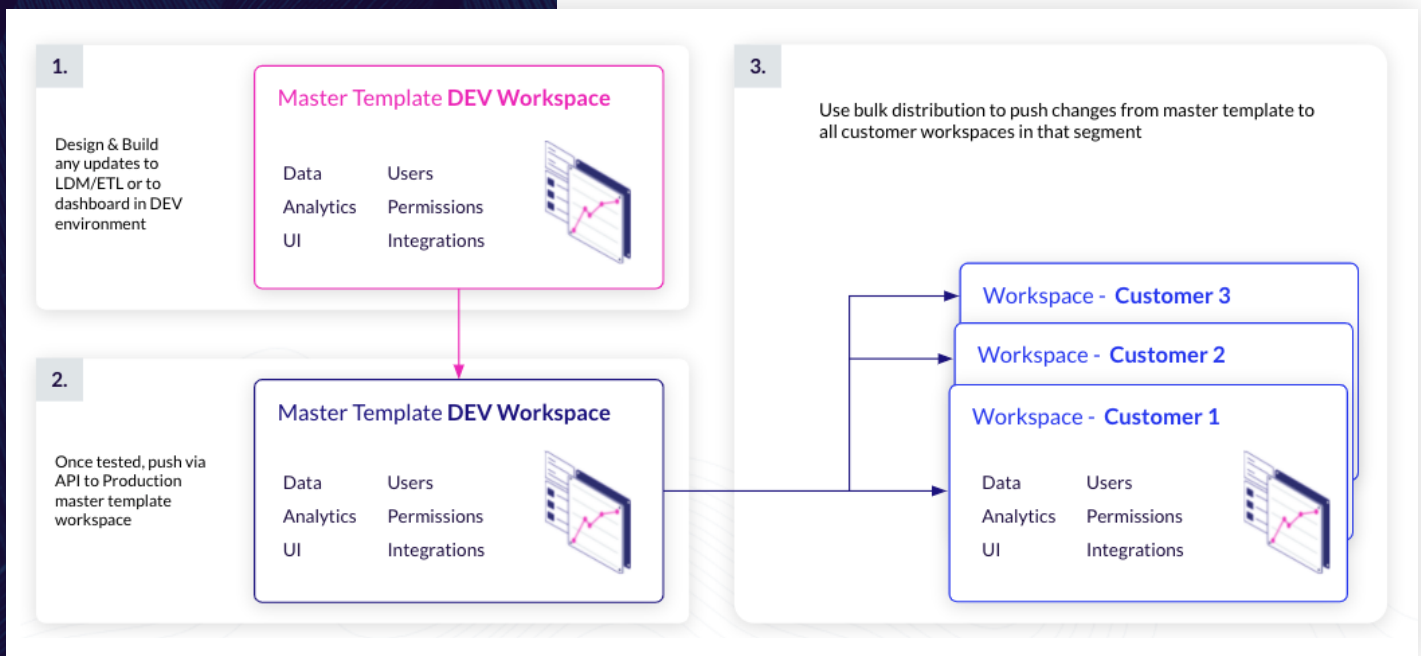
While creating the first versions of an analytics experience, with different variants, is straightforward, the harder challenge is to roll out a separate version to each of your customers' application instances at regular intervals — without overwriting changes they made within self-service mode.

The automation tools of your analytics platform should therefore primarily cover the following governance processes:

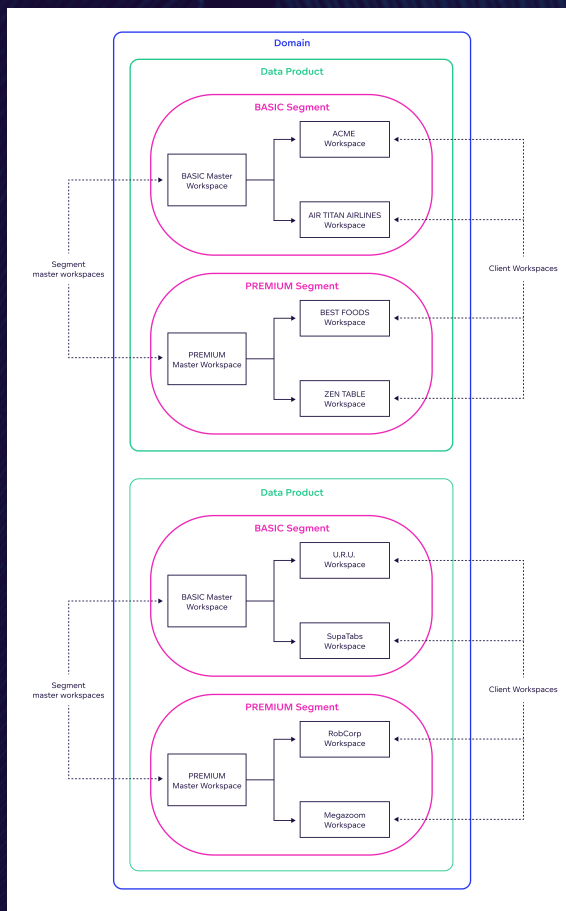
- Agile change management enables the fast release of insight application changes to the hundreds or thousands of customized analytical workspaces.
- Dynamic provisioning of workspaces and users' roles and permissions.

Ideally, the automation tasks should be scheduled and available via APIs and SDKs that can be used for integrating change management and provisioning with a continuous delivery process.

Moreover, the automation tools should enable agile change management with the inclusion of a development environment, a testing environment, and a production environment, providing the ability to propagate changes to clients in production without disrupting the end-user experience. The agile change management automates the entire change management process of moving the data model, metrics, insights, and other metadata mentioned above, from dev, to test, and production, all while maintaining localized customizations.



While the clients still have the ability to develop custom insights and dashboards inside their own workspace, the automation processes also allow you to create standardized reports, distribute them to clients, and manage all their later changes.



Furthermore, your platform should include a suite of rollout tools that enable the segmentation of customers by subscription tier or use preferences. With this capability, you are able to roll out different versions of your data analytics to different users, creating tiered pricing models to the benefit of both your business and your customer base. The rollout of these different versions of analytics won't invalidate or overwrite the customizations and unique user preferences that your users may have already created, further underlining this feature's added value.

Onboarding new clients can be a fully automated process. If you can provide the list of clients as part of the data feed, the platform will create and decommission workspaces without any manual intervention. The same goes for users, which can be automatically added and removed from workspaces based on declarative input.

Automated user provisioning tools are another important aspect of automation. The goal is to integrate with pre-existing user management tools using standard security protocols to automate the entire user provisioning process in real time, by means of an extensive API set. The tooling can also be used without integrating with pre-existing user management tools for instantly adding or terminating profiles and updating access rights.

Get a Hand From Professionals

Regardless of the implementation's straightforward process, your company just might not have the resources, capacity, or team to focus on implementation of analytics instead of focusing on your core business.

The ideal embedded analytics provider offers the expertise of a professional services team that will partner with you in order to launch a successful analytical experience within your product. They can help you through inception, development, testing, launch, and maintenance to ensure success across the entire data and analytics production lifecycle. There are four main ways in which the professional services team can help you:

- **Acceleration:** Complete your initial implementation in weeks, not months or years. If you miss analytical expertise in your team, the processes of defining requirements might slow you and your roadmap down.
- **Expertise:** Leverage the business and technical expertise of the provider to define a path for the ongoing success of your analytics projects.
- **Collaboration:** Extend your team with a dedicated group of experts by designing value-driven partnership programs that exceed your expectations.

- **Risk Mitigation:** Increase the likelihood of achieving the desired outcomes and dramatically reduce the chance of time and cost overruns with focused implementation and success plans.

Whether you seek guidance to implement your data projects or entrust the provider to take the lead, the approach should help you to design, build, launch, and improve projects with significant time to value.

The provider should be able to help you move up the analytics maturity curve by enhancing your existing data projects, providing world-class round-the-clock technical support, and offering extensive training programs to get you up and running in no time.

Transparent and Predictable Pricing for Your Growth

Inability to transparently predict future costs can quickly erase your margins if calculated wrong. There are three common ways by which analytical platforms charge for their services, none of which will help you predict your future costs:

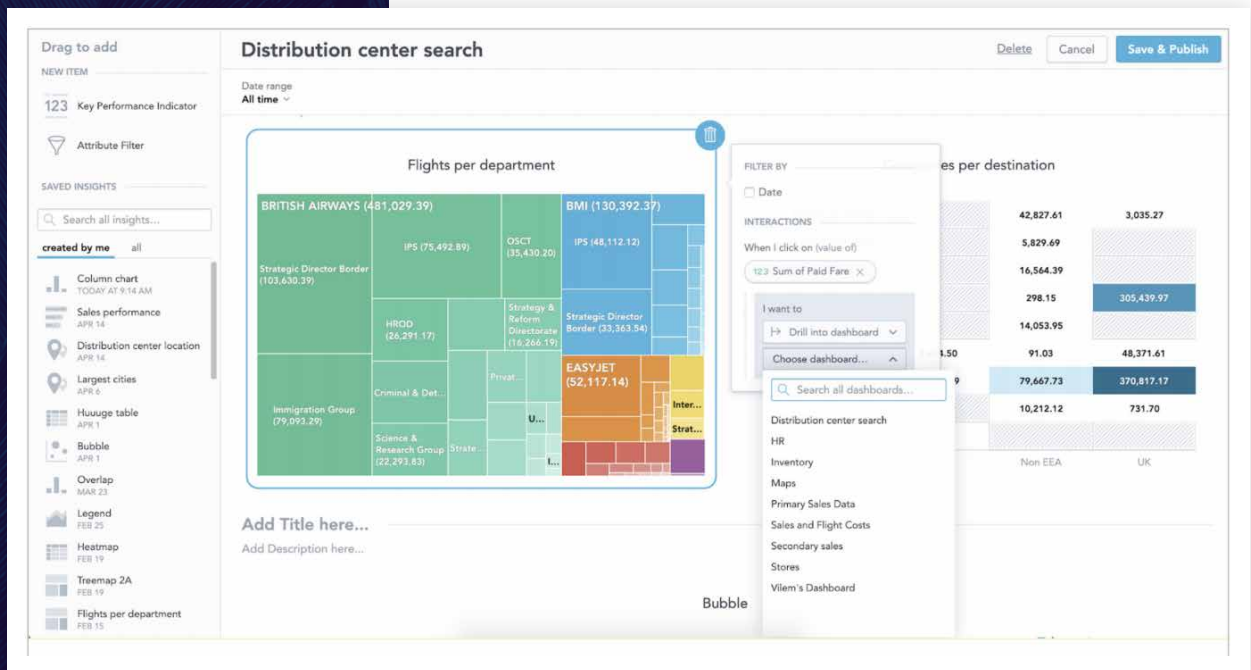
- **Pay per user:** With this type of pricing, you will have to charge per every user of every customer. Unless you are going to charge your customers for using analytics per user, too, your costs will grow unpredictably (based on the current customer user base) and your margin calculations will become increasingly difficult — a nightmare for your CFO.
- **Pay per query:** While user numbers may be predictable at the time of customer contract signing, traffic won't be. Whenever an end user displays a dashboard or insight, or simply explores information from the data in a self-service analytical tool, they will hit a pay-as-you-go database (a requirement from almost every query). Again, this means unpredictable costs, finance team headaches, and lost margins.
- **Pay per customer workspace:** This enables you to bring analytics to each of your customer's end users, predict the price per customer, and update your pricing accordingly. The pricing can then be easily calculated by number of workspaces, need of additional data storage, and specific feature add-ons.

Learn more at www.gooddata.com/pricing.

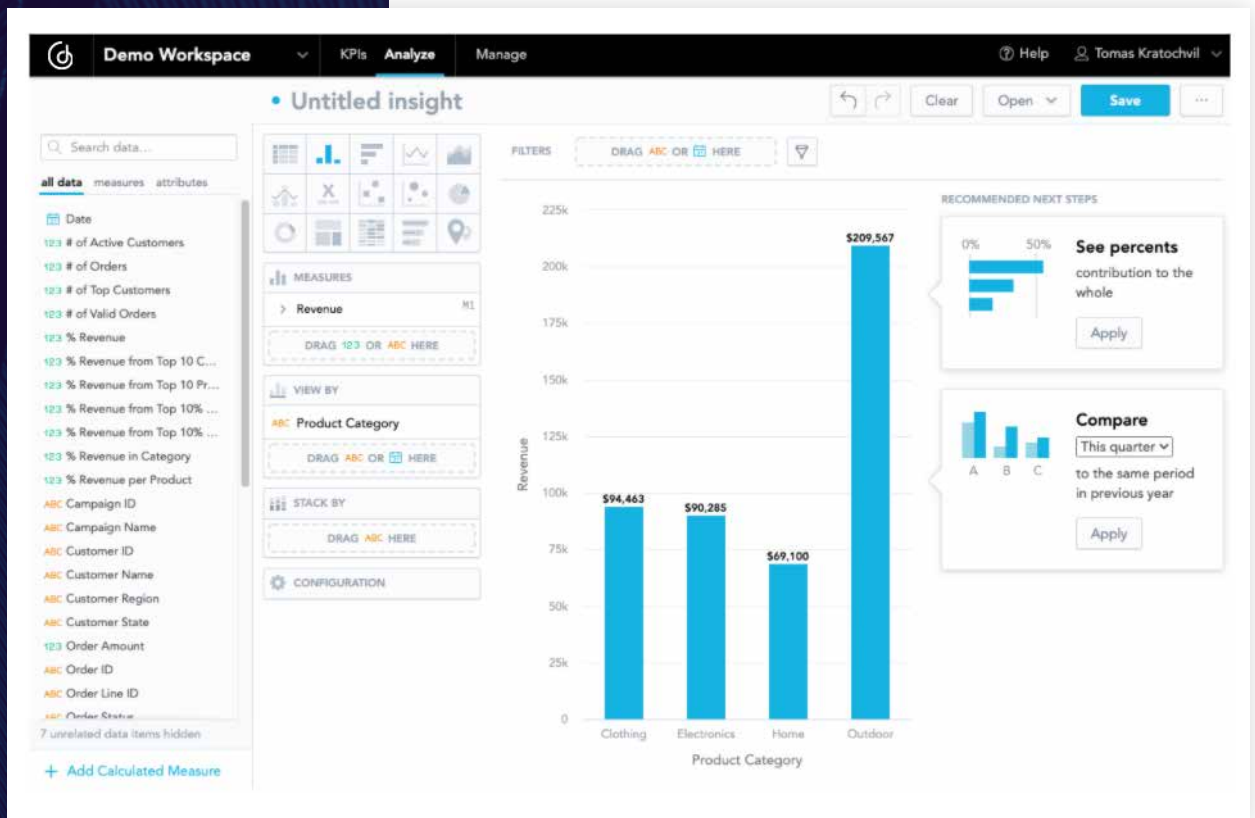
Self-Service: Prevent Change Request Overload

Depending on the industry and the customer base, it is very likely that a wide variety of users will need to be catered to — and therefore requiring a wide variety of use cases and user requirements. As such, the SaaS embedded analytics needs to be accessible to every type of user that the application supports. Your customers and their variety of users will always require a different scope of insights than your product allows out of the box. From an additional data source and the changing of visualization type, to offering additional filters, slicing by a different dimension, and a completely new dashboard — all of these requests will lead to a backlog of change requests that will probably not yield any reasonable profit, but burden your team with huge workloads instead.

The embedded analytics platform should provide self-service capabilities that will allow your customers to create their own reports, insights, or dashboards on top of your out-of-the-box analytics — the same toolset used for building an analytical experience in your product, with your customers simply building on top of it.



Business users will be able to reuse your pre-existing metrics and insights and tailor them with the drag-and-drop tool. Data analysts on your customer's side will drill deep into the datasets searching for fresh insights that can, in turn, be digested and used business-wide to fuel decision-making and wider business strategy. Both roles will benefit from advanced features, with the unrestricted ability to manipulate reports, visualizations, and key metrics. The capacity to create entirely new reports from scratch and share them with their teams right from inside your application is also of key importance to them.



Your platform should ideally contain a certain level of intelligent tooling within the user interface in order to guide users throughout the analysis. The tools should have the ability to suggest recommended reports to be created as well as showing only the relevant attributes according to the data model, therefore making it impossible to create a meaningless report by mistake. This, in particular, helps less data-literate users build reports on combinations of metrics and attributes that make sense.

For your product, customizations made by your customers in self-service mode, or data pipeline adjustments made by your team, shouldn't present an integrity risk, as a well-designed embedded analytics platform will detect all customizations when releasing a new version of your product. Thanks to self-service analytics, you can focus on your core product, while your customers create and share their own reports within your application without having to export data externally.

Build a Solid Analytical Model Fast, No Matter Your Customer Count

Your team doesn't have to worry about how complicated writing queries is, as long as you have just a few static reports with a few end users. However, when building an analytics solution with self-service for your users and insights integrated right at the point of work, you will need to keep your time investments at a reasonable cost.

Reports that cannot be reused sufficiently slow down the overall performance and productivity of your teams.

For every simple variance of an existing report that's defined by attributes, metrics, and filters, you need to change as few lines of code as possible. The less you need to change, the less you can break, and the less complicated maintenance will be. So, if your metrics have to be written again just to accomplish simple report requests, your solution will quickly become difficult to maintain as your user base grows. A query language and semantic model that abstracts the physical data and creates an understandable layer for common business users is invaluable.

The properties of such a query language include:

- Easy to understand expressions for business users
- Context-awareness — for example, one metric can be reused for 10,000 reports, generating the specific query base needed for a specific context. In other words, you don't need to create individual metrics or queries for each report - create your base technical metrics once and make them widely reusable for end-users in different contexts with the ability create their own custom metrics expressions, from existing ones, to save and use later as needed

One defined metric can replace plenty of SQL query scripts, making it easier to write and maintain your data model. This alternative type of query language allows you to assemble a new metric from existing ones without the need to know how underlying data is structured and how related metrics are connected. Therefore, insights built in a platform of this type are context-aware, meaning that a specific metric performs different computations on entities with a many-to-many relationship to prevent double-counting, further enabling ease of use for metric creation and reusability.

Let's take a look at an example of MAQL — a query language used in GoodData — and compare it to a standard SQL-based query. Your goal is to get the sales data from your database. In SQL, the query might be the following:

SQL: `SELECT SUM(Amount) FROM sales_data;`

The result will be 1 000 000 - the total value of products you have sold.

With MAQL, it's much easier to deliver the same results:

MAQL: `SELECT SUM(Amount)`

So, where is the sales_data table? With MAQL, it's not necessary; all relevant amounts are summed from the appropriate datastore, according to the logical data model.

The complexity difference is best seen in detailed queries — for instance, how much was generated from shoe sales?

SQL: `SELECT SUM(amount) FROM sales_data JOIN product_data ON sales_data.product_type id=product_data.product_type_id WHERE product_data.name = "shoes";`

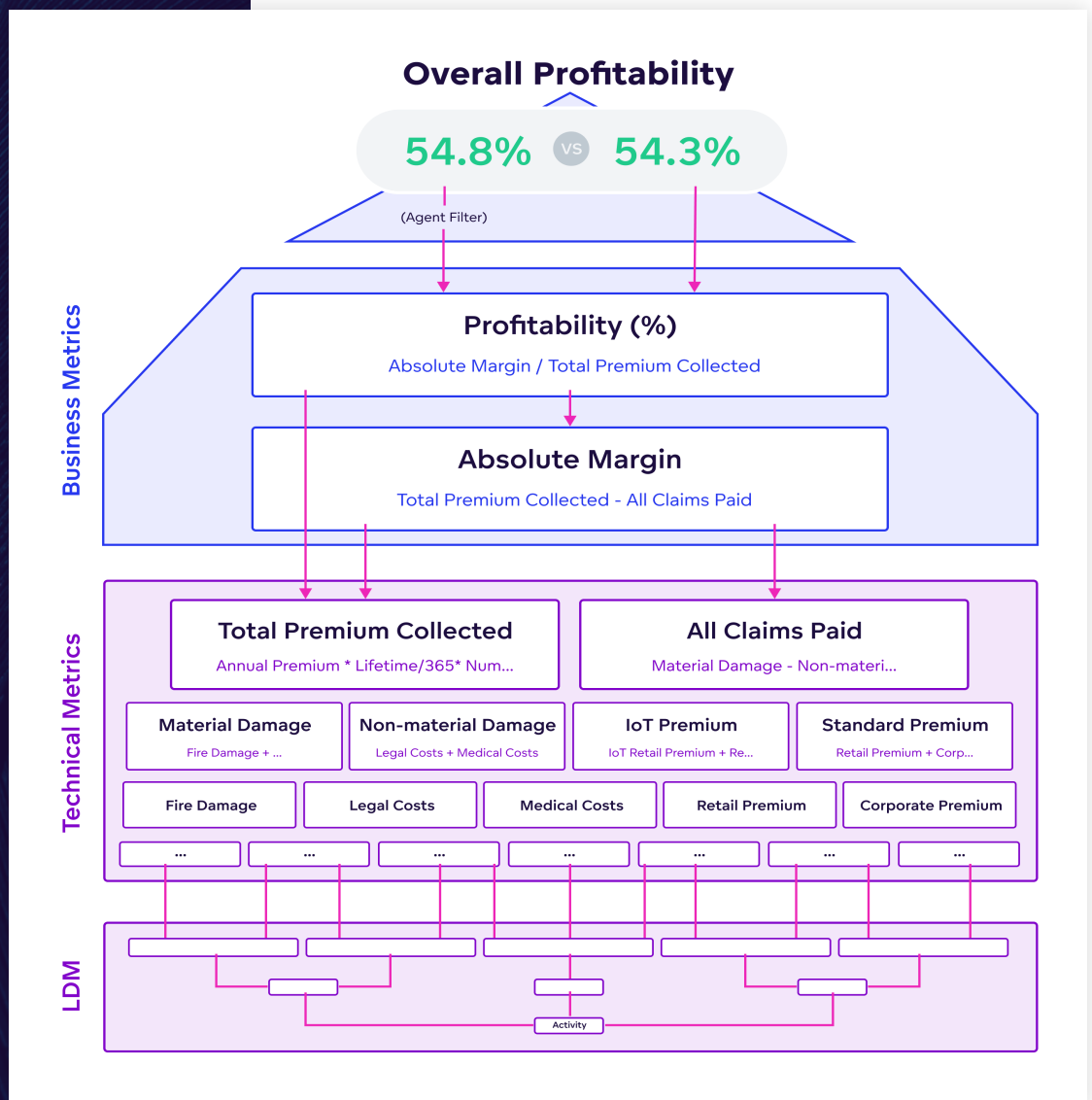
The result is 500 000. To get the number in a SQL-based environment, you need to understand the table and column structure, plus understand the relational algebra and how data is stored in your database.

With MAQL, the difference is even more obvious:

MAQL: `SELECT SUM(Amount) WHERE Product Type = shoes`

As you can see, you don't have to remember how to join tables to get the results; you can easily reuse the metrics as you want with business semantics.

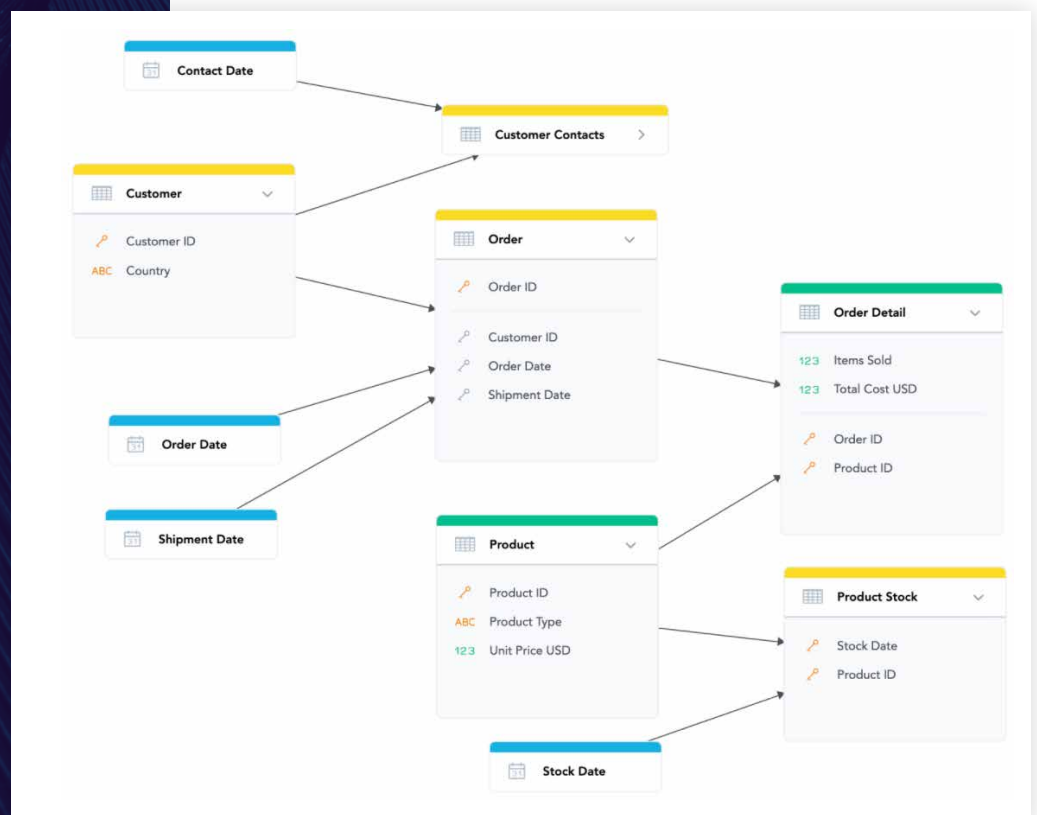
So, what is the semantic model? It brings meaning to physical data and allows end users with low data literacy to work in self-service mode without the help of experienced analysts.



The semantic model serves as a common set of definitions for everyone, including developers, who work on the product. The semantic model includes a contextual logical data model (LDM) and defined calculated measures. The LDM describes dimensions, facts, and their relationships in the semantic model and simplifies complex physical data models into a compact, aggregated analytical model that helps end users to understand the data. The LDM describes the way your data is structured in the workspace and defines objects and their relationships. Therefore, it also defines what will be possible to calculate and what will not be possible. Also, it allows the reuse of metrics in different contexts as well as ad-hoc analysis and, as such, is a key part of the analytical solution's design.

The LDM modeler, ideally web-based so that it's integrated within the product — coupled with the possibility of data modeling with CSV files, or datasets from your cloud data warehouse — lets you quickly and easily prototype, edit, and publish a data model in the analytics platform. This allows your solution to get up and running quickly, mapping the logical model to physical data, without the lengthy time frame of connecting an external data warehouse or writing long queries.

The web-based LDM modeler brings drag-and-drop functionality to your data analysts and allows them to easily communicate and explain the data model to the rest of the team. And due to it all being integrated into a single interface, data engineers, data analysts, and business users can all collaborate and work together in one unified environment. As a result, you spend less time on setting up analytics within your project, and more time testing and refining your product before rollout.



Seamless User Interface Integration

Making analytics an inherent part of your product is your ultimate goal. Your customers and users should not see any boundaries between your product and analytics. The easiest way to achieve this is to use a single platform with a single SLA and support that brings all the necessary tools for integration of analytics into your product. We have already covered areas that ensure the smooth delivery and operation of the solution. So, the questions that follow are “How do we get our data to the analytics platform?” and “How do we integrate the UI seamlessly into our product and infrastructure?” In turn, this forces you to ask the following questions:

- What data sources do we have, and how are planning to query them?
- Do we already have a data warehouse or data lake that runs on Snowflake, Redshift, BigQuery, S3, Athena or something else?
- Do we need an integrated login to our product and analytics, or is a separate login to a data portal sufficient?
- How should the analytics UI fit to our brand? What styles, fonts, or colors will we need?
- Do we need to control interactions between our product and analytics, such as controlling filters or drilling within the product menu?

Unless you are building a standalone [analytical application for your business partners](#) that will work fine with a separate login, you will probably want to eliminate any visual or experience boundaries between your product and analytics. This is primarily made possible by two things:

Single-sign-on (SSO): With SSO, the analytics that runs within your product, but is operated by a third-party platform, doesn't require a separate login. The analytical platform ensures automated user and access rights provisioning, so that all users are connected properly. The most commonly used types of SSO that an embedded analytics platform should support are SAML 2.0-based authentication: Fluig, Okta, Auth0, Salesforce, AD FS. Optionally, the platform might support a PGP-based authentication mechanism based on PGP keys.

Flexible UI embedding: This includes rich styling and localization options that allow for customization of interactions, styling, and overall look and feel. Plus, they should meet the UX guidelines of your product perfectly. With the ideal embedded analytics platform, customers won't be able to recognize that you are OEMing analytics.

Moreover, not only will you be able to integrate analytics that are consumed, but also a set of self-service tools. This provides users the power to create their own reports and dashboards. This will be an innovative and competitive advantage for your product. And, of course, everything is responsive, so you'll build it once and it will work for both desktop and mobile devices.

Following are the different ways of integrating analytical UI into your core application.

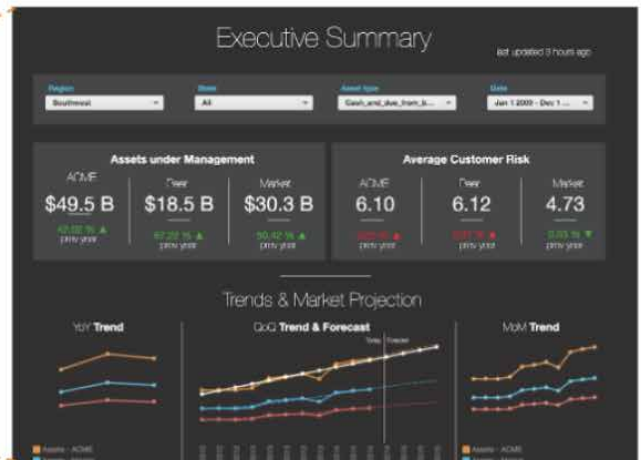
Stand-Alone Analytical Portal (White-Labeled)

The first and easiest option is a standalone white-labeled analytical portal adapted to your brand's look and feel, allowing you to skip UI integration. This is the right choice for companies building analytics portals as standalone products for their business partners or employees, because they don't have software where the analytics could be integrated. Here, you will rely primarily on dashboards and drag-and-drop self-service tools for building reports.

Embedding via iFrame

Another option is embedding via iFrame into your application with SSO. You will embed the Analytical Designer to support self-service creation of reports and dashboards, presenting analytical sections within your product. Embedding via iFrame is the right choice if you wish to enrich your existing product with analytical sections or tabs in the menu.

Iframe with JavaScript messaging



Fast and simple way how to put analytics into any web application

Limited styling, layout and interactivity options

JavaScript-Based UI SDK

An SDK for front-end development is the state-of-the-art way by which to deeply integrate analytics into your application at the source code level: Insights are embedded as JavaScript front-end components, and custom visualizations are based on the data that was retrieved by executions. This type of SDK is designed to help application developers quickly create and evolve interactive data analytics applications that are tailored to the needs of your users.

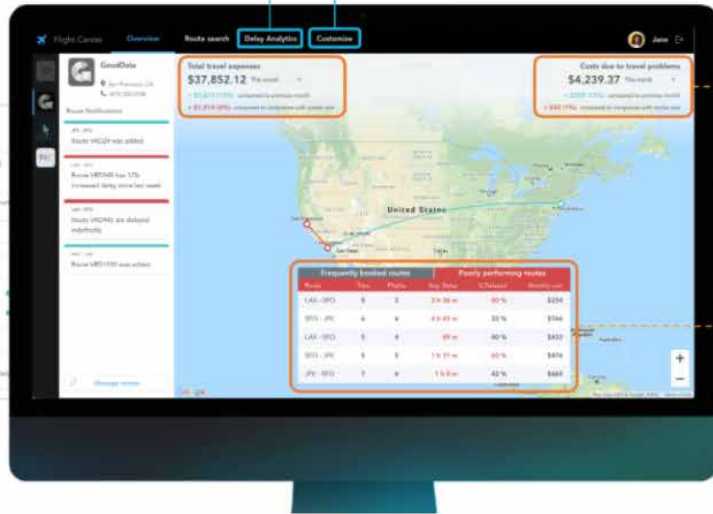
Dashboards



Analytical Designer



GoodData.UI Components



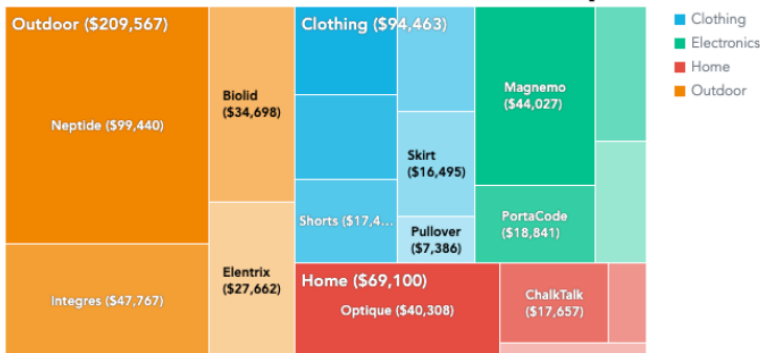
JavaScript React / React Native



GoodData Confidential. GoodData Corporation. All Rights Reserved 2020.

With pre-built components that connect directly to the analytical platform and query engine, your productivity isn't impacted by waiting on your back-end engineers. You will be up and running quickly with a set of JavaScript (such as React) components with Typescript types, granular packaging, detailed documentation, and interactive code samples.

Revenue Treemap



```
1 // (C) 2020 GoodData Corporation
2 import React from "react";
3 import { Treemap } from "@gooddata/sdk-ui-charts";
4 import * as Ldm from "../ldm";
5 import Hint from "./Hint";
6
7 export default () => (
8   <>
9     <h1>Revenue Treemap</h1>
10
11     /* Try editing the component below 📍 */
12     <div style={{ height: 300 }}>
13       <Treemap
14         measures=[[Ldm.Revenue]]
15         viewBy=[Ldm.ProductCategory]
16         segmentBy=[Ldm.Product]
17         //config={{ legend: { position: "top" } }}
18       />
19     </div>
20
21     <Hint />
22   </>
23 );
```

Integrate With Your Production Data Quickly and Reliably

Data integration is the most crucial aspect when choosing third-party analytics for your software. You are primarily choosing an analytics provider based on their fit with your existing data infrastructure, so that you limit any additional investments.

As mentioned previously, the LDM either can be created from scratch or derived from your data and then modeled according to your needs. So, how can you connect your data and map it to your data model? A reliable embedded analytics platform supports various types of data sources:

- ▶ Cloud data warehouses, such as Amazon Redshift, Google BigQuery and Snowflake
- ▶ CSV files in cloud storage, such as Amazon S3 bucket or manually loaded
- ▶ Dedicated pre-built connectors to various systems, platforms, and applications

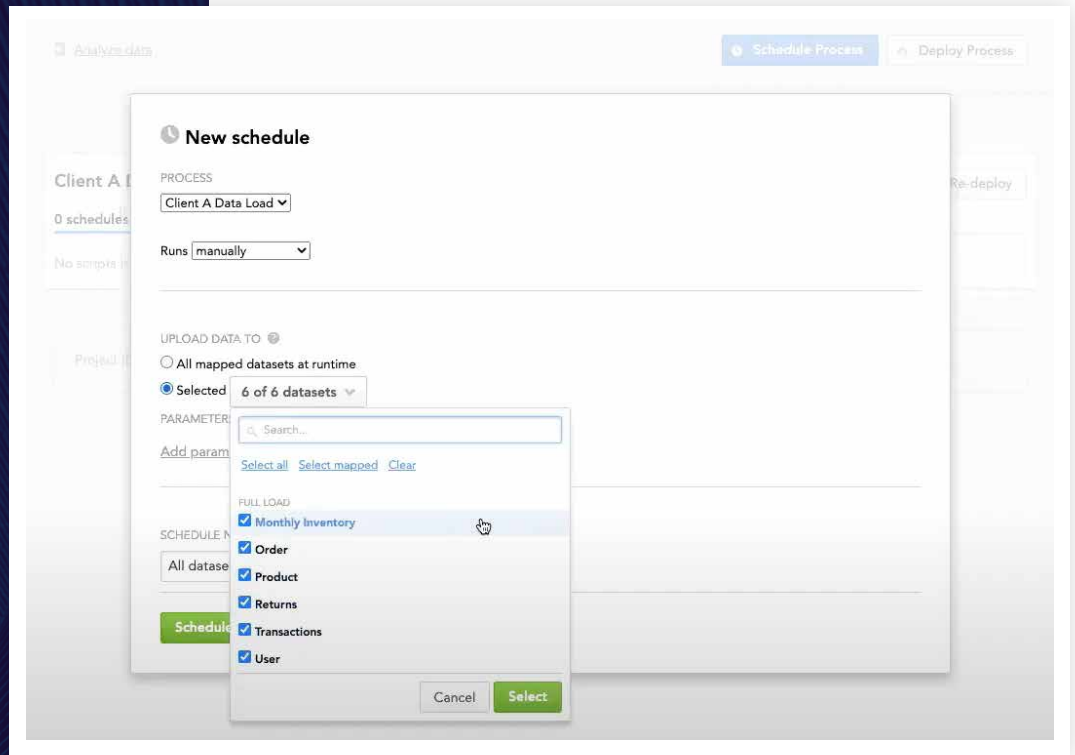
With cloud data warehouse integration, the platform should help you to adjust your source data structure according to the data model, so that data mapping between LDM datasets and tables/views in your data warehouse requires as few steps as possible and can be fully automated.

The screenshot displays a 'Data Integration Console' interface. On the left, a 'Shopboard' shows a tree view of database objects for 'ECOMMERCE_DEMO', including tables like 'OUT_MONTHLYINVENTORY', 'OUT_ORDER', and 'OUT_PRODUCT'. A 'Data Preview' window shows a table with columns 'row', 'F_MONTHLYQUANTITY', 'D_INVENTORYMONTH', and 'R_PRODUCT'. The preview data is as follows:

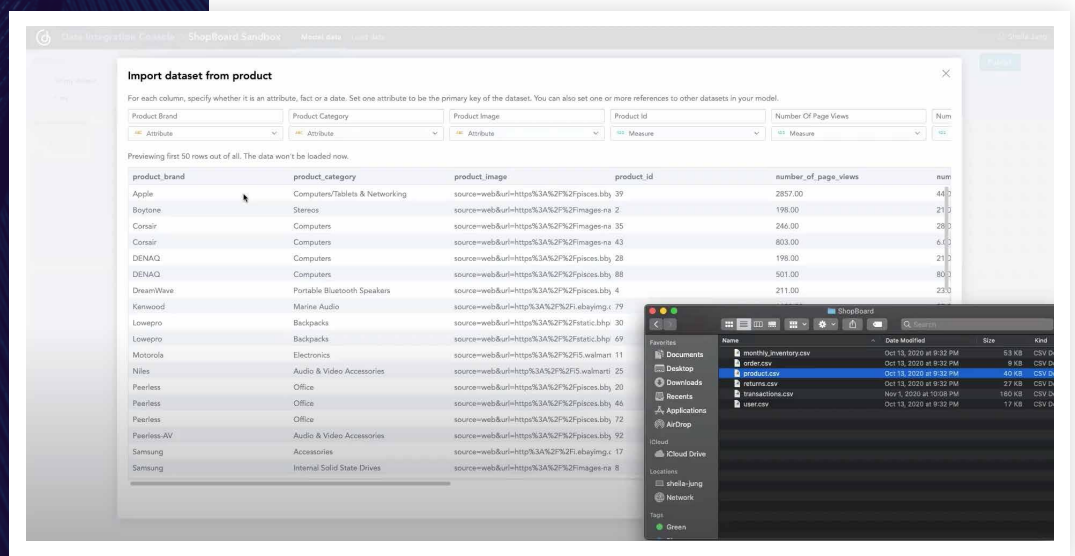
row	F_MONTHLYQUANTITY	D_INVENTORYMONTH	R_PRODUCT
1	2.00	2018-03-01	72
2	2.00	2020-04-01	39
3	2.00	2020-08-01	68
4	2.00	2020-09-01	47
5	2.00	2020-10-01	75
6	3.00	2019-05-01	48
7	3.00	2020-08-01	22

On the right, the 'Snowflake Connection' configuration form is shown. It includes fields for 'Data Source type' (Snowflake), 'Connection URL' (jdbc:snowflake://yourcompany.snowflakecomputing.com), 'Username' (admin@yourcompany.com), 'Password', 'Database' (DEMO_PROJECT), 'Warehouse' (EXPLORER), 'Schema' (DEMO), and 'Table prefix' (out_). A 'Test connection' button is visible at the bottom of the form.

After creating the data model and publishing it to your desired data product, you will set up automated data distribution to regularly update your data model with mapped data from your cloud data warehouse.



A similar process should apply for CSV files, which can be manually uploaded or uploaded via an automated distribution from cloud storage, such as your Amazon S3 bucket, where your CSV files are stored. In fact, CSV file integration is the way to integrate your embedded analytics platform with any data source without the need to configure a specific connector. You can simply export your datasets to an S3 bucket and let the platform load them automatically. The platform's data modeling tool should also suggest a structured template for your files according to your model and automatically map the fields to the attributes within it.



Are you interested in how GoodData works with CSV files to create rich insights? Read [our blog post to learn more](#).

When working with dedicated pre-built connectors, it's likely that you won't operate your own data warehouse. Therefore, the ideal embedded analytics platform should offer data warehousing services to store large amounts of business data coming from your connectors to your operational systems or other data sources. Such warehousing services should be ideally designed for high performance operations to store the full history of your data along with quick data extracts. A typical example are column-oriented database systems.

Learn More About GoodData

Now that you have learned about the best practices for building data products with embedded analytics, why not learn more about how GoodData solves the most crucial aspects of an analytics solution for your SaaS software? We have several types of resources that will help you get the answers.

GoodData Community

Join the [GoodData community](#) to talk to other customers, evaluators and GoodData employees. Read interesting posts, or ask questions directly. This is especially helpful if you don't want to talk to our experts directly.

GoodData University

With the [GoodData University](#), you can take courses that will guide you through the key concepts of GoodData and building analytics.

Documentation

You can also find your answers in our [detailed documentation](#) or [our developer portal](#) to learn more about our SDKs and data integration.

Why the World's Top Companies Chose GoodData

The core of what I love about GoodData has been the same throughout our partnership: its customizability and powerful insights. I'm able to manipulate the data and make projection models in a way that's accessible and easy to manage, and highlights the key KPIs we're tracking toward. I know GoodData will continue to be a crucial partner for us in the years to come.



 CompareNetworks

Case Study

GoodData is the perfect distribution channel for us. It's got authentication measures and allows us to silo data in different ways so that competing customers cannot access sensitive information. Plus, beyond presenting the data in flexible dashboards, there's the ability for users to create their own specific reports, as well.



Case Study

The GoodData Advantage

Business

1. One platform for all: internal teams, client companies, external partners
2. Self-service analytics for business users
3. Your own branding
4. Predictable pricing to suit your business, no pay-per-user
5. The highest data privacy and security certifications

Technical

1. Automated scaling to different departments and companies
2. Embedded dashboards in your application or software product
3. Streamlined multi-tenant change management
4. Abundant data-source options
5. Fully hosted or deployed as a container in your application

Dive deeper into the GoodData platform

[Request a demo](#) and let our experts take you on a guided tour of the GoodData platform.