

# Beyond the Modern Data Stack: Tabnine's AI-Ready Data Foundation with RudderStack

Tabnine, an enterprise AI coding platform, has spent more than a decade building tools that support developers as they write software, establishing them as one of the original AI coding assistants. Today, Tabnine is trusted by millions of developers and thousands of companies worldwide, helping engineering teams deliver better software – faster, safer, and at scale.

Early on, Tabnine leaned on in-house engineering expertise to build their own customer data infrastructure. But as Tabnine evolved from an early AI assistant into a sophisticated, enterprise-grade platform, it became clear that their DIY customer data infrastructure wasn't built to scale. Hard-coded pipelines and siloed tooling caused fragmented data, which left them with little visibility into how developers discovered, adopted, and used their platform.

To support their continued growth, Tabnine needed a customer data platform built for the AI era.

**300+**

product events per developer per hour

**10+**

downstream destinations activated

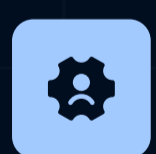
**5 Minutes**

to onboard new sources rather than weeks of engineering work

**Features:** Event Streaming, Transformations, Reverse ETL, and Audiences

**Sources:** JavaScript SDKs, Node, Python, webhooks, HTTP endpoints, Snowflake

**Destinations:** Snowflake, Mixpanel, HubSpot, Customer.io, Google Analytics, Slack, Amazon S3, Redis, VWO, GTM, Facebook Pixel



## The Challenge

Tabnine accelerates the software development lifecycle by delivering AI-assisted code suggestions as developers work. It runs continuously alongside them in their own IDEs, generating more than 300 code-completion events per developer per hour.

"It isn't something developers have to manually open and use; it lives alongside them throughout their entire workday. Understanding how developers interact with something that's always there, in the background, requires a completely different approach to data and identity!"

Because Tabnine isn't a traditional application with obvious events like logins, clicks, views, and searches, event tracking and analytics are uniquely complex. The system they built in-house could track events, but it lacked the capabilities necessary to give them a complete picture of how developers were actually using Tabnine. Moreover, data flowed through hard-coded pipelines that were difficult to maintain as new tools and use cases were added. The DIY infrastructure that got the job done early on became a bottleneck as data volume, complexity, and downstream needs grew. Tabnine faced challenges that their existing data infrastructure wasn't built to handle:

## Tracking developer engagement

Because Tabnine runs continuously inside developers' IDEs, understanding developer behavior is significantly more complex than in traditional applications. In most common SaaS or other applications, users go into the specific application to perform a specific task. Unlike these cases, Tabnine lives inside developer IDEs, and in the case of the Tabnine CLI, it actually becomes the agent they use all day to write code, for any purpose. Whether you use an agent to chat with your organization's codebase, get code suggestions in your IDE, or find answers to architectural questions, supporting and understanding the myriad ways users use Tabnine in the SDLC is a true challenge.

## Building a single view of developers who work across multiple environments

The always-on nature of the platform introduced another layer of complexity: developers frequently use Tabnine across multiple IDEs and environments, causing a single individual to appear as multiple users in the data. Because their DIY infra did not resolve identities, the developer lifecycle remained fragmented and incomplete.

As their GTM strategy evolved, Tabnine needed to understand what truly drove meaningful, long-term engagement, but the limitations of their DIY data infrastructure made that level of visibility nearly impossible to achieve. The team needed a data foundation designed for flexibility and control without ongoing engineering overhead.



## The Solution

Using RudderStack, Tabnine was able to rebuild their data foundation around a centralized, warehouse-first architecture, gaining the flexibility of a DIY approach while dramatically simplifying their infrastructure.

## Under the Hood: Tabnine's Data Architecture

RudderStack now acts as the central command for the entire customer data lifecycle between Tabnine's platform, warehouse, and downstream tools. Customer events are collected once and routed through RudderStack. After collection, RudderStack Transformations handles PII masking, geolocation enrichment, event normalization, and the delivery of reliable, trustworthy data across the entire stack.

"If someone needs to add a new source, we can usually do it in a few minutes. It's a simple process now."



Nimrod Astarhan (Nimo)  
Head of Data/Platform

Events are delivered directly to Snowflake, where Tabnine runs dbt models to derive more granular user and company attributes (such as programming languages used, feature adoption patterns, and product intent signals). Then, using RudderStack Reverse ETL, they sync those warehouse-derived traits back into downstream tools like Mixpanel, Customer.io, HubSpot, and advertising platforms, keeping every team aligned with real product behavior.

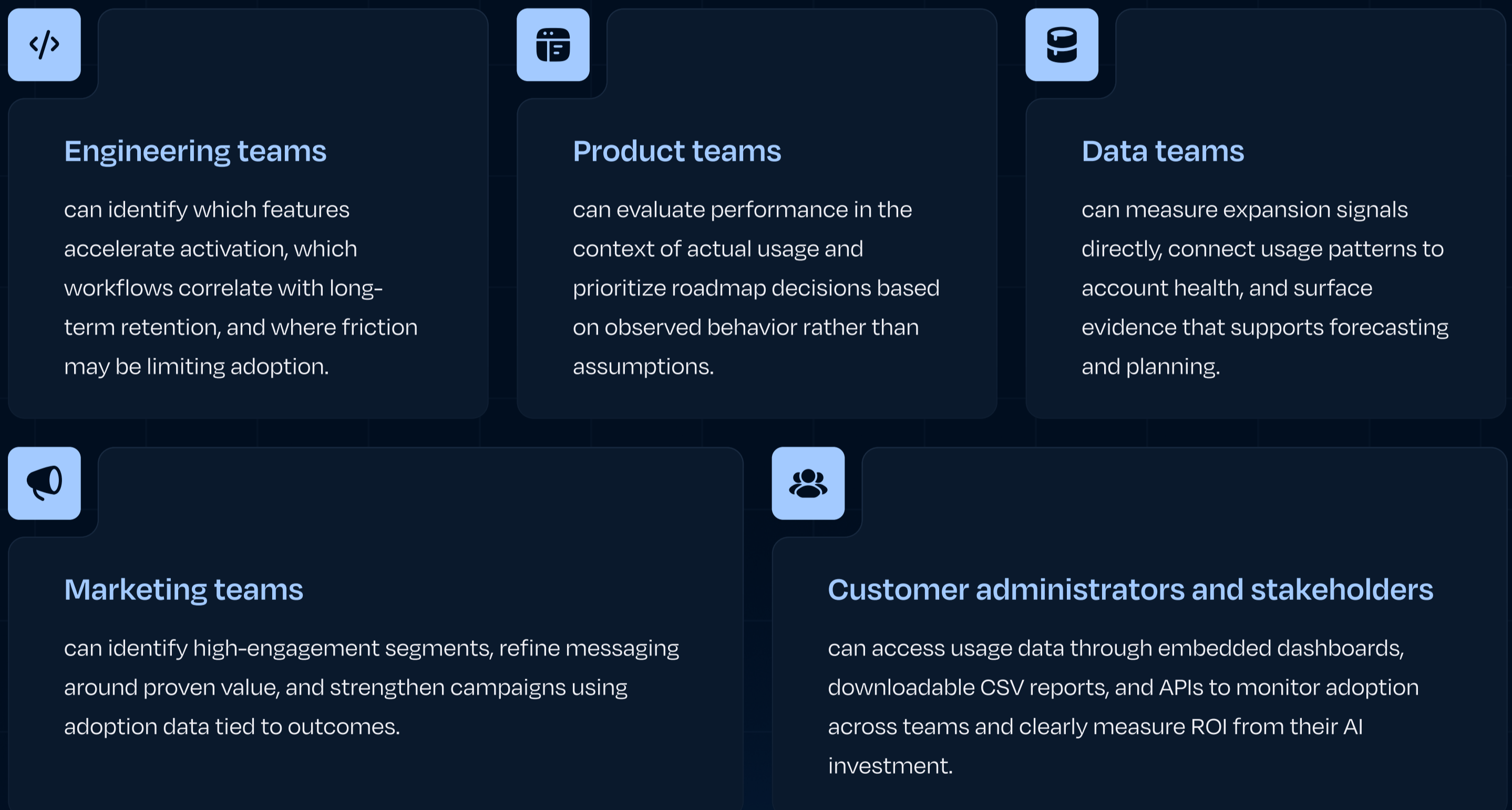
"The best thing about our data infrastructure is that you rarely hear about it. It just works."



Nimrod Astarhan (Nimo)  
Head of Data/Platform

## The Impact: Deep Product Intelligence and Decisions based on Evidence

For Tabnine, product interaction data isn't just about tracking feature clicks or monitoring adoption. It's the foundation for understanding the full developer journey, and turning that understanding into smarter decisions and customer value. The impact shows up across the entire company:



What started as an effort to build a scalable, reliable data foundation became so much more. Now Tabnine has a clear, trusted view of how developers actually engage with AI-assisted development. With RudderStack, every team has trustworthy data, unlocking richer product analytics, more effective marketing, and an expansion strategy built on a data foundation they trust.