tecton + prima

Prima's Journey From In-House Solution to Enterprise Feature Platform

Challenge:

Sustain growing demand and accelerate innovation across global markets

To sustain growing demand and accelerate innovation as Prima expanded, the company aimed to help its data scientists and engineers work more efficiently. With this in mind, Prima first implemented a centralized method through an internal package to more efficiently bring the work of their data scientists into production. Unfortunately, this package caused issues in production because it was designed to work offline, and the online systems were written in a different language.

Solution:

Move from an in-house centralized feature solution to an enterprise feature platform

Prima decided to test the value of an enterprise feature platform by rethinking the architecture of their pricing engine. The engineering team set up a simulated environment that replicated the expected conditions in a new country and used this MVP environment to validate Tecton's functionality by mocking input data rather than actual customer data.

Results:

Increased consistency, productivity, and collaboration across the organization

Not only does Prima's new data platform, of which Tecton is a core component, allow batch and real-time retrieval of features for their pricing engine in low latency, but it also has allowed Prima to centralize its feature transformations, ensure their accuracy and consistency across training and production environments, and allowed data scientists and engineers to collaborate on the design and deployment of new features.

Prima is a data-driven insurance company specializing in the motor insurance market. The company distributes the policies of Great Lakes Insurance (Munich RE group) and iptiQ (Swiss RE) in Italy through its website and a network of brokers and agents. Prima has over 800 employees distributed among its offices in Milan, Rome, London, and Madrid. In 2022, the company expanded policy sales to the United Kingdom and Spain. As of January 2023, Prima has over 2.5 million customers.

> *"By including Tecton in our stack, we've almost entirely automated our release process through CI/CD pipelines. This new process saves time for our machine learning engineers, enabling them to focus on creating and pushing new features to production rather than worrying about the details of the release process."*
>
> — Senior Vice President, Infrastructure & Engineering, Prima

Introduction

Prima is a data-driven insurance company with over 2.5 million customers in the Italian motor insurance market and is expanding into new markets across Europe. At the heart of Prima's success is a talented team that uses predictive models and in-house software to deliver the best possible service to its users.

Prima develops many predictive products, including pricing and underwriting. Approximately 220 engineers and 50 data scientists collaborate on developing and maintaining these products, from design to deployment. Nearly 30 engineers and 40 data scientists work specifically in the pricing area.

As Prima expanded into different regions, the company aimed to help its data scientists and engineers work more efficiently to sustain growing demand and accelerate innovation. With this in mind, Prima first implemented a centralized method through an internal package to more efficiently bring the work of their data scientists into production. Unfortunately, this package caused issues in production because it was designed to work offline, and the online systems were written in a different language.

To address this and restructure the organization, the platform team investigated the use of feature stores and compared other solutions.

Choosing the right feature solution

After considering various options, Prima's engineering team spent months comparing and testing different data platforms and feature stores, creating a list of pros and cons for each. They ultimately chose Tecton's feature platform because of its git-centric management of feature definitions that allowed easy integration with their CI/CD tools and because the platform supports and orchestrates transformations as first-class citizens.

On the other hand, other feature store options were mainly web interfaces where very little was known under the hood. Another advantage the Prima team found is that **Tecton's feature platform encompasses all aspects of the ML feature lifecycle**, such as:

- Defining feature definitions in code and storing those definitions in a centralized repository
- Transforming raw data into feature values
- Storing and serving those feature values
- Monitoring and automating the associated pipelines

In contrast, other feature store options only store and serve feature data, and do not provide a way to re-execute the transformation in the future.

Furthermore, the Prima team felt that Tecton was the best choice since it was the first company to make feature stores mainstream, and the platform was more mature than other providers in the market. They also believed working with the Tecton team would be more beneficial due to Tecton's extensive experience in a new space.

To confirm Tecton's effectiveness, the team at Prima decided to run a POC by rethinking the architecture of their pricing engine. The engineering team set up a simulated environment that replicated the expected conditions in a new country. They developed a framework that was intended to be more general than the specific needs of a single country to create a reusable minimum viable product (MVP). They used this MVP environment to validate Tecton's functionality by mocking input rather than customer data.

Using the MVP environment allowed them to test Tecton without the added complexity of actual customer data while avoiding the additional complications related to legacy code and complex feature pipelines that would have been required in existing geographies. The team started from scratch with a proper feature store, designing a system that would leverage Tecton as a core element for feature engineering.

Prima and Tecton, in practice

Prima's pricing engine

Prima's pricing engine uses price comparison websites, which all have their own suggested fields, as the main entry point for collecting requests for quotations. The engine then gathers all the necessary information, calculates the features, and uses a machine-learning model to compute prices. The price comparison website then displays Prima's prices to the user as an offer. If the user is interested, they can click on the offer to be redirected to Prima's website and continue the process.

From Raw Data to Risk Score

Raw

id	type	year
1	car	1990
2	moto	2005
3	car	2015

Features

type	age
car	32
moto	17
car	7

Risk Model

Out

price
200
100
400

Challenges

Prima built the original architecture for the pricing engine in Elixir. The teams extracted features in Python, then replicated them in Elixir. But because most features require live data processing to materialize values on demand, translating from Python to Elixir occasionally introduced bugs. On the other hand, SQL queries directly on operational data simulated the features in a "batch" way, but they were actually computed and processed online. The only requirement for using the package was the ability to write extractions in SQL and transformations in Python.

Designed to be used offline and configured to extract values based on definitions provided by data scientists, the system's primary function was to centralize these definitions so that users could access and retrieve feature values from various services as needed. The system served as a centralized location for feature definitions and helped automate some processes like extraction based on the provided definitions. In other words, it acted as an in-house feature store. Although this setup ensured a standardized definition for features, it also introduced many challenges.

First, feature definitions vary by the person designing them. For instance, there are multiple ways to evaluate a vehicle's age, such as mileage, rust, etc. Each method may yield slightly different results, especially when approximating the age to the year. When Prima first started working on the pricing engine, they had around 10 data scientists who each had his or her definition of what vehicle age meant and wanted their own version in production. However, in production, there can only be one definition. This issue was not limited to features; it also extended to other practices, such as how data scientists handled environments and models.

Second, the system was incompatible with on-demand or online feature computation. For the pricing engine to work, it needs to retrieve on-demand and pre-computed feature values exceptionally quickly. In Spain, the engine receives 0.3 requests per second, and in Italy, it receives ~3 requests per second. The desired latency for model prediction, feature computation, and other processes is in the hundreds of milliseconds to avoid a slow network experience for the user. The pricing engine relies on on-demand access to:

- **Real-time feature values:** The main categories of information fetched include vehicle and demographical features. The license plate number is an example of something fetched in real-time or on demand. Because these features are on-demand, the engine cannot pre-compute and store their values. Instead, it needs a system to centralize and execute the transformation process online, as needed.
- **Pre-computed feature values:** Other information that is fetched from either pre-computed or batch sources include vehicle features and geographical features. Vehicle features include information such as the powertrain and model of the car, while geographical features include the average number of claims in a specific area and meteorological data. These features can help sell insurance for specific purposes, such as windshield insurance, by knowing the weather patterns in a specific area.

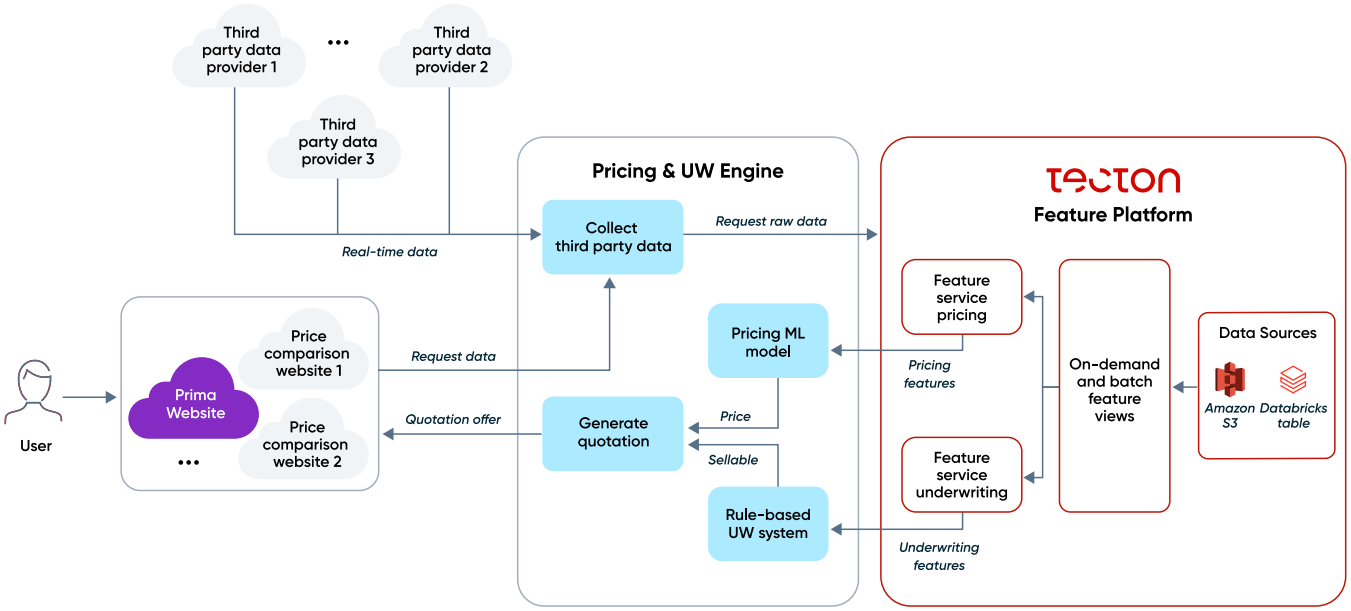
Finally, while this package was helpful for offline work, the team encountered many challenges when implementing new features into the pricing engine and deploying them to production environments. Because data scientists wrote their feature extractions in SQL or Python, but the pricing engine was built in Elixir, they were forced to bring the features to production through specification documents. These files were often riddled with errors and caused significant problems in production. In the best-case scenario, it took a long time to bring new features into production—and in the worst case, the team was unaware that there was a misalignment between what they intended to do and what was actually happening in production. This resulted in sometimes selling policies at the wrong price.

Solution

The Prima team first created a list of complex items to do in their current architecture and tried to replicate those tasks in Tecton using the same data sources. This process took several months but ultimately helped them gain confidence in Tecton's capabilities.

After customizing and improving the architecture, they conducted various tests, including load testing and integration testing, to ensure Tecton was ready for production. The tests were successful, and Tecton was successfully deployed in two clusters: one for production and one for staging and development.

Pricing & Underwriting Engine Architecture



With this new architecture, Prima's release process is almost entirely automated through CI/CD pipelines. Before releasing a data scientist's latest feature, the engineers manually ensure it works correctly using synthetic and functional tests. If the feature performs well in testing, it is deployed to the staging environment. If it performs well in staging, the engineers merge it into the master branch of the Git repository before deploying it to production. This streamlined process saves time for the machine learning engineers, enabling them to focus on creating and pushing new features to production rather than worrying about the details of the release process.

Results

Overall, adopting Tecton has helped Prima improve the efficiency and reliability of their pricing engine and significantly accelerated the path to production of the model in new geographies.

Today, Tecton is a core component of Prima's data platform. Not only does Tecton's feature platform allow batch and real-time retrieval of features for their pricing engine in low latency, but it has also allowed Prima to centralize its feature transformations and ensure their accuracy and consistency across training and production environments.

- Tecton average response time: 27 ms
- Engine average latency (without considering calls to external data providers): 155 ms
- End-to-end avg latency (considering calls to external data providers): 1.6 s

After the POC, Prima settled on Tecton to launch their pricing engine in a new geography. Tecton's opinionated approach to defining features, maturity, and integration with Git and CI/CD were critical factors in the decision. Tecton also allowed on-demand feature extraction in a language the data scientists and engineers were familiar with, which was crucial for Prima's needs.

Prima's new data platform, of which Tecton is a core component, allows data scientists and engineers to collaborate on the design and deployment of new features. Data scientists create new features that the engineers review to ensure the code is maintainable and efficient. The engineers also provide examples of different features to help the data scientists understand how to create similar features in the future.

For instance, when data scientists see an example of a new feature, they can replicate it and create dozens of similar features quickly, significantly reducing the time spent on the initial development of a new feature. Let's say the data scientists need to add a new feature with complex logic, such as streaming. In that case, the engineers can easily use Tecton to create a template and show the data scientists how to replicate it, allowing them to develop similar features efficiently.

Tecton also allows for advanced features like backfilling and time travel, which can be helpful when re-applying transformations or working with new data. Using such features with Prima's original system meant relying on the accuracy of the feature tables created by others and losing the lineage of where the values originated. With Tecton, users can now track where data comes from and see the transformations applied to it.

Tecton also powers Prima's Underwriting Product. Tecton computes features from raw data and Prima uses these features to determine the specific products to offer each user for their car insurance. Currently, this decision is made using a rule-based system that considers geographical and user-related features rather than a machine-learning model. However, Prima is considering using a machine-learning model for this process in the future.

Learn more at tecton.ai