DOMINO

The True Cost of Building a Data Science Platform

WHITEPAPER

Table of Contents

Executive Summary

As data science evolves and becomes more integral to your company's success, your data science team will likely make requests around "data science platform" requirements. It can be tempting to try and quickly meet these requirements by expanding on your current Docker, Kubernetes, and Git initiatives, but that can be a very costly and time-consuming path.

We have seen IT departments build data science platforms that support basic requirements around IDEs (Integrated Development Environments), tools, and scalable compute. But inevitably, these "built from scratch" platforms are under scoped and do not satisfy internal customer requirements. Nor do they improve the processes around the data science lifecycle. They require significant ongoing maintenance and support to keep up with the quickly evolving data science tooling and framework ecosystem. Unfortunately, what happens is these "basic" platforms quickly bloat into multi-year, multi-million dollar projects that are outside the IT department's core competencies and don't meet the needs of the data scientists. After investing significant time and money, we regularly see organizations scrapping these projects and turning to a commercial solution.

Introduction

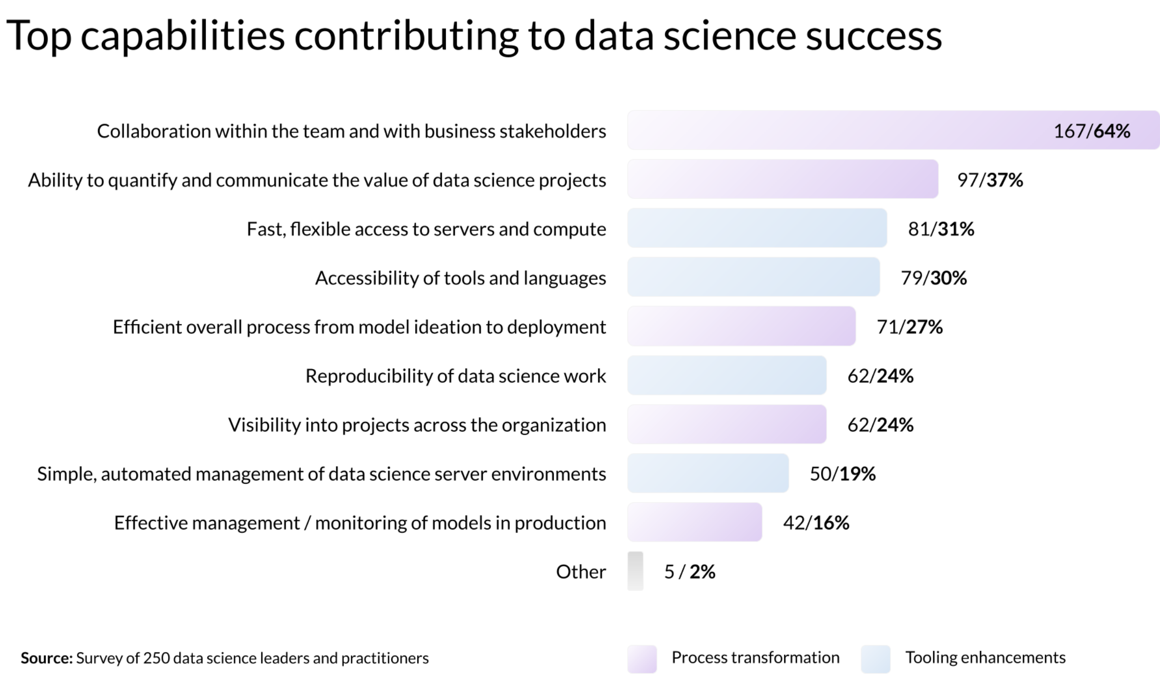
Data science is an integral discipline to many industries. Manufacturing, finance, insurance, technology, agriculture, and many other industries now benefit extensively from data science, particularly machine learning. According to a recent DataIQreport, one in four businesses expect data science to impact topline revenue by more than 11%! It is also becoming a competitive necessity, with 76% of executives surveyed by Accenture believing their organization will go out of business if they don't scale the use of data science.

Many companies already have one or more basic platforms for data science. These companies' IT departments have cobbled together basic solutions that involve Docker, Kubernetes, Jira, and/or Git, and can support data scientists' basic tooling/IDE and scalable compute requirements. But if you want data science to scale and become highly impactful in your company, it's vital to move beyond those basic platform capabilities and support all aspects of the data science lifecycle. It's also important to not confuse the criticality of data science to your long term success with the need to build the underlying platform. Data science is your competitive advantage, not the platform!

The Key Capabilities for Data Science Success

In a 2018 poll we conducted of 250 data science leaders and practitioners, we categorized their responses to what contributed to data science success into 2 buckets: tooling enhancements and process transformations. And success, as was noted above, is about scaling data science across the business.

As you can see, while access to compute, tools, and languages is important to the success of data scientists, process transformation capabilities such as collaboration, communication and business value are ranked higher overall.



Based on our experience working with clients, key process capabilities include:

- Data science leaders need organization-wide visibility to all projects and their status to more effectively manage and ultimately justify scaling their workforce.
- Data engineers need to assist the data science team with getting access to the various datasets they need for projects.
- Business domain experts need to work with the data science team to ideate and clarify business problems at all points in the data science lifecycle.
- Model validators need to collaborate with the data science team to challenge assumptions, re-run experiments, and ensure models meet ethical standards.

- Machine learning engineers need to collaborate with the data science team to deploy and monitor the models.

Data Science Wasn't Built to Scale

In most organizations, data science has evolved as a grassroots effort that was distributed in pockets across the business. Today, even if your data science team has access to centralized tooling and compute, it's very likely they still do not have a seamless way to expose their work to the many stakeholders that need to be involved. This leads to a series of process breakdowns that slow data science to a crawl:

- If data science practitioners do not have the necessary tools to learn from each other and prior artifacts, they end up with duplicative projects and inefficient work.
- If data science teams frequently find it difficult to expose their work to business stakeholders, they waste valuable time re-packaging their work into emails or slides.
- If model validators have to re-create the data science team's entire project environment from Git repositories, file shares, emails, and PDFs in a separate system, they waste months with error-prone manual efforts to get models deployed
- If data science leaders do not have the tools they need to gain easy insight into model development progress and model performance, they cannot effectively manage data science portfolios.
- If there is no central repository of data science knowledge, knowledge is lost when key people leave the organization
- If there is no centralized way to provision tools and compute, CIOs and IT leaders struggle to meet the constant demands to manage and secure all the software and hardware stacks with limited budgets. Data scientists can't get access to the tools and compute they need without wasting significant time on DevOps tasks.
- If the data science development environment is separate from the production environment, the handoff for model deployment can be time-consuming and error-prone.

When you have one or more of these process breakdowns it can have a dramatic impact on your data science success and throughput. According to the DataIQ reportt, one-third of organizations needed months to get models deployed into the hands of end-users due to ad hoc processes and legacy infrastructure.

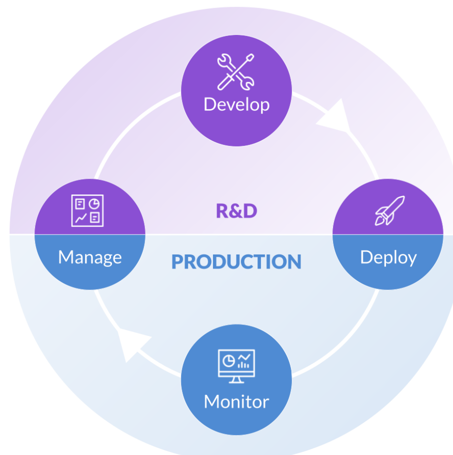
The only way to tackle these challenges is to move beyond a siloed set of tools and processes that create friction and chaos for data science.

Scale Data Science With an Enterprise MLOps Platform

We believe data science platforms need to incorporate eight critical capability levels to allow data science to scale and deliver on its promise. Our approach was developed through working with data science teams in over 20 percent of the Fortune 100. Many of them tackled building a data science platform from scratch and ultimately realized they would be better off focusing their time and efforts on the competitive value from data science rather than the platform. That's when they turned to Domino to deliver the full level of functionality needed to operate at scale.

Platforms that deliver on all eight capabilities can be defined as Enterprise MLOps platforms. They encompass all aspects of the data science lifecycle, making it easier for companies to manage, develop, deploy, and monitor business-critical models *faster*, at enterprise scale, with the requisite security, governance, compliance, reproducibility, auditability, etc. that are required to do this *safely* and *universally*. That is what we designed our Enterprise MLOps platform to deliver.

Data Science Lifecycle



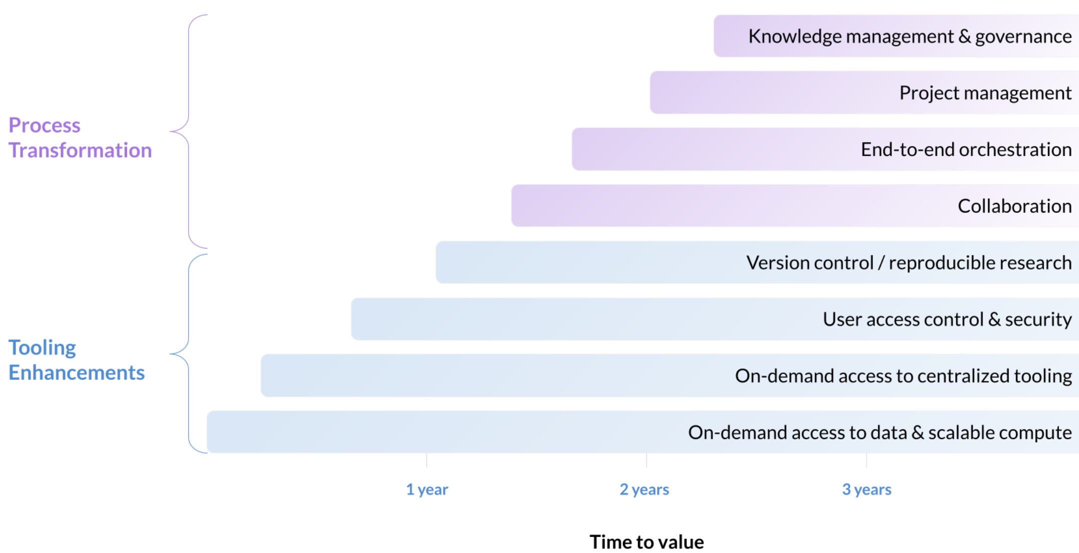
The value from an Enterprise MLOps platform can be significant. For example, in early 2021, Forrester completed a Total Economic Impact (TEI) Study of our Enterprise MLOps Platform and identified over $30 million in total economic value over 3 years, with a 542% ROI for the average customer.

> "The biggest piece is collaboration. Having multiple people working in the same project and having all your data and code in the same box has been huge. It's a big way that people share code and insights across their teams, especially data science teams, and that's been a really nice thing."
>
> **—Cloud Analytics Lead, Agriculture, Forrester TEI**

The Eight Data Science Platform Capability Levels

In our experience, we have found it easy to underestimate what is needed in a data science platform. The requirements are complex and they constantly evolve. This makes building one from scratch hazardous and time consuming. Because it takes time to gather requirements and build a platform, it can be years before the benefits start to accrue to the users. It is even easier to overlook the higher-level capabilities that transform data science processes and deliver the greatest value to the organization.

Each of the eight capability levels add increasing value to the organization and must exist to *safely* and *universally* scale data science. When they all exist, you get a true Enterprise MLOps Platform.



Tooling Enhancements

Level 1: On-demand Access to Data & Scalable Compute. Data is at the heart data science! At its most basic level, a data science platform must provide its users with governed, self-service access to diverse datasets. This accelerates projects and dramatically reduces risks related to data security. Data scientists also need self-service access to scalable compute to perform their development tasks, such as model training on large datasets or image classification with GPUs.

Level 2: On-demand Access to Centralized Tooling. There are tremendous advances happening in the open-source community and commercial data science applications, which means the desired toolset will continually evolve. Today, data scientists often download open source tools to their laptop or a local server in order to get access to the latest capabilities. By doing so they inhibit reproducibility and collaboration because there is no consistency around tool versions, as well as introduce significant security and governance risk. A data science platform needs to satisfy data scientist needs for on-demand access to the tools they use today such as SAS, Jupyter, and RStudio, and whatever new open-source IDE comes along in the future, without compromising security.

> "The biggest benefit we've seen is to auto-onboard somebody into our data science analytics environment in about 20 minutes of a quick run through a tool versus weeks it would take previously for somebody to get configured on their own."
>
> **—Senior Director of Decision Sciences, Software Services, Forrester TEI**

Many companies mistakenly define a data science platform to include only these two levels. That's why it's not surprising that many IT teams attempt to build their own data science platforms using the multitude of open-source tools that are readily available. Early progress can be so tantalizing, but a data science platform with only these capabilities omits the higher levels of functionality that are needed to make data science a scalable capability in your organization.

Level 3: User Access Control & Security. The data science team frequently requires access to the most sensitive data within an enterprise. Therefore it is absolutely critical that a data science platform gives IT a way to properly permission user access, track activity, and provide secure environments for development and production.

Level 4: Version Control/Reproducible Research. As organizations move beyond a few data scientists and models, version control and reproducibility become critical. Version control for traditional software projects can typically be satisfied by checking code in and out of a Git repository, such as GitHub or GitLab. Data science work, however, requires code, data, results, model artifacts, software versions, libraries and collaborative comments to be version-controlled in order for the research to be reproducible. These additional artifacts simply cannot be managed via an external Git repository.

This level of functionality resolves many of the daily DevOps challenges that data scientists spend much of their day tackling. It also provides the necessary capabilities for easily reproducing work for audit, compliance or regulatory purposes. Without reproducibility, IT and data science teams have to manually try to reproduce all components in a model, which can be a very time-intensive task.

Process Transformations

Level 5: Collaboration. To scale research, collaboration is key for ideation and improvement. Simply not reinventing the wheel brings tremendous productivity increases. It also makes it easy to bring new team members up to speed quickly. They need to expose their work to lots of other non-technical stakeholders. Collaborative conversations need to be captured within the platform, as opposed to dozens of email and Slack threads, as they are relevant to the overall data science project and lifecycle. The team will want to retain conversations around the goals, analyses, and experimentation that took place, key decision points, and so on.

> "It's so much easier to share models and see what everybody is working on. In our homegrown environment, it was up to the data scientist to figure out what artifacts and data they needed to save, but in Domino, that's all built into the platform. You can select any model run or experiment and see what version of the model and what data were used, the dependencies and what packages of software were used."
>
> **—Erich Hochmuth, Director of Engineering, Data and Analytics Group, Climate Corporation**

Level 6: End-to-End Orchestration. As enterprises seek to scale data science in production from dozens to hundreds or even thousands of models, they will benefit from many of the same engineering and operational practices that DevOps brought to software development. The discipline of MLOps accelerates the entire model lifecycle process and ultimately allows organizations to industrialize and scale machine learning.

The key is to have an enterprise-grade MLOps platform that can eliminate manual, inefficient workflows across all the activities of the data science lifecycle. If the lifecycle phases are broken up into different systems, it creates handoffs that induce process friction and break model lineage. This becomes especially important as production models often have to be re-trained, sometimes by a different data scientist.

Level 7: Project Management. As data science scales to support a greater number of projects worked on by large, geographically-dispersed teams of data scientists, it becomes increasingly difficult to manage effectively. We regularly see Chief Data Scientists, VPs of Data Science, and other senior leaders spending significant amounts of their time on status calls with individual members of their team to track progress, and document roadblocks on spreadsheets. They have higher value work they should be doing.

Project management capabilities within a centralized data science platform allow data science leaders to set standard processes that need to be followed by all participants. This capability can also track projects throughout their lifecycle, including the ability to link key artifacts to project goals to document progress, and detail roadblocks that can be addressed by the data science leader or worked on collaboratively with other members of the data science team.

Level 8: Knowledge Management & Governance. The final capability that is needed to scale is a comprehensive view and governance of your data science program delivered through a system of record that captures the intellectual property created by anyone involved in data science in your organization.

A system of record for data science can bring many of the same benefits that a sales leader gets from Salesforce, or an HR leader gets from Workday:

- Documentation of all aspects of a project in a centralized and searchable repository to accelerate future research, prevent re-work, and document best practices.
- Capturing critical knowledge so it isn't lost when data scientists or other users leave a company or lose their laptop.
- Tracking of all projects in process, their phases, models in production, and costs.
- Workload management.

These capabilities ultimately lead to consistent value delivery to the business, quantification of project and model impact, and effective management of the data science workforce.

The True Cost of Building a Data Science Platform

Lower level data science platform functionality can clearly be cobbled together using a combination of open-source tools and internal development. For some organizations this may be an acceptable solution if your requirements are modest and you're willing to accept the trade-offs that come with open-source software (e.g., little/no support, limited documentation, unclear roadmap/future).

However, with machine learning increasingly becoming key drivers of organizational performance, businesses must be able to manage, develop, deploy, and monitor models consistently and at scale. This will necessitate the higher levels of functionality be present in a platform, which unfortunately also brings increasing development complexity. Additionally, there is the cost of ongoing support – including training, documentation, regression testing, and more.

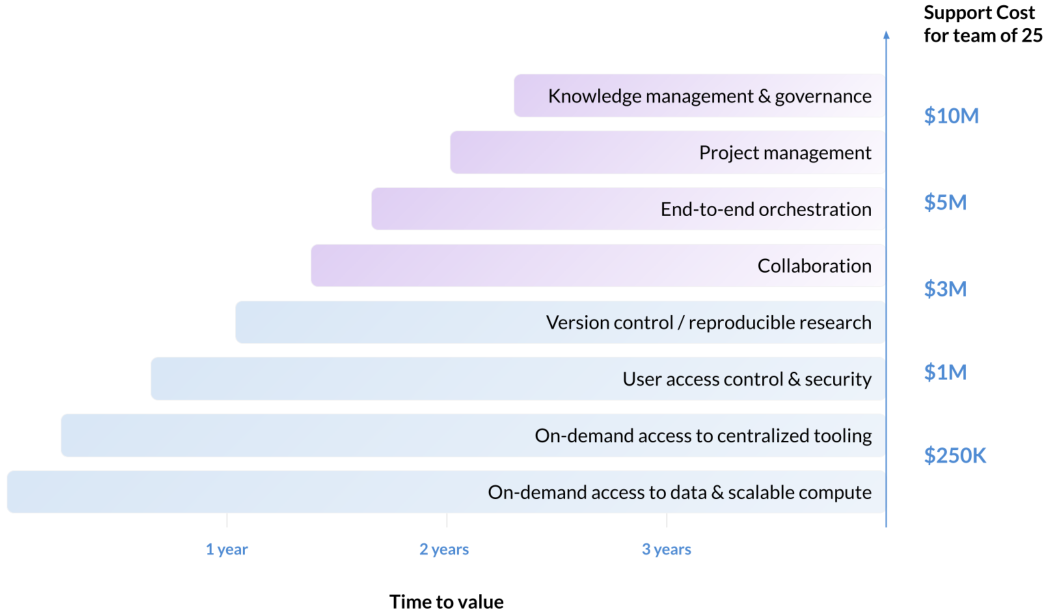
You also have to account for the rapid innovation happening in data science to continue to meet the needs of your data scientists. Just a few years ago, GPUs, TensorFlow, and Spark were rarely used for data science. Today they are common. Since then, new technologies, such as TPUs, PyTorch, and Dask have been introduced and rapidly adopted. The speed of change is staggering – for Python alone today there are over 137,000 libraries, with new ones constantly being added.

> "If we hadn't invested in Domino, first, I wouldn't have been able to set up a team at all, because you can't hire a high-skilled data scientist without providing them with the state of the art working environment."
>
> **—Chief Analytics Officer, Insurance, Forrester TEI**

Building and maintaining a high-performing data science platform is no easy task. More and more headcount are required to build out the platform as you add higher-level functionality, and to keep up with rapidly-changing data science technologies. It takes time to build out a platform, which means those features of highest value to your data scientists won't be delivered for multiple years. And when they are delivered, the need may have changed. A high-performing data science platform also requires the best practices of a software company, like roadmap development, user testing, and performance testing. Do you or your IT department really want to take on all these new responsibilities?

> "Our leadership directive is, if it's not a differentiating capability, we shouldn't be building it; we should be looking to buy it. In my experience there is initial excitement about building in-house tools, and they're great for two years, and then by year three, nobody cares about maintaining them anymore."
>
> **—Senior Director of Decision Sciences, Software Services, Forrester TEI**

Increasing Cost to Build Higher Value Features



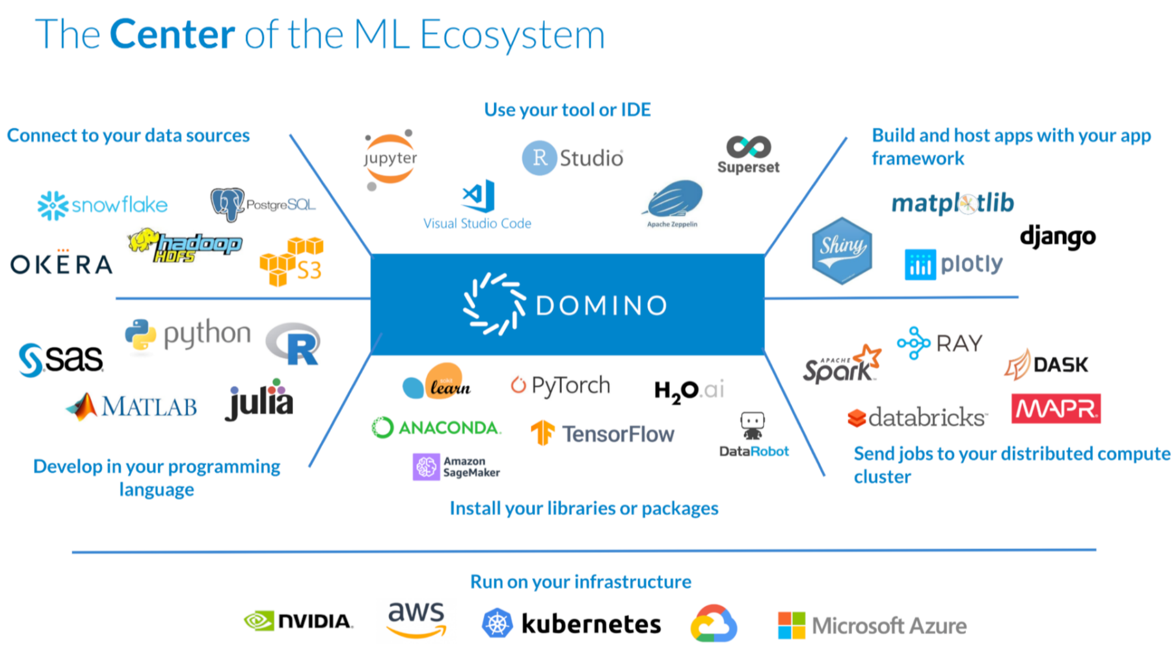
There are many additional risks when IT tries to build its own data science platform. Frequently the solution is under-scoped and does not meet the needs of all users. A common area ignored is the needs of data engineers, ML engineers, model validators, and other contributors to the data science process. Another is the product and knowledge management capabilities that streamline the governance, oversight, and value documentation processes. Transitions between steps in the data science lifecycle are also problematic if there are handoffs between tools, which can add friction to workflows.

If a platform doesn't meet the needs of everyone, it only serves to propagate silos of tooling and knowledge. Another major risk is the strain IT teams experience when they try to manage and govern multiple systems, which often results in costly budget overruns.

Focus on the Highest ROI and Fastest Time-to-Market Approach

The most successful IT teams Domino has worked with understand the complexity of building and deploying a full-featured data science platform. Given the urgency behind data science, and the time and cost associated with scoping, developing, and supporting a data science platform, the most agile departments are choosing to buy Domino's Enterprise MLOps platform. This is because Domino's platform provides market leading Enterprise MLOps capabilities with seamless integration throughout the data science lifecycle so that your data science team can get an instant productivity boost, rather than being beta testers over the course of a phased, multi-year rollout of in-house software.

Domino also includes a wide variety of integrations with the most popular tools. This means that rather than start with a blank slate, you can work with our experienced team to integrate Domino into your unique set of backend systems and data to make it work optimally in your environment. We have worked with our Fortune 100 customers to support deployment of Domino into some of the most challenging environments imaginable.



Once those integrations are done, the benefits are substantial. The Forrester Total Economic Impact Study identified that the average customer would see:

- **Reduced time-to-configure compute resources worth over $9.7 million**. With Domino, data scientists can instantaneously spin up new data science environments to begin working immediately, saving on average 70 hours per instance.
- **Accelerated time-to-onboard new team members worth over $984K**. Domino's intuitive interface and support for familiar tools means that little or no training is necessary. On average, each data scientist is productive in just one day as opposed to two weeks in prior environments.
- **Increased data science efficiency through the use of new tools in a shared environment worth over $5 million.** With Domino, data scientists quickly collaborate on models, share common solutions, and build prototypes to code against. Managers estimate savings of 200 hours per year per data scientist.

- **Incremental profits of over $5.1 million.** Increased collaboration and interconnections during the development cycle lead to data science solutions that are better aligned to business needs and directly impact revenue.

Those averages are proven out in Domino's customer's results:

- Lockheed Martin unified cross-functional teams with an open platform, which in turn streamlined development and deployment of deep learning models. They realized over **$20 million in annual benefits** from **10x data scientist productivity and efficiencies, 90%** DevOps **savings**, and faster onboarding and offboarding.

- Bayer improved visibility and collaboration across hundreds of projects, and also leveraged best practice templates. Through these efforts, **they reaped a 400% improvement in model delivery speed, and $100 million in net present value** generated over three years.

- Moody's Analytics centralized some of its data science projects on a data science platform, and thereby dramatically increased the efficiency of model development, and expanded its ability to build collaborative models with clients and partners. Moody's Analytics realized over **a 50% reduction in model development time, six times faster deployment of models, and four times more frequent validation and updates of models**

Case Study

A Chief Data Officer at an insurance company hired a consultant to build a custom data science platform. The platform was pretty basic – it focused on tooling only, and there were constant struggles with upgrading the platform to support new technology and increasing the volume of users. Initial savings and excitement quickly gave way to user frustration because the platform over-promised and under-delivered.

While IT scrambled to bolt on new features to a platform that ultimately wasn't open enough, users went back to laptops and other shadow IT practices – the very things that the system was hoping to prevent. This Chief Data Officer ended up taking a role with another insurance company where she faced a similar challenge around scaling data science. This time, with her experiences (and scars) fresh in mind, she brought in Domino for an "off-the-shelf", best-in-class Enterprise MLOps platform. The Domino solution came with full maintenance and support, as well as a roadmap that continues to support the evolving data science ecosystem.

The Chief Data Officer focused her team on building out the custom integrations to Domino that were unique to her company, and within weeks the data scientists were already seeing significant productivity gains. Ultimately, her company was able to realize significant revenue and operational improvements that simply would not have been possible had teams been distracted by in-house development efforts vs. data science experimentation and deployment.

Conclusion

Many IT departments that start building a data science platform never see the benefits they expected. Some of the mistakes they make include:

- They do not consider the needs of multiple classes of users.
- They focus on short-term tooling enhancements and ignore much more impactful process transformation opportunities.
- They get locked into an ecosystem that doesn't support future requirements.
- They do not anticipate the demands of software development and ongoing maintenance efforts.
- They compromise security and collaboration by building multiple tools instead of one cohesive platform.

The data science lifecycle benefits most from improved processes and collaboration with business stakeholders that come from a data science platform that encompasses all eight levels of capabilities. Domino's Enterprise MLOps platform provides not only a scalable compute grid to execute development and production workloads, but also collaboration, knowledge management, model deployment, governance, reporting features, and model monitoring. Domino provides a single place for data scientists to access data, develop in their language of choice with their tooling of choice (from SAS to RStudio to any future IDE), and manage all their packages.

With Domino, IT can be the hero of data science by building custom integrations around an open platform that makes the life of the data scientist easy. IT no longer has to worry about the rapidly evolving data science ecosystem. Domino future-proofs IT with our open platform and broad support. Lastly, for IT teams, Domino is able to consolidate multiple existing software/hardware stacks into a single system, which massively reduces the management complexity and cost burden that falls on IT.

About Domino

Appendix: Checklist of Features for the Eight Data Science Platform Capability Levels

Each level should include the following core functionality to build a minimum viable product.

Level 1: On-demand Access to Data & Scalable Compute

- ❑ Self-service data source access (S3, Hadoop, CSV, SQL, Snowflake, etc.)
- ❑ Self-service elastic compute access (CPU, GPU, Spark, etc.) with horizontal and vertical scalability.
- ❑ Distributed compute framework support (Spark, Dask, Ray, single machine, TensorFlow).
- ❑ GUI and CLI access.
- ❑ Cloud and on-premises infrastructure integration (with the ability to support a hybrid cloud).
- ❑ Model containerization in scalable environments such as Kubernetes.
- ❑ Cluster orchestration.
- ❑ File transfer and remote code execution.
- ❑ Fault tolerance and retries.
- ❑ Heterogeneous resource management.
- ❑ Job prioritization.
- ❑ Intelligent machine assignment and project caching.
- ❑ Horizontal scalability and load balancing.
- ❑ High availability.
- ❑ Zero-downtime upgrades.
- ❑ Concurrency.
- ❑ Logging.

Level 2: On-demand Centralized Tooling

- ❑ Broad language support (Python, R, SAS, MATLAB, etc.).
- ❑ Broad IDE/tool support (Jupyter, RStudio, Apache Zeppelin, etc.).
- ❑ Broad package support (Scikit Learn, PyTorch, Anaconda, H20.ai, TensorFlow, DataRobot, etc.).
- ❑ Multiple languages, tools, and packages supported within a single project.
- ❑ Multiple experiments execute simultaneously.

- Side by side comparison of experiment results across all tools and packages).
- Tool stack consolidation.
- Rapid provisioning of new/new versions of open-source tools and packages.
- New versions of tools and packages enabled globally, individually or by group without affecting the user's current workflow.

Level 3: User Access Control & Security

- Dataset access controls (e.g. read, edit/modify, copy).
- Project and notebook execution access/permissions controls.
- Role-based access controls (administrator, private/individual, public/anyone, collaborators/team, specific user, etc).
- End-to-end encryption.
- Active Directory integration.
- LDAP integration.
- Administrative monitoring (track users, projects, deployments).
- Secure package repository.
- User ID management.
- Multi-tenant repository support with access restriction controls.

Level 4: Version Control/Reproducible Research

- Code, data, tool, package, environment, parameter, and associated discussion tracked automatically.
- Model version control (set-up, manage and maintain.)).
- Enable reproducibility of prior results by tracking code, parameters, and environment setup.
- Model copy/fork from any version within the repository.
- Prior project version restoration.
- Version control repositories integration (GitHub, GitLab, Bitbucket, etc.).
- Automatic package federation to all projects in an environment.
- Automatic environment versioning.

Level 5: Collaboration

- Streamline and facilitate data scientist collaboration (share and discuss code, results, experiments, notebooks, and datasets).
- Flag blockages to solicit help from colleagues.
- Comment creation and response.
- Data and information sharing with business users or stakeholders, and include them in the model development process.
- Best practice documentation (processes, project templates, and pre-written code snippets).
- Capture and retain model features.
- Feature calculation catalog and management.
- Associate feature functions to a source code repository.
- Enterprise planning tool integration (e.g. JIRA).
- Shareable data science containers and revisioned datasets definition.

Level 6: End-to-End Orchestration

- Model deployment as apps to support use by business users (Flask, Shiny, etc.).
- Model deployment as real-time API endpoints for integration into enterprise applications.
- Model export as Docker images to CI/CD pipelines or other production infrastructure.
- Batch process support.
- JSON to/from in-memory R and Python processes marshaling.
- A/B testing.
- Re-training task scheduling.
- Deployed model monitoring.
- Data drift monitoring.
- Model drift and model quality determination and remediation.
- User alerting for data drift and/or model quality problems.
- Validation workflow management (including partnership with Quality Assurance teams).
- Traditional IT statistics monitoring and reporting (response rates, uptime, throughput/prediction volume, etc.).

Level 7: Project Management

- Project goal tracking and documentation.
- Project portfolio tracking across lifecycle stages.
- Model tracking across the production lifecycle.
- Project metadata tracking (top users per project, machine and tool usage, time spent, etc.).
- Compute cost visibility from a user, project and group perspective.

Level 8: Knowledge Management & Governance

- Capture and catalog all model development activities for knowledge sharing and retention.
- Production workload environment control.
- Project-level resource usage dashboards.
- User-level resource usage dashboards.
- Compute spend usage auditing, monitoring and attribution.
- Full model provenance log.