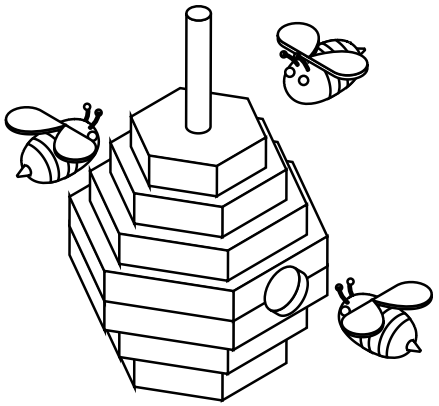
Building Production-Ready Predictive Analytics

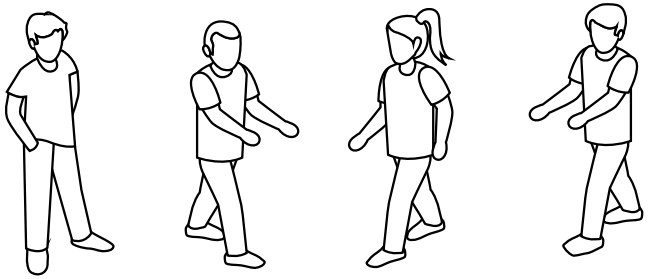
www.dataiku.com

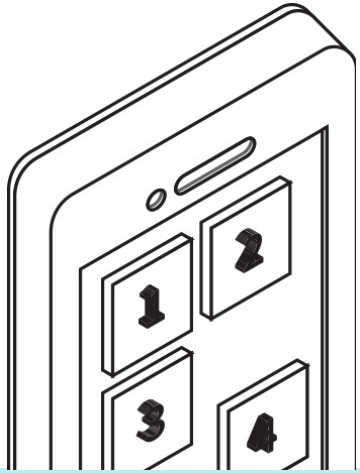


There's a part of data science that you never hear about: the production. Everybody talks about how to build models, but not many people worry about how to actually use those models.

Yet production issues are the reason many companies fail to see value come from their data science efforts.

We wondered how companies handled their production processes and environments to build production-ready data products, and we figured the easiest way to find out was to ask them.





We conducted a worldwide survey and asked thousands of companies. And we got our answers.
After analyzing those answers, we isolated four different ways companies are dealing with production today, and we put together a series of recommendations on how to build production-ready data science projects.

OUTLINE

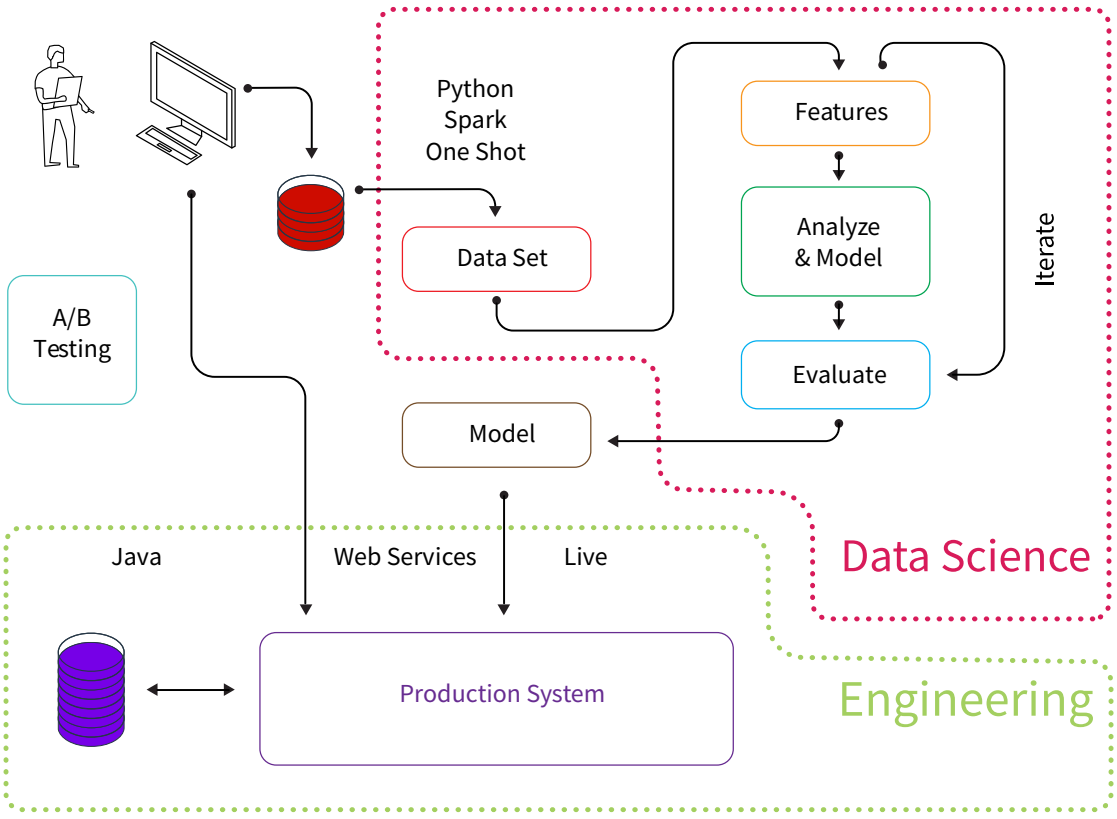
What Does It Mean To Go “Into Production”?

The data science process is extensively covered by resources all over the web and known by everyone. A data scientist extracts some data, splits it, cleans it, builds features, trains a model, deploys it to assess performance, and iterates until he’s happy with it. That’s not the end of the story though. Next, you need to try the model on real data and enter the production environment.

These two environments are inherently different because the production environment is continuously running. Data is constantly coming in, being processed and computed into KPIs, and going through models that are retrained frequently. These systems, more often than not, are written in different languages than the data science environment.

You can see a simplified schema of the interactions between the two environments on the right.

The challenge of deploying into production first arises obviously when the model is deemed sufficient and has to be deployed onto the existing production environment. It also arises for every iteration on these environments, whether it’s new analytical opportunities or changes in data.



What Does It Mean To Go “Into Production”?

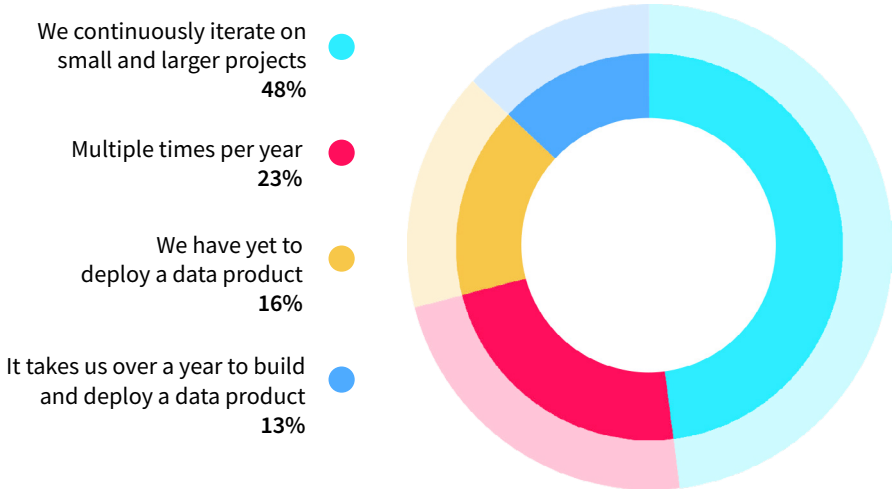
It’s fundamental to look into how companies have handled the transition between data science and production environments to ensure data projects will be successful.

There are several things to look at when assessing how effective your set-up is at bridging the gap between development and production:

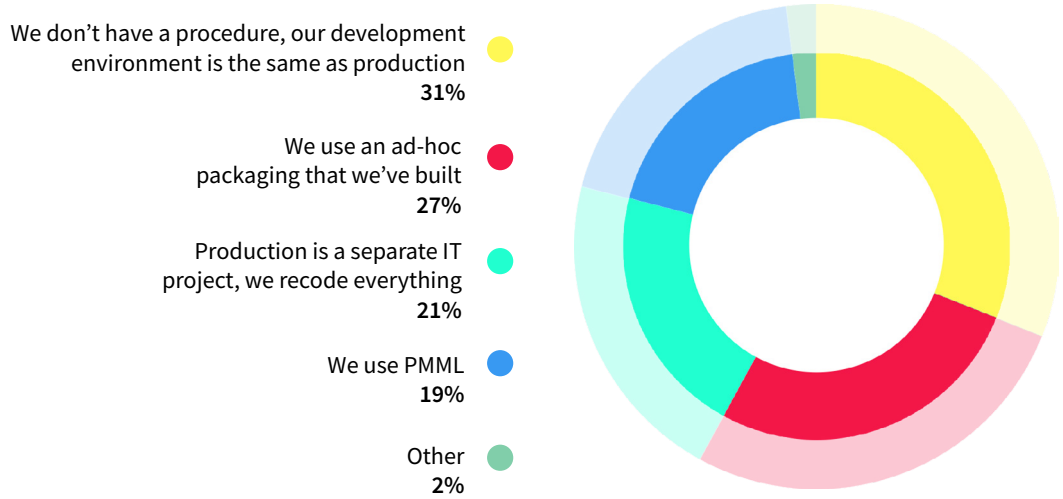
- Best operating procedures: Managing environmental consistency, data scalability, and consistent code and data packaging.
- Risk management for unforeseen situations: Roll-back and failover strategies.
- On-going iteration: Continuous retraining of models, A/B testing, and multivariate optimization .
- Implementing communication strategies: Auditing and functional monitoring.
- Impact of complicated technological stacks: Real-time scoring and online learning.

Consistent Packaging and Release

How often do you deploy data products into production?



What kind of procedure do you follow to package a data transformation or model before release?



A data project is a messy thing. It's lots of data in loads of different formats stored in different places, and lines and lines (and lines!) of code and scripts in different languages turning that raw data into predictions.

Packaging all that together can be tricky if you do not support the proper packaging of code or data during production, especially when you're working with predictions.

How do companies handle this?

The first option is to constantly deploy small iterations that have little chance of compromising the project. Creating ad-hoc packaging is a costly (but good) solution since it means you control the process from beginning to end. A typical release process should be:

1/ Have a versioning tool in place to control code versioning.
2/ Create packaging scripts to package the code and data in a zip file.
3/ Deploy it into production.

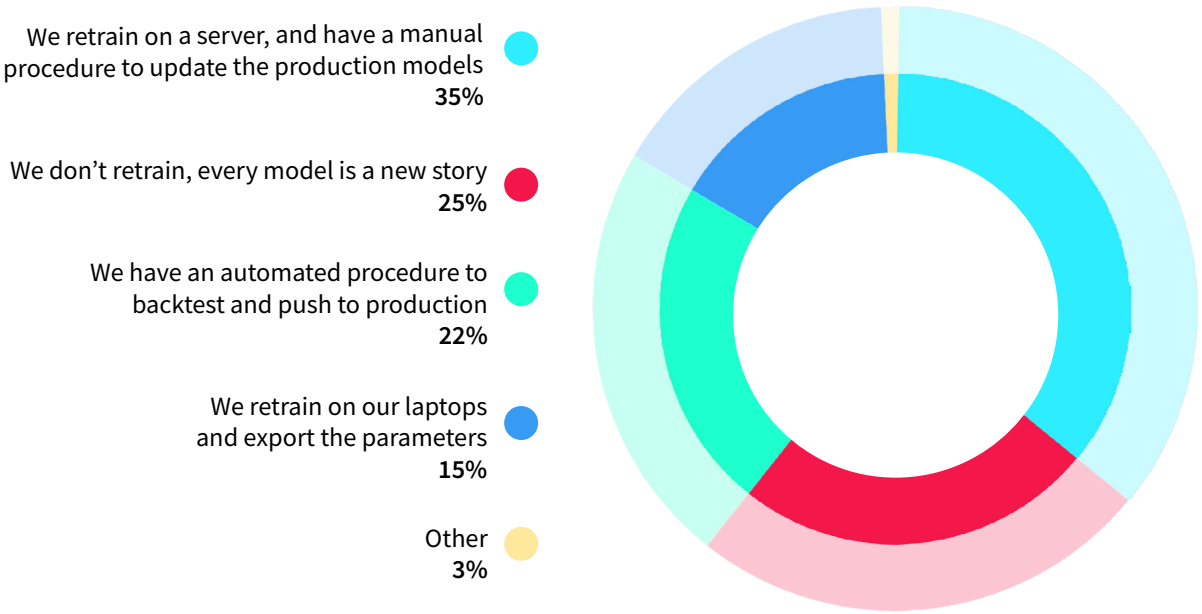
BIG TREND : ***IT Controls Production***

50% of all respondents do not have a specific data science production procedure.

Data production and process is an IT-lead project (only 17% use PMML).

Continuous Retraining of Models

What kinds of strategies have you implemented to retrain models?



Small iterations are key to accurate predictions in the long term, so it's critical to have a process in place for retraining, validation, and deployment of models.

Indeed, models need to constantly evolve to adjust to new behaviors and changes in the underlying data.

The majority of respondents retrain manually on a server or on a laptop, and only 22% automatically push model updates into production. Also, 38% of respondents state that they perform online machine learning.

The key to efficient retraining is to set it up as a distinct step of the data science production workflow. In other words, an automatic command that retrains a predictive model candidate weekly, scores and validates this model, and swaps it after a simple verification by a human operator.

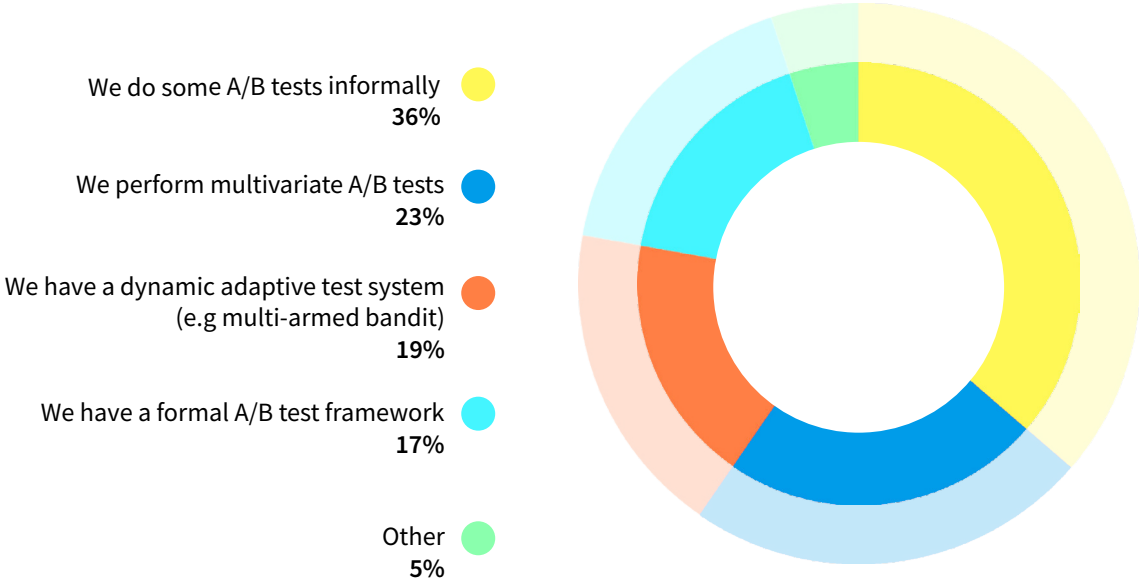
This is most successfull with the implementation of automated model scoring to automatically compare metrics of old and new models. In more advanced data projects, multivariate testing can be used.

***BIG TREND:** Backtesting*

Backtesting has yet to become mainstream, with only 22% of companies reporting automated procedures to backtest and push to production.

Multivariate Optimization

What kind of multivariate testing or strategies do you have in place for predictive models?



A model has to be validated and monitored after it has been deployed into production because real life data as well as the model itself will almost always differ from your design environment.

Of respondents, 76% perform A/B testing to evaluate multiple models in parallel.
This means maintaining previous versions of your data product with different versions of your model and comparing the end performance depending on the model. This is one of three options to validate, along with multi-armed bandit testing and multi-variable armed bandit testing with optimization (algorithms that select the best model).

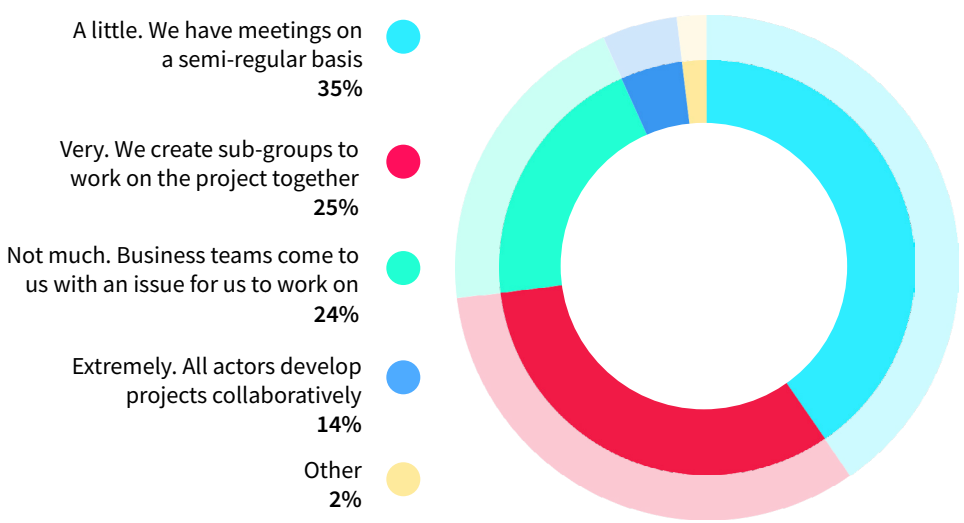
In any case, as you set up your data science production process you should be aware of the importance of 1/ Comparing your model against a baseline; 2/ Comparing two live versions of your algorithm; 3/ Eventually continuously optimizing and trading off different versions of your model.

***BIG TRENDS:** A/B testing is the rule*

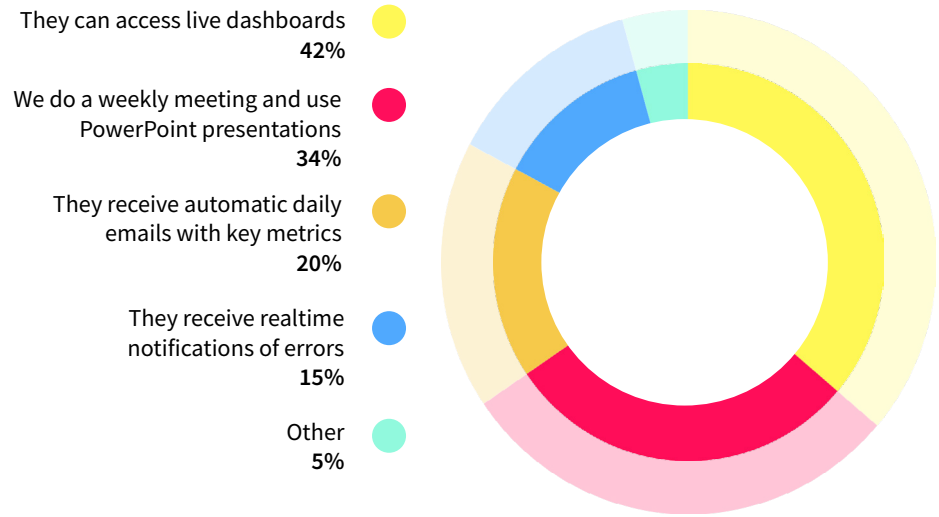
The logic of A/B testing for optimization dominates (76% of respondents). However, over half of the respondents built a dedicated framework to perform these tests.

Functional Monitoring

How involved are business teams in the data science process?



How can business people monitor the performance of the application?



A critical challenge in any data science project is getting everyone on the same page, including the business sponsors and end users of the project.

This is critical during the development of the project to ensure that the end product is understandable and usable by business users. Respondents seem to be aware of this but somewhat reluctant to push it very far, with only 40% reporting a strong involvement.
Once the data product is in production, it remains critical for business users to assess the performance of the model, since they base their work on it.
There are several ways to do this; the most popular is setting up live dashboards to monitor and drill down into model performance. Automatic emails with key metrics can be a safer option to make sure business teams have the information at hand.

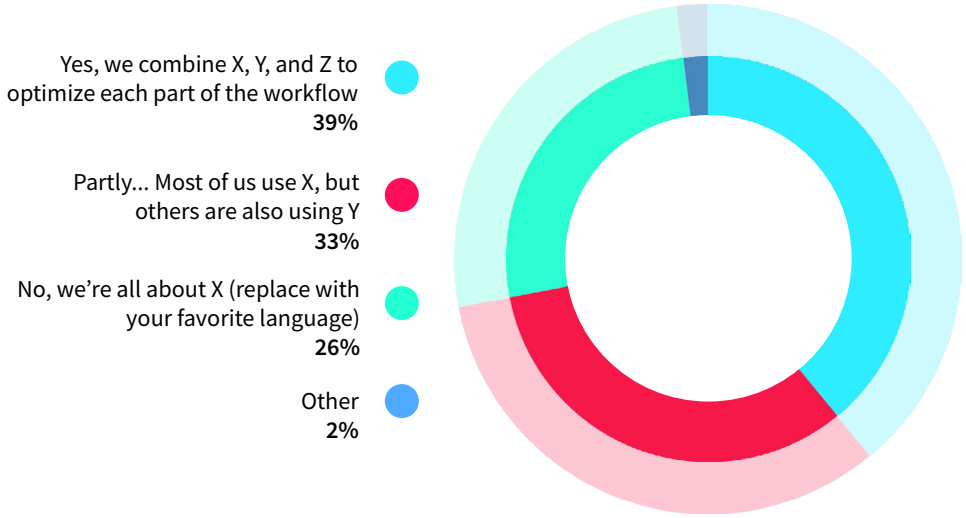
It's surprising that only 15% of respondents report the real-time notification of errors such as model performance drifts, data sync failures, etc. This can be concerning for real-time use cases as well as for pricing or churn prediction.

BIG TREND:* *Business Collaboration

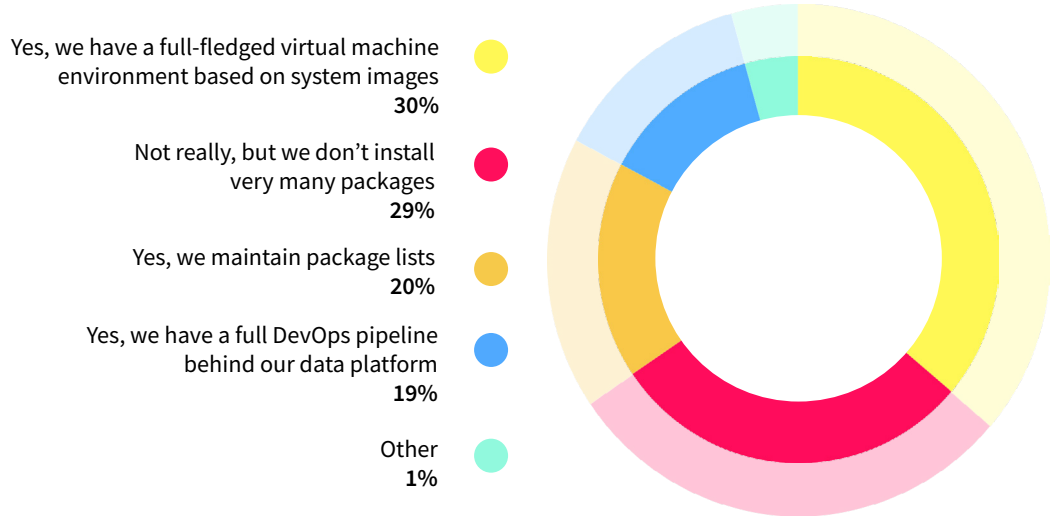
Only 33% describe collaboration between business and data science teams positively, with the main mode of communication on data projects still being PowerPoint or live dashboards (for 70% of respondents).

IT Environment Consistency

Do you support multiple languages and frameworks in your environment?



Do you have a specific strategy for IT environment consistency?



Of our respondents, 75% declare the use of multiple languages and frameworks in their data science environment.

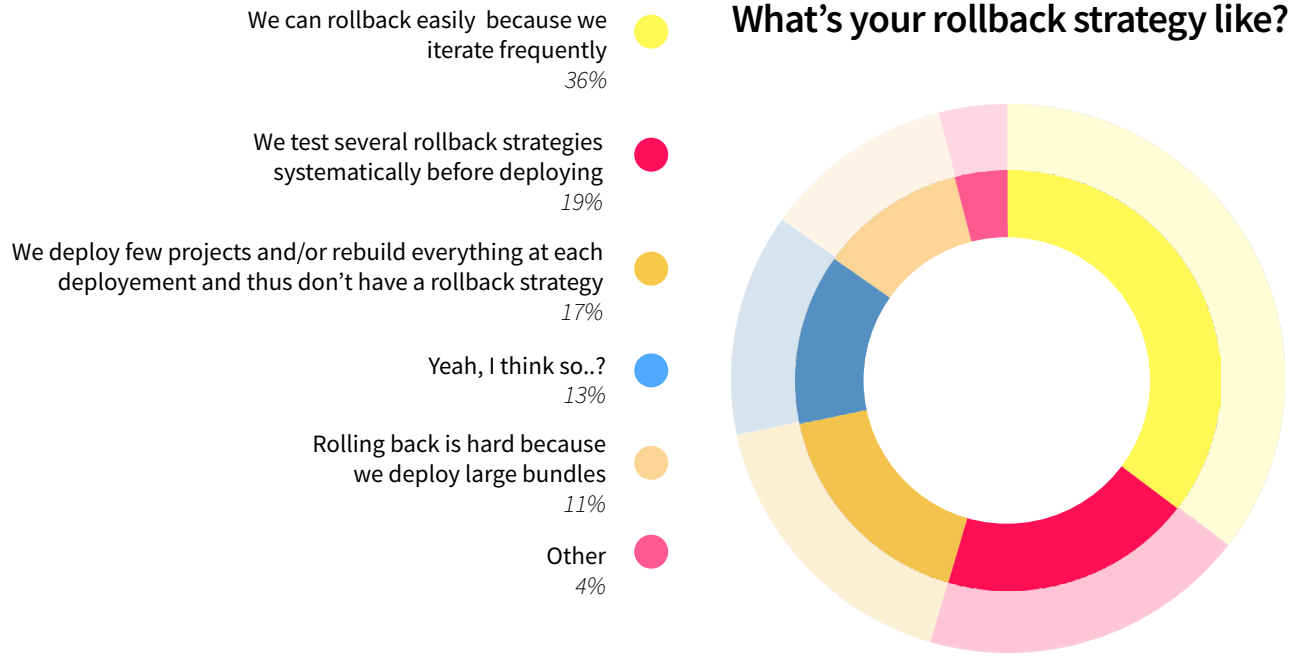
Modern data science relies on the use of several technologies such as Python, R, Scala, Spark, Hadoop, along with open source frameworks and libraries. This can cause an issue when production environments rely on technologies such as JAVA, .NET, and SQL databases, which could require complete recoding of the project.

The multiplying of tools also poses problems when it comes to maintaining the production as well as the design environment with current versions and packages (a data science project can rely on up to 100 R packages, 40 for Python, and several hundred Java/Scala packages). To manage this, two popular solutions are to maintain a common package list (20%) or to set up virtual machine environments for each data project (30%).

BIG TRENDS: ***Business Collaboration***

Multiple languages as a norm: 80% of people have a polyglot development environment.

Rollback Strategy



With efficient monitoring in place, the next milestone is to have a rollback strategy in place to act on declining performance metrics.

A rollback strategy is basically an insurance plan in case your production environment fails.

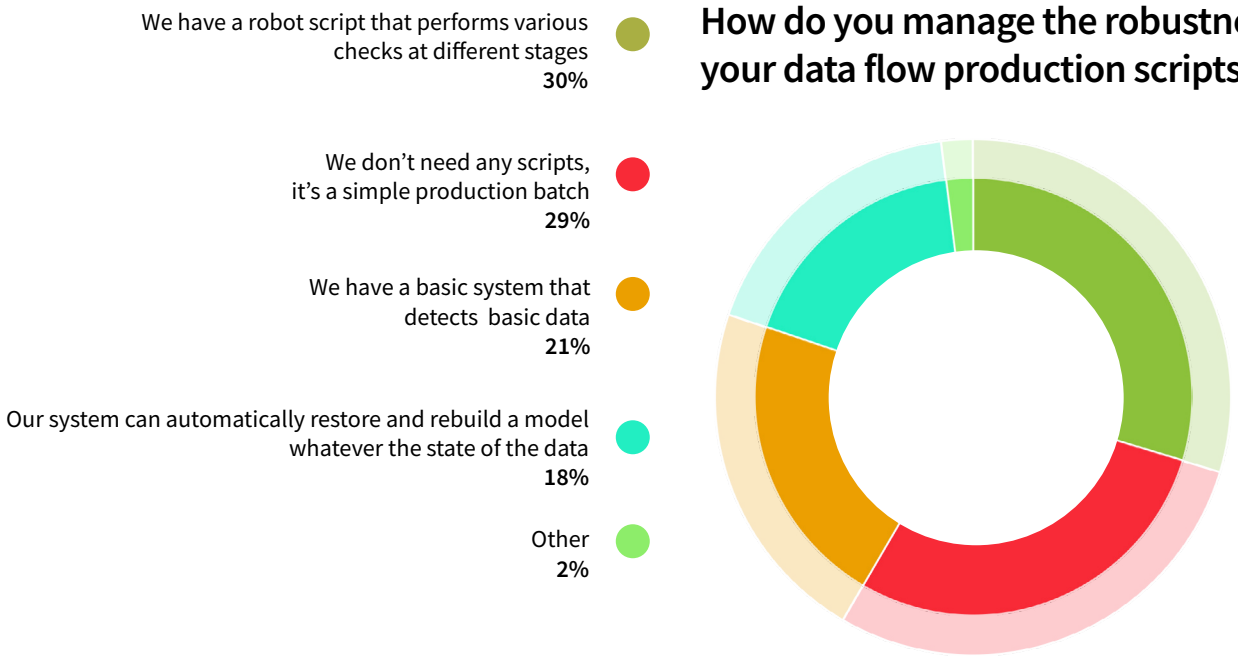
Yet 41% of respondents are facing trouble with rollback strategies or may not have one set up. Only 19% have tested rollback strategies they have implemented.

A good rollback strategy has to include all aspects of the data project, including the data, the data schemas, transformation code, and software dependencies.

It also has to be a process accessible by users who aren't necessarily trained data engineers to ensure reactivity in case of failure.

Failover Strategy & Robust Scripts

How do you manage the robustness of your data flow production scripts?



In case of system failure, a robust failover strategy has to be built in to take over in certain scenarios. An effective failover strategy has to do the following:

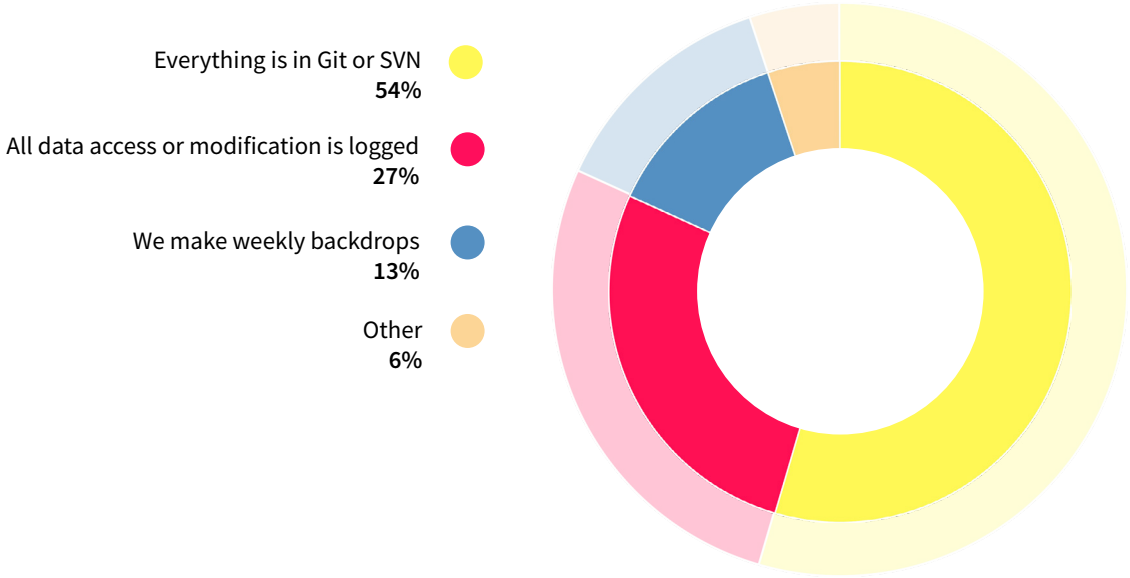
- Integrate all of the events in the production system
- Monitor the system based on an exhaustive overview of possible failover scenarios
- Immediately alert IT as well as users in case of failure
- Automatically re-run and recover in case of failure

Only 18% of respondents have a global failover strategy, while 29% don't have a failover strategy in place. This is explained by the unique challenges big data environments bring to failover strategy formulation.

The volume of data makes it impossible to merrily rebuild, meaning the workflow has to re-execute intelligently.

Auditability & Version Control

What are your version control capabilities?



Being able to audit to know which version of each output corresponds to what code is critical. Tracing a data science workflow is important if you ever need to trace any wrongdoing, prove that there is no illegal data usage and privacy infringement, avoid sensitive data leaks, or demonstrate quality and maintenance of your data flow.

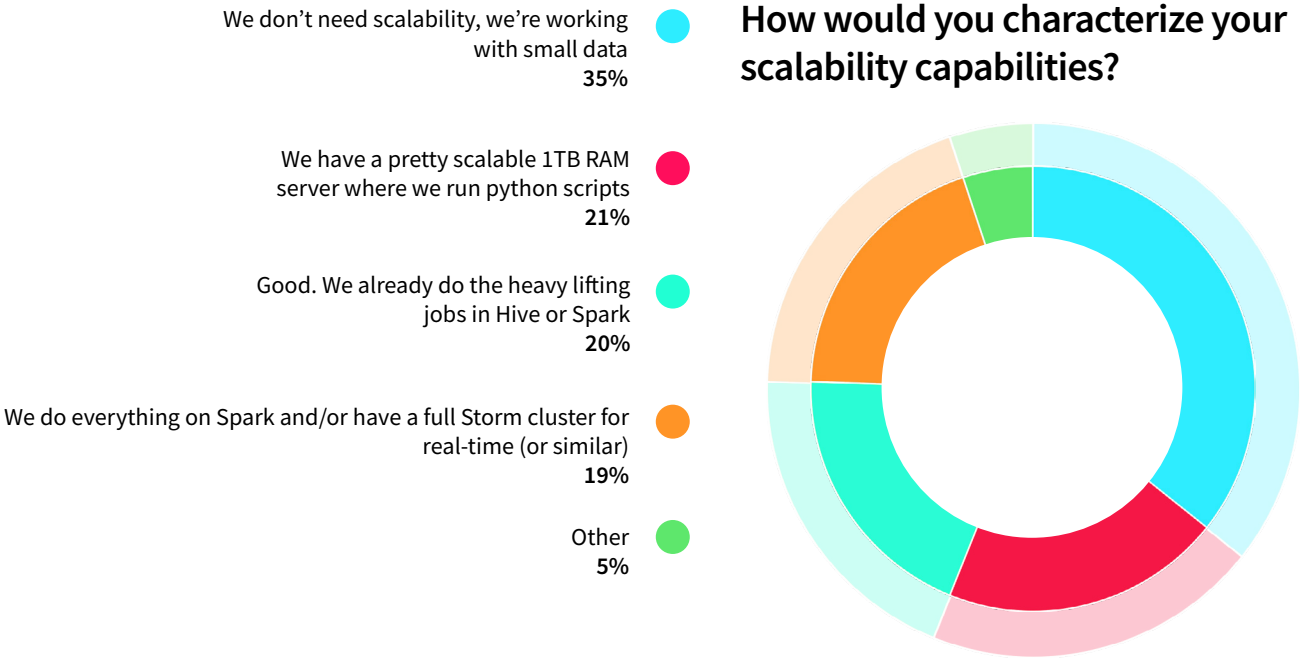
The most common way to control versioning is (unsurprisingly) Git or SVN, popular with over 50% of respondents. However, keeping logs of information about your database systems (including table creation, modifications, and schema changes) is also a best practice, adopted by a third of respondents.

***BIG TREND:** King Git*

50% of companies use classic configuration management tools, such as Git, for their production projects.

Performance and Scalability

How would you characterize your scalability capabilities?

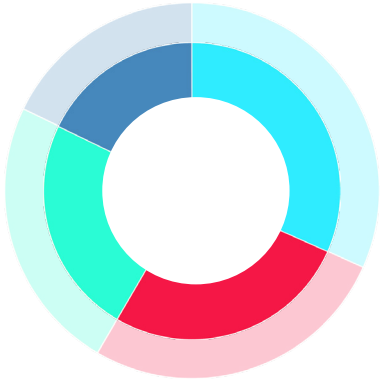


As your data science systems scale with increasing volumes of data and data projects, maintaining performance is critical. This means setting up a system that's elastic enough to handle significant transitions, not only in pure volume of data or request numbers, but also in complexity or team scalability.

A sizable 35% of respondents are content with their system because they're not working with big data.
But scalability issues can come unexpectedly from bins that aren't emptied, massive log files, or unused datasets. They're prevented by having a strategy in place to inspect workflows for inefficiencies or monitoring job execution time.

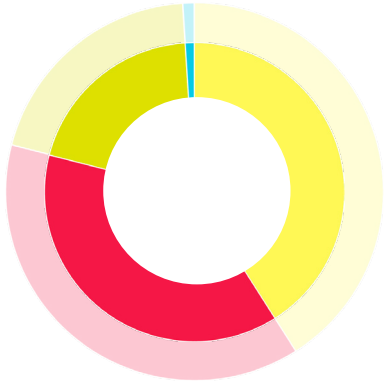
Real-time and Online Learning

What type of scoring do you do?



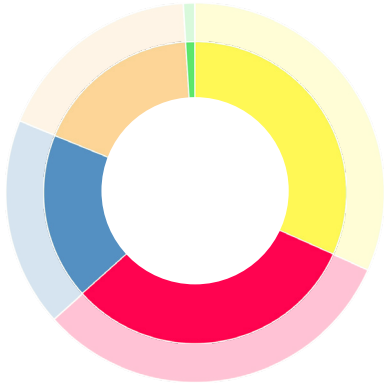
- We do some of both **32%**
- Batch (i.e., we recompute customer segments every night) **27%**
- Real-time (i.e., we have a streaming engine or an API setup with an app) **23%**
- I don't do scoring **18%**

Do you use offline or online learning?



- We only have algorithms **41%**
- We use a combination of online and offline machine learning **38%**
- We only do batch calculations offline (mostly PAC) **19%**
- Other **1%**

How have real-time scoring or online machine learning impacted your production process?



- I don't do either, so this question doesn't apply to me **32%**
- We have had to buy a distinct design test deployment and monitoring pipe **31%**
- Not much; we have roughly the same processs for all our systems **18%**
- We had to adapt our process to better monitor performance and drift **18%**
- Other **1%**

Real-time scoring and online learning are increasingly trendy for a lot of use cases including scoring fraud prediction or pricing. And 70% of respondents who do scoring use a combination of batch and real-time, or even just real-time scoring. In addition, 38% of respondents report using online machine learning.

These technologies lead to complications in terms of production environment, rollback and failover strategies, deployment, etc. However, half of the respondents using one or the other declared that they didn't have to develop a distinct process and only made monitoring adjustments.

Where Do You Fit In?

Because we work with data every day, we thought we'd take our work to the next level and we trained a KMeans algorithm to find clusters in our respondents.
We found four ways of handling data science production processes, and a few outliers.

Small Data Teams - 23%

Focus on building small projects fast: Standard machine learning packages with a unique server and technical environment for all analytics projects.

> **3/4** Do either Marketing or reporting.
> **61%** Report having custom machine learning as part of their business model.
> **83%** Use either SQL or enterprise analytic databases.

These teams, as their name indicates, use mostly small data and have a unique design/production environment. They deploy small continuous iterations and have little to no rollback strategy. They often don't retrain models and use simple batch production deployment, with few packages.
Business teams are fairly involved throughout the data project design and deployment.

Average level of difficulty of deployment: 6.4.

Packagers - 27%

Focus on Building a Framework (the software development approach): Independent teams that build their own framework for a comprehensive understanding of the project.

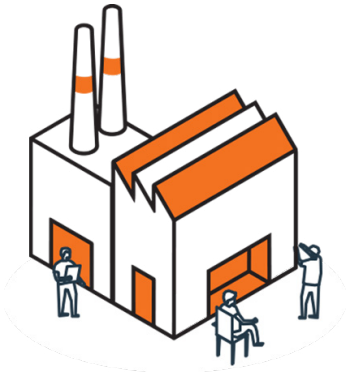
> **48%** have set-up Advanced Reporting.
> **52%** of respondents mix storage technologies.
> **63%** use SQL and open source.

These teams have a software development approach to data science and have often built their framework from scratch. They develop ad-hoc packaging and practice informal A/B testing.

They use Git intensely to understand the globality of their projects and dependencies, and they are particularly interested in IT environment consistency.
They tend to have a multi-language environment and are often disconnected from business teams.

Average level of difficulty in deployment: 6.4.

Where Do You Fit In?



Industrialization Maniacs - 18%

Focus on Versioning and Auditing: IT-driven teams think in terms of frequent deployment and constant logging to track all changes and dependencies.

> **61%** have Logistics, Security, or Industry Specific use cases.
> **30%** have deployed Advanced Reporting (vs 50% of all respondents).
> **72 %** use NoSQL and Cloud.

These data teams are mostly IT-led and don't have a distinct production environment. They have complex automated processes in place for deployment and maintenance. They log all data accesses and modifications, and have a philosophy of keeping track of everything. In these setups, business teams are notably not involved in the data science process and monitoring.

Average level of difficulty in deployment: 6.9.

The Big Data Lab - 30%

Focus on Governance and Project Management: Mature teams with a global deployment strategy, rollback processes, and preoccupation with governance principles and integration within the company.

> **66%** of companies have multiple use cases in place.
> **50%** do advanced Social Media Analytics (vs 22% of global respondents).
> **53%** use Hadoop, and two thirds of them only use Hadoop.

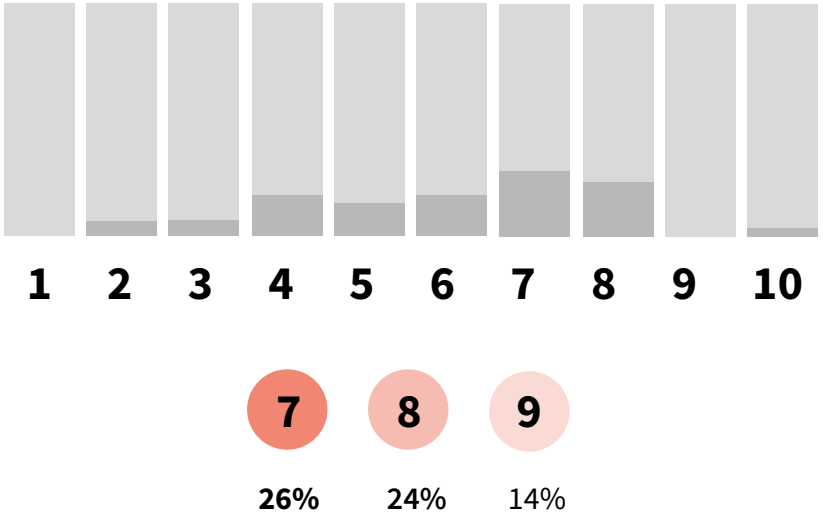
These teams are very mature with more complex use cases and technologies.
They use advanced techniques such as PMML, multivariate testing (or at least formal A/B testing), have automated procedures to backtest, and robust strategies to audit IT environment consistency.

In these larger, more organized teams, business users are extremely involved before and after the deployment of the data product.

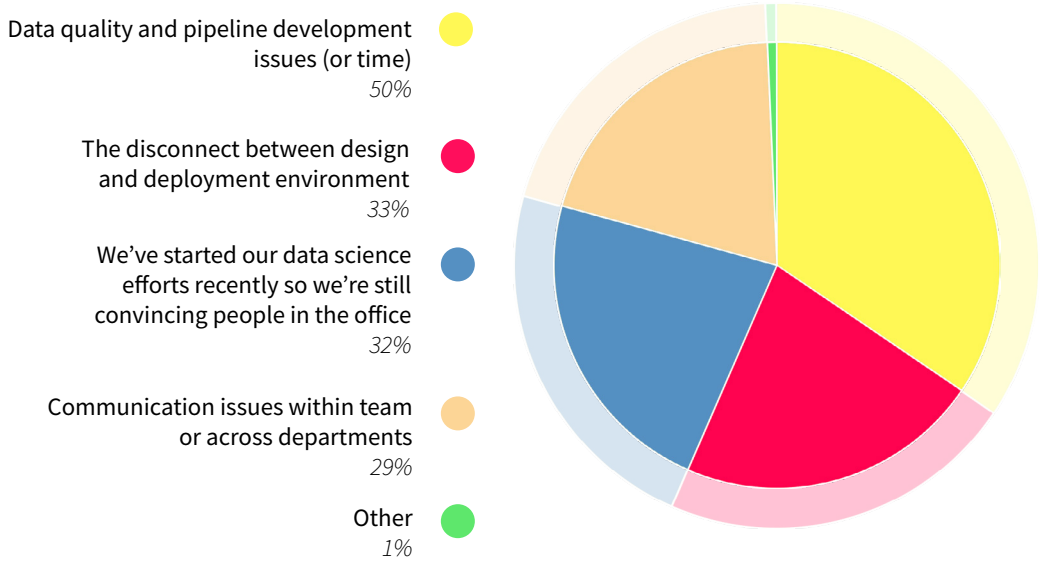
Average level of difficulty in deployment: 5.6.

Building Production-Ready Data Products is Hard

On a scale of 1 to 10, what level of difficulty is involved in getting a data product in production?



What are the barriers potentially keeping you from deploying your data projects?



The average reported difficulty of deploying a data project into production is 6.18 out of 10, and 50% of respondents state that, on a scale of 1 to 10, the level of difficulty involved in getting a data product in production is between 6 and 10. The main reported barrier to production (50% of respondents) is data quality and pipeline development issues (or time).

The second cause is a three-way tie between "We've started our data science efforts recently so we're still convincing people in the organisation of the use cases," "the disconnect between design and deployment environments," and "communication issues within teams or across departments."

BIG TREND: **Data Quality**

A whopping 50% of companies agree that the number one barrier to data deployment is data quality and pipeline development issues (and time to do so).

CONCLUSION

WHERE'S THE PROBLEM?

One a scale of one to ten, **50% of our respondents stated the level of difficulty involved in getting a data product in production was more than six.** So we wondered, more than the difficulties they identified, out of all the topics we asked them, which answers were most susceptible to correlate with a higher level of difficult. And so we trained a random forest algorithm to see what we could find.

Unsurprisingly, the most important feature in our model was the statement "We've started our data science efforts recently and are still convincing people in the organization of the use cases."

Respondents who reported doing real-time scoring as well as combining online and offline learning where also much more susceptible to a higher level of difficulty. Companies with cloud or object storage also face more difficulties.

Companies who reported their main difficulty to be "Data quality and pipeline development issues" have a harder time deploying into production.

Finally, two important causes were, firstly, business users accessing only live dashboards to monitor applications and, secondly, the lack of business team involvement ("they just come to us with a problem"). **The implication of business teams in the development and the monitoring of data science applications is as important as technology in terms of getting your projects into production.**

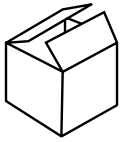
Our Final Recommendations

Considering the results of our survey, here are a few principles to keep in mind on how to build production-ready data science products:

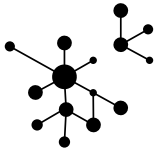
- Getting started is tough. Working with small data on SQL databases does not mean it's going to be easier to deploy into production.
- Multi-language environments are not harder to maintain in production, as long as you have an IT environment consistency process. So mix'n'match!
- Real-time scoring and online machine learning are likely to make your production pie more complex. Think about whether the improvement to your project is worth the hassle.
- Working with business users, both while designing your machine learning project and after when monitoring it day to day, will increase your efficiency. Collaborate!

Checklist

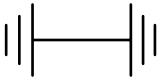
When you set-up your data science production process, you should think about:



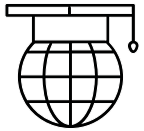
☐ Your packaging strategy



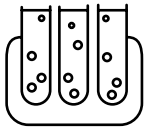
☐ The auditability of your projects and processes



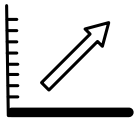
☐ Your model optimization and retraining



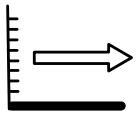
☐ Whether your use cases call for real-time scoring or online machine learning



☐ The involvement of business teams



☐ The scalability of your system and processes



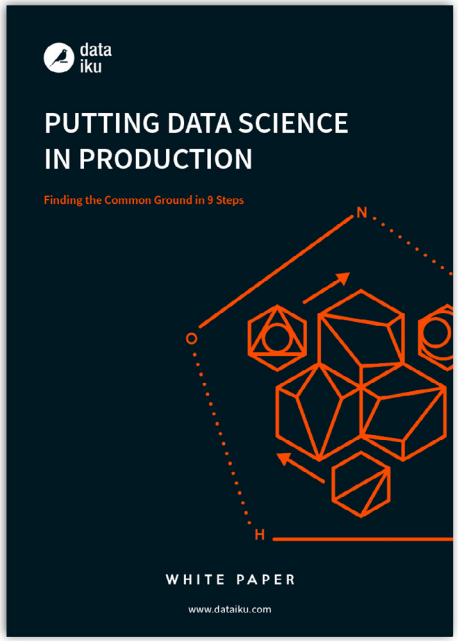
☐ Your IT environment consistency strategy

☐ Your rollback and failover strategies

Next Steps...

Get more detailed advice on how to optimize your data science deployment strategy by reading our Production Guidebook.

http://pages.dataiku.com/development-to-production-guide



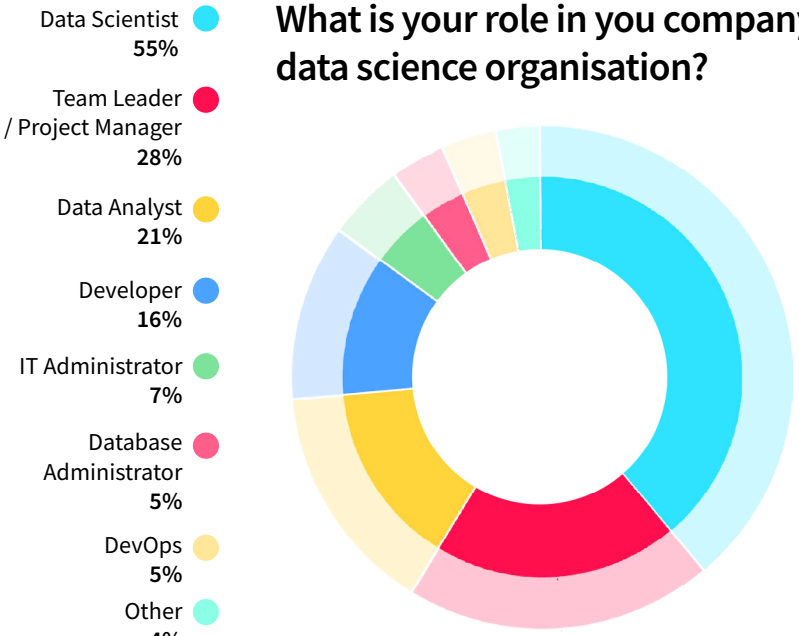
data
iku

If you're ready to explore simple design-to-production integration, and advanced personalized monitoring, explore how Dataiku Data Science Studio can help:

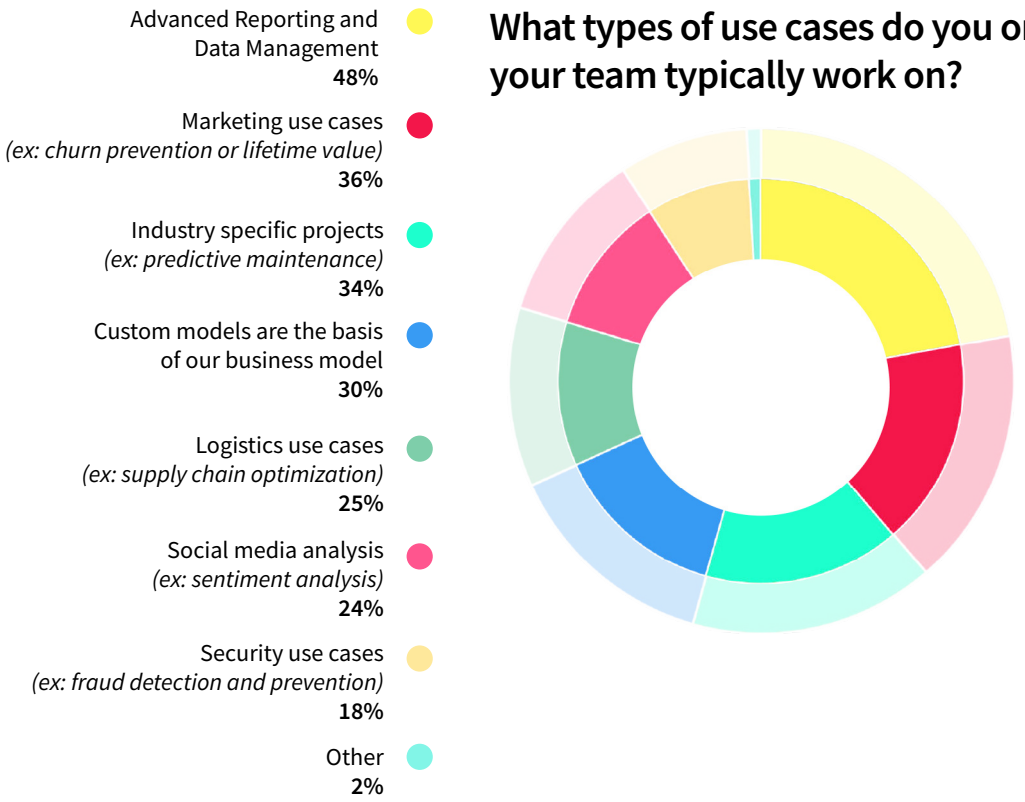
http://www.dataiku.com/dss/features/deployment/

About the Survey

What is your role in you company's data science organisation?



What types of use cases do you or your team typically work on?



We asked several thousand people in companies worldwide of different sizes and in different industries. The majority of respondents to our survey are data scientists (55%), and a third were in managing positions.

The most common reported use case that they work on is advanced reporting and data management, with almost 50% of respondents. The second most common use case were marketing related, with 51% of respondents working on marketing and/or social media analysis.

About the Survey

What is your current type of data storage system?

SQL (open-source or enterprise grade DBMS) **42%**

Hadoop and other distributed storage ecosystems **30%**

NoSQL and document-oriented (including graph or time-series databases) **27%**

Analytics databases (such as Oracle, Exadata, Teradata, Vertica etc) **24%**

Cloud and object storage **24%**

Other **3%**

Almost 60% of respondents reported using either open source SQL or enterprise analytics databases.

Hadoop and distributed systems are the second most popular storage system with 30% of users, and a little fewer have NoSQL databases.