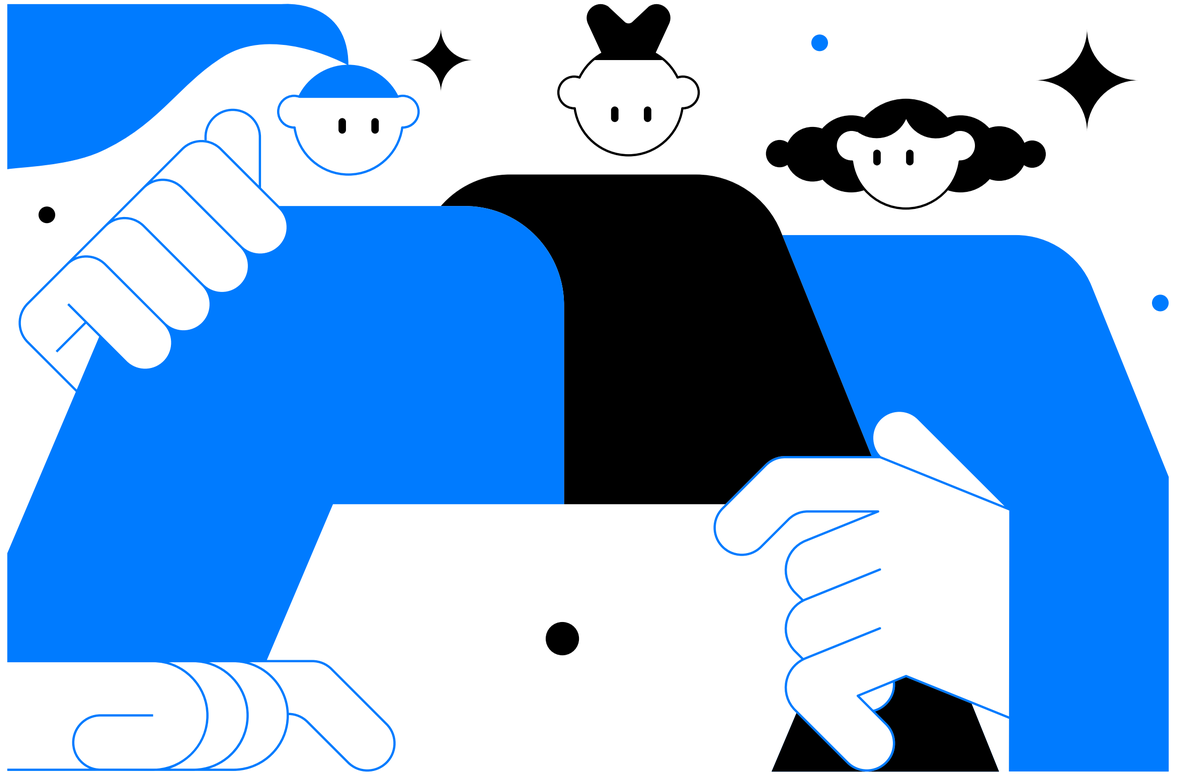


3 Pillars for a Successful Data Team

Who is this whitepaper for?

This whitepaper could be relevant for:

- Data science and data analytics managers in small and mid-size teams
- CTOs in startups
- Senior data specialists who want to pursue the management career track
- Project and product managers in data-driven teams

Abstract

Leading teams that are working at the intersection of science, technology, and innovation can be challenging – data teams are not an exception here. Not so many best practices have been established yet, as the domain is still young, but the growing number of data specialists requires effective management, collaboration practices, and motivation methodologies.

This whitepaper will give you ideas about how to:

- Structure data teams
- Introduce collaboration practices
- Measure your data team's performance

We hope this paper will inspire you on your data team management journey and help you make better decisions from the very beginning.

About JetBrains

JetBrains is a global software vendor specializing in the creation of intelligent, productivity-enhancing tools for software developers and teams.

It maintains its headquarters in Prague, Czech Republic, with offices located in Munich, Berlin, Amsterdam, Shanghai, and Boston.

JetBrains employs more than 1,500 people and is grown organically, with no external funding.

Its product catalog includes award-winning developer tools such as IntelliJ IDEA, ReSharper, PyCharm, and WebStorm, plus a comprehensive set of team tools including Datalore (a collaborative data science platform for teams), TeamCity (for continuous integration and continuous deployment), YouTrack (for issue tracking and project management), and Space (an all-in-one solution for software teams).

Data team structure

Deciding on your team's structure is the foundation for a productive workflow. In this chapter, we'll discuss an organizational framework that could be particularly relevant for data teams of 5–20 people.

As your data team grows, you might start gradually shifting from the startup culture of "everyone doing everything" towards a culture of "everyone doing what they specialize in". This is where a few challenges might arise, such as the following:

- You need to design a team structure that enables collaboration from the very beginning of each project. Otherwise, you might find at the end of the project that data scientists' solutions have been declared not viable production-wise or business-wise and that stakeholders are unhappy with moved deadlines.
- You will need more management for newcomers as the data team grows.
- You need to make sure the team members develop professionally and feel happy and inspired at work. Otherwise, you could see a lot of ash from burned out data specialists and a high turnover rate.

Now we'll introduce a team structure concept which you could use to inspire your team, help them adapt, and tackle challenges like the ones listed above.

Dr. Jodie Burchell

Jodie is a Developer Advocate for Data Science at JetBrains and was previously the Lead Data Scientist in audience generation at Verve Group Europe. After finishing a PhD in psychology and a postdoc in biostatistics, she has worked in a range of data science and machine learning roles across search improvement, recommendation systems, natural language processing, and programmatic advertising.

About the Author ↑

Skill-centric units and project squads

The idea is to split the data team into skill-centric units, e.g. a Data Science unit, a Data Engineering unit, a Business Intelligence unit, etc., and further mix and match people from the units into project squads, always having at least one middle or senior specialist from each skill unit on high priority projects. Further you can reassess each quarter whether the project team is performing well together and rearrange people if needed.

How might this approach work to tackle the challenges above?

First, you **stimulate collaboration from the very beginning of the project and avoid data science decisions with implementation implications.** For example, you can save data scientists' time by deciding together early on that ease of implementation or prediction speed is more important than model accuracy, meaning the data scientists can focus on models such as linear or logistic regression, rather than more precise models such as neural networks. An engineer in a project team can point out that neural nets will require a separate API with a response time of more than five seconds, whereas a linear regression could easily be programmed with almost any programming language.

Tip from Jodie: Don't try to hire a Machine Learning engineer if your team is small. These specialists are extremely expensive and hard to hire, and very soon one ML engineer would not be enough to maintain all the deployed models and you'd end up having to hire more. Instead, try to spread the engineering responsibilities between existing engineering teams, e.g. Backend Engineers or Data Engineers.

Skill-centric units and project squads

Second, **smaller teams tend to be better at self-organization.** Inside skill-centric units, you can establish code review practices and assign mentoring roles to senior and middle specialists.

Inside project teams, you can also choose a senior person who would be responsible for managing project stakeholder relations. Ideally, you would have at least a 1 Senior – 2 Middle – 2 Junior ratio in each skill-centric unit, as more juniors would require much more management and mentoring, and because senior and middle specialists would be less focused on contributing to their projects.

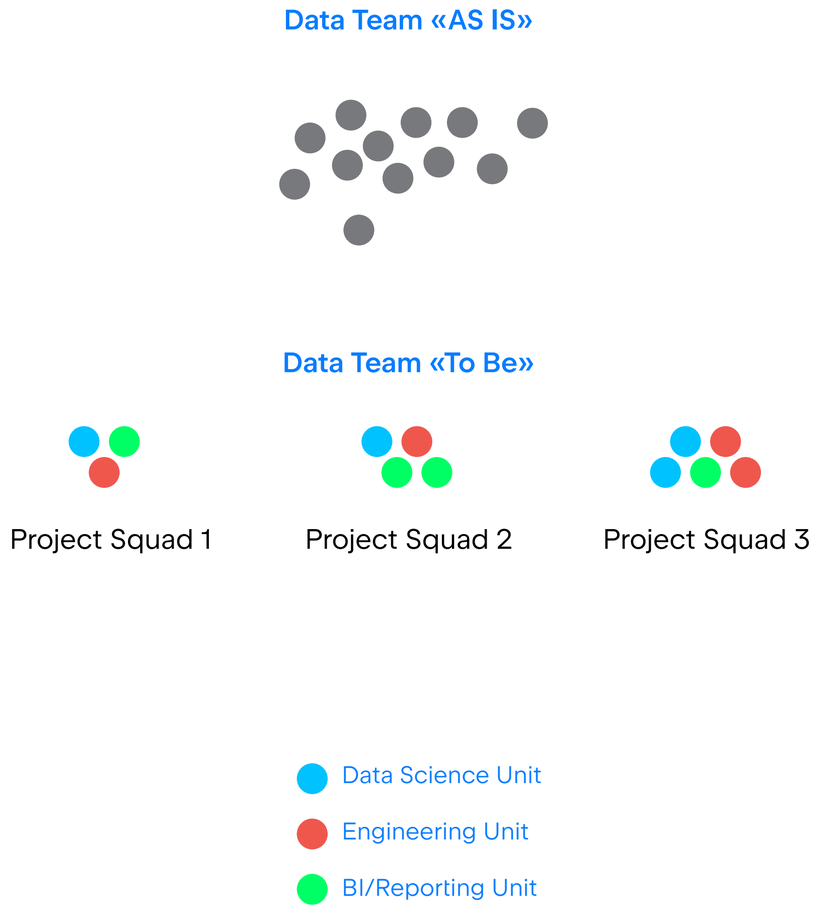
Tip from Jodie: Generally there are two career tracks for data professionals: individual contributor and manager. It is important to be careful when assigning management roles to senior data specialists in growing mid-size teams, as their personality might not fit these responsibilities well. It is preferable to have a non-data scientist manager for the data science team if no data scientists fit this role. However, in that case, you should do your best to make sure the senior data scientists are forming a culture of mentorship for the junior team members, and also make sure your non-DS manager is thinking about how to foster career development for DS team members.

Skill-centric units and project squads

Third, having skill-centric units will help data professionals **feel that they belong to a small professional community** and:

- Stimulate professional development through the transfer of knowledge from more experienced to less experienced specialists.
- Reduce the bus factor, as unit members will regularly discuss their projects together. However, to fully transfer knowledge between team members, at least two people from the same unit should be working on the same project at least part-time.
- Make specialists less stressed, as they will always have people to reach out to in case they need help.
- Foster creativity by talking through their ideas.

After cultivating data team collaboration to happen within your team structure, you will need to choose the right tooling to bring your collaboration workflows to life. We will talk about data science project collaboration in more detail in the next chapter.



Establishing productive collaboration

Data science projects can be complex, consisting of many parts, such as notebooks, data, environments, and scripts, and it can be challenging for data science teams to work together on them effectively.

In this chapter, you will learn five modern collaboration techniques for data-driven teams to improve productivity and lower stress.

Alena Guzharina

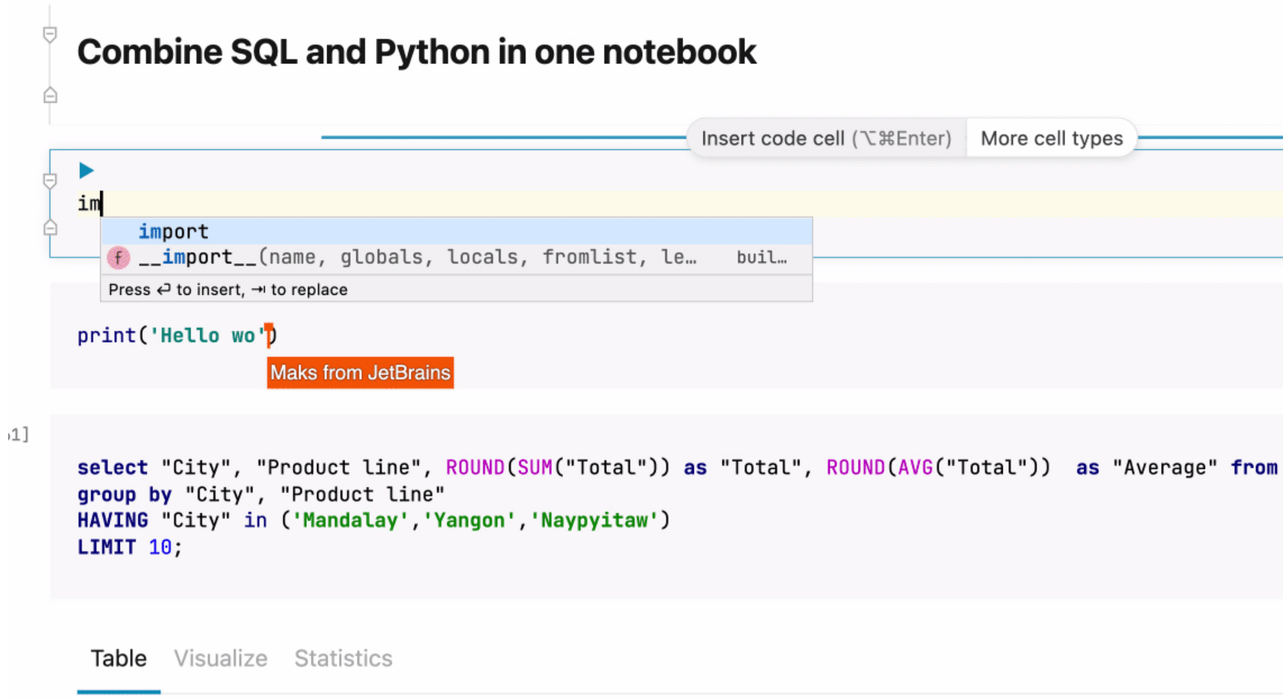
Alena is a Product Marketing Manager for JetBrains Datalore. Alena holds a degree in data science and has been responsible for the product marketing of a data science platform at JetBrains for two years.

About the Author ↑

Share code and essential artifacts easily

Collaboration on data science projects is prone to fail right from the beginning simply because it is so cumbersome to share materials with other team members. Sharing a Jupyter notebook requires you to share a boatload of context as well: the specific environment, data, and data connections. And that seems like overkill when all you need is help with a data transformation. Wouldn't it be great if sharing Jupyter notebooks were as easy as sharing Google Docs?

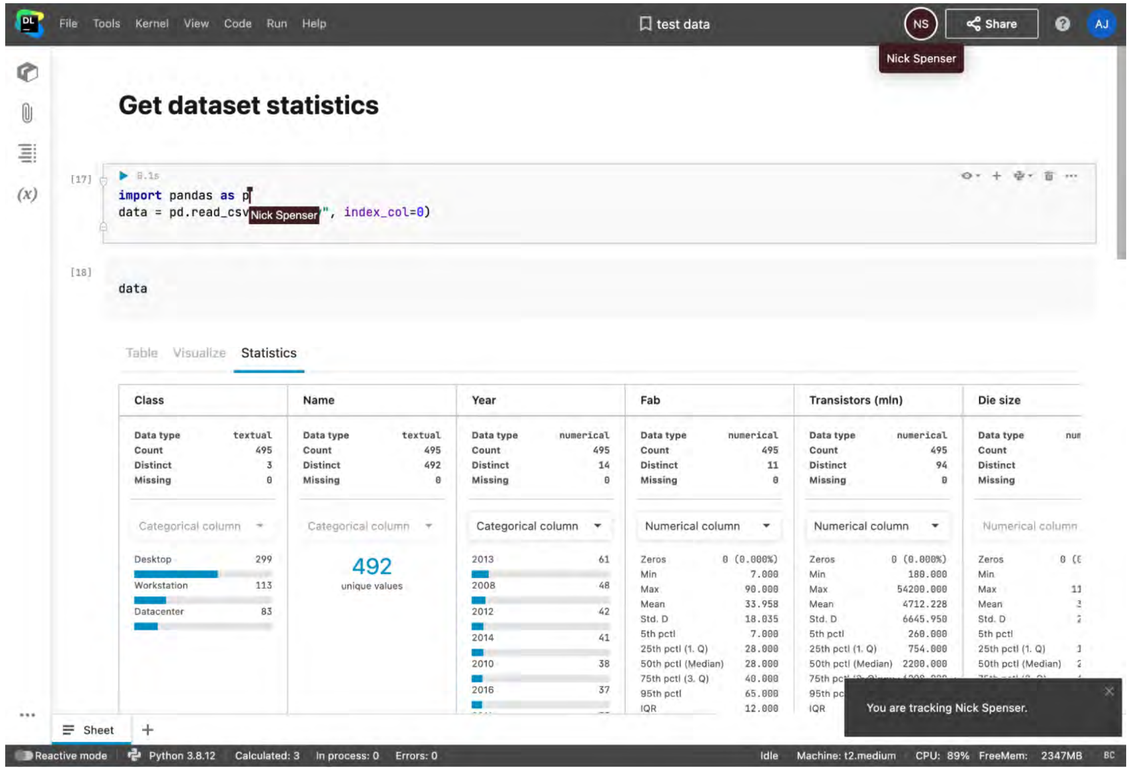
With **Datalore**, you can share notebooks with either view or edit access simply via link or email invitation, and all of the attached data and data integrations, environment, and computation states will be shared automatically. This could be particularly handy when you've been training a machine learning or deep learning model for a long time and want to share your progress in real time.



Datalore is a collaborative data science and BI platform for teams. You can try Datalore Community and Datalore Professional online, hosted by JetBrains, or you can install Datalore Enterprise as a self-hosted solution in your private cloud or on-premises.

Share code and essential artifacts easily

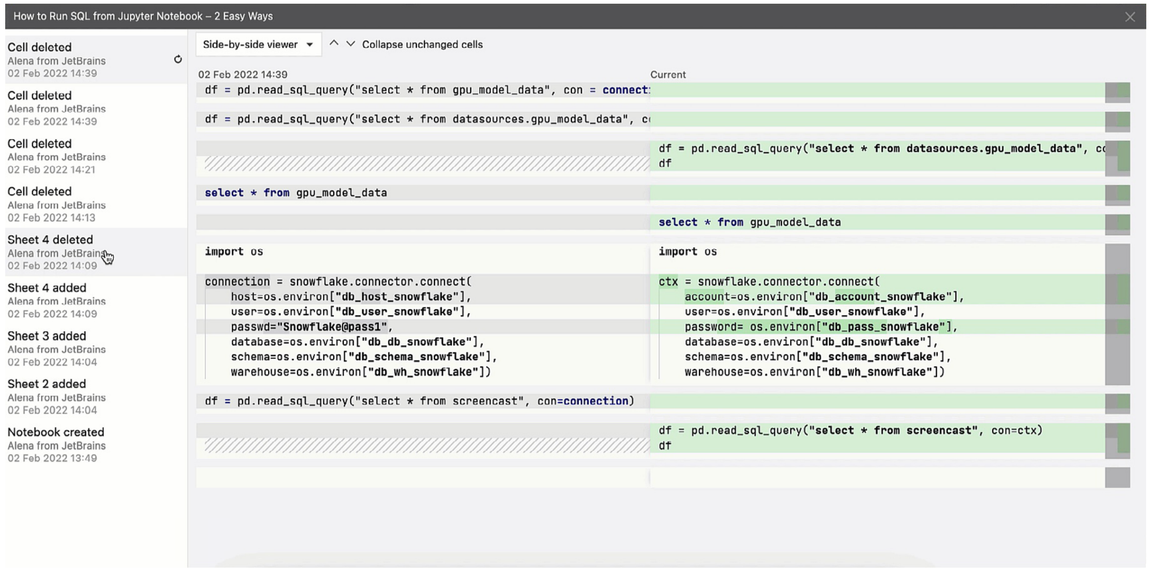
When your colleague enters the notebook, you will see their icon and cursor in real time. By clicking on their icon, you'll be able to start tracking and following along with them automatically. You can collaborate in real-time on Python scripts and data files attached to the notebook as well.



Share code and essential artifacts easily

It is possible to access a shared notebook both in real-time and when the other team member is offline. You don’t need to worry about breaking something in the notebook, as the actions are saved in the History tool, meaning you can always track changes and revert to a custom or automatic checkpoint.

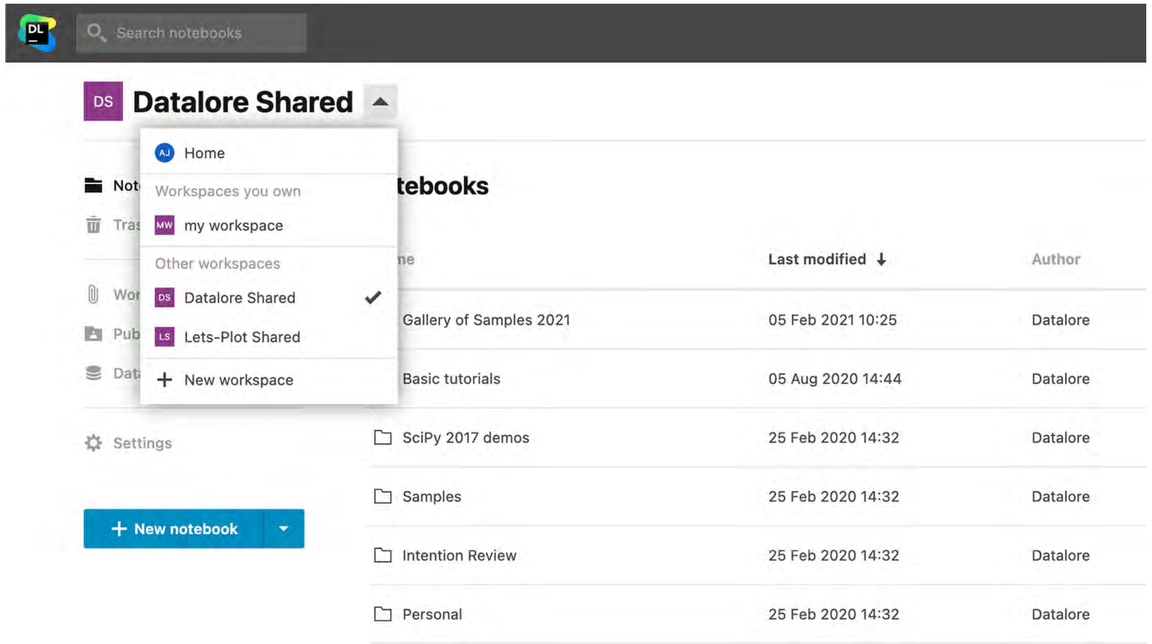
If you prefer to use open-source Jupyter notebooks, you can share them via link and collaborate after installing a Yjs plugin on your server. However, this plugin lacks role permissions and doesn’t have real-time collaborator tracking and version diffs, and your database passwords or other credentials are exposed and can be retrieved by your team members.



Create a knowledge base for data science projects

If your team members often do repetitive tasks, you might think of creating a knowledge base with notebook templates. This is a simple way to prevent a member of your team from wasting hours reinventing a process that another team member has already developed.

In Datalore, you can create a shared team workspace and store all of your essential notebook templates and datasets. Thanks to Datalore's all-in-one setup, these templates can include a configured environment, proper markdown descriptions, some documented template code, and even connections to relevant databases or cloud storage. Data scientists will then be able to clone these notebooks to their home workspaces and start building on them.



Such a knowledge base also streamlines the onboarding of new team members, as all essential datasets, notebooks, and environment setups are available in one place.

Write code with reproducibility in mind

Understanding each other's code can be tough, but resolving your colleagues' bugs is even harder. Below you can find a brief checklist to introduce to your team to help with reproducibility:

- Describe every 2–3 code blocks with a Markdown cell.
- Click "Run all" and make sure the notebook is recomputed with no errors before publishing your work as a report or putting it in a shared workspace. Alternatively you can use Reactive mode in Datalore to make the notebook state consistent. You can read more about it here.
- Share the environment and data along with the notebook. Datalore handles this by default, but if you are using open-source Jupyter, you'll need to do it explicitly.

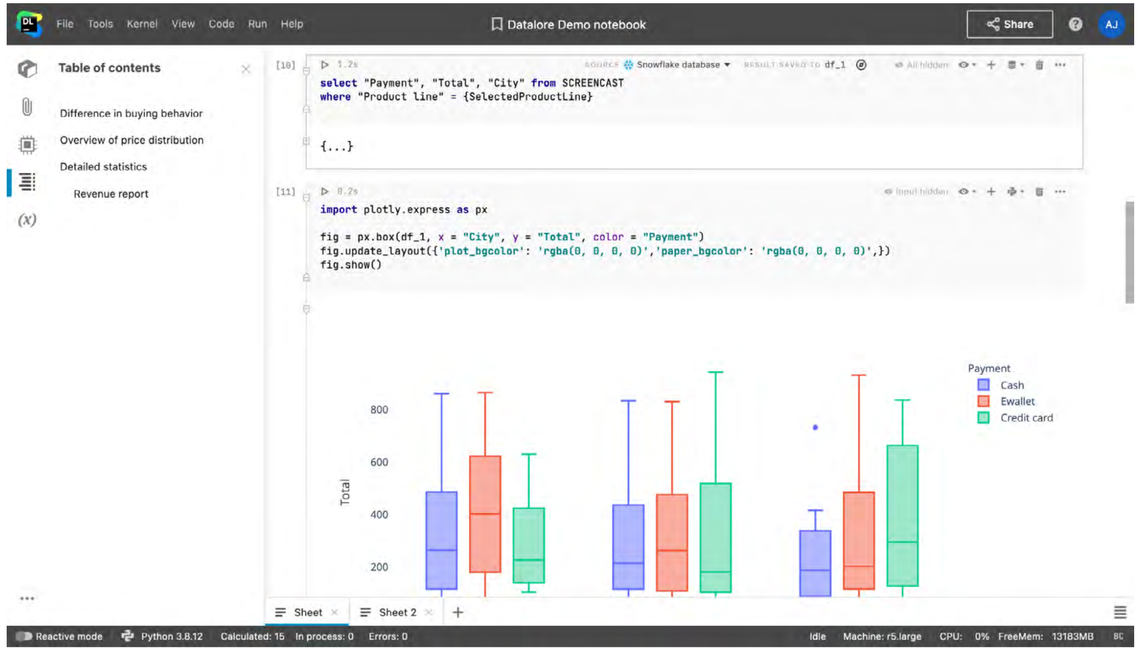
If reproducibility is important to you, make sure to watch our recent webinar with Dr. Jodie Burchell on five tips for reproducible research.

Effectively communicate your findings to foster data-driven decisions company wide

While notebooks are an excellent tool for conducting data science research, they are not the most effective means to communicate the results.

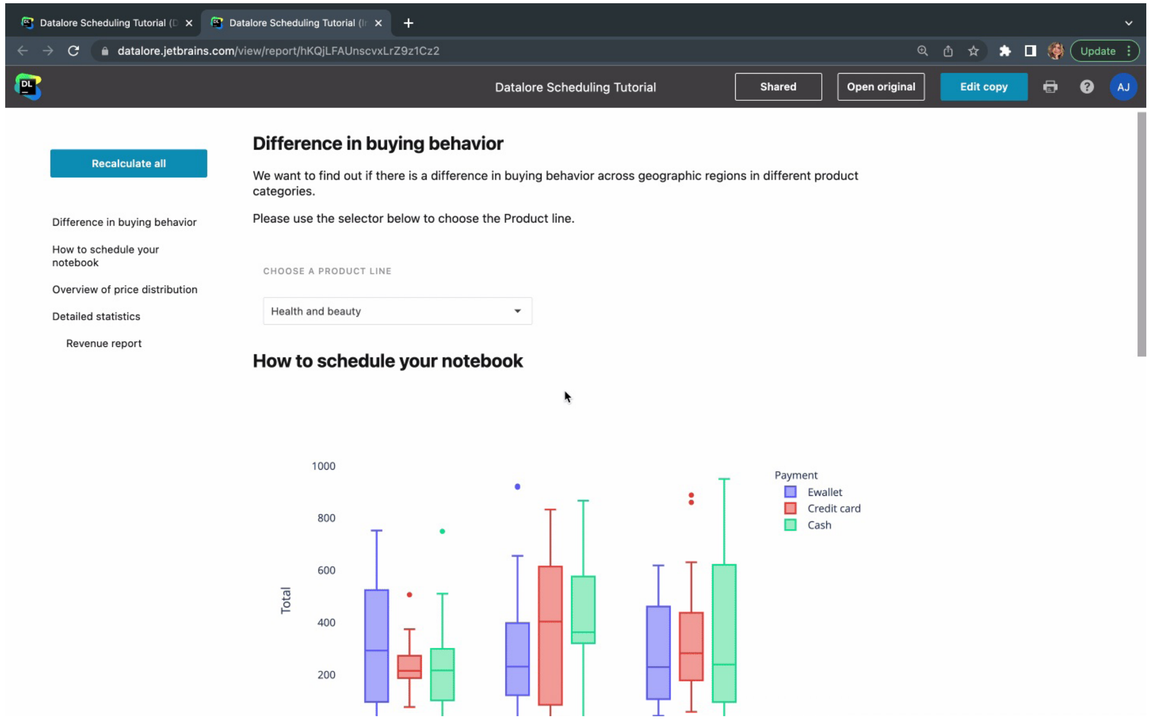
Raw notebooks with large chunks of code are bound to include a lot of information that is irrelevant to both technical and non-technical stakeholders. They usually just want the story of what you did, why you did it, and what your findings were.

However, creating reports using tools like Tableau or Power BI, dashboarding packages like Dash and Streamlit in Python or Shiny in R, or even Google Docs and Microsoft Word, is a lot of extra work. It also removes the connection between the notebook and the report, meaning that any changes you make to the notebook need to be manually updated in the report.



Effectively communicate your findings to foster data-driven decisions company wide

These pain points can be easily addressed with Datalore. Notebooks in Datalore can be converted directly into reports, with the ability to hide as much of the notebook input and output as you would like. Stakeholders can access these reports without a Datalore account and interact with dropdowns, sliders, and plots. If stakeholders have a Datalore account and basic Python or SQL knowledge, they will be able to dive into the analysis by editing a copy of the report.



Take a look at the report example here.

From local to cloud-based

Are you using locally installed Jupyter notebooks? Check out the comparison table below for a few reasons why you should consider moving to a cloud-based data science platform.

	Local Jupyter, installed individually.	**Cloud platform, hosted by your company or a SaaS provider.**
Collaboration	Only via Git. The connection to data and environments can be lost, you might forget to commit the latest state of the project, and pushing notebooks with outputs can bring additional clutter into the Git repo.	Real-time collaboration on notebooks and shared workspaces, with all of the artifacts attached (available in Datalore).
Working with big data	It takes a lot of time to pull big data from the server and your local machine might run out of memory.	You can scale the cloud machine and pull the data without having to rely on internet speed.
New team member onboarding	Every new team member spends time installing Jupyter, configuring the environment, and pulling data by themselves.	One-click access to team projects with everything pre-installed.
Computation machine access	Users need to spin up a machine manually and SSH to it.	Easy-to-run computations on powerful servers with one click.
Environment setup	Each user has different environments which can be hard to manage. A new package might break the whole application and it will be hard to debug.	Teams can create multiple base environments with pre-installed dependencies. The app won't be broken, as each notebook's environment is isolated.

Teams can install Datalore in their private clouds or on-premises with Docker or Kubernetes technology. Docker is perfect for small teams, whereas Kubernetes offers maximum infrastructure scalability. You can use **Datalore** for free in a team of up to 4 people or request your 30-day free trial for 1000 users **here**.

Data science KPIs: How to measure teams' performance

To find the way to measure your team's effectiveness and establish the KPIs, it is important to understand what projects the team is involved in.

For the sake of simplicity, let's separate data teams into a few categories:

- Data science teams working on product features
- Data science teams doing internal research
- Data analytics teams
- Data engineering teams

Of course, this list is not comprehensive, but it can be used to demonstrate four different types of KPI. If you have a different data science team structure, you can easily take ideas from this paper and apply them to your specific case.

Now let's dive into the team-specific KPIs.

Nikita Povarov

Team Lead for the Data Analytics and Machine Learning teams at JetBrains

About the Author ↑

Data Science teams working on product features

This is probably the category where it is easiest to measure whether the team is performing well. The goal of such teams is usually to drive the major product metrics with data science.

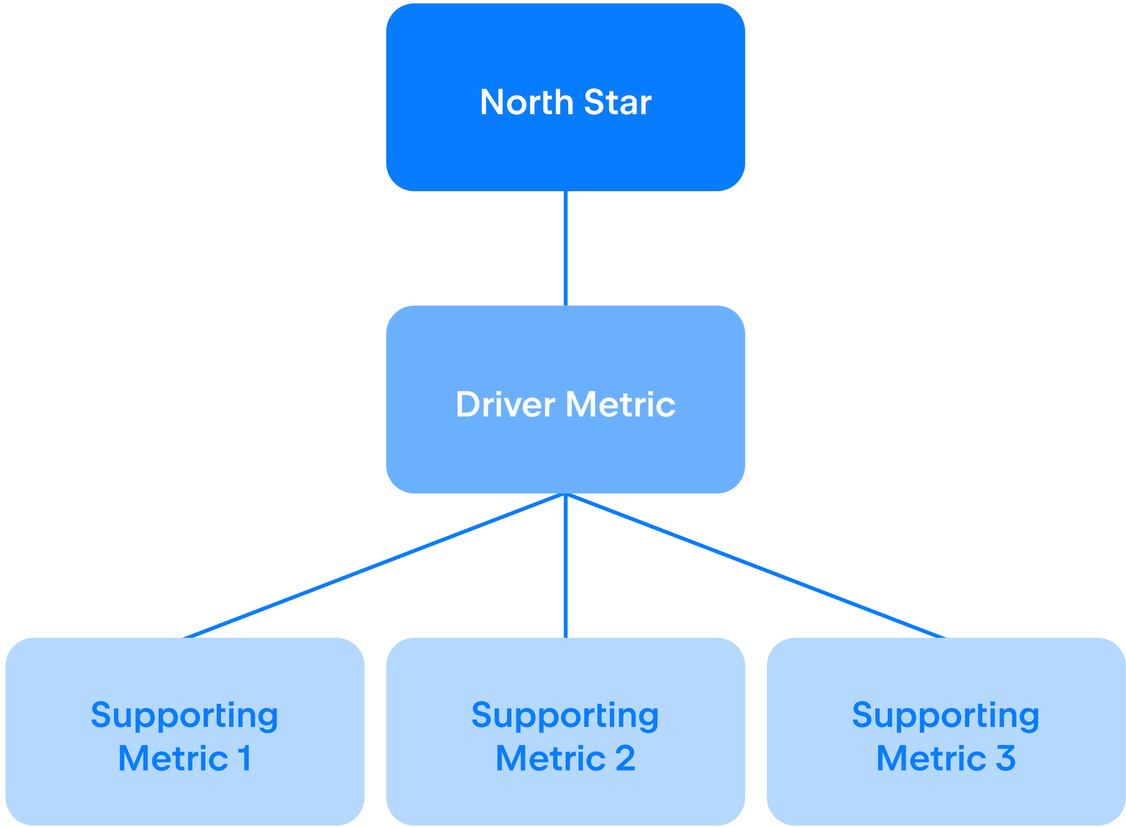
For example, such teams might work on improving app stickiness with high-quality recommender systems or lowering a taxi app's wait times to improve customers' NPS score.

Usually there is more than one product metric, and data science teams can choose what metric to work on. They could decide to drive the major North Star metric of the product, or try to improve particular metrics in the metric hierarchy, which will ultimately influence the North Star metric.

Before choosing a product metric to work on, make sure it satisfies the product metric criteria below:

- The metric shouldn't grow by itself. Or if it does, teams shouldn't use the absolute number but should instead look at the growth rate (for example, year on year or month on month).
- It should be possible to visualize the metric on a timeline.
- It should be possible to see product releases on this timeline.
- There shouldn't be any occasional noise (like explosive growth or drops not connected to product changes).

Teams can conduct AB tests to determine whether their data science model improvements have had an effect on the metric.



Data Science teams doing internal research

These kinds of teams might be involved in digital marketing decisions, conducting predictive analytics and participating in customer retention programs. And a major goal for such teams might be **increasing the utility of the ROC curve.**

Let’s explain this metric with a sample example.

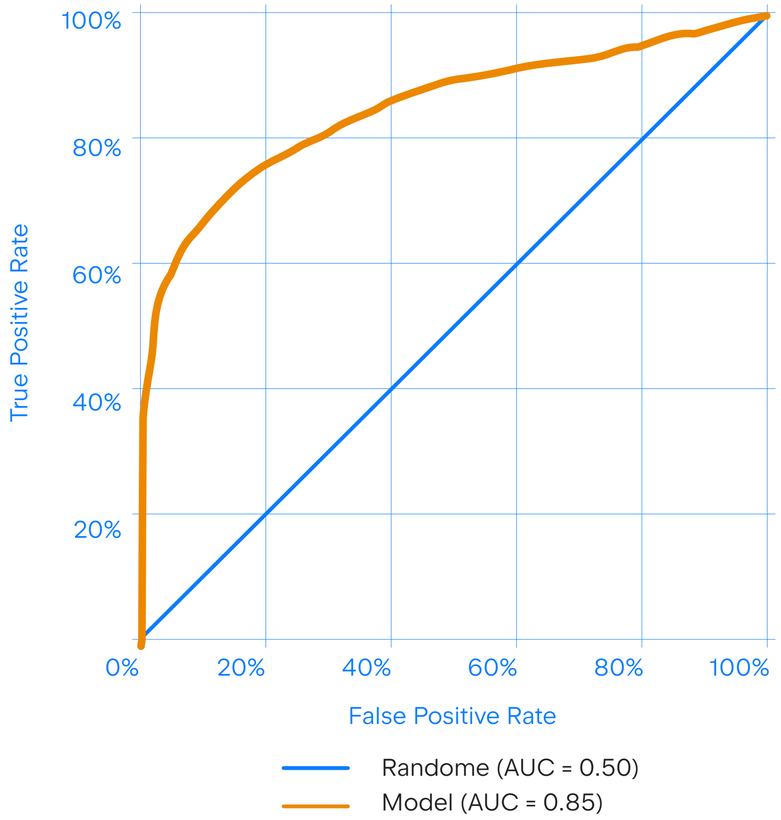
Imagine, you need to figure out whom you should give a 5% discount to in order to increase monthly revenue. If you give the discount to everyone, you will lose the Average Order Value · 5% · the number of people who would purchase without a discount (True Negative). If you don’t give the discount at all, you will lose the Average Order Value · (100%-5%) · the number of people who wouldn’t purchase without a discount (False Negative).

So the data scientist’s task is to analyze user behavior, predict the probability that a discount will stimulate the purchase, and set a threshold, t, above which probability they will send the discount.

First, let’s calculate TPR = TP/(TP+FN) and FPR = FP/(FP+TN), where:

TP = True Positive = discount was sent, and it was the purchase driver.
TN = True Negative = discount was sent, and it was not the purchase driver.
FP = False Positive = discount was not sent, and it was not the purchase driver.
FN = False Negative = discount was not sent, and it could have been the purchase driver.

Second, let’s draw a line of relation between TPR and FPR, also known as the ROC curve.



Data Science teams doing internal research

Third, let’s draw the utility line for the ROC curve.
This is the line that has the same utility at each FPR-TPR point.

Let’s assume that usually 30% of our user base makes a purchase every month.
In this case:

Utility(t)=(95% · AOV·TPR(t)·30%)-(5% · AOV·(1-TPR(t))·30%)-(95·AOV·(1-FPR(t))·70%)

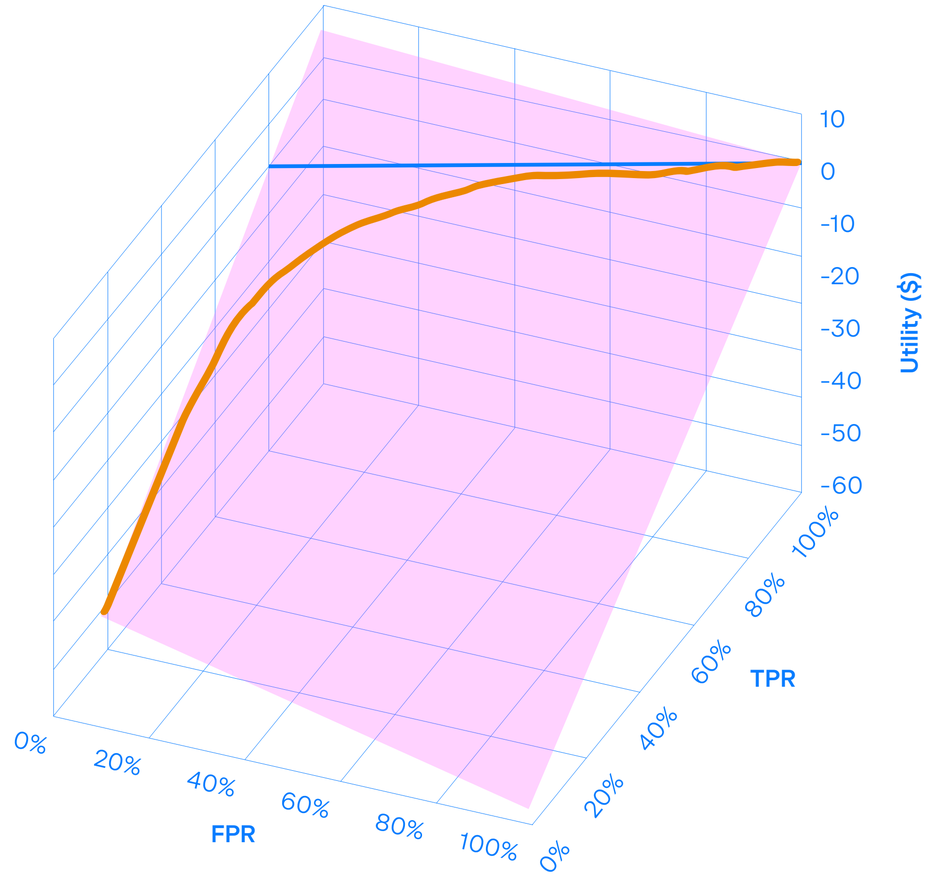
		Actual condition of purchase	
		Purchase happened	**Purchase didn’t happen**
Threshold t	Positive	utility = +95% · AOV rate(t)=TPR(t)·30%	utility = 0 rate(t)= (FPR(t))·70%
	Negative	utility = -5% · AOV rate(t)=(1-TPR(t))·30%	utility = - 95%·AOV rate(t)=(1-FPR(t))·70%

utility(t)=(95% · AOV·TPR(t)·30%)-(5% · AOV·(1-TPR(t))·30%)-(95%·AOV·(1-FPR(t))·70%)

Data Science teams doing internal research

The data scientist’s task is then to make sure that the additional utility of the model is not negative and covers the costs for the project team.

Here is a very good article on this topic that you might want to take a look at.



Sharing data science research is easy with Datalore. Data scientists can hide specific cell inputs and outputs in their notebooks, add interactivity with widgets, and publish results as interactive reports with one click. Stakeholders can further change the controls and get the report results recomputed on the fly simply in the browser.

Data Engineering teams

As these teams are mostly responsible for making the data accessible for other teams, there could be two different methodologies for measuring their performance.

The first methodology is establishing **health check metrics**, such as the number of hours per month the data was accessible and up to date, or the number of hours per month that ETL processes completed without errors. Here you could also assign different traffic light categories:

- Green = The data was fully accessible.
- Yellow = The data was partially accessible.
- Red = The data was inaccessible.

The second methodology is introducing **a 360-degree feedback evaluation system**. This way you can mitigate the risk of a 2d order error, which occurs when the data is being added and the data flow is growing but it is not yielding business results. The 360-degree feedback could involve asking people from other data science teams who work directly with data engineers about the data quality, data accessibility, delivery and reaction speed, etc.

Data analytics teams

If a team is involved in pure data analytics, their major goal could be **increasing the overall data literacy in the company.** At first, this goal sounds quite vague, but let us explain why it is the most important effectiveness metric for this sort of team.

If the data literacy is low, data analytics teams have a lot of extra work to do. They get asked very simple questions like, “what is the 2d month user retention?” or “How many users tried this feature?”

Is completing a lot of tasks a good indicator for a team in this situation? Probably not. However, some data analytics teams try to optimize their workflow using the query theory.

The magic starts happening, when the company’s general data literacy increases and employees know the exact answers to their questions or know where to find them. Then the data analytics team starts getting complex tasks, which in turn helps lead to more conscientious business and product decisions.

How could you measure the data literacy in your company? You could conduct **quarterly surveys and quizzes** for the employees, asking about the most important metrics. This will help you get a sense of whether the company’s analytics team is moving the company in the right direction with their education efforts.

Conclusions

The three pillars described in this whitepaper should help you improve your data team's productivity and become a more confident leader.

Providing your team with a **solid structure** will help you deliver project results faster and further easily maintain them, reduce the bus factor, and foster mentoring and the professional development of your team members. With the introduction of **collaboration practices**, you will encourage knowledge sharing and make it easy for your team members to get help from colleagues whenever they need it. And finally, **identifying relevant KPIs** will motivate your data science teams to deliver the results that matter most to your business.

To support team collaboration and introduce the techniques mentioned in this paper, it is also important to choose the **right tooling**, taking your infrastructure and security requirements into account. Datalore can be installed in the team's private cloud and on-premises, helping you take advantage of your existing computational hardware resources and making your data security engineers happy.

Next steps

To learn more about how Datalore can help your data team become more productive, visit https://www.jetbrains.com/datalore/enterprise/ . You can use Datalore for free in a team of up to 4 people or request your 30-day free trial for 1000 users here.

If you have any questions about Datalore or feedback regarding the recommendations outlined in this whitepaper, please get in touch with us via email at datalore-enterprise@jetbrains.com.