Data Science Life Cycle PDF

The data science life cycle has stages: business understanding, data collection, data preparation, exploration, modelling, evaluation, deployment and monitoring. It is iterative, so findings in later stages often send you back. CRISP-DM is a widely used framework that organises these steps for planning and communication.

Key takeaways

- Projects loop back: poor evaluation results often mean returning to data preparation.
- Business understanding comes first and defines success measures.
- Data preparation is usually the longest stage.
- Deployment is not the end; models drift and need monitoring.
- Each stage should produce a written deliverable that others can review.

What is data science life cycle?

The data science life cycle is the repeatable sequence of stages a project follows, from defining the business question through data work and modelling to deployment, monitoring and improvement.

What are the stages of the life cycle?

Most descriptions agree on eight stages, though names differ.

Stage	Key question	Typical deliverable
1. Business understanding	What decision are we supporting?	Problem statement, success measure
2. Data collection	What data exists and can we use it?	Data inventory, access list
3. Data preparation	Is the data clean and consistent?	Cleaned dataset, data dictionary
4. Exploration	What patterns and issues appear?	Charts, summary notes
5. Modelling	Which method fits the problem?	Baseline and candidate models
6. Evaluation	Does it meet the success measure?	Metrics, error analysis
7. Deployment	How will people use it?	API, report or dashboard
8. Monitoring	Is it still working?	Alerts, retraining plan

What is CRISP-DM?

CRISP-DM is a process model with six phases: business understanding, data understanding, data preparation, modelling, evaluation and deployment. It shows arrows going back, because projects are iterative. Teams use it as a shared vocabulary, even when they adapt the steps to their own needs. It is a guide, not a rule.

How do I define the problem well?

Ask what decision will change if the model works, who will use the output and how success will be measured. Convert the idea into a question with a target and a time frame, such as "predict which customers will cancel within thirty days". Agree on a baseline, such as the current manual method, so improvement can be measured fairly.

Why does data preparation take so long?

Real data has missing values, inconsistent formats, duplicates, unclear definitions and changes over time. You must also join sources and decide how to treat outliers. Documenting each decision in a data dictionary saves time later and helps reviewers trust the work. Skipping this stage causes errors that no algorithm can fix.

How should I model and evaluate?

Start with a simple baseline, then try one or two stronger models and compare them with the same split and metric. Use cross-validation for stability. Study errors by segment to see where the model fails. Check that the metric matches the business cost, for example recall for screening problems. Keep a test set untouched until the final comparison.

What does deployment involve?

Deployment may be a scheduled batch file, a dashboard, a web service or a feature inside an app. Consider speed, cost, security and who responds when it breaks. Save the model, the preprocessing code and the library versions together so results can be reproduced.

Why monitor a model?

Data changes. Customer behaviour, prices and sensors shift, so a model that worked last year may degrade. Track input distributions, prediction quality when labels arrive and business outcomes. Define a retraining trigger and an owner. Monitoring turns a one-off project into a reliable service.

Which pitfalls appear most often?

Skipping the business question, leaking future data into features, ignoring the baseline, over-tuning on the test set, deploying without a monitoring plan and failing to document decisions. Review each stage with a short checklist before moving on.

How do teams document each stage?

Short, regular documents prevent confusion later. Keep a one-page problem brief, a data dictionary that explains every column and its source, an experiment log that records each model, parameters and score, and a model card that states intended use, known limits and fairness checks. Store them beside the code in version control so changes are traceable. When a new person joins, these documents let them understand decisions without long meetings, and they help auditors and managers trust the work.

How does the cycle differ for small and large projects?

In a small project, one person may move through all stages in a few weeks using notebooks and a simple report. In a large project, separate people own data pipelines, modelling, deployment and monitoring, and approvals, security reviews and service agreements add steps. The stages are the same, but the amount of process grows with risk and scale. Match the process to the stakes: a quick analysis needs light notes, while a model that affects customers needs stronger testing and review.

What example shows the whole cycle?

Imagine a subscription business that wants to reduce cancellations. The team defines the goal as flagging customers likely to cancel next month, collects usage and billing data, cleans it, explores which behaviours precede cancellation, builds a baseline and a better classifier, checks recall on recent months, delivers a weekly list to the retention team and tracks whether the flagged group actually cancels less. If drift appears, the team retrains. Each step produced a document that others could review.

Frequently asked questions

What are the stages of the data science life cycle?

They include business understanding, data collection, preparation, exploration, modelling, evaluation, deployment and monitoring, with loops back when results are weak.

What is CRISP-DM?

CRISP-DM is a widely used process model with six phases that helps teams plan and describe data mining and data science projects.

Which stage takes the most time?

Data preparation usually takes the most time because real data is messy, incomplete and inconsistent across sources.

Is the life cycle linear?

No. It is iterative, and findings in evaluation often send the team back to data preparation or even to redefining the problem.

Why monitor models after deployment?

Data and behaviour change over time, so a model can lose accuracy; monitoring detects this early and triggers retraining.

Sources and further reading

- Python documentation
- scikit-learn user guide
- MLflow documentation

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550