Data Science Portfolio Guide PDF

A data science portfolio is a small set of well-presented projects that show how you think. Pick two or three, write a clear README for each, keep notebooks clean and runnable, explain the problem, method, result and limits, and host everything in a tidy repository. Quality and clarity matter more than quantity.

Key takeaways

- Two or three polished projects beat many unfinished ones.
- Each project needs a README with problem, data, method, results and limits.
- Notebooks should run top to bottom and tell a story with headings.
- Never publish private data, secret keys or licensed datasets without permission.
- Show variety: one modelling project, one analysis and, if possible, one deployed demo.

What is data science portfolio?

A data science portfolio is a public collection of projects, code and write-ups that lets employers see how a candidate frames problems, handles data, builds models and communicates results.

What should a portfolio contain?

A good portfolio has a short profile page, two or three featured projects and links to code. Choose projects that differ: a predictive model, an exploratory analysis with strong charts, and, if you can, a small deployed demo. Each project should show a different skill, not the same pattern repeated.

How do I choose which projects to show?

Pick work that you can explain deeply. Prefer projects with a clear business question, a real dataset and a measurable result. Remove tutorial copies and half-finished experiments, or move them to an archive folder. A reviewer will open only a few repositories, so make them count.

What goes into a strong README?

Use this outline:

1. **Title and one-line summary.**
2. **Problem:** the question and why it matters.
3. **Data:** source, size, licence and key columns.
4. **Method:** cleaning, features, models and metric.
5. **Results:** a small table and one chart.

6. **Limits and next steps:** honest weaknesses.
7. **How to run:** environment file and three commands.

How should notebooks be written?

Give the notebook a title and short markdown sections: question, data loading, cleaning, exploration, modelling, results and conclusion. Keep cells small, remove dead code, set random seeds and make sure it runs from the first cell to the last. Clear outputs that contain private information. A reader should understand the story without running anything.

How do I keep the repository tidy?

Use a clear folder structure such as data, notebooks, src and reports. Add a requirements file and a .gitignore. Write meaningful commit messages. Avoid committing large data files; link to the source or use a small sample. Pin your best repositories on your profile.

How do I tell the story well?

Lead with the result, then explain how you got there. For example, begin with "The model identifies about two thirds of customers likely to leave, which would let a team prioritise calls", and then show the steps. Use plain language, and describe a decision you made and why, such as choosing recall over accuracy.

What about deployed demos and blogs?

A small web app or dashboard lets reviewers interact with your model and shows you can package work. A short blog post about one lesson, such as handling leakage, shows communication skill. Keep both simple and make sure they stay online and accurate.

What should I avoid?

Avoid inflated claims, unexplained numbers, huge notebooks with no narrative, copied work without credit and broken links. Do not include personal data. Check licences of datasets and code you reuse and give credit where due.

How do I get feedback?

Ask peers and mentors to open a repository and tell you in two minutes what the project does. If they are unsure, simplify the README. Update the portfolio after each learning milestone.

How do I show teamwork and process?

If you worked in a group, say what you personally did, how you shared tasks and how you resolved disagreements. Show pull requests, issue lists or short design notes that reveal your process. Employers value people who communicate and review work, not only those who write code alone.

Keep a short log of decisions, such as why you dropped a feature or changed a metric, because explaining decisions is a regular interview topic.

How often should I update the portfolio?

Review it every quarter. Remove outdated experiments, fix broken links, update library versions in the requirements file and add one new item that reflects a recent skill. Re-run notebooks to confirm they still work. A portfolio that is current and tidy signals care and reliability, which matter as much to a hiring manager as the models themselves.

What should the profile or about page say?

Write a short paragraph with your focus, the tools you use and the kind of problems you enjoy. Add your contact method and links to your repositories and, if you wish, a short note on what you are learning now. Keep the tone factual and friendly, avoid exaggerated claims, and include a photo only if you are comfortable doing so. A clear about page helps visitors decide quickly whether to read further.

How long should I spend on each project?

Plan two to four weeks for a project, including one week for the write-up. Time spent on a clear README and tidy notebook is rarely wasted.

What is a good first step this week?

Pick your strongest project, rewrite its README using the outline above and ask one friend to review it.

Frequently asked questions

How many projects should a portfolio have?

Two or three polished projects with clear READMEs are better than many unfinished ones, because depth shows real understanding.

Where should I host my portfolio?

GitHub is common for code, and you can add a simple page or blog; make sure links are public, tidy and kept up to date.

What should a project README include?

Include the problem, data source, method, results, limits, and instructions to run the project so reviewers can understand and reproduce it.

Can I use public datasets?

Yes, if the licence allows it. Record the source, avoid personal data and do not upload files you do not have permission to share.

Does a portfolio ensure a job?

No. It helps you show skills, but hiring depends on the role, competition and many other factors beyond the portfolio.

Sources and further reading

- GitHub documentation
- Jupyter 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