📝 How to Write Documentation (Team Guide)

This is your go-to guide for writing great documentation within our ML/AI teams. We organize everything into 3 main sections:

1. **Business Documentation** – for context, goals, and decisions
2. **Technical Documentation** – for architecture, systems, and code
3. **User Manuals** – for guides and usage help

Before you write anything, ask yourself:

> "What is the **purpose** of this information, and **who** is it for?"

1. 📊 Business Documentation

Audience: Product owners, managers, external stakeholders, and non-technical team members
Goal: Explain the **what** and **why** of our work

✅ Include:

- Project goals and KPIs
- Use cases
- High-level architecture overview
- Design decisions and trade-offs (why we chose X over Y)

📌 Writing Tips:

- Remember that you are writing for managers, and they do not have technical background
- Write in plain language (no code or internal jargon)
- Use diagrams to illustrate concepts
- Keep paragraphs short and focused
- Link to relevant technical pages if needed

---

2. 🧱 Technical Documentation

Audience: Engineers, DevOps, data scientists
Goal: Explain **how** the system works and **how to maintain it**

Audience: Data Scientists, Developers, Maintainers, Reviewers
Goal: Explain the internal **code**, **architecture**, and **technical decisions**. Show **how** the system works and **how to**

maintain it

✅ Include:

- Detailed Code Descriptions (and docstrings)
- Code architecture diagrams and internal design
- Model training/inference pipelines
- Data flow diagrams and structure
- Dependency explanations (e.g., HuggingFace, FAISS, etc.)
- Version control setup (Git, DVC, etc.)
- Model registry and data lineage
- APIs and services if applicable
- CI/CD setup if applicable

📌 Writing Tips:

- Be precise and complete
- Always include file paths, commands, and examples
- Document default values, configs, and edge cases
- Keep it up-to-date with the code!

---

3. 👨‍💻 User Manuals

Audience: Internal users, DevOps, Backend/ Frontend devloeprs, QAs, Reviewers **Goal**: Help users **run and use the system** without needing to understand or read the code

This section should **not** include technical internals. It's about giving users the tools and steps to get things done.

✅ Include:

- How to install and set up the environment
- How to run pipelines, scripts, or workflows
- Expected inputs/outputs (e.g., CSV formats, samll table reperenting sample input/output)
- Command-line examples
- Where to find results and logs
- Troubleshooting and FAQ

📌 Tips:

- Be step-by-step and beginner-friendly
- Use real commands that users can copy-paste
- Include example inputs and outputs
- Add screenshots if helpful

---

4. 🧪 Experiments

Audience: Team members running ML experiments, Team Leads **Purpose**: Track ML experiments with results, metrics, and decisions

Each experiment should be documented in a clear, repeatable format.

✅ Include:

- The purpose of experiment
- Dataset & code version
- Version of submmodules (for example which cleansing and translation strategy was used)
- Hyperparameters
- Evaluation metrics
- Observations and conclusions
- Next steps
- reference to MLflow webpage

📌 Tips:

- Use a consistent format across all experiment logs
- Auto-export from MLflow if possible
- Include tables for easy comparison

🧠 **REMEMBER:**
- **Business Docs** = *What & Why*
- **User Manuals** = *How to Use*
- **Technical Docs** = *How It Works*
- **Experiments** = *What We Tried*

Find detailed markdown tutorial here.