Data Science Transition Plan: Astrophysics to Industry (Munich Market)

Target Role: Senior Data Scientist / Research Engineer / Machine Learning Engineer
Background: PhD in Astrophysics + 10 Years Post-doc (HPC & Hydrodynamical Simulations)
Timeline: 2 Years (Part-time preparation)
Location Focus: Munich, Germany (e.g., BMW, Munich Re, Airbus, Siemens, Celonis)

Part 1: High-Level Strategic Roadmap (0–24 Months)

Phase 1: The "De-Academization" & Tooling (Months 1–6)

Goal: Transition from "Researcher Code" to "Production Software."

- **Focus:** Modular programming, SQL databases, Unit Testing, and CI/CD.
- **Outcome:** You stop writing scripts that only run on your laptop. You start building installable, testable packages.

Phase 2: The "Hard" Certificates (Months 7–12)

Goal: Add recognizable industry verification to your academic CV.

- **Focus:** Cloud Computing (AWS or Azure).
- **Action:** Obtain **Azure Data Scientist Associate (DP-100)** (preferred for Munich corporate) or **AWS Certified Solutions Architect (SAA-C03)**.
- **Outcome:** Proof that you understand deployment, scalability, and MLOps.

Phase 3: Strategic Portfolio (Months 13–18)

Goal: Build 2–3 "Hero Projects" bridging Astrophysics and Business.

- **Focus:** End-to-end applications (Data Ingestion -> Model -> API -> Docker).
- **Ideas:**
    1. **Digital Twin:** Surrogate modeling for fluid dynamics (HPC cost reduction).

2. **Predictive Maintenance:** Time-series anomaly detection on sensor data.

- **Outcome:** A GitHub profile that looks like a Software Engineer's, not a student's.

Phase 4: Market Entry & Networking (Months 19–24)

Goal: Networking and Interview Preparation.

- **Focus:** PyData Munich, "Data Geeks" meetups, and translating CV language (e.g., "HPC" instead of "Simulations").
- **Outcome:** Securing interviews at top-tier Munich tech firms.

Part 2: Detailed Weekly Schedule (Months 1–6)

Month 1: Professionalizing Your Python (The "Un-learning" Phase)

Goal: Adopt industry-standard software engineering practices.

- **Week 1: The Modern Stack**
  - **Task:** Setup VS Code, configure Linter (Ruff/Pylint), and Formatter (Black).
  - **Action:** Learn `venv` or `poetry`. Stop using global environments.
  - **Deliverable:** Refactor one old script into three modules: `loader.py`, `calc.py`, `main.py`.
- **Week 2: Version Control (Collaboration)**
  - **Task:** Git beyond `commit/push`.
  - **Action:** Learn Branching (Feature branches -> Merge to Main) and Pull Requests.
  - **Resource:** Udacity Version Control with Git.
- **Week 3: Code Quality & Type Hinting**
  - **Task:** Make code readable and robust.
  - **Action:** Add Type Hints (`def func(x: float) -> float:`) to previous code. Enforce strict linting.
- **Week 4: Advanced Python Features**

  - **Task:** Performance and elegance.
  - **Topics:** Generators (for large data streams), Decorators (for logging/timing), Context Managers (`with`).

Month 2: The Data Engineer's Toolkit (SQL & Wrangling)

Goal: Fluency in SQL is the primary filter for Data Science jobs.

- **Week 5: SQL Basics Refresher**
  - **Topics:** `SELECT`, `GROUP BY`, `HAVING`, `JOINS` (Left vs Inner).
  - **Resource:** Mode Analytics SQL Tutorial.
- **Week 6: Advanced SQL (The Interview Questions)**
  - **Topics:** Window Functions (`RANK`, `LEAD`, `LAG`), Common Table Expressions (CTEs).
  - **Action:** Solve 5 "Medium" SQL problems on LeetCode.
- **Week 7: Pandas for Production**
  - **Task:** Efficient data manipulation.
  - **Topics:** `.groupby()`, `.merge()`, vectorization (avoiding loops).
- **Week 8: Interfacing Databases with Python**
  - **Task:** ETL (Extract, Transform, Load).
  - **Action:** Use `SQLAlchemy` or `psycopg2` to query a local PostgreSQL DB, process in Pandas, and write back.

Month 3: Classical Machine Learning (Scikit-Learn)

Goal: Mapping Physics Math to ML Vocabulary.

- **Week 9: The Scikit-Learn API & Pipelines**
  - **Topics:** `fit()`, `transform()`, `predict()`.
  - **Critical:** Use `sklearn.pipeline` to chain preprocessing and modeling (prevents data leakage).
- **Week 10: Supervised Learning**
  - **Topics:** Linear/Logistic Regression, Random Forests, XGBoost.

  - **Focus:** Understand hyperparameters (depth, learning rate) intuitively.

- **Week 11: Model Evaluation**
  - **Topics:** Precision, Recall, F1-Score, ROC/AUC.
  - **Context:** Understand the business trade-off between False Positives and False Negatives.

- **Week 12: Unsupervised Learning**
  - **Topics:** K-Means, PCA (Principal Component Analysis).
  - **Relevance:** Dimensionality reduction for complex datasets.

Month 4: Software Engineering for Data (Testing & APIs)

Goal: The differentiator. Most applicants lack these skills.

- **Week 13: Unit Testing**
  - **Tool:** `pytest`.
  - **Action:** Write tests for data cleaning functions (check for NaNs, negative values, types).

- **Week 14: Docker & Reproducibility**
  - **Task:** Containerize your environment.
  - **Action:** Write a `Dockerfile` for a Python script. Build and run the image.

- **Week 15: APIs (FastAPI)**
  - **Task:** Expose your model as a service.
  - **Action:** Build a "Hello World" API using FastAPI.

- **Week 16: Review & Refactor**
  - **Action:** Refactor Month 1 code using Tests and Docker.

Month 5: Hero Project 1 — End-to-End Pipeline

Goal: Your first major portfolio piece.

- **Week 17: Scoping & Ingestion**
  - **Action:** Select dataset (e.g., NASA Turbofan). Init Git repo. Write `README.md`. Write ingestion scripts.
- **Week 18: EDA & Modeling**
  - **Action:** Exploratory Notebooks. Then, train XGBoost model via script. Save model with `joblib`.
- **Week 19: Deployment Wrapper**
  - **Action:** Wrap the model in a FastAPI endpoint (Input: JSON data -> Output: Prediction).
- **Week 20: Containerization**
  - **Action:** Dockerize the API. Verify it runs locally. Push to GitHub.

Month 6: Cloud Prep & CI/CD

Goal: Preparing for the Enterprise.

- **Week 21: Cloud Concepts 101**
  - **Topics:** IaaS/PaaS, Storage (S3/Blob), Compute (EC2/VMs).
- **Week 22: Certification Selection**
  - **Action:** Commit to **Azure DP-100** or **AWS SAA-C03**. Purchase study materials.
- **Week 23: Continuous Integration (CI/CD)**
  - **Action:** Set up GitHub Actions to run `pytest` automatically on every push.
- **Week 24: Half-Year Review**
  - **Action:** Update CV with new tech stack. Assess progress.

Part 3: Recommended Resources

Books

- **Clean Code** by Robert Martin (The philosophy of good software).
- **Designing Machine Learning Systems** by Chip Huyen (The reality of MLOps).
- **The Elements of Statistical Learning** by Hastie et al. (The math—which you will find easy, but useful for vocabulary).

Courses & Practice

- **YouTube:** ArjanCodes (Software Design in Python).
- **SQL Practice:** LeetCode (Database Section), Mode Analytics.
- **Cloud:** Microsoft Learn (Azure) or A Cloud Guru (AWS).

Munich Networking

- **Meetups:** PyData Munich, Munich Datageeks.
- **Conferences:** PyCon DE (often held in Berlin but relevant).