ChatGPT

60-Day Data Science Career Roadmap for Beginners

This intensive 9-week plan is organized in five phases: **Foundations → Hands-on Projects → Advanced Tools & Cloud → Interview Prep → Job Sprint**. Each day (4–5 hours of focused learning) includes step-by-step tasks and practice, emphasizing project-based learning and portfolio-building. Core data-science topics (Python, SQL, Statistics, ML, Excel, Tableau/Power BI) are covered first, followed by tools (Git/GitHub, Docker, MLflow, LangChain/prompt engineering, FastAPI, REST APIs) and cloud (focus on Azure, with awareness of AWS/GCP). We also integrate soft skills, resume/portfolio building, LinkedIn optimization and mock interviews. Throughout, use platforms like **Kaggle**, **HackerRank/LeetCode**, **YouTube tutorials**, and **Azure ML Studio** for practice. This plan assumes no prior experience and culminates in a targeted job-application sprint.

Phase 1: Foundations (Days 1–21)

The first three weeks build essential skills. We start with **Python**, the preferred beginner language (easy syntax, powerful libraries) [1], then cover **Git/GitHub basics** for version control, **SQL** for data querying (SQL is "the universal database query language" every data scientist needs [2] [3]), **Excel**, **Power BI/Tableau**, and **Statistics** fundamentals. Each day lists 4–5 hours with specific topics and practice.

- **Day 1:**
- *9:00–10:00:* Python fundamentals – variables, data types (strings, lists, dicts). Python's "simple and intuitive" syntax makes it great for beginners [1].
- *10:00–11:00:* Python control flow – `if`/`else`, loops (`for`, `while`).
- *11:00–12:00:* Practice basic Python exercises on HackerRank (print, loops).
- *1:00–2:00:* **Git/GitHub setup:** Install Git, create GitHub account. Learn `git init`, `git add/commit/push`. Git version control is crucial for collaborating on code [4].
- *2:00–3:00:* Clone a sample repository; write a simple Python script and commit it to GitHub (practice the workflow of coding and versioning).

- **Day 2:**

- *9:00–10:00:* Python data structures – lists, tuples, dictionaries, sets; list comprehensions.
- *10:00–11:00:* Practice problems (e.g. HackerRank/YouTube guided exercises) on data structures.
- *11:00–12:00:* Introduction to **NumPy** arrays (basic operations).
- *1:00–2:00:* **Git practice:** Branching and merging with GitHub. Understand `git clone`, `git branch`, `git merge`. (Use the branch workflow for development.)
- *2:00–3:00:* Python functions – writing reusable functions; practice writing a simple function and testing it.

- **Day 3:**

- *9:00–10:00:* Continue Python: file I/O, exception handling.
- *10:00–11:00:* **Pandas** introduction: loading CSV, basic DataFrame operations.
- *11:00–12:00:* Pandas practice: filter and summarize data in a small dataset (e.g. sales or Titanic data).
- *1:00–2:00:* **SQL fundamentals:** Learn SELECT statements on a sample database. SQL is indispensable – every data scientist "needs to access and retrieve data... and hence... will need SQL" [3] .

- *2:00–3:00:* SQL practice on HackerRank or MySQL Sandbox: SELECT, WHERE, LIMIT on a toy table.

- **Day 4:**

- *9:00–10:00:* Python for data analysis: DataFrame indexing, grouping, merging.
- *10:00–11:00:* Continue Pandas – handling missing data, basic aggregations.
- *11:00–12:00:* **SQL JOINS:** INNER JOIN, LEFT/RIGHT JOIN on two tables.
- *1:00–2:00:* SQL practice: combine multiple tables, GROUP BY and HAVING clauses.

- *2:00–3:00:* Set up a simple **SQLite** or **PostgreSQL** local database and run the above queries on real data (e.g. CSV loaded into tables).

- **Day 5:**

- *9:00–10:00:* **Statistics fundamentals:** descriptive statistics (mean, median, variance) and data distributions. Statistics is the "backbone of data science" [5] , so understanding these is critical.
- *10:00–11:00:* Probability basics: events, random variables, normal/binomial distributions.
- *11:00–12:00:* **SQL functions:** aggregate functions (COUNT, SUM, AVG) in practical queries.
- *1:00–2:00:* **Excel fundamentals:** spreadsheets, formulas (SUM, AVERAGE, pivot tables). Excel is highly accessible; it's a "highly useful tool in data science, especially for its ease of use and accessibility" [6] .

- *2:00–3:00:* Build a simple budget tracker in Excel: use basic formulas and create a chart.

- **Day 6:**

- *9:00–10:00:* Continue Excel – PivotTables and charts (bar/line charts).
- *10:00–11:00:* **Data visualization basics:** theory of charts, best practices.
- *11:00–12:00:* **Tableau introduction:** Connect Tableau to a data file, create first visualizations. (Power BI is similar in concept; we'll cover it soon.)
- *1:00–2:00:* **SQL practice:** More JOINs and subqueries on sample datasets.

- *2:00–3:00:* Quiz yourself on Python and SQL via HackerRank/DataCamp quizzes.

- **Day 7:**

- *9:00–10:00:* **Tableau/Power BI:** Build a simple dashboard in Power BI Desktop (e.g. sales dashboard). Power BI and Tableau are "crucial BI technologies... to collect, integrate, analyze, and present business information" [7] .
- *10:00–11:00:* Continue creating interactive charts in Power BI (slicers, filters).
- *11:00–12:00:* Review: Python and SQL summary; redo any practice areas.

- *1:00–2:00:* **Statistics:** introduction to hypothesis testing (p-values, significance).
- *2:00–3:00:* Practice statistical questions (e.g. on Khan Academy/YouTube) and review Excel/Pivot problems.

- **Day 8:**

- *9:00–10:00:* **Machine Learning intro:** overview of supervised vs unsupervised learning; real-world use cases. (No code yet – conceptual understanding.)
- *10:00–11:00:* **Scikit-learn** basics: splitting data, fitting a simple model (e.g. Linear Regression on a toy dataset).
- *11:00–12:00:* Python practice: implement Linear Regression with scikit-learn; compute predictions and errors.
- *1:00–2:00:* **GitHub portfolio:** Create repositories for your sample projects. Commit your Python, SQL, Excel work. Include README files describing each project (this builds your portfolio).
- *2:00–3:00:* Post on GitHub and write a short description. GitHub is the industry standard for sharing code [4].

- **Day 9:**

- *9:00–10:00:* **Machine Learning:** K-Nearest Neighbors (classification); use scikit-learn to fit and predict.
- *10:00–11:00:* Python practice: apply KNN to Iris dataset in scikit-learn; visualize results in Python (matplotlib).
- *11:00–12:00:* **SQL performance:** Learn about indexing and query optimization (conceptual).
- *1:00–2:00:* **Power BI deep dive:** create a second dashboard (e.g. customer segmentation) to practice.
- *2:00–3:00:* Reflect on learning: update your LinkedIn profile with projects, skills so far. **Soft skill:** Practice explaining one Python project to a friend to improve communication.

- **Day 10:**

- *9:00–10:00:* **Statistics:** regression analysis and correlation (Excel or Python to calculate).
- *10:00–11:00:* **Machine Learning:** Decision Trees (fit a classifier/regressor).
- *11:00–12:00:* Python practice: train and visualize a Decision Tree on a sample dataset.
- *1:00–2:00:* **Azure introduction:** Create a free Azure account. Explore Azure Machine Learning Studio interface. (Azure ML Studio offers a graphical IDE for ML workflows, with drag-and-drop simplicity [8].)
- *2:00–3:00:* Follow an Azure Quickstart to create and deploy a simple model (e.g. linear regression) in Azure ML Studio.

- **Day 11:**

- *9:00–10:00:* **Machine Learning:** Basics of model evaluation (accuracy, confusion matrix).
- *10:00–11:00:* **Kaggle setup:** Sign up on Kaggle. Browse beginner competitions (e.g. Titanic).

- *11:00–12:00:* **Hands-on:** Download Titanic dataset from Kaggle; load it in Python. Start cleaning (fill missing values). Kaggle is an ideal platform for practice.
- *1:00–2:00:* Continue Titanic data prep; write Git commits as you refine code.
- *2:00–3:00:* **Soft skill:** Describe your Titanic project plan (to a peer or mirror) — practice explaining technical work in clear terms [9].

- **Day 12:**

- *9:00–10:00:* **Kaggle Titanic:** Feature engineering in Python (create new columns, encode categories).
- *10:00–11:00:* Train a classification model (e.g. RandomForest) on Titanic data, evaluate accuracy.
- *11:00–12:00:* **Project:** Visualize a key result (e.g. survival vs class) using Power BI or Tableau.
- *1:00–2:00:* **Documentation:** Update your GitHub README to explain the Titanic project (context, approach, results). Good portfolios let employers easily see your skills [10] [11].
- *2:00–3:00:* Continue Azure work: try Azure Automated ML (AutoML) on a small dataset to see how it suggests models.

- **Day 13:**

- *9:00–10:00:* **Machine Learning:** Clustering (e.g. K-Means) – learn concept and apply to a toy dataset.
- *10:00–11:00:* Python practice: use scikit-learn KMeans, plot clusters.
- *11:00–12:00:* **Excel advanced:** Use Solver or Data Analysis Toolpak for simple optimization examples.
- *1:00–2:00:* **Communication skill:** Prepare a one-minute explanation of one of your projects (e.g. Titanic) as if to a non-technical audience [9].
- *2:00–3:00:* **Review:** Go over any weak spots (e.g. retry problems on HackerRank, re-read Python tutorials on YouTube).

- **Day 14:**

- *9:00–10:00:* **Tableau:** Build a final interactive dashboard combining data from multiple sources (e.g. merge two Excel sheets).
- *10:00–11:00:* Polish and publish one of your Power BI dashboards to Power BI Service (online).
- *11:00–12:00:* **Soft skills:** Read about the business context of a dataset you worked on (e.g. Titanic survival factors). Understanding "why" you analyze data builds business acumen (data scientists must communicate findings to stakeholders) [7] [9].
- *1:00–2:00:* **GitHub:** Ensure all code from the past 14 days is committed. Create a professional README to tie your foundation projects together.
- *2:00–3:00:* Plan Phase 2: outline two small projects (e.g. a regression on a Kaggle dataset, a BI report for a case study).

Phase 2: Hands-on Projects (Days 15–28)

In this phase, the emphasis is on **building real projects** to apply the foundations. Continue using **Kaggle, Azure ML Studio, etc.** Each project serves as portfolio material. You will alternate between coding projects (Python/SQL/ML) and business-analytics projects (Excel/Tableau/PowerBI).

- **Day 15:** *Project 1 start (Kaggle Titanic or similar)*
- *9:00–10:30:* Work through a Kaggle "Titanic – Machine Learning" tutorial video to understand the steps.
- *10:45–12:15:* Finish and improve your Titanic model: try a different algorithm or feature.
- *1:00–2:00:* **Share:** Post Titanic notebook/code on GitHub and link it on your LinkedIn.
- *2:00–3:00:* **Practice:** Solve a related Kaggle exercise (e.g. submit your Titanic result on Kaggle to see your score).
- **Day 16:** *Project 1 (cont'd)*
- *9:00–11:00:* Finalize Titanic project; write up key findings (e.g. which features mattered most).
- *11:00–12:00:* **Presentation:** Create a simple 5-minute slide deck or README highlighting the Titanic project.
- *1:00–2:00:* **Quiz:** Review statistics (e.g. do a quick Q&A on regression theory).
- *2:00–3:00:* **LinkedIn:** Post an update about finishing the project (builds online presence).
- **Day 17:** *Project 2 start (BI/Excel case study)*
- *9:00–10:30:* Pick a business dataset (e.g. sales or marketing data). Outline questions to answer (e.g. seasonal trends, KPIs).
- *10:45–12:15:* **Excel/SQL:** Load data; use SQL or Excel to clean and aggregate data.
- *1:00–2:30:* **Power BI/Tableau:** Create a dashboard addressing your questions.
- *2:45–3:00:* Write a summary of insights (as bullet points or mini report).
- **Day 18:** *Project 2 (cont'd)*
- *9:00–11:00:* Finalize BI dashboard; ensure charts are clear.
- *11:00–12:00:* **Publish:** Share the dashboard (e.g. Power BI publish or Tableau Public).
- *1:00–2:00:* **Documentation:** Add this project to your portfolio (GitHub repo with slides or report).
- *2:00–3:00:* **Soft skill:** Practice explaining the dashboard to a friend or mirror.
- **Day 19:** *Project 3 start (Regression analysis)*
- *9:00–10:30:* Take a Kaggle regression dataset (e.g. House Prices). Explore features.
- *10:45–12:15:* Develop a regression model in Python (split data, train, evaluate).
- *1:00–2:00:* **Visualization:** Plot residuals or feature importance using Python charts.
- *2:00–3:00:* **Git:** Commit the regression code to your repo; update README.

- **Day 20:** *Project 3 (cont'd)*

- *9:00–11:00:* **Refinement:** Tune the regression (try polynomial features or another model).
- *11:00–12:00:* Prepare a one-page analysis report (PDF or Markdown) on your findings.
- *1:00–2:00:* **Practice:** Attempt 1–2 SQL/HackerRank problems in Python related to data manipulation.
- *2:00–3:00:* **Review:** Ensure all Phase 2 projects are in your portfolio; ask a peer to review code or give feedback.

- **Day 21:** *Integrated project (combine skills)*

- *9:00–10:30:* Start a capstone mini-project: pick any dataset and do end-to-end analysis (SQL query, Python modeling, and dashboard).
- *10:45–12:15:* Execute the pipeline: write SQL to get data, Python to analyze, Excel/Tableau to present.
- *1:00–2:00:* **Git:** Push this project to GitHub; ensure link is in your resume draft.
- *2:00–3:00:* **Soft skill:** Summarize the project in 2–3 bullet points – practice communicating results clearly.

(After Day 21, pause to consolidate learning. Update your resume and LinkedIn with new projects and skills. Begin practicing problem-solving on HackerRank or LeetCode to prepare for interviews.)

Phase 3: Advanced Tools & Cloud (Days 22–42)

Now cover modern tools and cloud platforms to boost your resume. These sessions integrate hands-on practice.

- **Day 22:** *Git/GitHub (Advanced)*
- *9:00–10:30:* Learn about **Git workflows** – branching strategies, merging, pull requests. Git/GitHub are "tools to overcome" development chaos [12] [4].
- *10:45–12:15:* Practice on GitHub: fork one of your repos, create a feature branch, make a change, open a Pull Request.
- *1:00–2:00:* Review GitHub's interface (issues, projects, wikis) as tools to manage your portfolio.
- *2:00–3:00:* **HackerRank:** Solve a Python coding problem; commit the solution via Git.

- **Day 23:** *Docker (Intro)*

- *9:00–10:30:* Install Docker. Learn containers vs VMs. Docker is key for deployment: "if your app works on your machine it will work on others" via containers [13].
- *10:45–12:15:* Hands-on: Write a simple `Dockerfile` for a Python app (e.g. a Flask app). Build and run the container locally.
- *1:00–2:00:* Document your Docker setup in a repo. Try sharing it with a friend or colleague (push to DockerHub).
- *2:00–3:00:* **Practice:** Follow a Docker tutorial for data science (e.g. building an image with Jupyter and libraries).

- **Day 24:** *Docker (Advanced)*

- *9:00–10:30:* Docker Compose: use it to run multiple containers (e.g. Python app + database).
- *10:45–12:15:* Containerize one of your ML projects: include data and code in a Docker image; test running it.
- *1:00–2:00:* Research "Docker in Data Science" best practices (avoid pushing large data into Git, use volumes) [14] .
- *2:00–3:00:* **Quiz:** Explore how Docker ensures **reproducibility**: recall that Docker "ensures your code runs consistently across different machines and platforms" [15] .

- **Day 25:** *MLflow (Intro)*

- *9:00–10:30:* Install and explore MLflow: an open-source tool to track ML experiments. MLflow "focuses on the full lifecycle" of ML projects [16] .
- *10:45–12:15:* Hands-on: Run an ML experiment (e.g. training model) and log parameters/metrics to MLflow Tracking.
- *1:00–2:00:* View the MLflow UI: see runs, compare models.
- *2:00–3:00:* Modify your code to save the best model in MLflow Model Registry.

- **Day 26:** *MLflow (Advanced)*

- *9:00–10:30:* Practice organizing a small MLOps project: use MLflow Projects to define reproducible runs.
- *10:45–12:15:* Try using MLflow to deploy a model as a REST endpoint (if feasible) or integrate with an MLflow-serving tutorial.
- *1:00–2:00:* **Cloud:** Push an MLflow-tracked project to Azure or AWS (e.g. store tracking data in cloud storage).
- *2:00–3:00:* **Review:** Document an example MLflow log so future you can recall how to use it.

- **Day 27:** *Azure Machine Learning (Basics)*

- *9:00–10:30:* Deep dive Azure ML Studio: create a new workspace, understand compute instances. Azure ML is a cloud service for building/deploying models [8] .
- *10:45–12:15:* **Azure ML Designer (no-code):** Drag-and-drop a simple pipeline (e.g. data prep + ML algorithm + evaluate).
- *1:00–2:00:* **Azure SDK:** Try running a Python MLflow experiment on an Azure compute instance.
- *2:00–3:00:* **HackerRank:** Complete an Azure fundamentals quiz on Microsoft Learn (reinforces cloud concepts).

- **Day 28:** *Azure (Advanced)*

- *9:00–10:30:* Learn about Azure ML **AutoML**: set up an AutoML experiment on a sample dataset.
- *10:45–12:15:* Deploy an Azure ML model: either use **Azure Container Instances** or **Azure Functions** to serve an MLflow model.
- *1:00–2:00:* **GCP/AWS overview:** Briefly review their ML tools (e.g. AWS SageMaker, GCP BigQuery ML). Focus remains on Azure, but mention that familiarity with any cloud is valuable [17] .

- *2:00–3:00:* **Quiz:** Why is cloud important? Recall that cloud skills are now *"as important as programming"* for data scientists [17]. List cloud tasks you completed.

- **Day 29:** *LangChain (Intro)*

- *9:00–10:30:* Learn LangChain basics: LangChain is "an orchestration framework for LLM-driven applications" (e.g. chatbots) [18].
- *10:45–12:15:* Set up a simple LangChain environment in Python. For example, call a GPT-like model via an API.
- *1:00–2:00:* Build a simple LangChain chain (prompt → model → output) to answer questions on a piece of text.

- *2:00–3:00:* **Practice:** Use LangChain to retrieve info from a short document (e.g. Q&A from text).

- **Day 30:** *Prompt Engineering*

- *9:00–10:30:* Study prompt engineering: craft clear prompts for an LLM. Good prompts "unlock the full potential of AI models" [19].
- *10:45–12:15:* Practice with ChatGPT or an open LLM: iteratively refine a prompt to get a better answer.
- *1:00–2:00:* Integrate a prompt into a small LangChain application (e.g. a summarizer or Q&A bot).

- *2:00–3:00:* **Soft Skill:** Reflect on how you phrase queries: clarity and context in prompts are communication skills.

- **Day 31:** *FastAPI*

- *9:00–10:30:* Learn FastAPI basics: FastAPI makes it easy to build web APIs for your models. As JetBrains notes, "FastAPI provides an easy way to convert your data science project into a working application" [20].
- *10:45–12:15:* Build a basic REST API in FastAPI that serves a dummy function (e.g. `/ping`). Test with Swagger UI (auto-generated docs).
- *1:00–2:00:* Extend the API: load one of your ML models (e.g. Titanic model) in FastAPI and set up an endpoint for prediction.

- *2:00–3:00:* **Deployment practice:** Containerize the FastAPI app with Docker and run it. This ties together Docker + FastAPI.

- **Day 32:** *REST APIs and Integration*

- *9:00–10:30:* Learn how to **call** external APIs in Python (e.g. using `requests`). Many data projects require integrating web APIs (e.g. pulling JSON data).
- *10:45–12:15:* Practice: Use a public API (e.g. OpenWeather) in Python; parse and analyze the JSON results.
- *1:00–2:00:* **Project:** Enhance your FastAPI app to accept JSON input and return JSON output (making it a usable service).

- *2:00–3:00:* **Documentation:** Write a brief guide on how to use your API (what endpoints exist, what data it returns).

- **Day 33:** *CI/CD & MLOps*

- *9:00–10:30:* Overview of CI/CD: learn how tools like GitHub Actions can automatically test/deploy models.
- *10:45–12:15:* Set up a simple GitHub Action: on each commit to a repo, automatically run tests or retrain a model.
- *1:00–2:00:* **Project:** Link your GitHub repo to Azure Pipelines or GitHub Actions for automatic Docker builds (if comfortable).

- *2:00–3:00:* Reflect on workflow: version control + containerization + automated tests equals modern MLOps best practice.

- **Day 34:** *Agile Project Management*

- *9:00–10:30:* Study Agile methodology for data projects. Agile (Scrum/Kanban) is "a perfect data science project management method" for iterative work [21].
- *10:45–12:15:* Create a Kanban board (e.g. Trello/Jira) for your projects: backlog, in-progress, done. Move tasks as you complete them.
- *1:00–2:00:* Plan a mock sprint: choose a small deliverable (like a mini project feature) and break it into tasks.
- *2:00–3:00:* **Teamwork:** (If possible) discuss your plan with a peer; adapt based on feedback. Emphasizes communication and collaboration skills [9].

(Days 35–42 overlap some topics or review as needed. By Day 42, you should have strong hands-on experience with all listed tools and cloud basics.)

Phase 4: Interview Preparation (Days 35–49)

With technical foundations in place, shift focus to interview skills, algorithm practice, and soft skills. Continue daily 4–5h but now include mock interviews and resume work.

- **Day 35:** *Data Structures & Algorithms*
- *9:00–10:30:* Solve easy problems on **LeetCode/HackerRank** (arrays, strings). Data scientists need basic DSA for coding rounds.
- *10:45–12:15:* Review Python implementations of common algorithms (sorting, searching).
- *1:00–2:00:* **Mock technical:** Time yourself solving a coding problem.

- *2:00–3:00:* **Soft skill:** Practice explaining your solution aloud.

- **Day 36:** *ML and Statistics Review*

- *9:00–10:30:* Go over key ML concepts: overfitting vs underfitting, cross-validation. Quiz yourself or explain these out loud.
- *10:45–12:15:* Practice writing code to compute confusion matrix and metrics manually.
- *1:00–2:00:* **HackerRank:** Complete a basic SQL challenge (to show SQL knowledge).

- *2:00–3:00:* **Problem:** Identify where you struggled in Phase 1–3; revisit that topic briefly (be it a Python trick, a SQL query, etc.).

- **Day 37:** *Behavioral & Business*

- *9:00–10:30:* Review common behavioral interview questions. Practice telling your "story": why DS, key projects, strengths/weaknesses. Communication is crucial [22].
- *10:45–12:15:* **Case study practice:** Read a short data case (e.g. how to boost sales using data). Outline how you'd approach it.
- *1:00–2:00:* **Mock HR:** Have a friend ask you standard interview questions; respond as you would in an interview.

- *2:00–3:00:* **Soft skill:** Pay attention to listening as well as speaking (a top communication tip [23]).

- **Day 38:** *Quant/Logic Practice*

- *9:00–10:30:* Do a short data-driven case (e.g. calculate ROI from a dataset in Excel). This tests analytical thinking.
- *10:45–12:15:* Time yourself taking a brief aptitude test or brainteaser (many DS interviews include a quant problem).
- *1:00–2:00:* **HackerRank:** Solve a medium-level Python question with time pressure.

- *2:00–3:00:* Review your answers and ensure you can clearly explain your reasoning and code.

- **Day 39:** *Project and Portfolio Prep*

- *9:00–10:30:* Polish your GitHub portfolio: ensure all projects have clear README, comments, and links. A portfolio lets recruiters "see you can do the job" [10].
- *10:45–12:15:* Finalize your one-page resume (emphasize skills: Python, SQL, ML; include GitHub link [11] and mention Azure).
- *1:00–2:00:* Optimize LinkedIn: add profile photo, write a summary focusing on data science projects, list skills (Python, SQL, Azure, etc.).

- *2:00–3:00:* **Mock interview:** Ask a peer or mentor to quiz you on your projects and skills. Practice answering technical questions.

- **Day 40:** *Interview Practice (Technical)*

- *9:00–10:30:* Take a practice technical interview on Pramp or with a friend: include a DS coding question.
- *10:45–12:15:* Work through the solution of that interview problem, noting improvements.
- *1:00–2:00:* **Soft skill:** Practice STAR method for behavior questions (Situation, Task, Action, Result).

- *2:00–3:00:* **Review:** Make flashcards of key algorithms, SQL queries, or ML concepts to quickly review daily.

- **Day 41:** *Applied Skills*

- *9:00–10:30:* Try an end-to-end mini case: from raw data to insight. For example, download a CSV, perform EDA, build a quick model, and summarize findings as if writing an email to a non-technical manager.

- *10:45–12:15:* Focus on one communication skill: try to explain a technical term (like “overfitting” or “regression”) in simple language.
- *1:00–2:00:* **HackerRank:** One last SQL problem and one Python problem to reinforce confidence.
- *2:00–3:00:* **Relax:** Review positive notes (projects done, progress made). Mental prep is important.

(Days 42–49 continue practice, or buffer for catch-up. By the end of Day 49, you should be ready to start actively applying to jobs.)

Phase 5: Job Application Sprint (Days 50–60+)

The final phase is about job-hunting logistics: applying, networking, and final preparation. Keep learning but focus on getting interviews and offers.

- **Day 50:** *Resume & Profile Finalization*
- *9:00–10:30:* Proofread and finalize your resume. Make sure your **GitHub link** and **LinkedIn** URL are prominent [11] .
- *10:45–12:15:* Use job sites (Naukri, LinkedIn, Indeed) to find 5–10 data scientist/analyst jobs matching your skill level. Save descriptions to tailor applications.
- *1:00–2:00:* Update LinkedIn with relevant keywords from job postings (e.g. “Machine Learning”, “Azure”, “Data Visualization”).

- *2:00–3:00:* Reach out to any contacts or mentors; let them know you’re looking and share your portfolio.

- **Day 51:** *Applications and Networking*

- *9:00–11:00:* Draft customized cover letters/email for 3 jobs and submit applications.
- *11:00–12:00:* **LinkedIn:** Engage in one data science group; share a relevant article or comment.
- *1:00–2:00:* Follow up on any pending leads or applications.

- *2:00–3:00:* **Reflection:** List areas for further improvement (e.g. continue learning any weak spots identified).

- **Day 52:** *Interview Prep Review*

- *9:00–10:30:* Do a final mock interview (technical + behavioral) under timed conditions.
- *10:45–12:15:* Analyze the mock interview feedback; improve any weak answers.
- *1:00–2:00:* **Practice:** Review your portfolio; be ready to discuss any project in depth (interviewers often probe your past work [24]).

- *2:00–3:00:* Continue coding practice on LeetCode/HackerRank (maintain algorithmic sharpness).

- **Day 53:** *Follow-up and Continued Learning*

- *9:00–10:30:* Follow up on applications sent (polite inquiry email if appropriate).
- *10:45–12:15:* Prepare for upcoming interviews: review the company’s domain/business, think of potential questions.
- *1:00–2:00:* Watch a YouTube video on interview tips or latest industry trends (stay current).

- *2:00–3:00:* Rest and mentally prepare for interviews; practice relaxation or mindfulness to reduce anxiety.

- **Day 54–60:** *Final Push*

- *Each day (10:00–3:00):* Repeat applications to new job postings, network, and practice interviews. Timebox each day as above. Continue coding and ML problems to stay sharp.
- Keep refining your resume/LinkedIn as you get feedback.
- Engage in one mock interview and one problem-solving session per day.
- Document every application sent and every interview planned (treat it like a project with tasks). Use Agile boards or to-do lists to track tasks (combine technical readiness with effective planning [21]).

By Day 60, you will have a solid portfolio, strong foundational knowledge, hands-on project experience, and polished interview skills. This structured, active timetable ensures you cover **all key domains** while also preparing to *present* your skills to employers. Good luck!

Sources: Guidance on Python's ease for beginners [1]; SQL's ubiquity in data science [2] [3]; the role of statistics [5]; machine learning fundamentals [25]; importance of visualization tools [7]; continued relevance of Excel [6]; the necessity of version control (Git/GitHub) [4]; reproducible environments via Docker [13] [15]; MLflow for the ML lifecycle [16]; the rise of LangChain and prompt engineering for LLMs [18] [19]; FastAPI for deploying models [20]; the criticality of cloud skills [17] [26]; Agile methods for DS projects [21]; and the value of soft skills and portfolios [9] [10] [11].

---

[1] Python for Data Science – Learn the Uses of Python in Data Science | GeeksforGeeks
https://www.geeksforgeeks.org/python-for-data-science/

[2] [3] Why SQL is THE Language to Learn for Data Science - KDnuggets
https://www.kdnuggets.com/why-sql-is-the-language-to-learn-for-data-science

[4] [12] Git for Data Science: What every data scientist should know about Git
https://valohai.com/blog/git-for-data-science/

[5] The Importance of Statistics in Data Science
https://www.linkedin.com/pulse/importance-statistics-datascience-muhammad-dawood

[6] 5 Reasons Data Scientists Should Use Excel
https://www.nobledesktop.com/classes-near-me/blog/why-data-scientists-need-excel

[7] Tableau vs Power BI: Which Is Better Data Visualization Tool!
https://www.simplilearn.com/tutorials/power-bi-tutorial/power-bi-vs-tableau

[8] A Beginner's Guide to Azure Machine Learning | DataCamp
https://www.datacamp.com/tutorial/azure-machine-learning-guide

[9] [22] [23] The Importance of Soft Skills in Data Science Careers | NYIT Data Science
https://online.nyit.edu/blog/the-importance-of-soft-skills-in-data-science-careers

[10] [11] [24] How to create a data science portfolio for job applications – Dataquest
https://www.dataquest.io/blog/career-guide-data-science-projects-portfolio/

[13] [14] Docker for Data Science | GeeksforGeeks
https://www.geeksforgeeks.org/docker-for-data-science/

[15] Docker for Data Scientists: Simplifying Reproducible Environments | by The Data Beast | Medium
https://medium.com/@thedatabeast/docker-for-data-scientists-simplifying-reproducible-environments-1ab07e89257c

[16] MLflow: A Tool for Managing the Machine Learning Lifecycle | MLflow
https://mlflow.org/docs/2.21.3/

[17] [26] Refonte Learning : Cloud Skills for Data Scientists: AWS, Azure, or Google Cloud?
https://www.refontelearning.com/blog/cloud-skills-for-data-scientists-aws-azure-or-google-cloud

[18] What Is LangChain? | IBM
https://www.ibm.com/think/topics/langchain

[19] What is Prompt Engineering? A Detailed Guide For 2025 | DataCamp
https://www.datacamp.com/blog/what-is-prompt-engineering-the-future-of-ai-communication

[20] How to Use FastAPI for Machine Learning | The PyCharm Blog
https://blog.jetbrains.com/pycharm/2024/09/how-to-use-fastapi-for-machine-learning/

[21] 7 Steps to Mastering Data Science Project Management with Agile - KDnuggets
https://www.kdnuggets.com/2023/07/7-steps-mastering-data-science-project-management-agile.html

[25] What is the Role of Machine Learning in Data Science | GeeksforGeeks
https://www.geeksforgeeks.org/role-of-machine-learning-in-data-science/