

90-Day Data Science / Machine Learning Roadmap for Beginners Targeting ₹10 LPA

This roadmap is designed for absolute beginners with no prior experience in programming, data science, or machine learning, aiming to secure entry-level roles like **Junior Data Scientist**, **Machine Learning Engineer**, or **Advanced Data Analyst** in India with a salary of ₹10 LPA within 90 days. Each day provides detailed tasks, explanations, time estimates (6 hours), resources, and tips to ensure you can follow along and build a strong foundation. The plan balances theory, hands-on practice, and project work, culminating in a robust portfolio, interview readiness, and active job applications.

Prerequisites and Setup

- **Hardware:** A laptop with at least 8GB RAM, running Windows, macOS, or Linux.
 - **Software:** Install Python 3.10+, Anaconda (includes Jupyter Notebook), and VS Code (all free).
 - **Accounts:** Create a **GitHub** account for project hosting and a **LinkedIn** account for networking.
 - **Mindset:** Commit to 6 hours daily, stay patient with errors, and engage with online communities (e.g., Stack Overflow, Reddit) for support.
 - **Tools:** Use a notebook or Notion to track progress and errors.
-

Month 1: Python, Math, EDA, and Visualization (Days 1–30)

Focus: Build a foundation in Python programming, essential mathematics, data manipulation, and visualization. By the end of Month 1, you'll complete two EDA projects to showcase your skills.

Week 1: Python Programming Foundations (Days 1–7)

Goal: Master Python basics for data science, including syntax, data structures, and file handling.

- **Day 1: Setup Environment and Python Basics**

- **Tasks** (6 hours):
 - **Install Tools** (1 hour): Download and install Anaconda (includes Python and Jupyter Notebook) and VS Code. Open Jupyter Notebook and run `print("Hello, World!")` to test.
 - **Learn Basics** (3 hours): Study Python syntax, variables (e.g., `x = 5`), data types (`int`, `float`, `str`, `bool`), and `print()`. Watch a beginner tutorial.
 - **Practice** (2 hours): Solve 10 simple problems (e.g., calculate area of a circle, swap two variables without a third variable).
- **Why It Matters:** Python is the primary language for data science, and setting up tools allows you to start coding.
- **Resources:**
 - [Python Official Website](#) for installation.
 - [FreeCodeCamp Python Tutorial](#) (first 1 hour).
 - [W3Schools Python Basics](#).
- **Tips:**
 - Use Jupyter Notebook for interactive coding (type code, see output instantly).
 - Create a GitHub repository named “DataScienceJourney” to save code.
 - Note errors and their fixes in a notebook.
- **Day 2: Control Structures**
 - **Tasks** (6 hours):
 - **Learn Conditionals and Loops** (3 hours): Study `if`, `elif`, `else` for decisions and `for/while` loops for repetition. Understand indentation (Python’s code structure).
 - **Practice** (3 hours): Solve 10 problems (e.g., check if a number is prime, print first 10 Fibonacci numbers).
 - **Why It Matters:** Control structures enable program flow control, essential for data processing.
 - **Resources:**
 - [Automate the Boring Stuff with Python](#) (Chapter 2, free online).
 - [HackerRank Python Challenges](#) (select “Easy” problems).
 - **Tips:**
 - Test code in small chunks to catch errors.
 - Use `print()` to debug loop outputs.
- **Day 3: Functions and Modular Coding**
 - **Tasks** (6 hours):

- **Learn Functions** (3 hours): Study defining functions (`def my_function():`), arguments, return statements, and lambda functions. Explore `map()` and `filter()`.
 - **Practice** (3 hours): Write 5 functions (e.g., calculate BMI, factorial, check palindrome).
- **Why It Matters:** Functions make code reusable and organized, a key coding practice.
- **Resources:**
 - [Real Python: Functions.](#)
 - [LeetCode Python Problems.](#)
- **Tips:**
 - Start with simple functions, then try lambda for concise code.
 - Add comments to explain function purpose.
- **Day 4: Data Structures**
 - **Tasks** (6 hours):
 - **Learn Data Structures** (3 hours): Study lists (ordered, mutable), dictionaries (key-value pairs), sets (unique elements), and tuples (ordered, immutable). Learn methods (e.g., `append()`, `pop()`, `keys()`).
 - **Practice** (3 hours): Solve 10 problems per structure (e.g., remove duplicates from a list, count word frequency with a dictionary).
 - **Why It Matters:** Data structures are critical for data manipulation in data science.
 - **Resources:**
 - [Python Crash Course](#) (Chapters 5–7, or find PDF online).
 - [GeeksforGeeks Python Data Structures.](#)
 - **Tips:**
 - Visualize dictionaries as tables.
 - Practice list comprehensions for efficiency.
- **Day 5: File Handling and Exceptions**
 - **Tasks** (6 hours):
 - **Learn File I/O** (3 hours): Study reading/writing text files (`open()`, `read()`, `write()`) and CSV files with pandas. Use with statements for safe handling.
 - **Learn Exceptions** (1 hour): Study `try`, `except`, `finally` for error handling.
 - **Practice** (2 hours): Read a CSV, handle errors (e.g., `FileNotFoundError`), and write processed data.
 - **Why It Matters:** Data science involves file-based datasets; error handling ensures robust code.
 - **Resources:**

- [Real Python: Reading and Writing Files.](#)
 - [Kaggle Python Course](#) (File I/O section).
- **Tips:**
 - Download a sample CSV from Kaggle.
 - Save error-handling snippets for reference.
- **Day 6: NumPy Fundamentals**
 - **Tasks** (6 hours):
 - **Learn NumPy** (3 hours): Install NumPy (pip install numpy) and study arrays, indexing, slicing, broadcasting, and matrix operations (e.g., dot product).
 - **Practice** (3 hours): Solve 10 problems (e.g., normalize an array, matrix multiplication).
 - **Why It Matters:** NumPy enables efficient numerical computations for large datasets.
 - **Resources:**
 - [NumPy Quickstart Tutorial.](#)
 - [DataCamp NumPy Cheat Sheet.](#)
 - **Tips:**
 - Use vectorized operations instead of loops.
 - Experiment with 2D arrays for matrices.
- **Day 7: Pandas Basics**
 - **Tasks** (6 hours):
 - **Learn Pandas** (3 hours): Install Pandas (pip install pandas) and study Series (1D) and DataFrames (2D). Practice head(), tail(), info(), and filtering (e.g., df[df['column'] > value]).
 - **Practice** (3 hours): Read a CSV, filter rows, and compute column stats (e.g., mean).
 - **Why It Matters:** Pandas is the go-to library for data manipulation.
 - **Resources:**
 - [Pandas Getting Started.](#)
 - [Kaggle Pandas Course.](#)
 - **Tips:**
 - Keep a Pandas cheat sheet handy.
 - Save processed DataFrames as CSV.

Week 2: Advanced Pandas + Statistics (Days 8–14)

Goal: Deepen Pandas skills, learn statistical concepts, and complete an EDA project.

- **Day 8: Advanced Pandas**

- **Tasks** (6 hours):
 - **Learn Advanced Functions** (3 hours): Study `groupby()` (aggregate data), `merge()` (join datasets), `pivot_table()` (summarize), and `apply()` (custom functions). Handle missing data (`isna()`, `fillna()`, `dropna()`).
 - **Practice** (3 hours): Use a dataset (e.g., Iris) to group, merge, and handle missing values.
- **Why It Matters:** Advanced Pandas skills enable complex data wrangling.
- **Resources:**
 - [Pandas User Guide](#).
 - [Towards Data Science: Pandas Tutorials](#).
- **Tips:**
 - Save intermediate DataFrames.
 - Test `merge()` with small datasets.
- **Day 9: Descriptive Statistics**
 - **Tasks** (6 hours):
 - **Learn Stats** (3 hours): Study mean, median, mode, variance, standard deviation, and interquartile range (IQR). Compute with Pandas.
 - **Practice** (3 hours): Compute stats on a dataset and visualize distributions with histograms.
 - **Why It Matters:** Statistics underpin data analysis and ML evaluation.
 - **Resources:**
 - [StatQuest YouTube: Descriptive Stats](#).
 - [Khan Academy Statistics](#).
 - **Tips:**
 - Differentiate population vs. sample stats.
 - Use `df.describe()` for quick insights.
- **Day 10: Probability Basics**
 - **Tasks** (6 hours):
 - **Learn Probability** (3 hours): Study rules (addition, multiplication), normal and binomial distributions, and expected value.
 - **Practice** (3 hours): Solve 5 problems (e.g., coin toss probability) and simulate distributions with SciPy.
 - **Why It Matters:** Probability is foundational for ML algorithms.
 - **Resources:**
 - [Probability for Data Science](#) (or online PDF).
 - [3Blue1Brown Probability Videos](#).
 - **Tips:**
 - Focus on distribution intuition.

- Use `scipy.stats` for simulations.

- **Day 11: Matplotlib Visualization**

- **Tasks** (6 hours):
 - **Learn Matplotlib** (3 hours): Install Matplotlib (`pip install matplotlib`) and create bar, pie, histogram, scatter, and subplots. Customize with labels and colors.
 - **Practice** (3 hours): Create 5 plots on a dataset (e.g., Iris).
- **Why It Matters:** Visualization communicates insights effectively.
- **Resources:**
 - [Matplotlib Tutorials](#).
 - [DataCamp Matplotlib Cheat Sheet](#).
- **Tips:**
 - Save plots with `plt.savefig()`.
 - Use `plt.tight_layout()` for clean layouts.

- **Day 12: Seaborn Visualization**

- **Tasks** (6 hours):
 - **Learn Seaborn** (3 hours): Install Seaborn (`pip install seaborn`) and create `pairplot()`, `boxplot()`, `heatmap()`, and `distplot()`.
 - **Practice** (3 hours): Create 5 visualizations on a dataset.
- **Why It Matters:** Seaborn simplifies professional visualizations.
- **Resources:**
 - [Seaborn Official Docs](#).
 - [Kaggle Visualization Course](#).
- **Tips:**
 - Use Seaborn for quick, aesthetic plots.
 - Combine with Matplotlib for customization.

- **Day 13: Kaggle Dataset for EDA**

- **Tasks** (6 hours):
 - **Select Dataset** (1 hour): Choose a beginner-friendly Kaggle dataset (e.g., Titanic, Iris).
 - **Explore** (3 hours): Load with Pandas, check `info()`, and identify key columns.
 - **Plan EDA** (2 hours): List 5–10 questions (e.g., “What affects Titanic survival?”).
- **Why It Matters:** EDA is the first step in data science projects.
- **Resources:**
 - [Kaggle Datasets](#).
 - [Kaggle Notebooks](#) for EDA examples.

- **Tips:**
 - Pick a dataset with 5–10 columns.
 - Save the dataset locally.
- **Day 14: Complete EDA Project**
 - **Tasks** (6 hours):
 - **Perform EDA** (3 hours): Clean data (handle nulls, outliers), compute stats, create visualizations (e.g., histograms, boxplots).
 - **Summarize** (2 hours): Write insights in Markdown cells.
 - **Publish** (1 hour): Upload Jupyter Notebook to GitHub with a README.
 - **Why It Matters:** This project showcases analysis and visualization skills.
 - **Resources:**
 - [Towards Data Science: EDA Guides.](#)
 - [GitHub Markdown Tutorial.](#)
 - **Tips:**
 - Use comments to explain code.
 - Include 3–4 visualizations.

Week 3: Advanced Python and Data Wrangling (Days 15–21)

Goal: Enhance Python skills, master data manipulation, and explore additional tools.

- **Day 15: Advanced Python Concepts**
 - **Tasks** (6 hours):
 - **List Comprehensions and Generators** (2 hours): Learn list comprehensions (e.g., `[x**2 for x in range(10)]`) and generators for memory efficiency.
 - **Modules and Packages** (2 hours): Create/import custom modules, use external packages (e.g., `requests`).
 - **Practice** (2 hours): Write 5 comprehensions, create a module with 2 functions.
 - **Why It Matters:** Advanced Python improves code efficiency.
 - **Resources:**
 - [Real Python: List Comprehensions.](#)
 - [Automate the Boring Stuff](#) (Chapter 8).
 - **Tips:**
 - Test generators with `next()`.
 - Save modules in GitHub.
- **Day 16: Data Cleaning Techniques**
 - **Tasks** (6 hours):

- **Learn Cleaning** (3 hours): Handle duplicates, inconsistent data (e.g., case sensitivity), and outliers with Pandas.
 - **Practice** (3 hours): Clean a messy dataset (e.g., Kaggle’s “Dirty Datasets”).
- **Why It Matters:** Clean data ensures accurate analysis.
- **Resources:**
 - [Kaggle: Data Cleaning Course](#).
 - [Towards Data Science: Data Cleaning](#).
- **Tips:**
 - Check `df.info()` before cleaning.
 - Save cleaned data as CSV.
- **Day 17: Advanced Visualization with Plotly**
 - **Tasks** (6 hours):
 - **Learn Plotly** (3 hours): Install Plotly (`pip install plotly`), create interactive plots (scatter, bar, heatmap).
 - **Practice** (3 hours): Create 5 interactive visualizations (e.g., Titanic dataset).
 - **Why It Matters:** Interactive plots enhance portfolios.
 - **Resources:**
 - [Plotly Python Docs](#).
 - [DataCamp Plotly Tutorial](#).
 - **Tips:**
 - Export plots as HTML.
 - Compare with Seaborn.
- **Day 18: Introduction to SQL**
 - **Tasks** (6 hours):
 - **Learn SQL Basics** (3 hours): Study SELECT, WHERE, ORDER BY using an online editor (e.g., SQLite).
 - **Practice** (3 hours): Solve 5 SQL problems (e.g., filter rows) on LeetCode.
 - **Why It Matters:** SQL is essential for querying databases.
 - **Resources:**
 - [Mode Analytics SQL Tutorial](#).
 - [LeetCode SQL Problems](#).
 - **Tips:**
 - Use an online SQL editor.
 - Save queries in a text file.
- **Day 19: Intermediate SQL**
 - **Tasks** (6 hours):

- **Learn Joins and Aggregations** (3 hours): Study JOIN (inner, left, right) and functions (COUNT, SUM, AVG).
 - **Practice** (3 hours): Solve 5 problems on StrataScratch.
- **Why It Matters:** Joins and aggregations are common in data tasks.
- **Resources:**
 - [W3Schools SQL Tutorial](#).
 - [StrataScratch SQL Questions](#).
- **Tips:**
 - Visualize joins as Venn diagrams.
 - Test on small datasets.
- **Day 20: Second EDA Project – Planning**
 - **Tasks** (6 hours):
 - **Select Dataset** (1 hour): Choose a Kaggle dataset (e.g., Wine Quality, Heart Disease).
 - **Explore and Plan** (3 hours): Load dataset, check structure, list 5–10 questions.
 - **Setup Notebook** (2 hours): Create a Jupyter Notebook with sections.
 - **Why It Matters:** Reinforces EDA skills for portfolio.
 - **Resources:**
 - [Kaggle Datasets](#).
 - [Kaggle Notebooks](#).
 - **Tips:**
 - Choose a dataset with mixed data types.
 - Outline EDA in Markdown.
- **Day 21: Second EDA Project – Execution**
 - **Tasks** (6 hours):
 - **Clean and Analyze** (3 hours): Clean dataset, compute stats, create visualizations (e.g., boxplots, pairplots).
 - **Summarize** (2 hours): Write insights in Markdown.
 - **Publish** (1 hour): Upload to GitHub with README.
 - **Why It Matters:** Adds depth to your portfolio.
 - **Resources:**
 - [Towards Data Science: EDA](#).
 - [GitHub Markdown Tutorial](#).
 - **Tips:**
 - Explain cleaning decisions.
 - Include 3–4 visualizations.

Week 4: Math for ML and Tools (Days 22–30)

Goal: Learn math for ML and explore tools like Git and Scikit-learn.

- **Day 22: Linear Algebra Basics**

- **Tasks** (6 hours):
 - **Learn Concepts** (3 hours): Study vectors, matrices, dot product, matrix multiplication.
 - **Practice** (3 hours): Use NumPy for 5 operations (e.g., matrix inverse).
- **Why It Matters:** Linear algebra underpins ML algorithms.
- **Resources:**
 - [3Blue1Brown: Linear Algebra.](#)
 - [Khan Academy: Linear Algebra.](#)
- **Tips:**
 - Visualize matrices as tables.
 - Use `np.linalg` for calculations.

- **Day 23: Calculus Basics**

- **Tasks** (6 hours):
 - **Learn Concepts** (3 hours): Study derivatives (rate of change) and integrals (area under curve). Focus on gradient descent.
 - **Practice** (3 hours): Solve 5 simple problems (e.g., derivative of x^2).
- **Why It Matters:** Calculus drives ML optimization.
- **Resources:**
 - [Khan Academy: Calculus.](#)
 - [StatQuest: Gradient Descent.](#)
- **Tips:**
 - Focus on intuition, not proofs.
 - Relate derivatives to slopes.

- **Day 24: Probability Distributions**

- **Tasks** (6 hours):
 - **Learn Distributions** (3 hours): Study normal, binomial, and Poisson distributions.
 - **Practice** (3 hours): Simulate distributions with SciPy, plot with Seaborn.
- **Why It Matters:** Distributions model uncertainty in ML.
- **Resources:**
 - [SciPy Stats Docs.](#)
 - [Towards Data Science: Probability.](#)
- **Tips:**

- Use `seaborn.distplot()` for visualization.
 - Understand distribution parameters.
- **Day 25: Git and GitHub**
 - **Tasks** (6 hours):
 - **Learn Git** (3 hours): Install Git, learn commands (`git init`, `git add`, `git commit`, `git push`).
 - **Practice** (3 hours): Create a repository, commit EDA projects, push to GitHub.
 - **Why It Matters:** Git enables version control and project sharing.
 - **Resources:**
 - [GitHub Guides](#).
 - [FreeCodeCamp: Git Tutorial](#).
 - **Tips:**
 - Write clear commit messages.
 - Organize your GitHub profile.
- **Day 26: Jupyter Notebook Best Practices**
 - **Tasks** (6 hours):
 - **Learn Practices** (2 hours): Study clean notebook writing (Markdown, comments, modular code).
 - **Practice** (4 hours): Refactor Day 14 and Day 21 EDA notebooks.
 - **Why It Matters:** Clean notebooks impress recruiters.
 - **Resources:**
 - [Jupyter Best Practices](#).
 - [Kaggle Notebooks](#).
 - **Tips:**
 - Use Markdown headings.
 - Test notebooks by restarting and running all cells.
- **Day 27: Scikit-learn Introduction**
 - **Tasks** (6 hours):
 - **Learn Scikit-learn** (3 hours): Install scikit-learn (`pip install scikit-learn`), study workflow (`fit()`, `predict()`, `score()`).
 - **Practice** (3 hours): Apply a simple model (e.g., KNN) on Iris dataset.
 - **Why It Matters:** Scikit-learn is the main ML library in Python.
 - **Resources:**
 - [Scikit-learn Getting Started](#).
 - [Kaggle: Intro to ML](#).
 - **Tips:**
 - Use small datasets for quick tests.

- Save model code.
- **Day 28: Data Preprocessing for ML**
 - **Tasks** (6 hours):
 - **Learn Preprocessing** (3 hours): Study scaling (StandardScaler), encoding (OneHotEncoder), and handling imbalanced data.
 - **Practice** (3 hours): Preprocess a dataset with ColumnTransformer.
 - **Why It Matters:** Preprocessing ensures ML model performance.
 - **Resources:**
 - [Scikit-learn Preprocessing](#).
 - [Towards Data Science: Preprocessing](#).
 - **Tips:**
 - Create a preprocessing pipeline.
 - Test on small data first.
- **Day 29: Mini Project – Data Cleaning and Visualization**
 - **Tasks** (6 hours):
 - **Select Dataset** (1 hour): Choose a dataset with issues (e.g., Kaggle’s “Credit Card Fraud”).
 - **Clean and Visualize** (4 hours): Clean data, create 3–4 visualizations (e.g., boxplot, heatmap).
 - **Publish** (1 hour): Document in a notebook, upload to GitHub.
 - **Why It Matters:** Reinforces cleaning and visualization skills.
 - **Resources:**
 - [Kaggle Datasets](#).
 - [Seaborn Visualization](#).
 - **Tips:**
 - Explain cleaning steps in Markdown.
 - Save visualizations as PNGs.
- **Day 30: Review and Consolidation**
 - **Tasks** (6 hours):
 - **Review** (3 hours): Revisit Python, Pandas, NumPy, visualization, and math.
 - **Practice** (3 hours): Solve 5 mixed problems (e.g., clean data, plot, write function).
 - **Why It Matters:** Ensures retention of Month 1 skills.
 - **Resources:**
 - [HackerRank Python](#).
 - [Kaggle Python Course](#).
 - **Tips:**
 - Create a cheat sheet of key functions.

- Update GitHub with revised code.
-

Month 2: Core Machine Learning, Projects, and Evaluation (Days 31–60)

Focus: Learn ML algorithms, evaluation metrics, and complete two mini-projects for your portfolio.

Week 5: ML Introduction & Regression (Days 31–37)

Goal: Understand ML basics and master regression.

- **Day 31: Machine Learning Basics**

- **Tasks** (6 hours):
 - **Learn Concepts** (3 hours): Study supervised vs. unsupervised learning, train-test split, overfitting, underfitting.
 - **Practice** (3 hours): Split a dataset with `train_test_split()`, explore structure.
- **Why It Matters:** Foundational for ML workflows.
- **Resources:**
 - [Scikit-learn User Guide](#).
 - [Coursera: ML by Andrew Ng](#) (Week 1, free audit).
- **Tips:**
 - Understand train-test split's role.
 - Use `random_state=42` for reproducibility.

- **Day 32: Linear Regression**

- **Tasks** (6 hours):
 - **Learn Linear Regression** (3 hours): Study predicting continuous variables, scikit-learn implementation.
 - **Practice** (3 hours): Implement on Boston Housing dataset, visualize predictions vs. actual.
- **Why It Matters:** Core ML algorithm.
- **Resources:**
 - [StatQuest: Linear Regression](#).
 - [Kaggle: Regression](#).
- **Tips:**
 - Standardize features with `StandardScaler`.
 - Plot residuals.

- **Day 33: Regression Metrics**

- **Tasks** (6 hours):
 - **Learn Metrics** (3 hours): Study MAE, MSE, RMSE, R^2 .
 - **Practice** (3 hours): Compute metrics for a regression model.
- **Why It Matters:** Metrics evaluate model performance.
- **Resources:**
 - [Real Python: Metrics](#).
 - [Scikit-learn Metrics](#).
- **Tips:**
 - Know when to use each metric.
 - Save calculations in a notebook.

- **Day 34: Polynomial and Regularized Regression**

- **Tasks** (6 hours):
 - **Learn Polynomial Regression** (2 hours): Study `PolynomialFeatures` for non-linear data.
 - **Learn Regularization** (2 hours): Study Lasso (L1), Ridge (L2).
 - **Practice** (2 hours): Implement both, compare results.
- **Why It Matters:** Handles complex data and overfitting.
- **Resources:**
 - [Scikit-learn: Polynomial Regression](#).
 - [StatQuest: Regularization](#).
- **Tips:**
 - Use Pipeline for preprocessing and modeling.
 - Try polynomial degrees 2–3.

- **Day 35: Mini Project – Housing Price Prediction**

- **Tasks** (6 hours):
 - **Setup** (1 hour): Select Kaggle's House Prices dataset.
 - **Preprocess and Model** (4 hours): Clean data, encode variables, train linear regression.
 - **Evaluate** (1 hour): Compute metrics, visualize predictions.
- **Why It Matters:** Showcases regression skills.
- **Resources:**
 - [Kaggle: House Prices](#).
 - [Towards Data Science: Regression](#).
- **Tips:**
 - Save preprocessing pipeline.
 - Document assumptions.

- **Day 36: Enhance Housing Project**

- **Tasks** (6 hours):
 - **Visualize** (3 hours): Add scatter plots, residual plots, feature importance.
 - **Document** (3 hours): Add comments, Markdown summary.
- **Why It Matters:** Makes project professional.
- **Resources:**
 - [Seaborn](#).
 - [Jupyter Best Practices](#).
- **Tips:**
 - Use Markdown headings.
 - Ensure visualizations are clear.
- **Day 37: Finalize Regression Project**
 - **Tasks** (6 hours):
 - **Clean Up** (3 hours): Remove unused code, organize notebook.
 - **Publish** (3 hours): Create GitHub repository, write README, upload.
 - **Why It Matters:** Polished project for portfolio.
 - **Resources:**
 - [GitHub README Templates](#).
 - [Kaggle Notebooks](#).
 - **Tips:**
 - Test notebook for errors.
 - Add project screenshot to README.

Week 6: Classification Techniques (Days 38–44)

Goal: Master classification and complete a text project.

- **Day 38: Logistic Regression**
 - **Tasks** (6 hours):
 - **Learn Logistic Regression** (3 hours): Study binary classification, sigmoid, scikit-learn.
 - **Practice** (3 hours): Implement on Iris dataset, visualize boundaries.
 - **Why It Matters:** Core classification algorithm.
 - **Resources:**
 - [StatQuest: Logistic Regression](#).
 - [Scikit-learn: Logistic Regression](#).
 - **Tips:**
 - Understand log-odds.
 - Use `predict_proba()` for probabilities.
- **Day 39: K-Nearest Neighbors (KNN)**

- **Tasks** (6 hours):
 - **Learn KNN** (3 hours): Study distance metrics, K tuning.
 - **Practice** (3 hours): Implement KNN, experiment with K values.
- **Why It Matters:** Simple, effective classifier.
- **Resources:**
 - [Real Python: KNN](#).
 - [Kaggle: Classification](#).
- **Tips:**
 - Scale features for KNN.
 - Plot accuracy vs. K.
- **Day 40: Decision Trees**
 - **Tasks** (6 hours):
 - **Learn Decision Trees** (3 hours): Study entropy, Gini, visualization.
 - **Practice** (3 hours): Implement, visualize with `plot_tree()`.
 - **Why It Matters:** Interpretable and widely used.
 - **Resources:**
 - [StatQuest: Decision Trees](#).
 - [Scikit-learn: Decision Trees](#).
 - **Tips:**
 - Limit tree depth.
 - Visualize trees.
- **Day 41: Random Forests**
 - **Tasks** (6 hours):
 - **Learn Random Forests** (3 hours): Study ensemble, bagging, feature importance.
 - **Practice** (3 hours): Implement, analyze feature importance.
 - **Why It Matters:** Improves accuracy over trees.
 - **Resources:**
 - [StatQuest: Random Forests](#).
 - [Scikit-learn: Random Forests](#).
 - **Tips:**
 - Use `n_estimators=100`.
 - Plot feature importance.
- **Day 42: Classification Metrics**
 - **Tasks** (6 hours):
 - **Learn Metrics** (3 hours): Study accuracy, precision, recall, F1-score, confusion matrix.
 - **Practice** (3 hours): Compute metrics, visualize confusion matrix.

- **Why It Matters:** Evaluates classifier performance.
- **Resources:**
 - [Real Python: Metrics.](#)
 - [Scikit-learn Metrics.](#)
- **Tips:**
 - Focus on F1 for imbalanced data.
 - Use `seaborn.heatmap()` for confusion matrix.
- **Day 43: Mini Project – Spam/Ham Detection**
 - **Tasks** (6 hours):
 - **Setup** (1 hour): Select Kaggle’s SMS Spam Collection.
 - **Preprocess and Model** (4 hours): Use `TfidfVectorizer`, train logistic regression.
 - **Evaluate** (1 hour): Compute metrics, visualize results.
 - **Why It Matters:** Introduces NLP and classification.
 - **Resources:**
 - [Kaggle: SMS Spam.](#)
 - [Towards Data Science: NLP.](#)
 - **Tips:**
 - Use `stop_words='english'` in TF-IDF.
 - Test with custom text.
- **Day 44: Finalize Text Project**
 - **Tasks** (6 hours):
 - **Enhance** (3 hours): Add visualizations (confusion matrix, word cloud).
 - **Publish** (3 hours): Document, clean notebook, upload to GitHub with README.
 - **Why It Matters:** Showcases NLP skills.
 - **Resources:**
 - [WordCloud Library.](#)
 - [GitHub Templates.](#)
 - **Tips:**
 - Discuss model limitations.
 - Share on LinkedIn.

Week 7: Unsupervised Learning + Portfolio (Days 45–51)

Goal: Learn unsupervised learning, build portfolio.

- **Day 45: K-Means Clustering**
 - **Tasks** (6 hours):

- **Learn K-Means** (3 hours): Study algorithm, elbow method, silhouette score.
 - **Practice** (3 hours): Implement on Iris dataset, visualize clusters.
- **Why It Matters:** Clusters unlabeled data.
- **Resources:**
 - [StatQuest: K-Means](#).
 - [Scikit-learn: Clustering](#).
- **Tips:**
 - Try multiple K values.
 - Scale features.
- **Day 46: Hierarchical Clustering**
 - **Tasks** (6 hours):
 - **Learn Hierarchical** (3 hours): Study linkage methods, dendrograms.
 - **Practice** (3 hours): Implement, visualize dendrogram.
 - **Why It Matters:** Provides tree-based clustering.
 - **Resources:**
 - [Real Python: Hierarchical](#).
 - [Scikit-learn: Hierarchical](#).
 - **Tips:**
 - Use scipy for dendrograms.
 - Compare with K-Means.
- **Day 47: Principal Component Analysis (PCA)**
 - **Tasks** (6 hours):
 - **Learn PCA** (3 hours): Study dimensionality reduction, explained variance.
 - **Practice** (3 hours): Implement on a high-dimensional dataset, plot variance.
 - **Why It Matters:** Simplifies data for modeling.
 - **Resources:**
 - [StatQuest: PCA](#).
 - [Scikit-learn: PCA](#).
 - **Tips:**
 - Standardize data.
 - Plot first two components.
- **Day 48: Project – Customer Segmentation**
 - **Tasks** (6 hours):
 - **Setup** (1 hour): Select Kaggle's Mall Customers dataset.
 - **Model and Visualize** (4 hours): Apply K-Means and PCA, visualize 2D clusters.

- **Document** (1 hour): Write insights in notebook.
- **Why It Matters:** Showcases unsupervised learning.
- **Resources:**
 - [Kaggle: Customer Segmentation.](#)
 - [Towards Data Science: Clustering.](#)
- **Tips:**
 - Interpret clusters for business.
 - Save visualizations.
- **Day 49: Model Persistence**
 - **Tasks** (6 hours):
 - **Learn Persistence** (2 hours): Study pickle and joblib for saving/loading models.
 - **Practice** (4 hours): Save/load a model from a project.
 - **Why It Matters:** Key for deployment.
 - **Resources:**
 - [Real Python: Pickle.](#)
 - [Scikit-learn: Persistence.](#)
 - **Tips:**
 - Use joblib for large models.
 - Test with new data.
- **Day 50: Flask API (Optional)**
 - **Tasks** (6 hours):
 - **Learn Flask** (3 hours): Install Flask (pip install flask), create a REST API.
 - **Practice** (3 hours): Serve a model via an endpoint, test with Postman.
 - **Why It Matters:** Shows deployment skills.
 - **Resources:**
 - [Real Python: Flask.](#)
 - [Towards Data Science: Deployment.](#)
 - **Tips:**
 - Keep API simple.
 - Save code in GitHub.
- **Day 51: Build Portfolio**
 - **Tasks** (6 hours):
 - **Organize GitHub** (3 hours): Create repositories for projects, write READMEs with visuals.
 - **Main Portfolio** (3 hours): Add main README to GitHub profile linking projects.
 - **Why It Matters:** Portfolio showcases skills to recruiters.

- **Resources:**
 - [GitHub Portfolio Examples.](#)
 - [Markdown Cheat Sheet.](#)
- **Tips:**
 - Highlight top 3 projects.
 - Include screenshots.

Week 8: Advanced ML and Exploration (Days 52–60)

Goal: Explore advanced ML, prepare for capstone.

• Day 52: Ensemble Methods

- **Tasks** (6 hours):
 - **Learn Ensembles** (3 hours): Study boosting (XGBoost), stacking.
 - **Practice** (3 hours): Implement XGBoost on a classification dataset.
- **Why It Matters:** Improves model performance.
- **Resources:**
 - [XGBoost Docs.](#)
 - [StatQuest: Boosting.](#)
- **Tips:**
 - Install xgboost (pip install xgboost).
 - Compare with random forests.

• Day 53: Cross-Validation

- **Tasks** (6 hours):
 - **Learn Cross-Validation** (3 hours): Study k-fold cross-validation.
 - **Practice** (3 hours): Apply `cross_val_score()` to a model.
- **Why It Matters:** Ensures robust evaluation.
- **Resources:**
 - [Scikit-learn: Cross-Validation.](#)
 - [Towards Data Science: Cross-Validation.](#)
- **Tips:**
 - Use `cv=5` for balance.
 - Save results.

• Day 54: Feature Engineering

- **Tasks** (6 hours):
 - **Learn Feature Engineering** (3 hours): Study creating features, encoding, selection.
 - **Practice** (3 hours): Create 3–5 features for a dataset.
- **Why It Matters:** Features boost model performance.

- **Resources:**
 - [Kaggle: Feature Engineering.](#)
 - [Towards Data Science: Feature Engineering.](#)
- **Tips:**
 - Test feature impact.
 - Document steps.
- **Day 55: Introduction to NLP**
 - **Tasks** (6 hours):
 - **Learn NLP** (3 hours): Study tokenization, stemming, TF-IDF.
 - **Practice** (3 hours): Process text with nltk or TfidfVectorizer.
 - **Why It Matters:** NLP is a key data science field.
 - **Resources:**
 - [NLTK Book.](#)
 - [Towards Data Science: NLP.](#)
 - **Tips:**
 - Install nltk (pip install nltk).
 - Start with simple text preprocessing.
- **Day 56: Time Series Basics**
 - **Tasks** (6 hours):
 - **Learn Time Series** (3 hours): Study trends, seasonality.
 - **Practice:** (3 hours): Analyze a time series dataset with Pandas (e.g., stock prices).
 - **Why It Matters:** Common in business.
 - **Resources:**
 - [Kaggle: Time Series.](#)
 - [Towards Data Science: Time Series.](#)
 - **Tips:**
 - Use pd.to_datetime() for parsing dates.
 - Plot trends.
- **Day 57: Capstone Project – Planning**
 - **Tasks** (6 hours):
 - **Select Topic** (2 hours): Choose a topic (e.g., loan default, sales forecasting).
 - **Find Dataset** (2 hours): Select Kaggle/UCI dataset.
 - **Plan** (2 hours): Outline objectives, preprocessing, in notebook.
 - **Why It Matters:** Capstone is portfolio centerpiece.
 - **Resources:**
 - [Kaggle Datasets.](#)

- [UCI Machine Learning](#).
- **Tips:**
 - Choose job-aligned topic.
 - Keep scope manageable.
- **Day 58: Capstone – Data Exploration**
 - **Tasks** (6 hours):
 - **Explore and Load** (3 hours): Load dataset, check structure, identify issues.
 - **Clean** (3 hours): Handle nulls, outliers, encode variables.
 - **Why It Matters:** Clean dataset for modeling.
 - **Resources:**
 - [Kaggle: Data Cleaning](#).
 - [Pandas Docs](#).
 - **Tips:**
 - Save pipeline.
 - Document issues.
- **Day 59: Capstone Project – EDA**
 - **Tasks** (6 hours):
 - **Perform EDA** (3 hours): Compute stats, create visualizations (heatmap, boxplots).
 - **Summarize:** (3 hours): Write insights in Markdown.
 - **Why It Matters:** EDA drives model decisions.
 - **Resources:**
 - [Seaborn](#).
 - [Towards Data Science: EDA](#).
 - **Tips:**
 - Focus on insights.
 - Save visualizations.
- **Day 60: Capstone – Model Building**
 - **Tasks** (6 hours):
 - **Train Models** (3 hours): Train 2–3 models (logistic regression, random forest, XGBoost).
 - **Evaluate:** (3 hours): Compare with cross-validation, metrics.
 - **Why It Matters:** Shows model selection skills.
 - **Resources:**
 - [Scikit-learn: Model Selection](#).
 - [XGBoost Docs](#).
 - **Tips:**

- Use `cross_val_score()`.
 - Save results in table.
-

Month 3: Capstone Project, Deployment, Interviews (Days 61–90)

Focus: Complete capstone, deploy model, prepare interviews, job search.

Week 9: Capstone Completion (Days 61–67)

- **Day 61: Hyperparameter Tuning**

- **Tasks** (6 hours):
 - **Learn Tuning** (3 hours): Study `GridSearchCV`, `RandomizedSearchCV`.
 - **Practice** (3 hours): Tune for capstone's best model.
- **Why It Matters:** Improves model performance.
- **Resources:**
 - [Scikit-learn: Tuning](#).
 - [Towards Data Science: GridSearchCV](#).
- **Tips:**
 - Limit search space.
 - Save best model.

- **Day 62: Final Model Training**

- **Tasks** (6 hours):
 - **Train Model** (3 hours): Train final model on full dataset.
 - **Visualize** (3 hours): Plot predictions, feature importance.
- **Why It Matters:** Ready for deployment.
- **Resources:**
 - [Scikit-learn: Persistence](#).
 - [Matplotlib Docs](#).
- **Tips:**
 - Test on holdout.
 - Save visualizations.

- **Day 63: Capstone Documentation**

- **Tasks** (6 hours):
 - **Document** (3 hours): Write notebook with intro, methods, results, conclusion.

- **Visualize** (3 hours): Add visuals, comments.
- **Why It Matters:** Professional presentation.
- **Resources:**
 - [Jupyter Best Practices](#).
 - [GitHub Templates](#).
- **Tips:**
 - Use storytelling.
 - Test notebook.
- **Day 64: Publish Capstone**
 - **Tasks** (6 hours):
 - **Clean Up** (2 hours): Remove unused code.
 - **Publish** (3 hours): Upload to GitHub with README, screenshots.
 - **Share** (1 hour): Post on LinkedIn.
 - **Why It Matters:** Portfolio highlight.
 - **Resources:**
 - [Kaggle Notebooks](#).
 - [GitHub Docs](#).
 - **Tips:**
 - Ensure reproducibility.
 - Test links.
- **Day 65: Streamlit Introduction**
 - **Tasks** (6 hours):
 - **Learn Streamlit** (2 hours): Install streamlit (pip install streamlit), create simple app.
 - **Practice** (4 hours): Add interactive elements (sliders, dropdowns).
 - **Why It Matters:** Enables web app deployment.
 - **Resources:**
 - [Streamlit Docs](#).
 - [Streamlit Tutorials](#).
 - **Tips:**
 - Test locally (streamlit run).
 - Use example apps.
- **Day 66: Deploy Capstone with Streamlit**
 - **Tasks** (6 hours):
 - **Build App** (3 hours): Create Streamlit app for capstone model (e.g., input features, predict).
 - **Deploy** (3 hours): Host on Streamlit Cloud, share link.
 - **Why It Matters:** Demonstrates deployment.

- **Resources:**
 - [Streamlit Deployment](#).
 - [GitHub Streamlit](#).
- **Tips:**
 - Ensure user-friendly.
 - Test deployed app.
- **Day 67: Capstone Demo**
 - **Tasks** (6 hours):
 - **Create Demo** (3 hours): Record 2–3 min video or write blog post.
 - **Share** (3 hours): Share on LinkedIn, add to GitHub.
 - **Why It Matters:** Makes project accessible.
 - **Resources:**
 - [Canva Video](#).
 - [Medium Blogs](#).
 - **Tips:**
 - Keep demo concise.
 - Highlight key features.

Week 10: Resume, LinkedIn, Job Prep (Days 68–74)

- **Day 68: Resume Building**
 - **Tasks** (6 hours):
 - **Create Resume** (3 hours): Build ATS-friendly resume (1-page, project-based, skills).
 - **Review** (3 hours): Check typos, use action verbs.
 - **Why It Matters:** Gets past screenings.
 - **Resources:**
 - [Novoresume ATS](#).
 - [Towards Data Science: Resume](#).
 - **Tips:**
 - Tailor for each job.
 - Quantify results.
- **Day 69: LinkedIn Profile**
 - **Tasks** (6 hours):
 - **Update Profile** (3 hours): Write summary, add skills, projects.
 - **Engage** (3 hours): Connect with 5 data scientists, comment on posts.
 - **Why It Matters:** Key for networking, jobs.
 - **Resources:**

- [LinkedIn Learning](#).
 - [DataCamp LinkedIn](#).
- **Tips:**
 - Professional photo, headline.
 - Link GitHub.
- **Day 70: Portfolio Site**
 - **Tasks** (6 hours):
 - **Build Site** (3 hours): Create GitHub Pages or Notion portfolio.
 - **Polish** (3 hours): Add links, ensure mobile-friendly.
 - **Why It Matters:** Consolidates work.
 - **Resources:**
 - [GitHub Pages](#).
 - [Notion Templates](#).
 - **Tips:**
 - Keep design clean.
 - Test on multiple devices.
- **Day 71: Initial Job Applications**
 - **Tasks** (6 hours):
 - **Apply** (3 hours): Apply to 10 entry-level jobs on LinkedIn, Naukri.
 - **Track** (3 hours): Create spreadsheet for applications.
 - **Why It Matters:** Tests job market.
 - **Resources:**
 - [LinkedIn Jobs](#).
 - [Naukri](#).
 - **Tips:**
 - Focus on “Junior” roles.
 - Tailor resumes.
- **Day 72: Interview Prep – ML Theory**
 - **Tasks** (6 hours):
 - **Review Theory** (3 hours): Study overfitting, bias-variance, ensembles.
 - **Practice** (3 hours): Write explanations for 5 concepts.
 - **Why It Matters:** Common in interviews.
 - **Resources:**
 - [StatQuest](#).
 - [ML Mastery](#).
 - **Tips:**
 - Explain simply.
 - Use diagrams.

- **Day 73: Interview Prep – Python**

- **Tasks** (6 hours):
 - **Review Python** (2 hours): Revisit lists, dictionaries, Pandas.
 - **Practice** (4 hours): Solve 5 HackerRank problems.
- **Why It Matters:** Coding tests are standard.
- **Resources:**
 - [HackerRank](#).
 - [LeetCode](#).
- **Tips:**
 - Focus on time complexity.
 - Save solutions.

- **Day 74: Interview Prep – SQL**

- **Tasks** (6 hours):
 - **Review SQL** (2 hours): Study SELECT, JOIN, CASE.
 - **Practice** (4 hours): Solve 5 problems on LeetCode.
- **Why It Matters:** Essential for queries.
- **Resources:**
 - [Mode SQL](#).
 - [StrataScratch](#).
- **Tips:**
 - Explain query logic.
 - Use online editor.

Weeks 11–12: Interview Prep & Job Search (Days 75–90)

- **Day 75: Mock Interview**

- **Tasks** (6 hours):
 - **Practice** (3 hours): Mock interview with friend or Pramp.
 - **Review** (3 hours): Record, analyze feedback.
- **Why It Matters:** Builds confidence.
- **Resources:**
 - [Pramp](#).
 - [YouTube Interviews](#).
- **Tips:**
 - Be concise.
 - Explain projects clearly.

- **Day 76: Case Studies**

- **Tasks** (6 hours):

- **Learn Cases** (2 hours): Study open-ended questions (e.g., “Reduce churn”).
 - **Practice** (4 hours): Solve 2 cases with CRISP-DM.
- **Why It Matters:** Tests problem-solving.
- **Resources:**
 - [Towards Data Science: Cases](#).
 - [StrataScratch](#).
- **Tips:**
 - Structure answers.
 - Verbalize approach.
- **Day 77: Networking**
 - **Tasks** (6 hours):
 - **Connect** (3 hours): Reach out to 10 LinkedIn professionals.
 - **Engage** (3 hours): Join groups, comment on posts.
 - **Why It Matters:** Leads to referrals.
 - **Resources:**
 - [LinkedIn Tips](#).
 - [DataCamp Networking](#).
 - **Tips:**
 - Personalize messages.
 - Follow up politely.
- **Day 78: Portfolio Polish**
 - **Tasks** (6 hours):
 - **GitHub** (3 hours): Fix typos, update READMEs.
 - **LinkedIn** (3 hours): Add capstone, refine profile.
 - **Why It Matters:** Strong first impression.
 - **Resources:**
 - [Grammarly](#).
 - [GitHub Checklist](#).
 - **Tips:**
 - Friend review.
 - Test links.
- **Days 79–90: Daily Routine**
 - **Tasks** (6 hours daily):
 - **Apply** (2 hours): 5 jobs daily (LinkedIn, Naukri, Instahyre).
 - **Review** (2 hours): 3–5 ML/stats/Python questions.
 - **Practice** (1 hour): 1 SQL, 1 Python problem (LeetCode, HackerRank).
 - **Network** (1 hour): Connect 2–3 professionals, engage.

- **Why It Works:** Maximizes opportunities.
 - **Resources:**
 - [Instahyre](#), [Wellfound](#), [Hirect](#).
 - [LeetCode](#), [StrataScratch](#).
 - **Tips:**
 - Track applications in spreadsheet.
 - Stay positive.
-

Outcome

- **Portfolio:** 5+ projects (EDA, housing, spam, segmentation, capstone).
 - **Assets:** Resume, LinkedIn, GitHub, Streamlit app.
 - **Applications:** 100+ tailored applications.
 - **Interviews:** Ready for technical/behavioral questions.
 - **Roles:** Junior Data Scientist (₹10 LPA).
-

Success Strategies

- **Time:** Use Notion/Trello, 2-hour blocks, weekly reviews.
 - **Community:** Join Reddit (r/datascience), Discord, LinkedIn groups.
 - **Errors:** Log errors, use Stack Overflow.
 - **Upskill:** Explore AWS/GCP, Docker if time permits.
 - **Mindset:** Stay resilient, celebrate wins, prioritize health.
-

Daily Schedule (6 Hours)

- **Morning:** 2 hours theory (videos, articles).
 - **Mid-Morning:** 2 hours coding practice.
 - **Afternoon:** 2 hours projects/datasets.
 - **Breaks:** 10 min hourly, longer lunch.
-

This roadmap is achievable with dedication. If you need clarification or additional resources, let me know!