Next-Gen Data Science

Trends, Tech & Transformation

Thursday, May 15, 2025

Anant Prakash Awasthi, PhD

Disclaimer

The views, opinions, and content shared during this session are solely my own and are intended for educational and discussion purposes only. They do not represent the official stance, policies, or perspectives of my organization or any affiliated entity. This is a candid, open conversation aimed at enriching your understanding and encouraging dialogue in the field of Data Science.

About Me 📊 ✨

Education

Amity Institute of Applied Sciences
2025 Ph.D. (Statistics)

University of Lucknow
2011 M.Sc. (Biostatistics)

University of Lucknow
2008 B.Sc. (Statistics)

Professional Journey (~14 Years)

Optum
Pharmacy Fraud, Data Products, Medication Adherence

genpact
Contact Center, Data Products, Operations Analytics

SUTHERLAND
Data Products, Capital Market, HR Analytics, Aviation

cognizant
Banking, Data Products, OEM Manufacturing

serco
Contact Center, Banking, Infrastructure

TCS TATA CONSULTANCY SERVICES
Banking – Cards & Payments

Xchanging
Insurance, Spend Analysis, Operations

Interests

- Generative AI
- Low Resource Machine Learning
- Privacy & Ethics in AI
- Responsible use of AI
- Explainable AI (XAI)
- Information Systems
- Cloud Computing
- AI for Edge Devices

A2025 Logistics

Online Infrastructure

- GitHub Account
- Databricks Community Account
- Google Colaboratory
- OpenAI Platform Account
- Pinecone API Account
- Hugging Face Account

Software (Local Environment)

Microsite and Resources

- Workshop Microsite

A2025 Agenda

- Introduction to Data Science
- Source Code Management & GitHub
- Building a Data Science Solution
  - A Workflow
  - Building a regression solution
  - Building a Classification solution
  - Using Agents for Data Science Developments
- Managing Data Science in a Big Data Environment
- Introduction to Generative AI
  - Building Generative AI Solution
  - Building Generative AI Solution (RAG Based)
  - Building an Agent
- Introduction to Cloud

A2025 Agenda

- **Introduction to Data Science**
- Source Code Management & GitHub
- Building a Data Science Solution
  - A Workflow
  - Building a regression solution
  - Building a Classification solution
  - Using Agents for Data Science Developments
- Managing Data Science in a Big Data Environment
- Introduction to Generative AI
  - Building Generative AI Solution
  - Building Generative AI Solution (RAG Based)
  - Building an Agent
- Introduction to Cloud

A2025 Introduction to Data Science

Just like a chef turns ingredients into delicious meals, a data scientist takes numbers, facts, and figures—and cooks up insights that help businesses solve problems, plan better, and serve people smarter.

It's a mix of common sense, curiosity, and computer help to understand what's happening, why it's happening, and what could happen next.

Data Science is Like Cooking a Delicious Meal

Ingredients	**Raw Data** (sales numbers, customer info, etc.)
Recipe	**Algorithm** (set of steps to solve a problem
Chef	**Data Scientist**
Kitchen Tools (knife, stove)	**Tools** (Excel, Python, Power BI)
Taste Testing	**Data Validation**
Final Dish	**Insight or Prediction**

A2025 Introduction to Data Science

Data Science (as a discipline)

Data Science is an interdisciplinary field that combines techniques from statistics, computer science, and domain knowledge to extract meaningful insights and patterns from structured and unstructured data.

It involves the entire lifecycle of working with data; including data collection, cleaning, exploration, modeling, interpretation, and communication to support decision-making, automate processes, or enable predictive and prescriptive analytics.

At its core, Data Science blends:

- Mathematics & Statistics – for quantifying uncertainty and making inferences
- Computer Science & Programming – for building tools, pipelines, and algorithms
- Domain Expertise – to ensure relevance and actionability of results

It is foundational to modern technologies such as machine learning, AI, business intelligence, and data-driven decision systems.

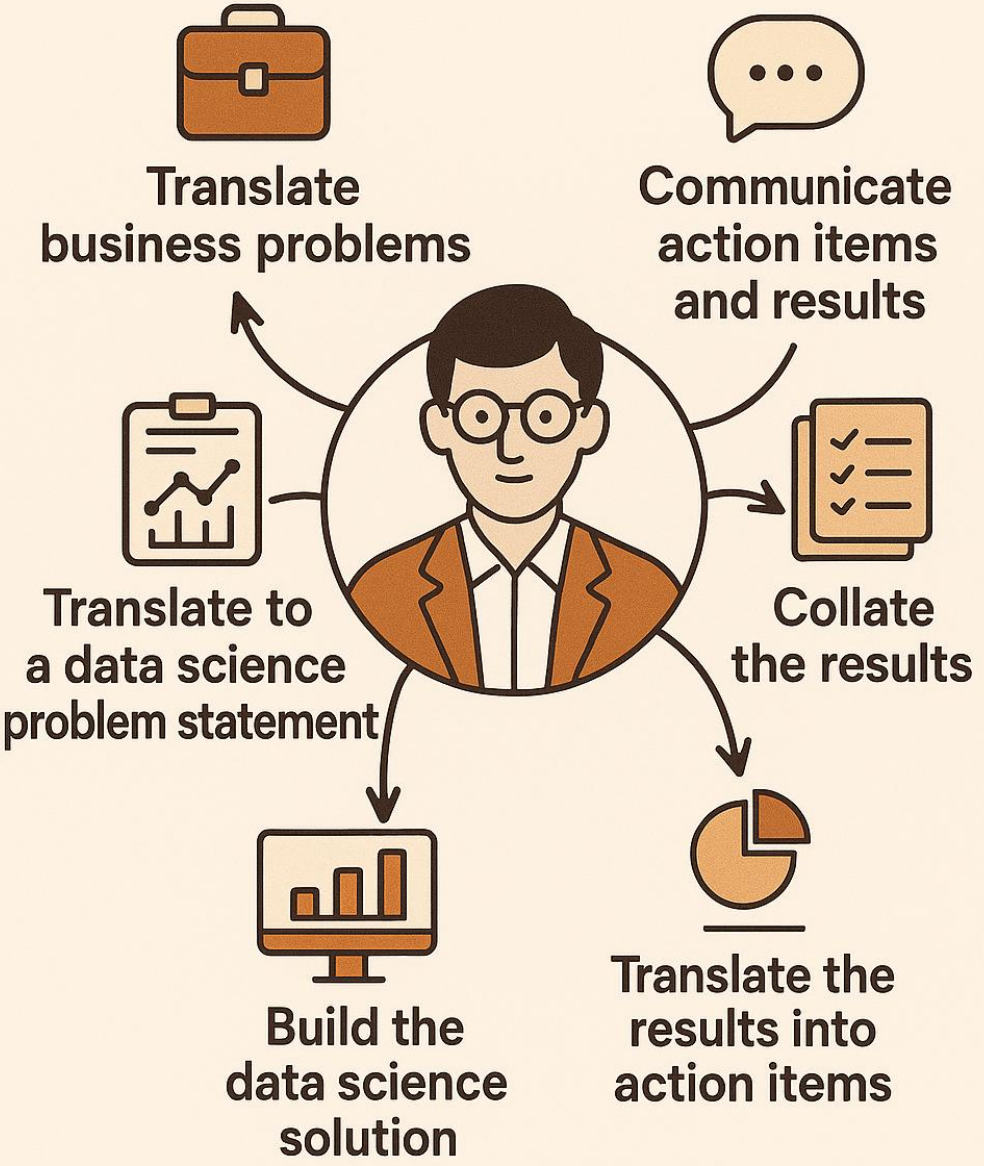
A2025 Introduction to Data Science

Who is a good data scientist?

*A person who can translate business problems to data science problem statement, translate data science statement to a data science solution, build the data science solution (**efficiently**), collate the results, translate the results into action items, and communicate the action items and results to business stakeholders.*

A Data Scientist expected to be expert of:

Statistics, programming, data management, story telling (sometime referred as data visualization), and stakeholder communication

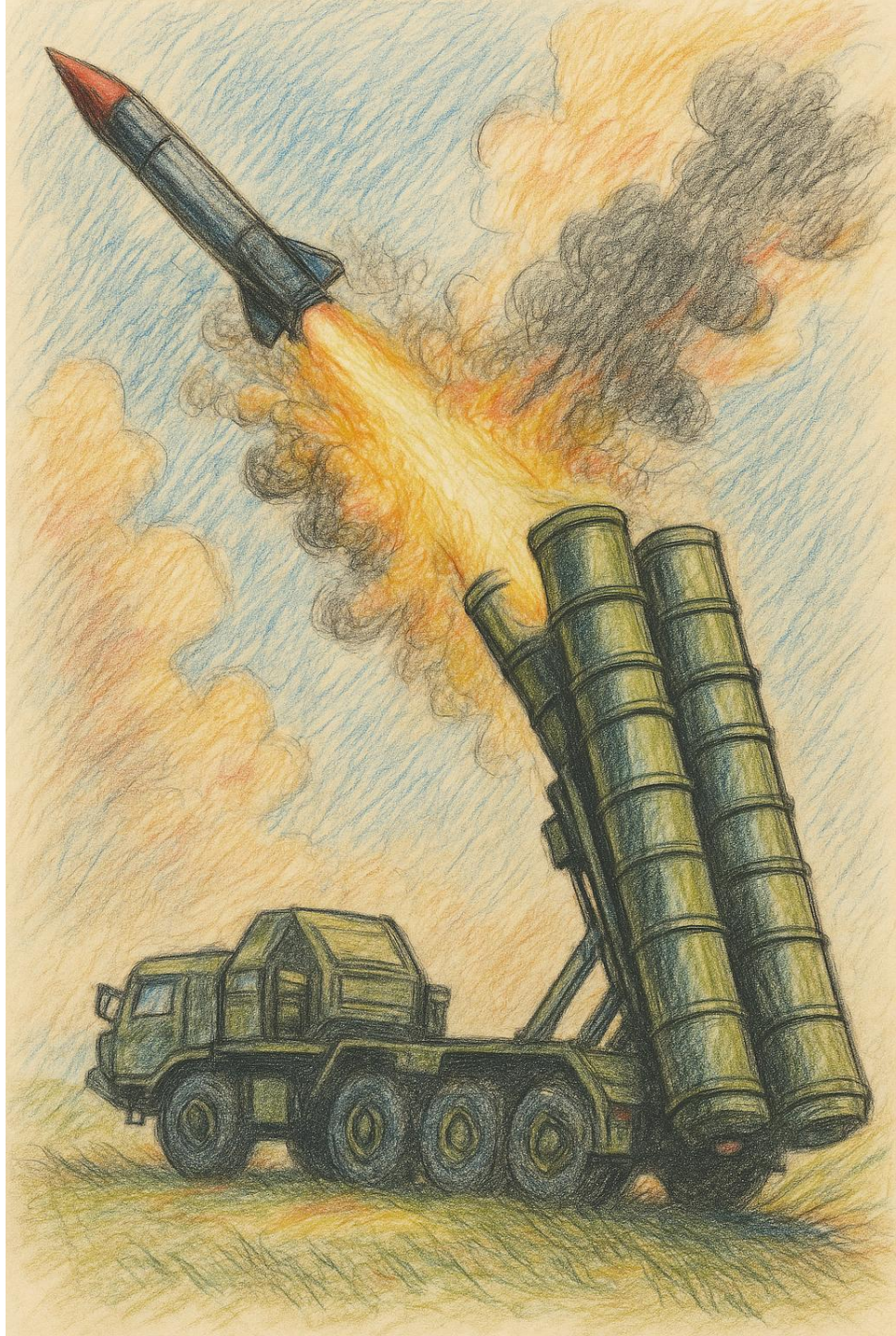


A2025 Skills of Data Scientist in 2025

- Statistical Analysis & Probability
- Programming (Python, R, SQL)
- Data Wrangling & Data Cleaning
- Machine Learning & Model Building
- Data Visualization
- Domain Knowledge
- Critical Thinking & Problem-Solving
- Communication & Storytelling
- Tool Proficiency
- Experimentation & Deployment (MLOps)

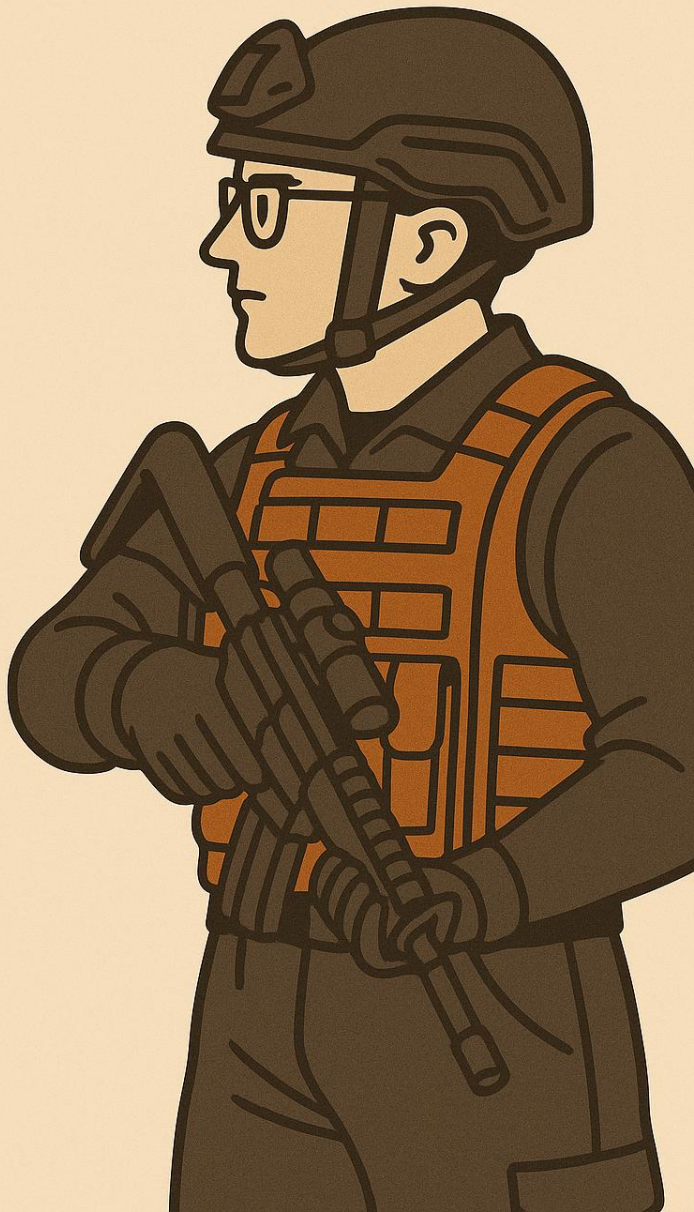
A2025 Arsenal of Data Scientist in 2025

Category	Key Tools	Purpose
Programming Languages	Python, R, SQL, Julia	Core for analysis, modeling, and data querying
IDE & Notebooks	Jupyter, JupyterLab, RStudio, VS Code, Deepnote, Hex	Interactive coding and reporting
Data Visualization & BI	Tableau, Power BI, Looker, Plotly, D3.js	Visualizing and presenting data-driven insights
Cloud Platforms & Big Data	AWS (S3, Redshift, SageMaker), GCP (BigQuery, Vertex AI), Azure (Synapse, ML), Databricks	Scalable storage, compute, and ML deployment
Data Engineering / Pipelines	Apache Spark, Hadoop, Airflow, Prefect, Luigi, Dagster, Metaflow	Data processing, orchestration, and scheduling
Databases & Storage	PostgreSQL, MySQL, MongoDB, Snowflake, Cassandra, Elasticsearch, Neo4j	Data storage: structured, unstructured, and distributed systems
ETL & Data Integration	Talend, Apache NiFi, Fivetran, Stitch, Matillion	Moving, transforming, and integrating data
Version Control & Collaboration	Git, GitHub, GitLab, Bitbucket	Code and project versioning, collaboration
Experiment Tracking & MLOps	MLflow, Weights & Biases, Neptune.ai, DVC, Kubeflow	Managing ML experiments and production deployments
Spreadsheets & Light Tools	Excel, Google Sheets, Airtable, Notion	Quick analysis, dashboards, collaborative planning



A2025 Data Scientist Career timeline (As Individual Contributor)

Designation	Years of Experience	Key Skillsets	Typical Responsibilities
Junior Data Scientist	0–2 Years	Python/R, basic statistics, SQL, data cleaning, data visualization (Tableau, Power BI), foundational ML	Assist in EDA, data wrangling, build basic models, support senior staff
Data Scientist	2–5 Years	Advanced stats, supervised ML, feature engineering, version control (Git), Hadoop/Spark basics	Own projects, build predictive models, translate business problems into DS tasks
Senior Data Scientist	5–8 Years	Deep learning, NLP, cloud (AWS/GCP), model tuning, A/B testing, experimentation framework	Lead complex projects, advise product teams, mentor juniors, contribute to product direction
Staff Data Scientist	8–12 Years	Specialized ML (e.g., RL, CV), production-grade ML engineering, scalable systems, API deployment	Set modeling standards, architect ML solutions, influence DS strategy, review code and project design
Principal Data Scientist	12+ Years	Strategic analytics vision, domain expertise, patents/publications, cross-industry leadership, responsible AI	Lead org-wide initiatives, represent in external conferences, shape strategy, evaluate new tech directions



A2025 Data Scientist Career timeline (Grows in Managerial Role)

Designation	Years of Experience	Key Skillsets	Typical Responsibilities
Data Scientist	0–3 Years	Python/R, SQL, data visualization, basic machine learning, statistical analysis	Analyze data, build predictive models, collaborate with cross-functional teams to derive insights
Senior Data Scientist	3–5 Years	Advanced machine learning, deep learning, big data tools (e.g., Hadoop, Spark), project leadership	Lead data science projects, mentor junior team members, contribute to strategic planning
Data Science Manager	5–8 Years	Team management, project management, stakeholder communication, budgeting, performance evaluation	Manage data science teams, oversee project execution, align data initiatives with business goals
Director of Data Science	8–12 Years	Strategic planning, cross-department collaboration, advanced leadership, policy development	Set data strategy, manage multiple teams or departments, drive organizational change through data initiatives
Vice President of Data Science	12+ Years	Executive leadership, enterprise data strategy, innovation management, governance	Lead the organization's data vision, oversee all data-related functions, represent data initiatives at the executive level

A2025 Agenda

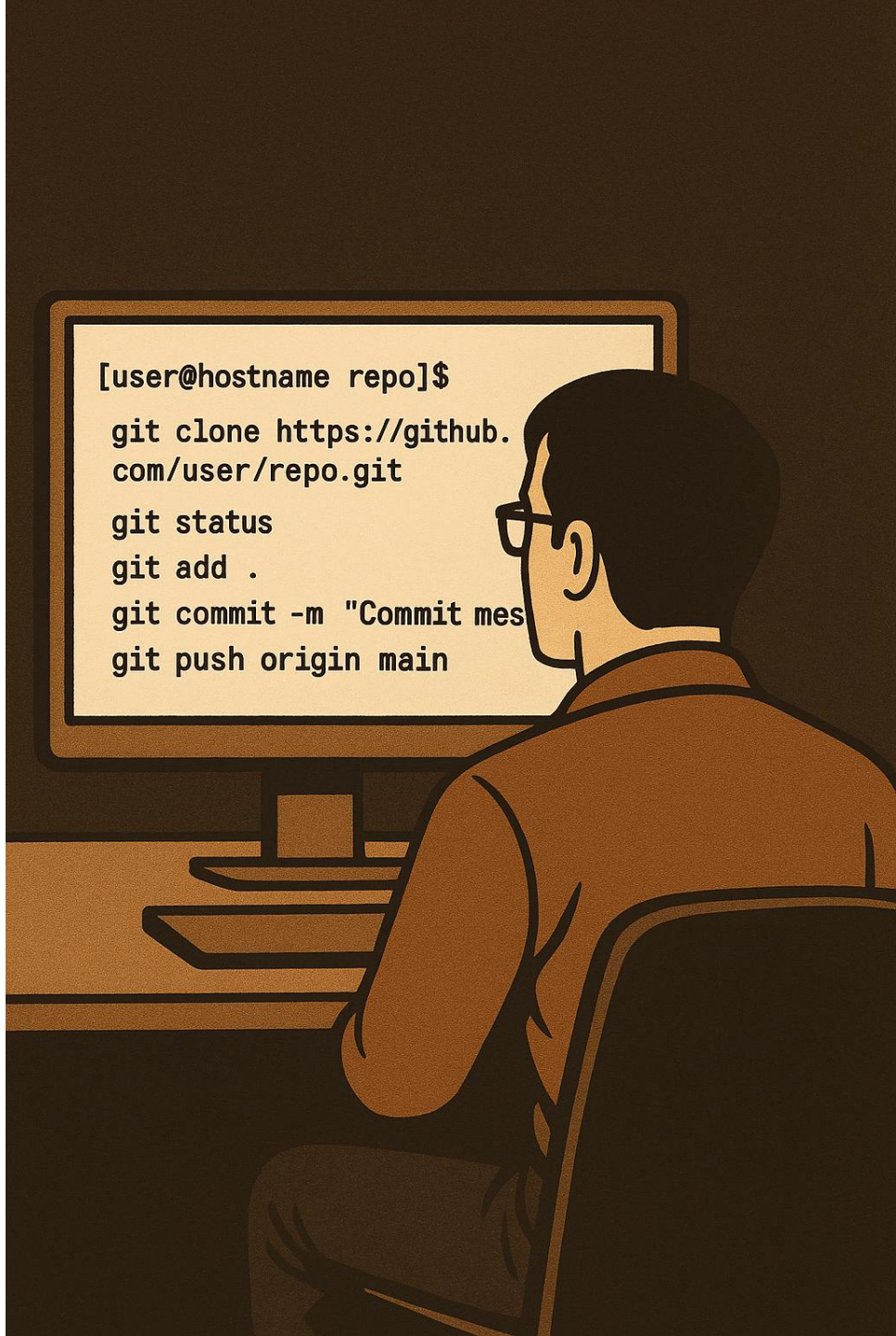
- Introduction to Data Science
- **Source Code Management & GitHub**
- Building a Data Science Solution
  - A Workflow
  - Building a regression solution
  - Building a Classification solution
  - Using Agents for Data Science Developments
- Managing Data Science in a Big Data Environment
- Introduction to Generative AI
  - Building Generative AI Solution
  - Building Generative AI Solution (RAG Based)
  - Building an Agent
- Introduction to Cloud

A2025 Introduction to Source Code Management

Before we moving forward

- *You have downloaded and installed **Git & GitHub Cli** (if not, go ahead and install it)*

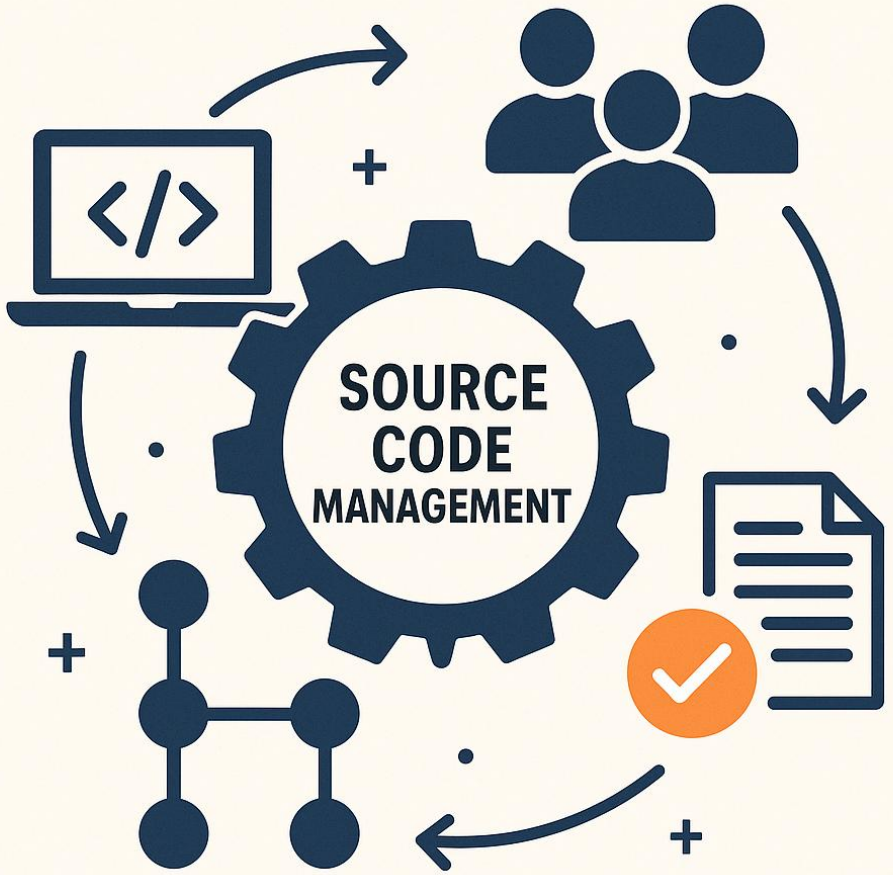
- *You have created a **GitHub account** (if not, please create one)*
  - GitHub Account



A2025 Source Code Management – An Introduction

Source Code Management (SCM) is a fundamental practice in software development that involves tracking, managing, and coordinating changes to code throughout its lifecycle. It enables developers to work collaboratively on the same codebase without overwriting each other's work, maintain a history of changes, and revert to previous versions if needed. SCM tools like Git, Mercurial, and Subversion allow teams to branch off, test new features, merge updates, and ensure the stability and integrity of the code. In essence, SCM serves as the backbone of modern software projects—enhancing productivity, accountability, and collaboration in development environments.

Source Code Management (SCM) is essential for data scientists because it brings structure, traceability, and collaboration to their work. While data science often revolves around experimentation, models, and data pipelines, without SCM tools like Git, managing code versions, tracking changes, and collaborating effectively becomes chaotic and error-prone.



A2025 Source Code Management – why learning SCM is crucial

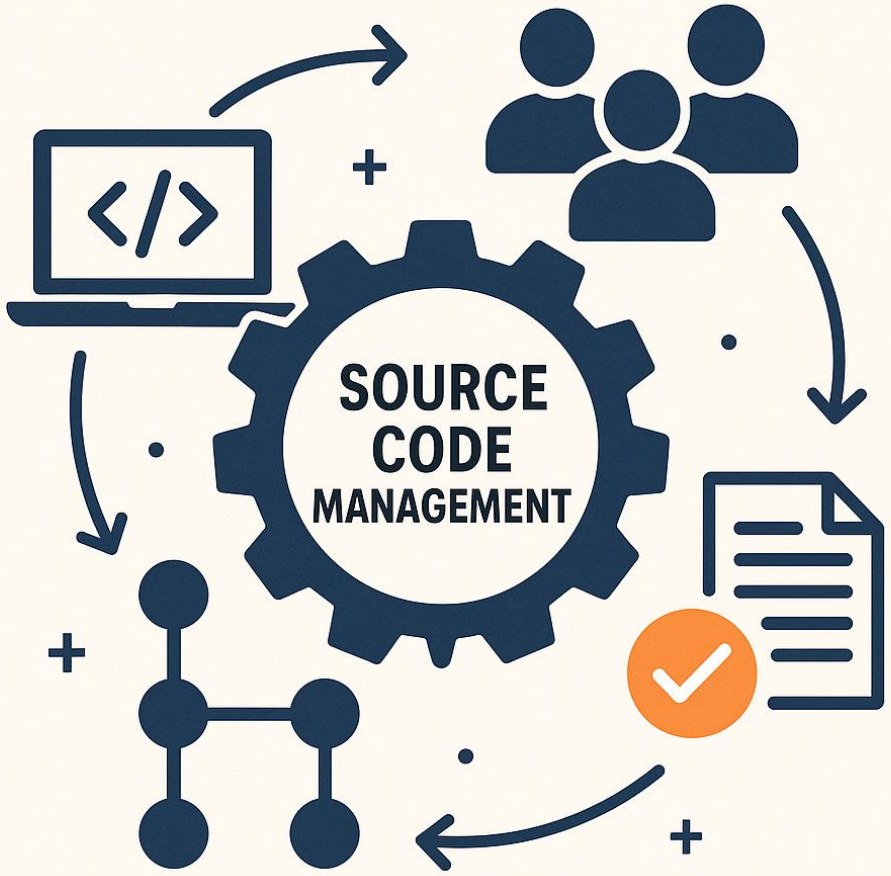
Version Control: Data science projects involve iterative changes to code, notebooks, and data processing scripts. SCM helps maintain a history of changes, making it easy to revert to earlier versions or compare changes over time.

Collaboration: In team settings, SCM allows multiple data scientists, analysts, and engineers to work on the same project without overwriting each other's work. Features like branching and merging support parallel development.

Reproducibility: SCM ensures that experiments and analyses can be reproduced reliably by tracking both code and configuration changes. This is critical in research and regulated industries.

Deployment & Integration: Modern ML projects are increasingly integrated into production pipelines. SCM is the foundation for CI/CD (Continuous Integration/Continuous Deployment), making it easier to deploy and maintain models.

Professionalism & Best Practices: Knowledge of SCM is expected in most data science roles. It aligns data scientists with software engineering best practices and improves their credibility in cross-functional teams.



A2025 Source Code Management – An Introduction

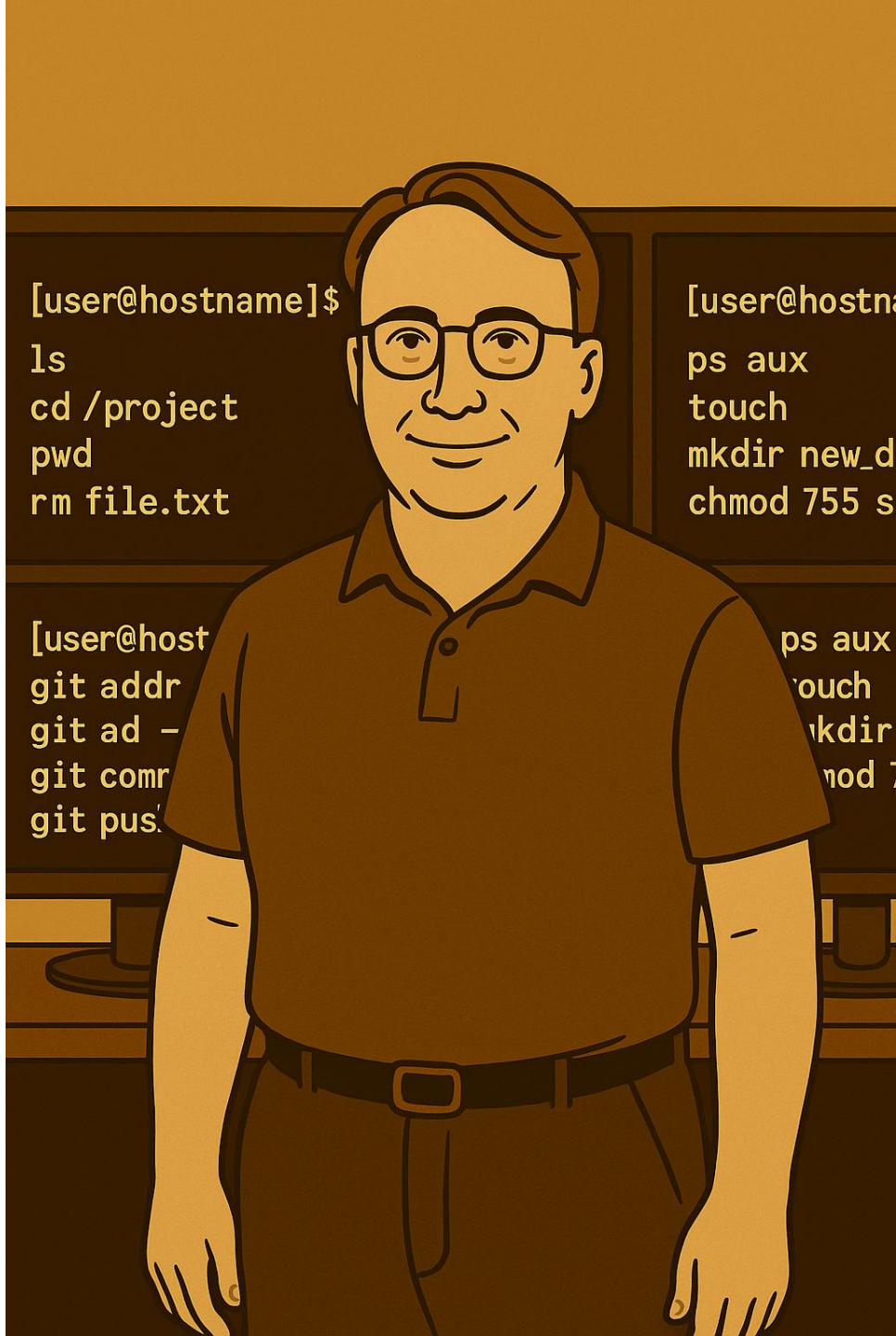
A Note of Gratitude to Linus Torvalds

Thank you, Linus Torvalds, for creating Git, a tool that revolutionized how developers collaborate, innovate, and build software together.

Your vision for a fast, distributed, and reliable Source Code Management system has not only empowered open-source communities but has also become the backbone of modern software development across the globe. With Git, you've enabled teams to work seamlessly, maintain cleaner histories, and push the boundaries of collaboration—whether they're individuals in garages or thousands-strong enterprises.

Your contribution continues to shape the future of technology.

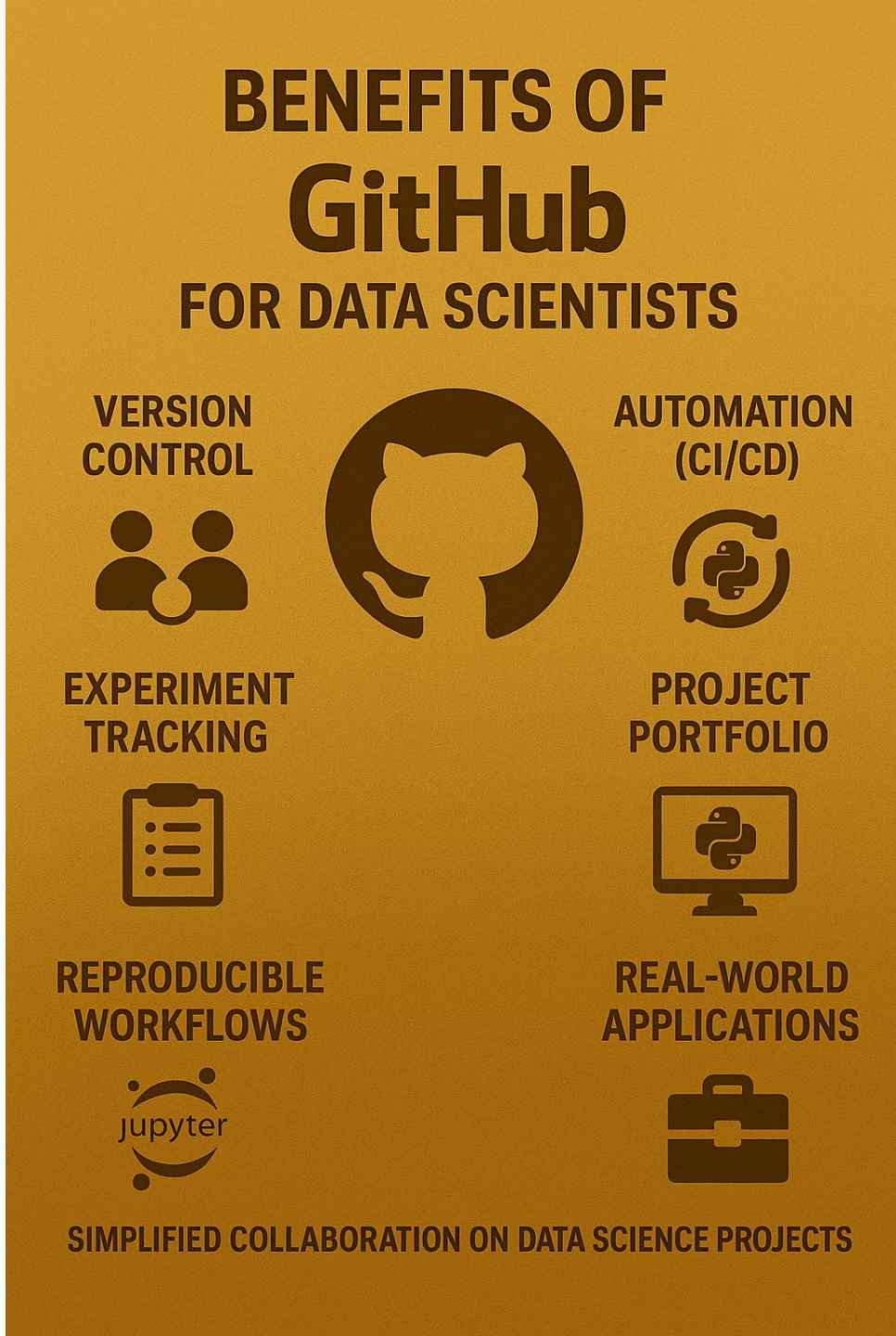
Thank you for Git. Thank you for empowering developers.



A2025 Source Code Management – The GitHub

GitHub is a cloud-based platform built on Git, designed to facilitate version control, code collaboration, and project management. Originally a tool for software developers, GitHub has become an indispensable resource for data scientists, offering a rich ecosystem to manage code, data, models, and workflows efficiently.

GitHub is a cloud-based platform built on Git, designed to facilitate version control, code collaboration, and project management. Originally a tool for software developers, GitHub has become an indispensable resource for data scientists, offering a rich ecosystem to manage code, data, models, and workflows efficiently.



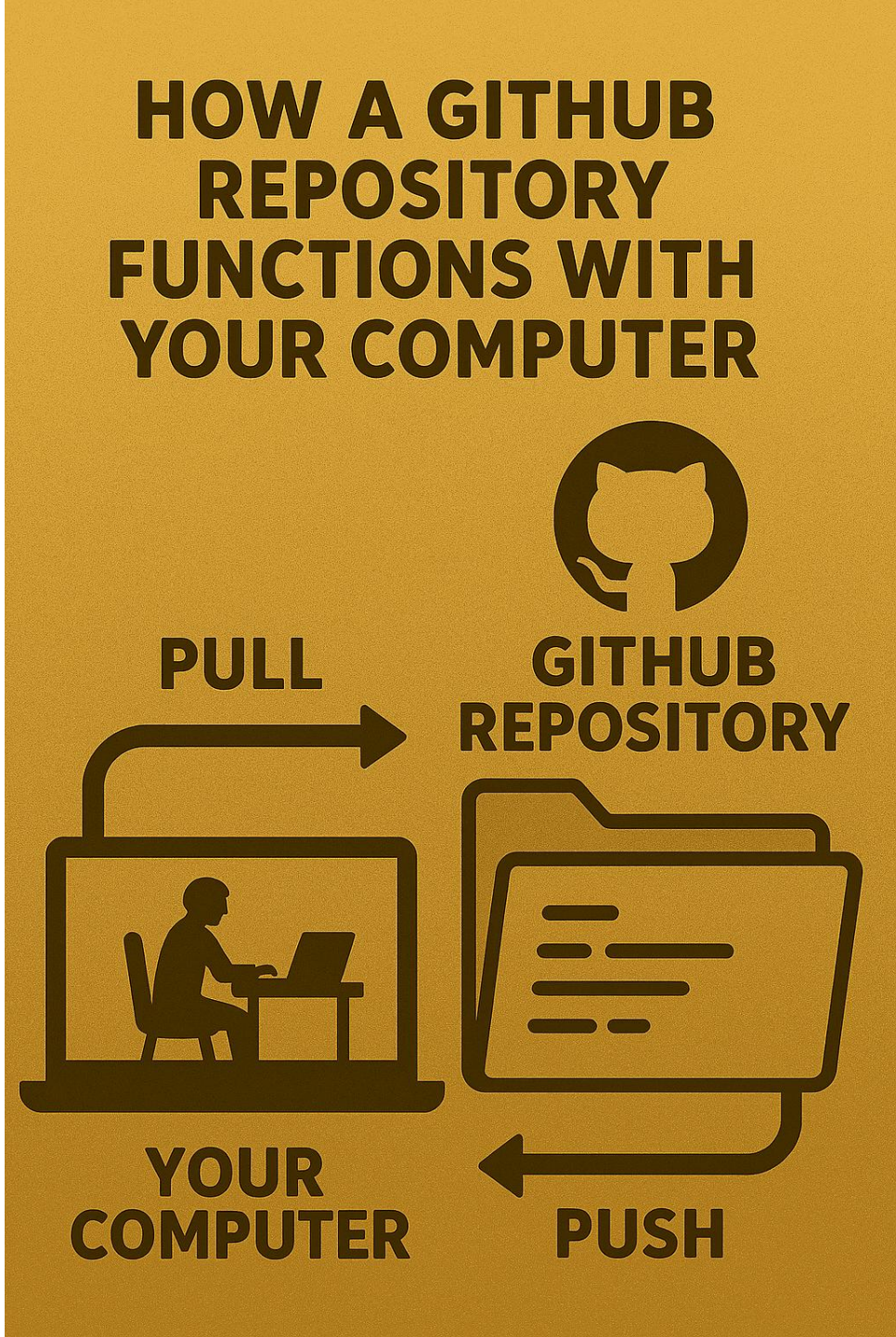
A2025 Source Code Management – The GitHub Ecosystem

GitHub is a cloud-based platform built on top of Git, a version control system. It helps you store, track, and collaborate on projects—especially code—in a secure and organized way.

At its core, GitHub functions like a smart online folder (called a repository) that keeps a complete history of your project. You can work on your project locally on your computer, then use GitHub to synchronize your changes using commands like push (upload your updates) and pull (download new updates from others).

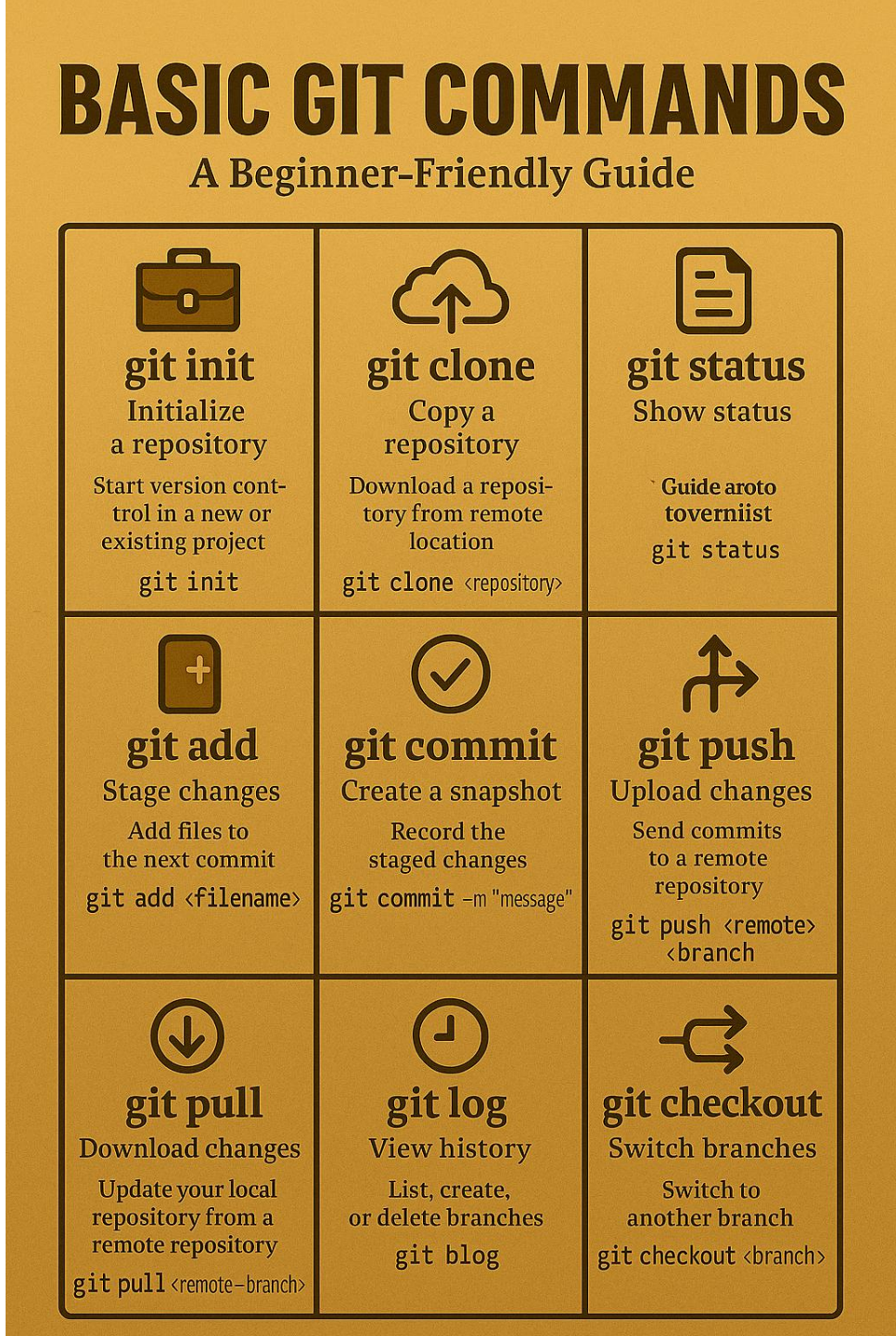
✅ **Key Concepts:**

- **Repository (Repo)**: A project folder on GitHub.
- **Commit**: A saved version or snapshot of your work.
- **Branch**: A separate workspace to try new ideas without disturbing the main project.
- **Push & Pull**: Push sends updates to GitHub; Pull gets updates from it.
- **Collaboration**: Multiple people can work on the same project without overwriting each other's work, thanks to GitHub's merging and review tools.



A2025 Source Code Management – GitHub Commands

Command	Purpose	Why It Helps You
git init	Start tracking versions of your project	Begin version control
git clone	Copy code from GitHub	Join team projects
git status	View current changes	Stay organized
git add	Select files to track	Choose what to save
git commit	Save a snapshot	Mark progress
git push	Share work on GitHub	Collaborate or back up
git pull	Get team updates	Stay in sync
git log	View history	Track project evolution
git branch	Create experiments	Test ideas safely
git checkout	Switch versions	Explore alternatives



A2025 Source Code Management – GitHub Case Study

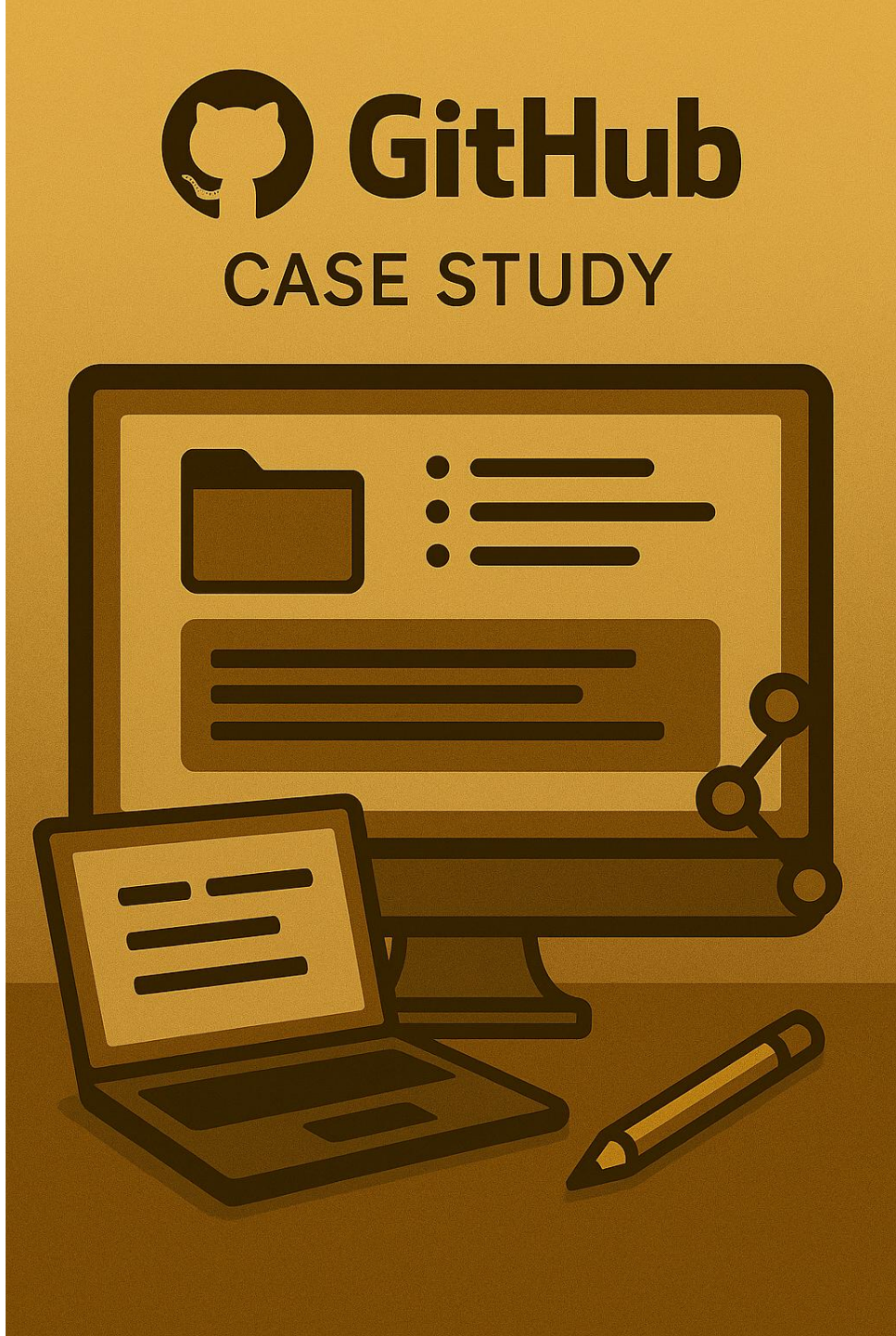
Hope your GitHub Environment is ready for action

Action Items:

1. Create a repository on GitHub
2. Setup the repository on Local Machine
3. Add a python file for printing “Hello World” to local repository
4. Add, commit, and push the local repository to remote repository

Action Items – if time allows:

1. Fork your friends repository
2. Setup on your local machine
3. Add or modify python file to this
4. Add, commit, and push the changes to remote repository
5. Raise a pull request for your friend’s action
6. Your friend merge the pull request



A2025 Agenda

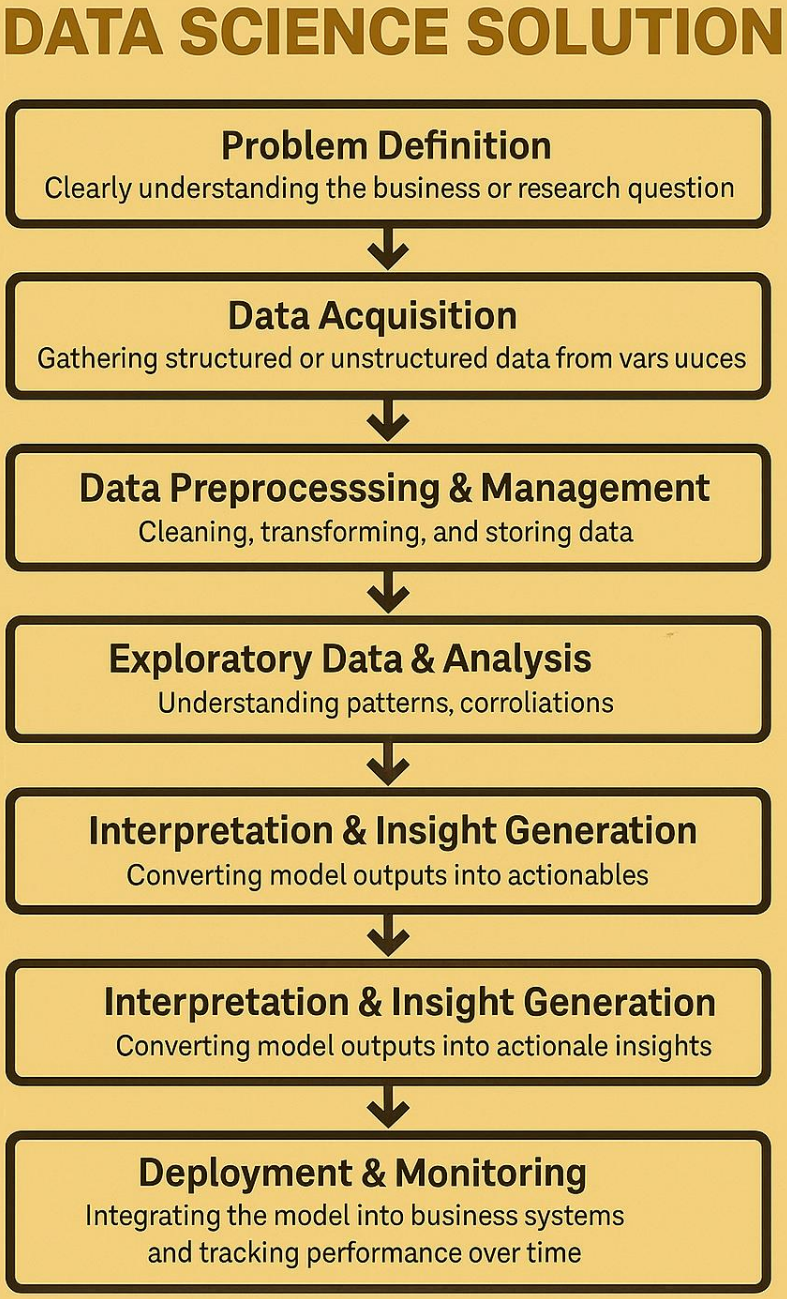
- Introduction to Data Science
- Source Code Management & GitHub
- **Building a Data Science Solution**
  - A Workflow
  - Building a regression solution
  - Building a Classification solution
  - Using Agents for Data Science Developments
- Managing Data Science in a Big Data Environment
- Introduction to Generative AI
  - Building Generative AI Solution
  - Building Generative AI Solution (RAG Based)
  - Building an Agent
- Introduction to Cloud

A2025 Building a Data Science Solution

A Data Science Solution is a structured, analytical approach that uses data, statistical techniques, and machine learning models to solve a real-world problem or drive decision-making. It typically involves collecting relevant data, analyzing it to discover patterns or insights, and then building predictive or prescriptive models that help optimize processes, improve efficiency, reduce risk, or uncover opportunities.

✅ **Key Components of a Data Science Solution:**

- Problem Definition
- Data Acquisition
- Data Preprocessing & Management
- Exploratory Data Analysis (EDA)
- Model Building & Evaluation
- Interpretation & Insight Generation
- Deployment & Monitoring



A2025 Building a Data Science Solution

Why It's Important for a Data Scientist to Understand the Workflow of Building a Data Science Solution

Understanding the end-to-end workflow of building a data science solution is crucial for a data scientist. Not just from a technical perspective, but also for ensuring real-world impact, stakeholder trust, and project success. Here's why:

- Provides Clarity and Structure
- Aligns Solutions with Business Goals
- Improves Data Quality Handling
- Strengthens Communication with Stakeholders
- Promotes Reproducibility and Best Practices
- Facilitates Iteration and Continuous Improvement
- Enables Deployment and Real-World Use

WHY UNDERSTANDING THE WORKFLOW OF DATA SCIENCE SOLUTIONS MATTERS

1 Provides Clarity and Structure
Ensures a sequential, organized approach, breaking complex projects into manageable steps.

2 Aligns Solutions with Business Goals
Ensures you're solving the right problem and optimizing relevant metrics.

3 Improves Data Quality Handling
Anticipates and addresses data-related challenges early on.

4 Strengthens Communication with Stakeholders
Builds trust and transparancy through clear explanation of each step.

5 Promotes Reproducibility and Best Practices
Encourages proper versioning, experiment tracking, and code modularity.

6 Facilitates Iteration and Continuous Improvement
Supports experimenting with new features, retraining models, and identifying lessons.

A2025 Building a Data Science Solution - Data Management

Data Management refers to the systematic process of collecting, organizing, storing, processing, and maintaining data throughout its lifecycle. It includes practices that ensure data is accurate, accessible, consistent, secure, and usable for analysis or decision-making.

For a data scientist, data management is not just a backend operation—it's a core competency that directly influences the quality and success of any data science project.

🧠 Why is Data Management Important for Data Scientists?

- Foundation for Analysis
- Efficiency and Reusability
- Scalability
- Reproducibility
- Compliance and Ethics
- Model Performance

A2025 Building a Data Science Solution - Data Management

Main Task	Subtasks	Purpose / Description
1. Data Loading	Load from CSV, Excel, JSON, SQL, APIs	Import raw data into your environment
2. Data Inspection	.head(), .info(), .describe(), .shape, .columns	Understand data structure and basic metadata
3. Subsetting	Row/column selection, filters, slicing, .loc[], .iloc[]	Focus on specific data segments
4. Data Cleaning	Handle missing values, remove duplicates, rename columns	Improve data quality
5. Type Conversion	Convert types like string to datetime, float to int	Ensure correct formats for processing
6. Transformation	apply(), map(), create derived features	Generate new insights or restructure data
7. Aggregation	groupby(), sum(), mean(), pivot tables	Summarize data across categories
8. Sorting & Ranking	sort_values(), rank()	Identify top performers, trends
9. Joining & Merging	merge(), join(), concat()	Combine data from multiple sources
10. Reshaping	melt(), pivot(), stack(), unstack()	Convert between wide and long formats
11. Exporting Data	to_csv(), to_excel(), to_sql()	Store transformed data for reuse or delivery
12. Time Series Handling	Parse dates, resample, lag features	Work with date-indexed or temporal data

A2025 Building a Data Science Solution

Data Management in Python using Pandas

Pandas is an open-source Python library used for data manipulation, analysis, and cleaning.

It provides high-performance, easy-to-use data structures such as:

- Series – a one-dimensional labeled array.
- DataFrame – a two-dimensional, tabular data structure (like an Excel sheet in code).

It is widely used by data scientists, analysts, and developers to handle structured data efficiently, especially in data preprocessing and exploratory data analysis (EDA).

🕰️ Brief History

- Developed by Wes McKinney in 2008 at AQR Capital to solve the lack of robust data tools in Python.
- Initially designed for financial modeling, but its flexibility led to widespread use in scientific computing, machine learning, and business analytics.
- The name "pandas" is derived from "panel data," an econometrics term referring to multidimensional structured data.

A2025 Building a Data Science Solution – Case study

Refer to GitHub Microsite for Case Study

A2025 Building a Data Science Solution

Data Visualization – An Introduction

Data Visualization is the art of turning raw numbers into pictures so that we can easily understand patterns, trends, and stories hidden inside the data.

It's like using maps to find directions, instead of getting lost in numbers, we use charts, graphs, and dashboards to get clear answers at a glance.

🕰️ Why Should a Data Scientist Master It?

- To Make Data Talk
- To Communicate Clearly
- To Spot Hidden Patterns
- To Drive Better Decisions

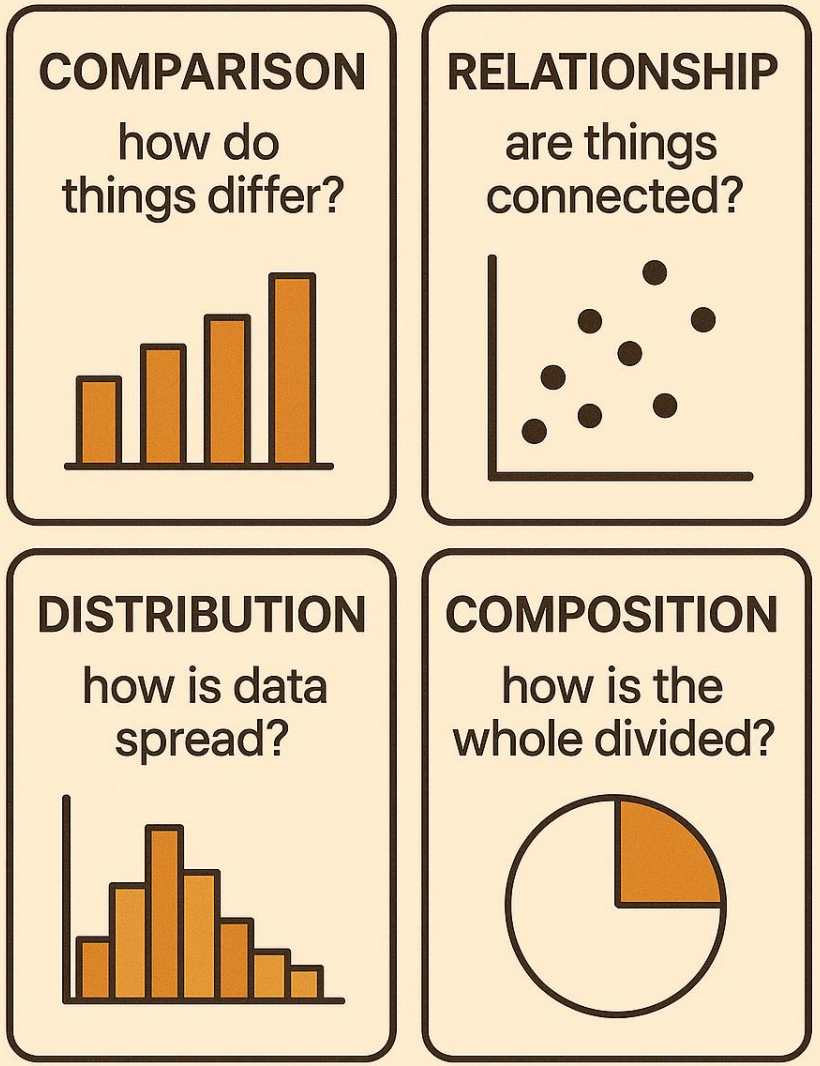
📝 In Short:

A data scientist who can visualize well is like a storyteller who turns data into insight. It's not just about doing the math — it's about showing the meaning behind it.

A2025 Building a Data Science Solution

Data Visualization – Questions to Answer

Problem Type	Question Solved	Common Charts
Comparison	How do things differ?	Bar, Column, Line
Relationship	Are things connected?	Scatter, Bubble, Heatmap
Distribution	How is the data spread?	Histogram, Box, Violin
Composition	How is the whole divided?	Pie, Stacked Bar, Area



A2025 Building a Data Science Solution

Data Visualization – Best Practices

✅ Do's of Data Visualization

Do's	Explanation / Benefit
Use clear titles	Helps the audience instantly understand the chart's story.
Label your axes	Always name X and Y to avoid confusion.
Use consistent colors	Same color = same meaning across visuals.
Highlight key data	Bold or accent important bars, points, or lines.
Use legends wisely	Only when there are multiple groups or variables.
Keep it simple	One chart, one clear message.
Use right chart types	Avoid fitting the data to your favorite chart.

❌ Don'ts of Data Visualization

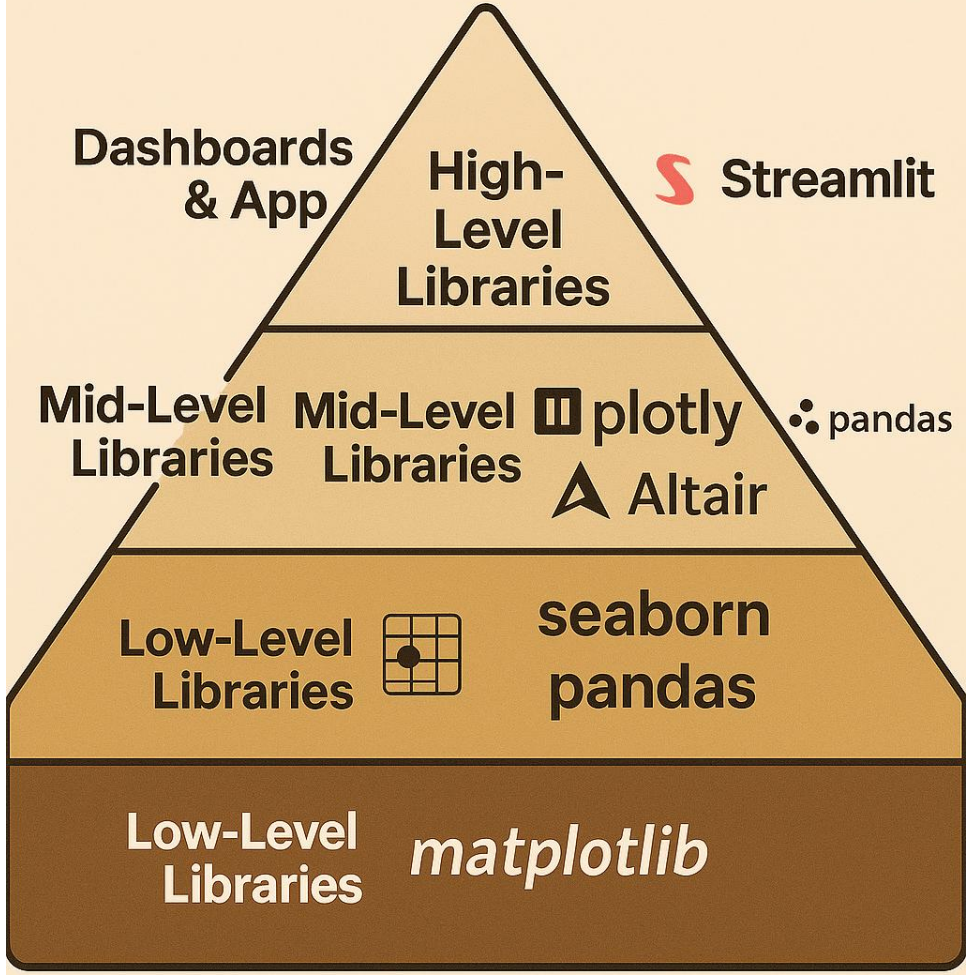
Don'ts	Why to Avoid
Avoid clutter	Too many elements = confusion.
Don't use 3D charts	They distort perception and don't add value.
Avoid rainbow color schemes	Hard to interpret and not colorblind-friendly.
Don't overload with text	Let visuals speak for themselves.
Avoid inconsistent scales	May mislead interpretation of trends.
Don't hide the baseline (0)	Especially in bar charts — it skews perception.

A2025 Building a Data Science Solution

Data Visualization – Python Ecosystem

Data Visualization is not just about creating pretty graphs, it's about making data understandable. In Python, there's a powerful ecosystem of tools and libraries that allows you to visualize everything from basic bar charts to interactive dashboards. This ecosystem has evolved to support exploration, communication, and decision-making — all from within a Python environment.

Library Level	Libraries	Purpose	Key Features	Use Cases
Low-Level	matplotlib	Foundational plotting	Highly customizable, static charts	Publication-quality plots, charts from scratch
Mid-Level	seaborn, pandas	Simplified static visualizations	Built on matplotlib, supports statistical plots	EDA, box plots, heatmaps, histograms
High-Level	plotly, altair, bokeh	Interactive, declarative visualizations	Tooltips, zoom, hover, multi-variable visualizations	Interactive dashboards, presentations
Dashboards & Apps	streamlit, dash, panel	App & dashboard creation using visual data	Web interface, interactive UI controls, Python-first	Analytics dashboards, rapid prototyping, storytelling apps



A2025 Building a Data Science Solution

Data Visualization – Case Studies

- Department-wise Age Distribution
- Attrition vs Monthly Income
- Gender Diversity across Departments
- Job Role vs Job Satisfaction
- Performance Rating vs Training Frequency

CASE STUDY

A2025 Building a Data Science Solution

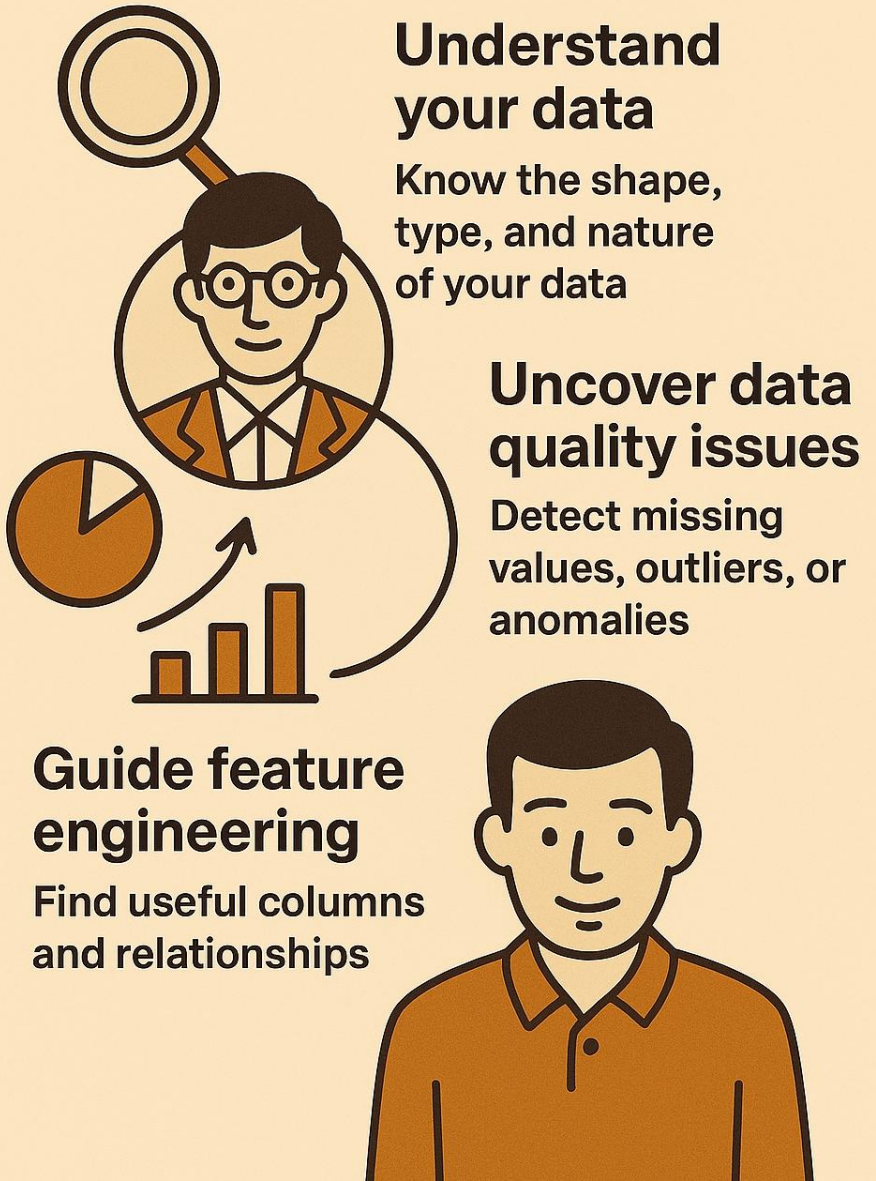
Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the process of investigating and visualizing data to discover patterns, spot anomalies, test assumptions, and gain insights — before applying any machine learning or statistical models.

Think of EDA as a detective's first walkthrough of a crime scene: before jumping to conclusions, the detective explores the environment, gathers facts, and asks better questions. Similarly, a data scientist uses EDA to "interview the data" and understand its structure, quality, and behavior.

Why Should a Data Scientist Master EDA?

- It Helps You Understand Your Data
- It Uncovers Data Quality Issues
- It Guides Feature Engineering
- It Builds Intuition for Model Building
- It Communicates Insights Clearly

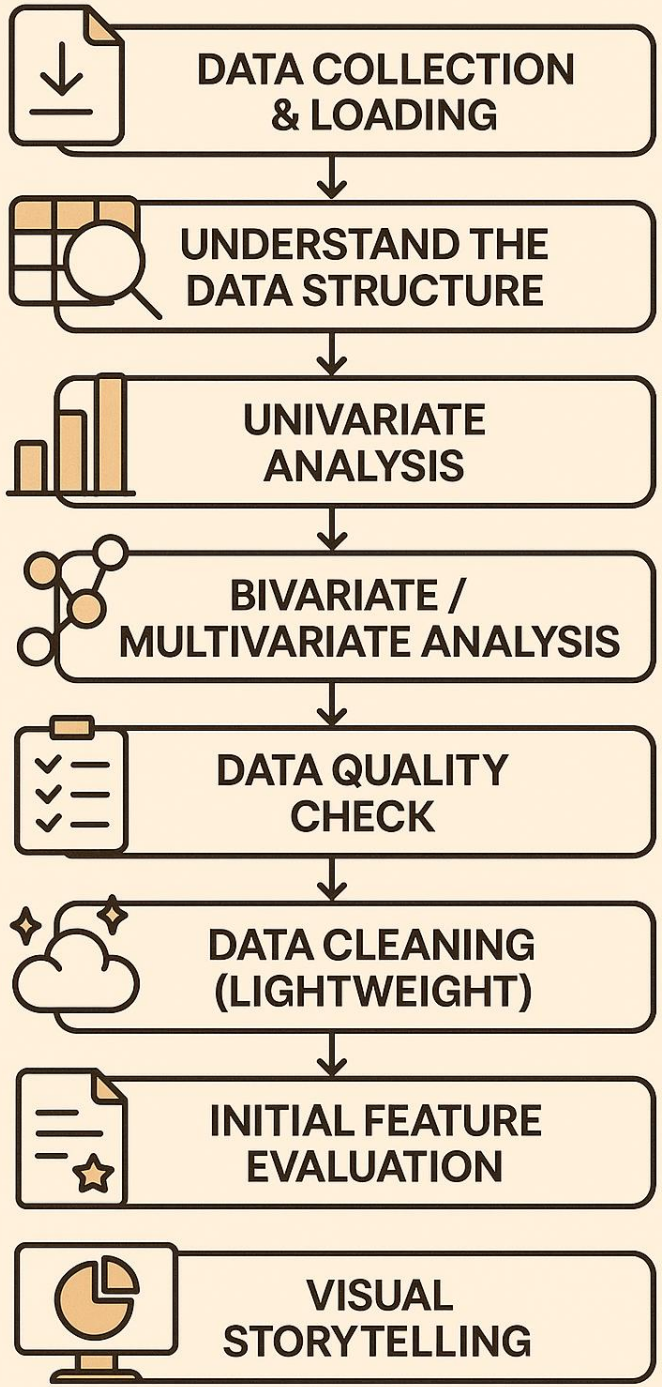


A2025 Building a Data Science Solution

Exploratory Data Analysis - Structure

Exploratory Data Analysis typically follows a structured, step-by-step approach that helps you explore, understand, and prepare your data for deeper analysis or modeling. Here's how you should think about it:

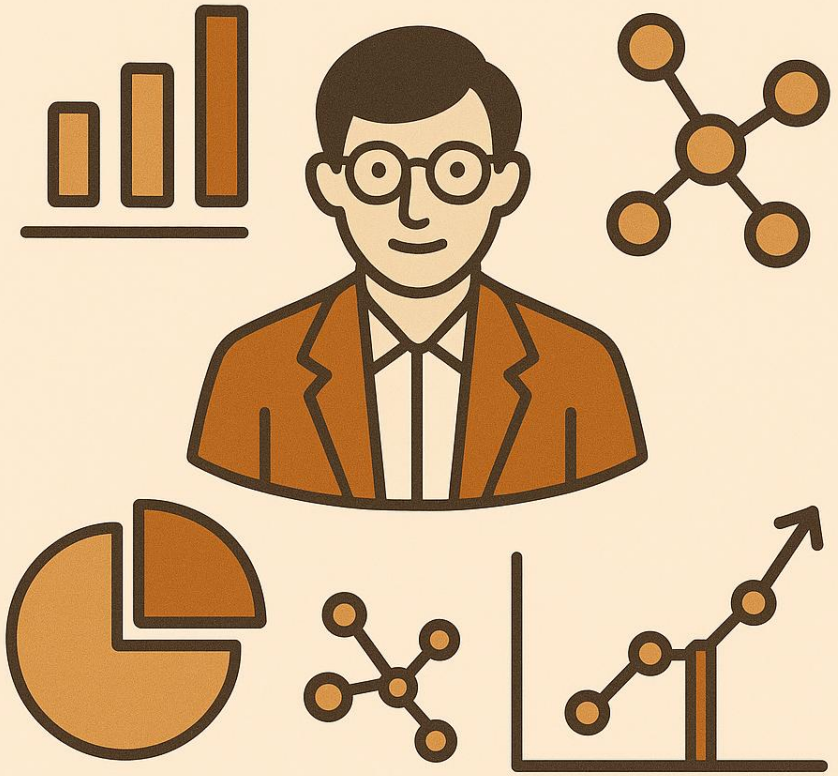
- Data Collection & Loading
- Understand the Data Structure
- Univariate Analysis
- Bivariate/Multivariate Analysis
- Data Quality Check
- Data Cleaning
- Initial Feature Evaluation
- Visual Storytelling



A2025 Building a Data Science Solution

Exploratory Data Analysis – Case Studies

- Age Distribution Across Treatment Groups
- Adverse Events by Treatment Group
- Outcome Score vs Follow-Up Blood Pressure
- Cholesterol Levels by Visit Status
- Gender-wise Distribution of Outcome Scores



A2025 Building a Data Science Solution

Feature Engineering

Feature Engineering is the process of creating, transforming, or selecting the right input variables (features) that can be used by machine learning models to improve their performance.

In simpler terms, it's about preparing the data in a smart way so that the model can learn better and make more accurate predictions.

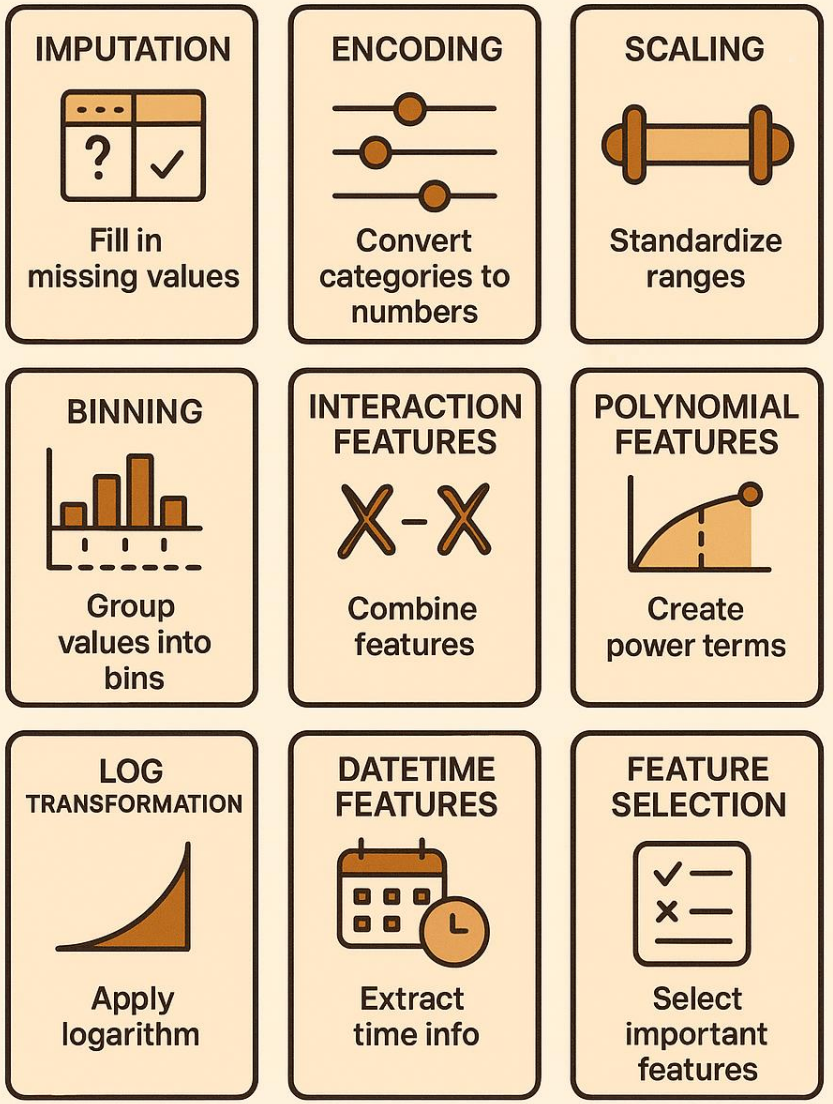
Why Should a Data Scientist Master Feature Engineering?

- Enhances Model Performance
- Bridges Data and Algorithms
- Improves Interpretability
- Helps Handle Data Issues
- Critical for Real-World Use Cases

A2025 Building a Data Science Solution

Feature Engineering – Feature Creation/improvement Techniques

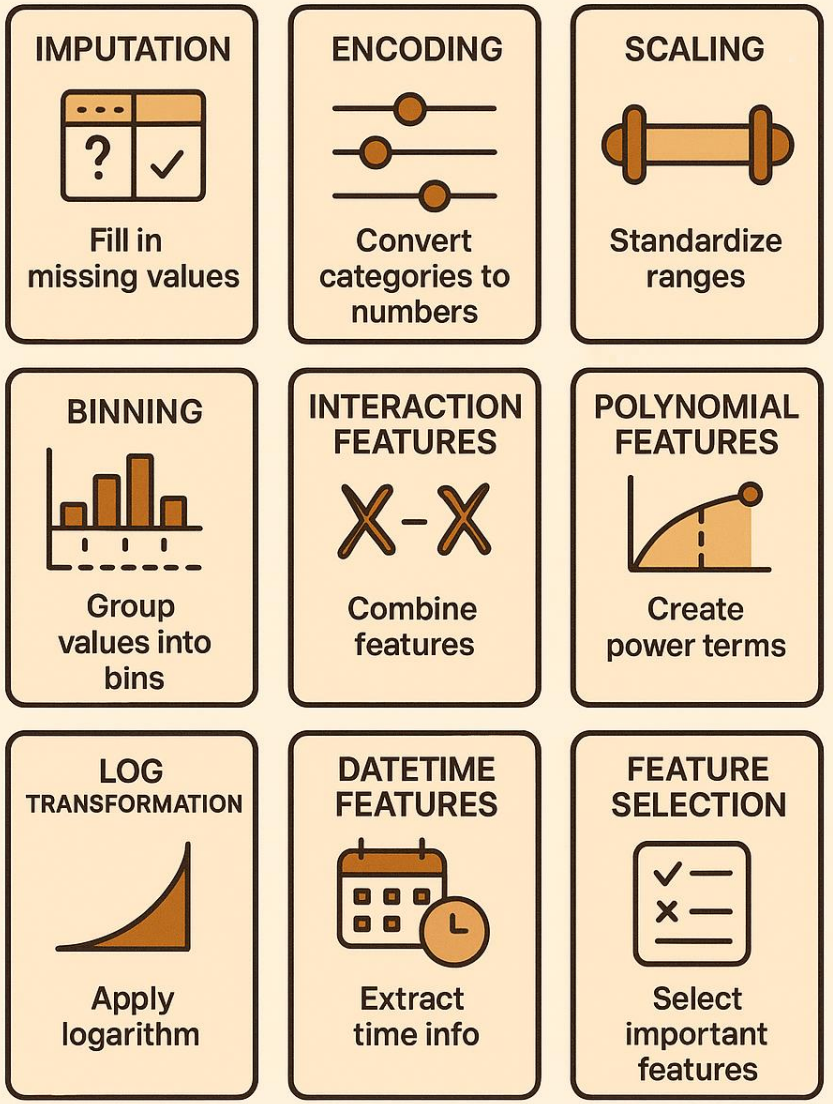
Technique	Real-World Use Case Example
Imputation	Filling missing blood pressure values in patient records
Label Encoding	Education Level: High School < Bachelor < Master < PhD
One-Hot Encoding	Encoding customer location for churn prediction
Binary Encoding	Converting ZIP codes with hundreds of levels
Frequency Encoding	Encoding car model names in insurance fraud prediction
Target Encoding	Mean loan default rate per employer
Scaling & Normalization	Equalizing salary and number of projects before modeling
Binning	Age grouping for marketing segmentation
Interaction Features	Spending per visit, BMI from height & weight
Polynomial Features	Temperature^2 in energy demand forecasting
Log Transformation	Skewed income data in housing price prediction
Datetime Feature Extraction	Extracting month/weekday from order dates
Text Feature Extraction	TF-IDF for customer review classification
Feature Selection	Dropping ID columns or unrelated demographic details



A2025 Building a Data Science Solution

Feature Engineering – Feature Selection Techniques

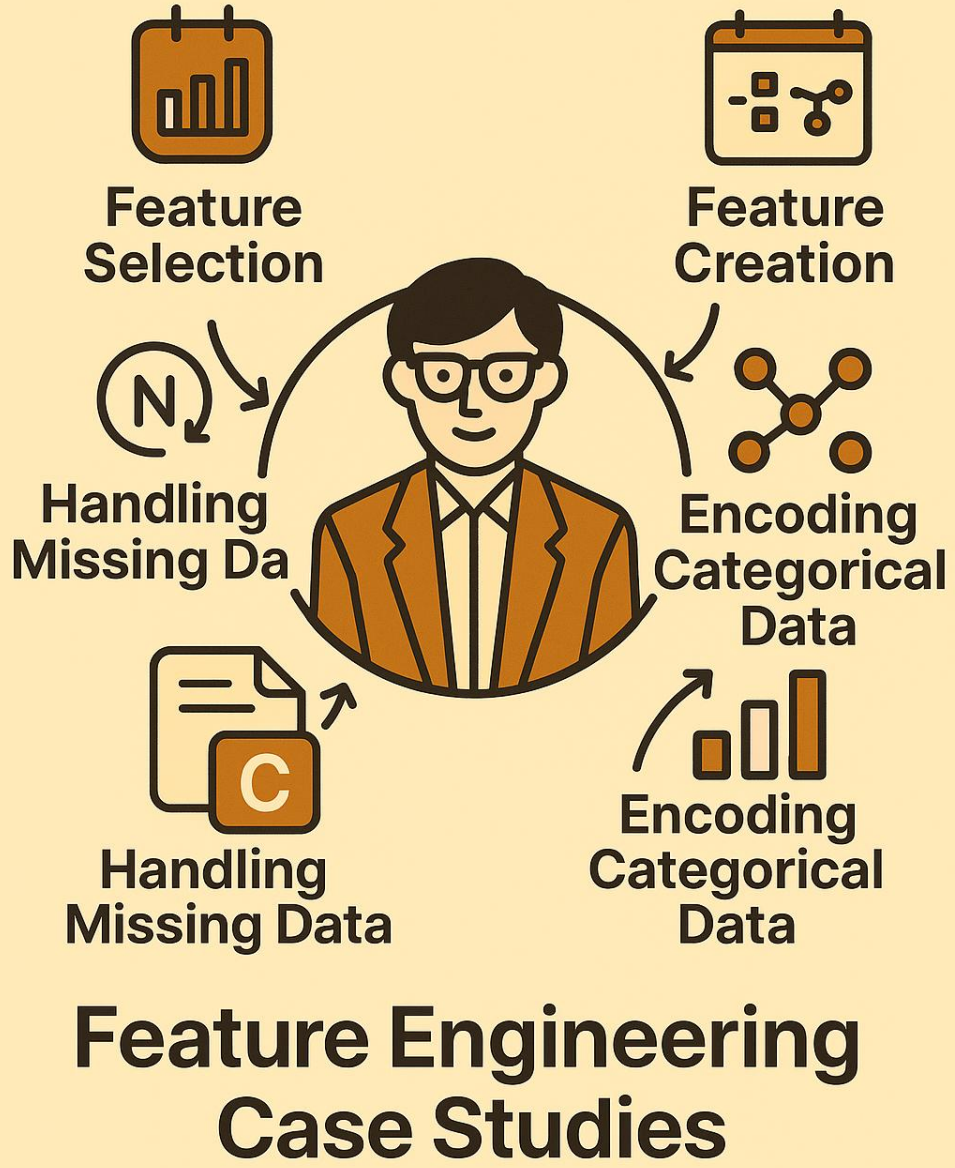
Technique	Real-World Use Case Example
Imputation	Filling missing blood pressure values in patient records
Label Encoding	Education Level: High School < Bachelor < Master < PhD
One-Hot Encoding	Encoding customer location for churn prediction
Binary Encoding	Converting ZIP codes with hundreds of levels
Frequency Encoding	Encoding car model names in insurance fraud prediction
Target Encoding	Mean loan default rate per employer
Scaling & Normalization	Equalizing salary and number of projects before modeling
Binning	Age grouping for marketing segmentation
Interaction Features	Spending per visit, BMI from height & weight
Polynomial Features	Temperature^2 in energy demand forecasting
Log Transformation	Skewed income data in housing price prediction
Datetime Feature Extraction	Extracting month/weekday from order dates
Text Feature Extraction	TF-IDF for customer review classification
Feature Selection	Dropping ID columns or unrelated demographic details



A2025 Building a Data Science Solution

Feature Engineering – Case Studies

- Imputing Missing Ratings
- Encoding Gender and Premium Membership
- Creating Recency Feature from Last Login
- Binning Total Spend into Customer Segments
- Frequency Encoding of Preferred Category
- Extracting Time-based Features from SignUpDate
- Log Transformation of Total Spend
- Interaction Feature: Spend per Order
- Device and Payment Method Encoding
- Feature Selection Using Correlation Analysis

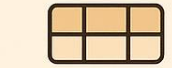
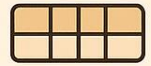
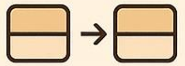
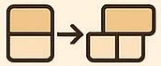
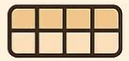


A2025 Building a Data Science Solution

Feature Engineering – Dataset Preparation for Model Development

Method	Ideal For	Pros	Cons
Holdout	Large datasets, baseline models	Simple, quick	May be unstable
Train-Validation-Test	Model selection + final eval	Easy tuning and test separation	Less data for training
K-Fold CV	Medium datasets	More robust, multiple validation	Slow, can overfit on validation
Stratified K-Fold	Imbalanced classification	Keeps class balance	Classification only
Leave-One-Out	Very small datasets	Maximum data usage	Very slow
Time Series Split	Forecasting, temporal models	Time-aware, avoids leakage	Needs careful config
Group K-Fold	Sessions/users/groups	Prevents group leakage	Risk of imbalance
Nested CV	Research-grade performance checks	Unbiased tuning + evaluation	Very heavy computational load

Dataset spliitting methods

Method	Purpose	Cons
 Holdout	Simple train-test split	May be unstable
 Train-Validation--Test	With validation set	Good for tuning
 Stratified K-Fold	Imbalanced classes	Computationally costly
 Leave-One-Out	Maximum data use	Very slow
 Time Series Split	Chronological splits	Respects time order
 Group K-Fold	Prevents leakage	Needs config
 Nested CV	Unbiased	Very intensive

A2025 Building a Data Science Solution

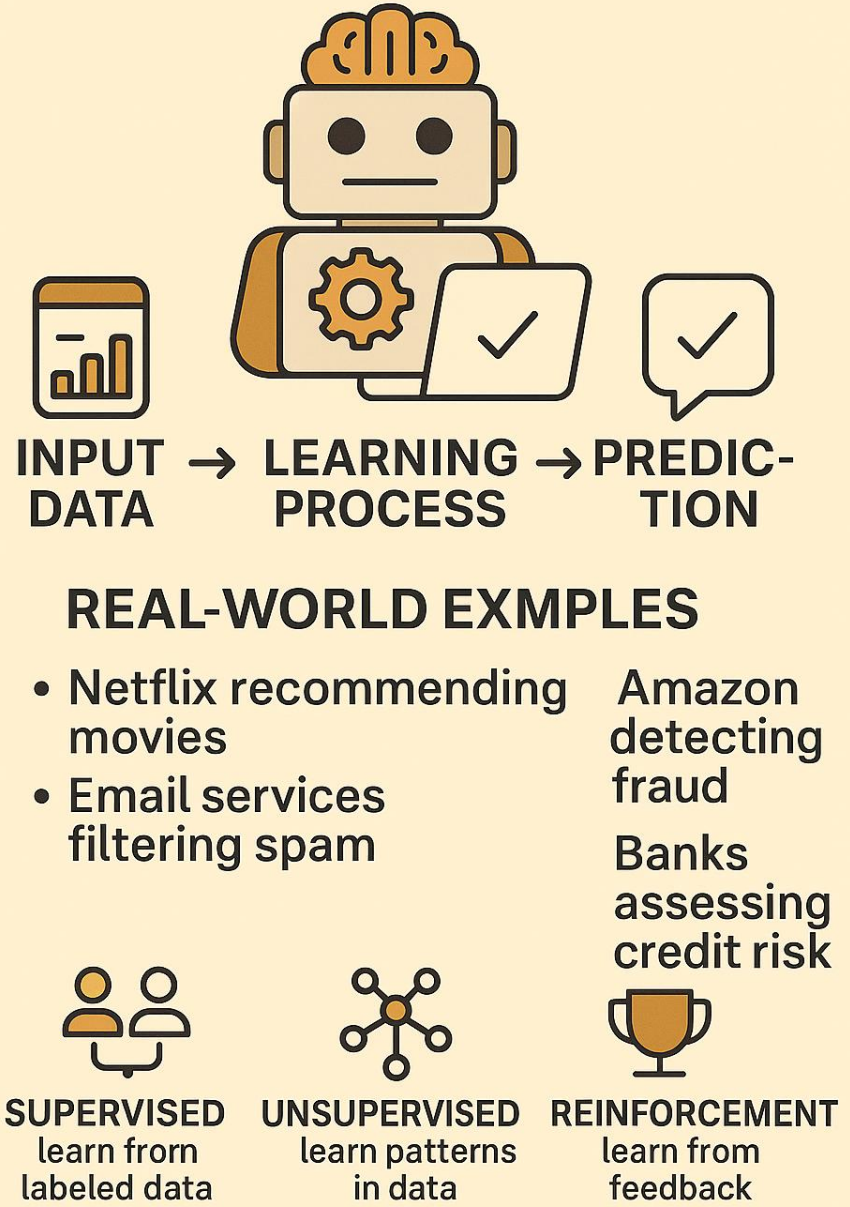
Introduction to Machine Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that focuses on building systems that can learn from data, identify patterns, and make decisions with minimal human intervention.

Rather than being explicitly programmed for every task, a machine learning system learns rules and relationships from examples (data), and uses those to predict outcomes or classify inputs.

How ML Works

- Input Data - Raw data like images, numbers, text, or transactions.
- Learning Process - Algorithms find patterns or relationships in the data.
- Model Building - A model is created that captures these patterns.
- Model Evaluation – Validation of Trained model
- Prediction/Inference - The model is used to make predictions on new data.



A2025 Building a Data Science Solution

Introduction to Machine Learning – Machine Learning Model Evaluation

Model Performance tells us how good or bad a machine learning model is at making predictions. It's like checking a cricket player's form: Are they scoring runs consistently? Similarly, we check if the model is accurate, reliable, and useful for solving the problem at hand.

⚙️ How is Performance Measured?

🧠 For Classification (Yes/No, Fraud/Not Fraud, etc.):

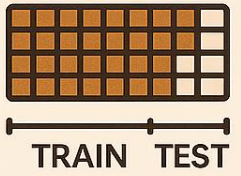
Metric	What It Means
Accuracy	% of correct predictions overall
Precision	Out of predicted positives, how many were correct
Recall	Out of actual positives, how many we caught
F1-Score	Balance between Precision and Recall

🔢 For Regression (Price, Score, Demand, etc.):

Metric	What It Means
MAE (Mean Absolute Error)	Average of absolute prediction errors
MSE (Mean Squared Error)	Penalizes big mistakes more heavily
RMSE	Square root of MSE, same unit as prediction
R^2 Score	How well the model explains the variation

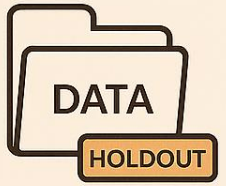
MODEL VALIDATION

Checking how well a machine learning model performs on new data.



Train/Test Split

Cross-Validation



Holdout Set

A2025 Building a Data Science Solution

Introduction to Machine Learning – Fairness in AI

Fairness means making sure AI systems treat everyone equitably, without bias or discrimination.

Imagine a loan approval model that denies more women or certain communities, even though they qualify. That's unfair AI — and it can happen if the training data contains historical biases.

🧭 **Why It Matters**:

- AI decisions impact real lives (jobs, loans, healthcare)
- Unfair systems can amplify social inequalities
- It's critical for trust, reputation, and legal compliance

🧱 **Common Causes of Unfair AI:**

- Biased data (e.g., more male resumes in training)
- Skewed labeling (e.g., old decisions based on biased policies)
- Design assumptions that don't consider diversity

FAIRNESS

WHAT IS FAIRNESS?

Ensuring a model's outcomes are unbiased and free from discrimination.

WHY IS FAIRNESS USEFUL FOR A DATA SCIENTISTT?

- Promotes ethical data practices
- Creates more inclusive outcomes
- Helps build trust in the model

A2025 Building a Data Science Solution

Introduction to Machine Learning – Explainability in AI (XAI)

Explainability is the ability to understand and explain why the AI made a decision.

Think of it like this: If an AI denies you a credit card, you should be able to ask: “Why?” — and get a human-understandable answer.

🎯 Why It Matters:

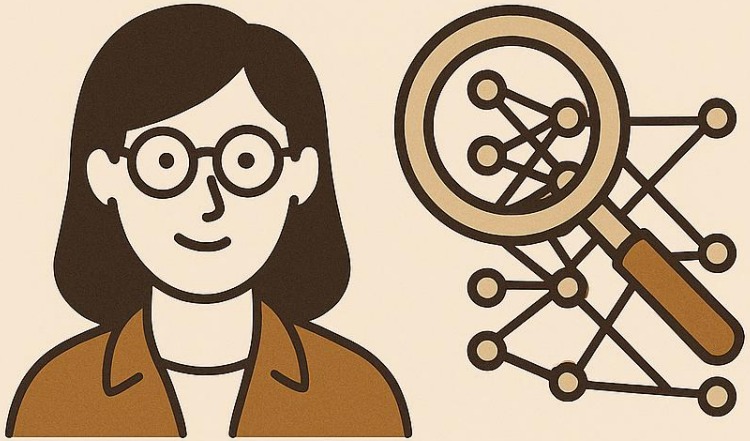
- Builds trust in AI systems
- Helps spot mistakes and biases
- Required in regulated industries (e.g., finance, healthcare)Essential for ethical and responsible AI

🛠️ Tools & Techniques:

Method	What It Does
LIME	Explains individual predictions using local examples
SHAP	Shows how much each feature contributed to a prediction
Decision Trees	Naturally interpretable (if shallow)
Feature Importance Charts	Helps visualize which variables matter most

EXPLAINABILITIY

The ability to understand and interpret how a model makes decisions.



WHY IT’S USEFUL

- Builds trust in the model
- Helps identify model limitations

A2025 Building a Data Science Solution

Agents for Data Science Development - Demonstration

Thank You

M: +91 88846 92929

E: anant.awasthi@outlook.com