Practical Approach for Developing AIML Projects

Units for Discussion

Unit - 1	Foundations of AIML
Unit - 2	Identifying & Framing Problems
Unit - 3	Model Building
Unit - 4	Evaluation and Deployment & Monitoring

DISCLAIMER

The content is curated from online/offline resources and used for educational purpose only.

Learning Objectives

- Understand the Complete AI/ML Project Lifecycle
- From identifying real-world problems to deploying and maintaining models.
- Gain Hands-On Insight into Data Preparation and Model Building
- Learn the importance of data collection, cleaning, feature engineering, and model selection.
- Apply Best Practices Using Tools and Frameworks Common in Industry
- Familiarize with Python, Jupyter, Colab, Scikit-learn, TensorFlow, MLflow, and deployment tools.

Source : www.freepik.com/

Learning Objectives

- Explore Real-World Applications in the Indian Context
- Examine use cases in agriculture, healthcare, fintech, and language processing tailored for India.
- Develop Critical Thinking Around Ethical and Practical Challenges in AI
- Understand fairness, privacy, bias, and compliance concerns relevant to the Indian market.

Source : www.freepik.com/

"You can have data without information,
but you cannot have information without data."

- Daniel Keys Moran

Source : www.freepik.com/

AI/ML: Shaping India's Future

Government Support and Innovation

India's government initiatives and a thriving startup ecosystem are accelerating AI and ML adoption across industries.

Transformation in Key Sectors

AI-powered solutions are revolutionizing agriculture, healthcare, and fintech by improving efficiency and service delivery.

Impact on Daily Life and Economy

AI and ML are enhancing daily life and fueling India's economic growth through smarter services and automation.

1. Global Trends in AI/ML

AI is transforming industries like **healthcare, finance, education, manufacturing**, and **transportation**.

Massive growth in AI adoption:

- Global AI market expected to reach **$1.8 trillion by 2030**.
- Rise of **Generative AI** (e.g., ChatGPT, DALL·E, Copilot) reshaping content creation, coding, and design.

Top technologies:

- **Computer Vision**, **Natural Language Processing (NLP)**, **Reinforcement Learning**, **Edge AI**.
- Countries like the **US, China, UK** are investing heavily in AI R&D and policy frameworks.

2. AI/ML Demand

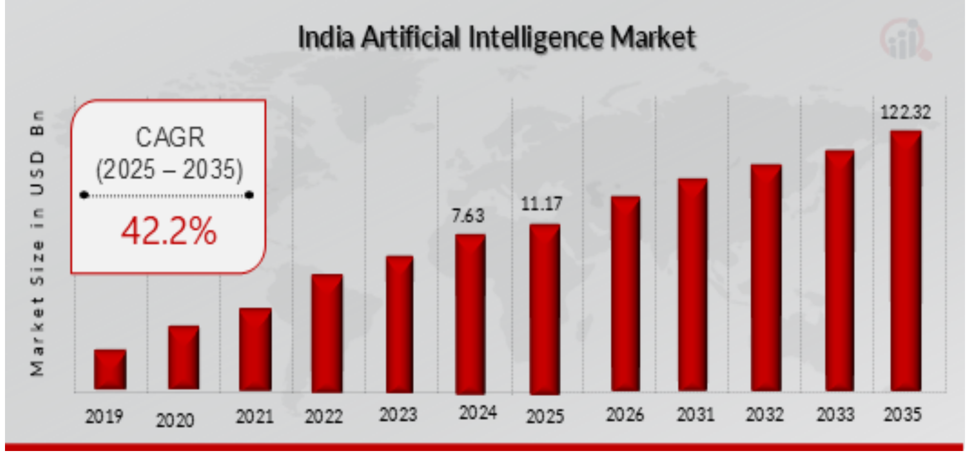
India is becoming a **global AI hub**, with talent, data, and a large digital economy.

Government initiatives:

- **National AI Mission** (NITI Aayog): Focus on healthcare, education, agriculture, smart cities.
- **IndiaAI portal**, **AI for All strategy**, and **AI Research Centres**.

Massive digital infrastructure:

- Aadhaar, UPI, DigiLocker, CoWIN – all AI-compatible platforms.



Booming startup ecosystem:

- Over **1,500+ AI startups** in India (2024), including **Gnani.ai, CropIn, Niramai**, and **SigTuple**.

Job growth:
AI/ML roles grew **30%+ YOY**; demand for data scientists, ML engineers, and AI product managers rising rapidly.

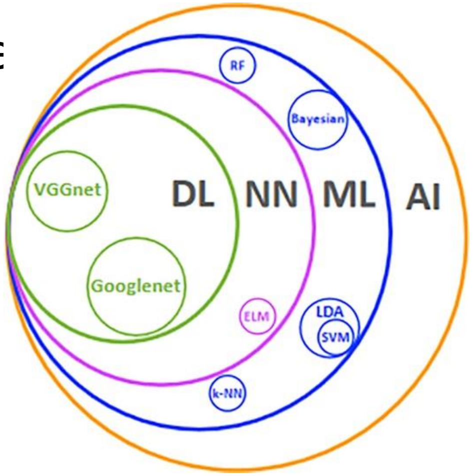
Source: https://www.marketresearchfuture.com/reports/india-artificial-intelligence-market-21411

3. What is AI, ML, and Deep Learning?

3.1 Artificial Intelligence (AI)

Definition:

- AI is the ability of machines to perform tasks that normally require human intelligence



Examples:

- Google Maps suggesting faster routes
- Chatbots like Alexa or Siri
- Facial recognition to unlock your phone

3.2 Machine Learning (ML)

Definition:

- ML is a subset of AI where machines **learn from data** and **improve automatically** without being explicitly programmed.

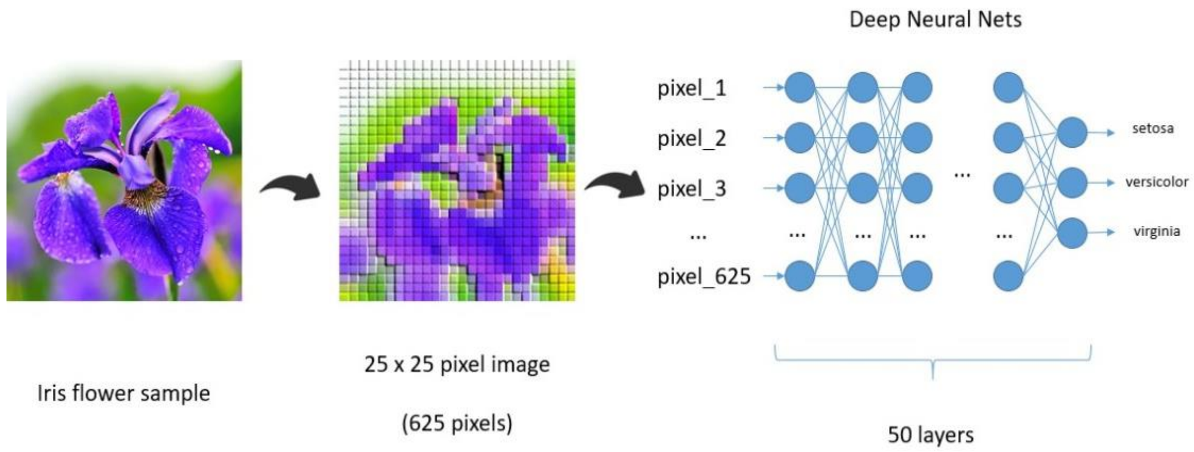
Examples:

- Netflix recommending shows
- Spam detection in Gmail
- Predicting student exam scores based on study habits

3.3 Deep Learning (DL)

Definition:

- Deep Learning is a specialized type of ML that uses **artificial neural networks** to mimic the human brain.



Examples:

- Voice assistants understanding Indian languages
- Self-driving cars identifying road signs
- Detecting diseases from X-ray or MRI images

Key Difference Summary

Topic	Concept	How it learns	Use cases
AI	Broad concept	Rules or data	Any intelligent task
ML	Subset of AI	Data	Predictions, recommendations
Deep Learning	Subset of ML	Large datasets	Images, speech, text understanding

4. The AI/ML Project Lifecycle

Problem Understanding	Data Collection & Preparation
Model Building	Model Deployment
Monitoring & Maintenance	Feedback & Iteration

5. Types of Problems Solved by AI/ML

Classification

- Categorize data into Groups
  - Email
  - Bank Customer
  - Xray

Regression

- Predict numerical value using i/p
  - Crop yield
  - House prices
  - Forecasting electricity bill

Clustering

- Group similar data points
  - Customer segmentation
  - Student segmentation on learning style
  - Identify disease patterns

5. Types of Problems Solved by AI/ML

Recommendation Systems

- Suggest item based on preference
  - Amazon product suggestion
  - Movies on Netflix or Hotstar
  - Career or Course recommendation

NLP

- Making machines to understand human language
  - Chatbots in regional languages
  - Sentiment analysis
  - Auto Translation

Computer Vision

- Machine to see & understand images/videos
  - Detecting fire/pothole/flood using drones
  - Scanning handwritten notes
  - Facial recognition

6. Why a Structured Approach Matters

Bias in Data or Model → Overfitting the model → Misalignment with Business needs → Privacy & Ethical risks → Wasted Time and Resources

7. Use Case Overview: Predicting Crop Yield with AI

Problem Statement

- Can we help farmers predict how much crop they will produce this season using technology?

Why It Matters?

- Agriculture is a major livelihood in India.
- Crop yield prediction helps with:
  - **Planning harvest & storage**
  - **Loan and insurance decisions**
  - **Reducing food waste and supply chain gaps**

Approach

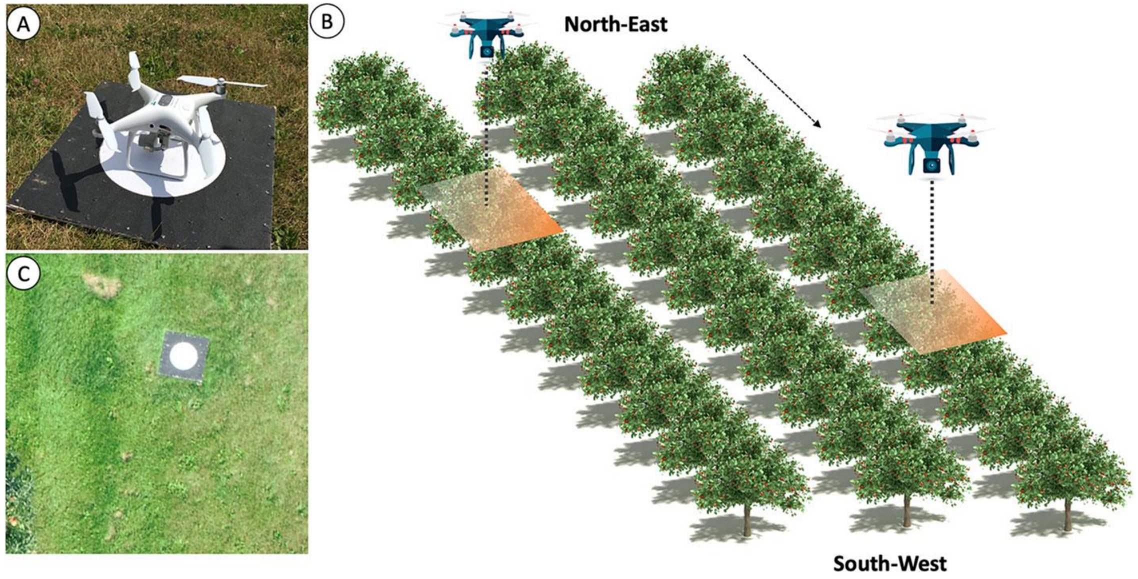
1.Data Collection:

1. Satellite images (NDVI, rainfall maps)	2. Soil data from sensors
3. Historical crop yields	4. Weather forecasts

7. Use Case Overview: Predicting Crop Yield with AI

1. **Modeling:**
    1. Train regression models (like Random Forest or XGBoost)
    2. Predict yield in kg/hectare for different crops like rice, wheat, sugarcane

2. **Output:**
    1. Visual dashboard for farmers or agri-officers
    2. Alerts via SMS in regional languages



Tools & Tech

- Python, Pandas, Scikit-learn
- Google Earth Engine or Bhuvan (ISRO satellite data)
- Dashboards using Streamlit

8. Problem Understanding/Ideation – Spotting Opportunities

Problem:

- **Healthcare Diagnostics:**

  Can we detect early signs of diseases like TB or anemia using smartphone apps?

- **Traffic Prediction:**

  Can we use past traffic data to reduce commute time in cities like Bengaluru or Hyderabad?

- **Language Translation:**

  Can AI help translate school textbooks into regional languages like Marathi or Tamil?

8.1 Business vs. Research Questions

Understand what kind of question you're answering — and why it matters.

1. Research Question (Academic Focus):

Can a CNN model detect diabetic retinopathy from retina scans?

2. Business/Impact Question (Real-world focus):

Can we reduce misdiagnosed diabetes cases by 30% using AI in tier-2 hospitals?

Why This Matters:

1. Business goals are **actionable, measurable**, and often tied to real outcomes.
2. Research questions focus on **novelty**, **accuracy**, and **understanding**.

Tip:

In competitions or real projects, always **connect your ML solution to a real-world impact.**

8.2 Feasibility Check

Just because it's a good idea doesn't mean it's ready for AI.

Check These First:

1.Data Availability

1. Do you have enough labeled and relevant data?
2. Example: Crop yield data from government sites like data.gov.in

2.Compute Power

1. Can your system (or cloud tools) handle the model training?
2. Tip: Use Google Colab if you lack a GPU system

3.Domain Expertise

1. Can someone explain what the data means?
2. Example: A doctor for a medical ML project or a teacher for an education app

8.3 Use Case Examples

Use Case 1: Automated Vehicle Classification at Toll Booths

1. Goal: Identify vehicle type (car, truck, bus) using AI for correct toll collection.
2. Tools: Computer vision + real-time camera feed.
3. Impact: Reduce toll booth errors and manual labor.

Use Case 2: Regional Language Chatbots

1. Goal: Build chatbots in Hindi, Bengali, Tamil, etc., for basic customer support.
2. Data: Local language Q&A, past support tickets.
3. Real Example: Chatbots for railway queries or hospital appointments.

9. Defining Success Metrics

Common Technical Metrics:

- ✓ **Accuracy**: % of correct predictions
- ✓ **Precision & Recall**: How good is your model at finding the right items?
- ✓ **F1 Score**: Balance between precision and recall
- ✓ **RMSE (Root Mean Squared Error)**: Used in prediction (regression) problems

Business or Project Metrics:

- ✓ Reduction in errors (e.g., 20% fewer fraud cases)
- ✓ Increased speed (e.g., chatbot replies 40% faster)
- ✓ Cost savings (e.g., ₹2 lakh saved per month by automation)

Note:

Always align your **technical metric** with a **real-world impact.**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

10. Common Mistakes in Problem Framing

1. Overambitious Scope

- Example: "Let's build an AI system that can diagnose all diseases" — unrealistic for a semester project

2. Misaligned Goals

- Solving a technical challenge that doesn't help anyone or isn't deployable

3. Ethical Blind Spots

- Using personal data without consent
- Ignoring bias in training data (e.g., English-only inputs in a multi-language society)

A well-framed problem is 50% of the solution in AI/ML!

11. What Makes Good Data?

Qualities of Good Data:

1. Quality – Clean, consistent, and accurate

Example: Crop data with verified harvest dates

2. Quantity – Enough samples to train and test the model

More data = better generalization

3. Relevance – Fits your specific problem

Don't use global weather data to predict Indian monsoons!

4. Diversity – Represents all use cases

Example: Include rural and urban dialects in a chatbot project

"Garbage in = Garbage out" - your model is only as good as your data.

12. Data Sources in Indian Projects

Government & Open Platforms:

- data.gov.in: Crop yields, health records, school data
- Bhuvan (ISRO): Satellite imagery
- PM-KISAN, Aarogya Setu (public data in parts)

Public & Research Datasets:

- Kaggle (many Indian-specific datasets)
- AI4Bharat: Language datasets for NLP
- UCI Machine Learning Repository (generic, still useful)

Private Sector & Startups:

- Startups like **CropIn**, **Razorpay**, and **Niramai** release anonymized datasets
- Telemetry data from mobile apps and sensors

Image Courtesy: https://blog.nextgrowthlabs.com/what-is-data-source

13. Data Collection Methods

Surveys (Manual Collection)

- Use tools like **Google Forms**, **KoboToolbox**, **SurveyMonkey**

Examples:

  - Farmer income surveys
  - Student performance tracking in rural schools

Sensor-Based Collection

- IoT devices used in agriculture, weather, health, traffic

Real-world examples:

  - Soil moisture sensors in Punjab
  - Smart meters in cities like Delhi for electricity usage
  - Air quality sensors in Mumbai

Image courtesy: https://www.vectorstock.com/royalty-free-vector/data-collection-icon-market-research-information-vector-31718556

Data Collection Methods

Web/Data Scraping

Python libraries: **BeautifulSoup**, **Selenium**, **Scrapy**

Use for collecting:

1. Product prices
2. Social media trends
3. Government reports

Note:

Follow **Data Protection Bill** and avoid scraping private or copyrighted data without permission.

14. Data Cleaning & Preprocessing

Cleaning:

- Remove **null/missing values** (or fill them with means/medians)
- Handle duplicates
- Fix format issues (dates, currencies, units)

Preprocessing:

- **Normalization/Standardization** – Scale values for ML models
- **Encoding Categorical Data** – One-hot or label encoding
- **Anonymization** – Mask or remove personally identifiable data
  - Example: Remove Aadhaar numbers or phone numbers from health records

Clean data improves accuracy more than complex models do!

15. Exploratory Data Analysis (EDA)

- Exploratory Data Analysis (EDA) is a technique used to analyze and summarize the main characteristics of a dataset, often through visual methods. It helps identify patterns, spot anomalies, and understand relationships between variables before formal modeling or hypothesis testing.

- Understand data patterns, distributions, and spot issues

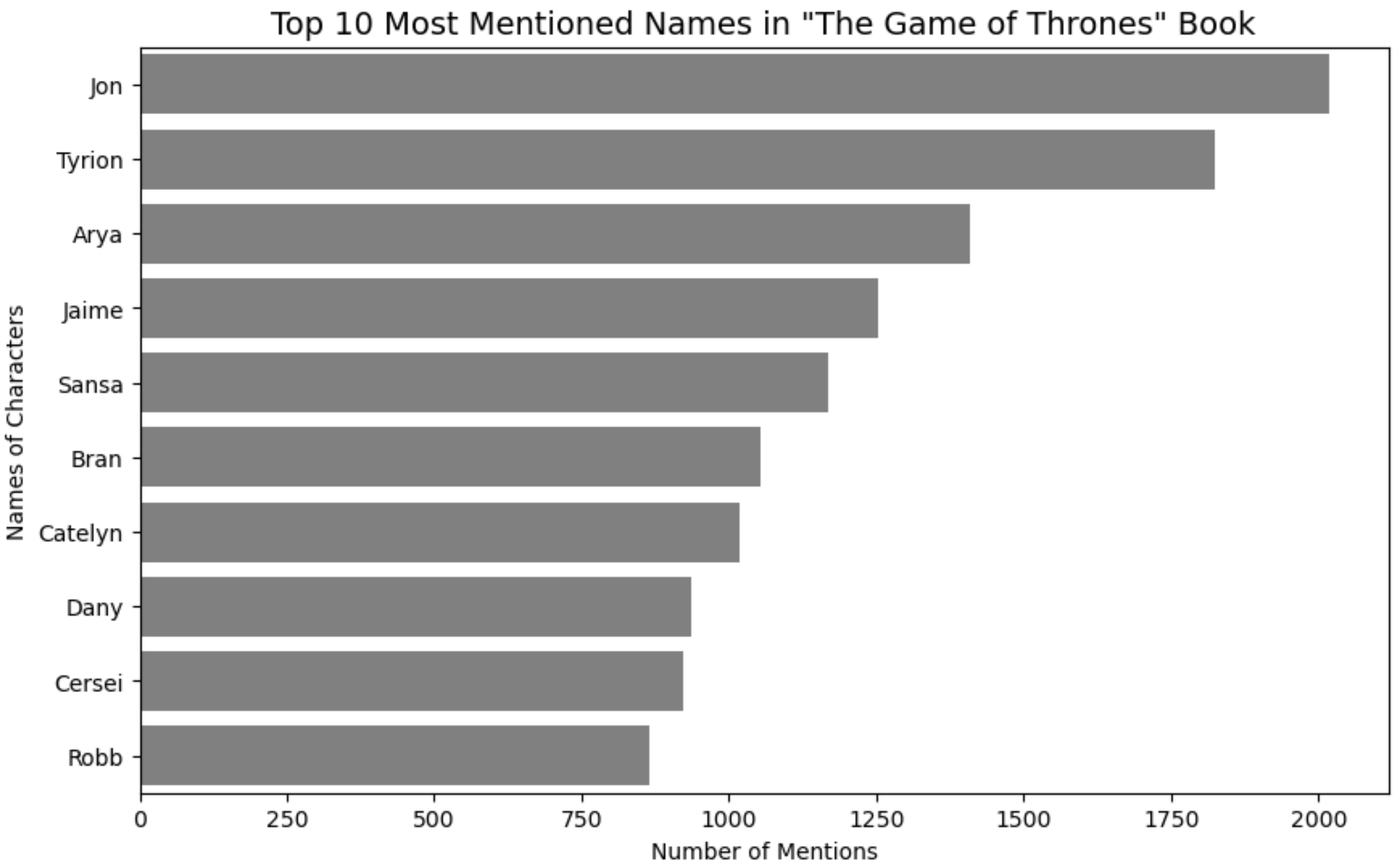
  Types of EDA

  1. Univariate Analysis – e.g., Histogram, Bar plot, Pie chart, Boxplot
  2. Bivariate Analysis – e.g., Scatterplot, Lollipop
  3. Multivariate Analysis – e.g., Bubble, Heatmaps, 3D Scatter

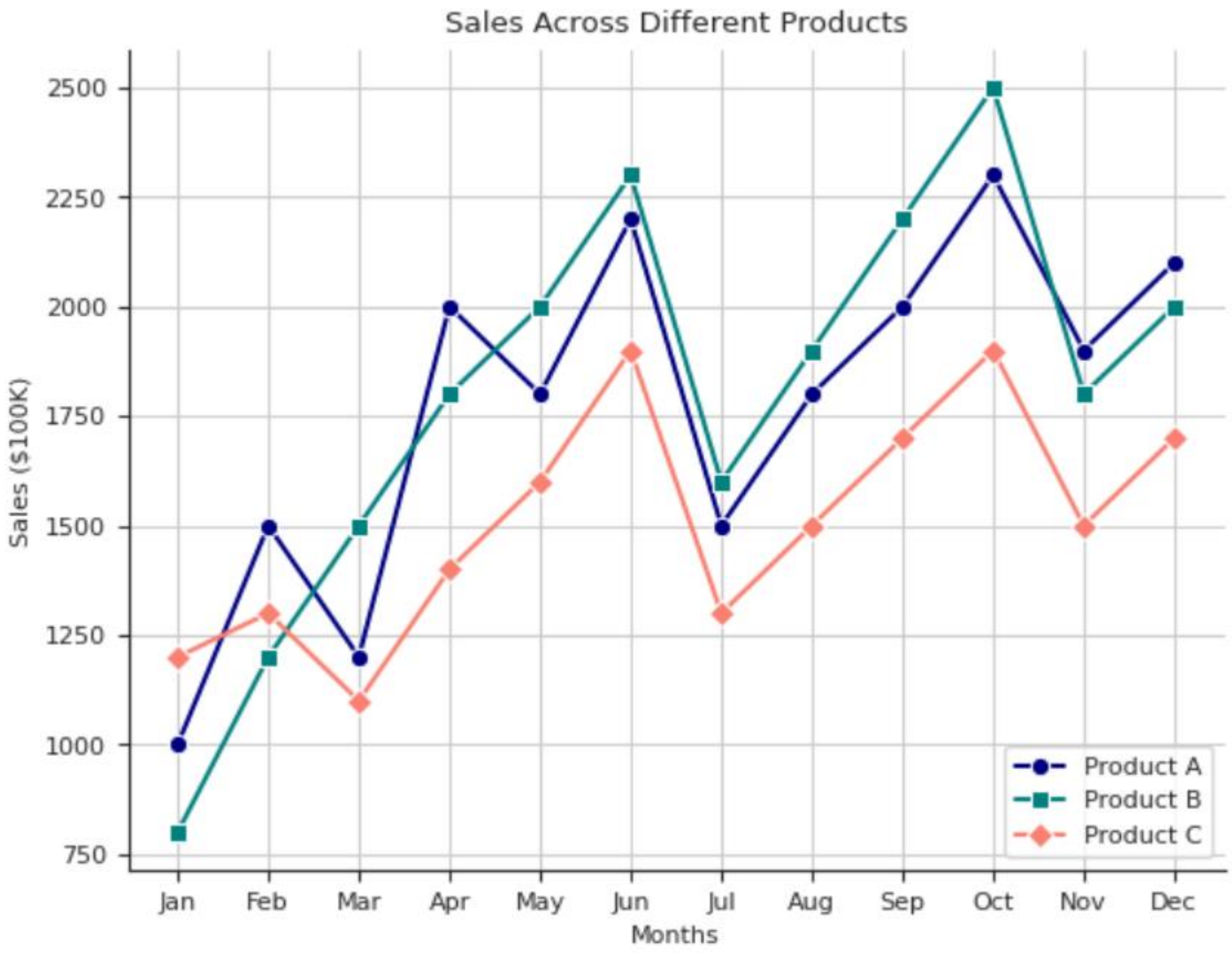
Key Tools (Python):

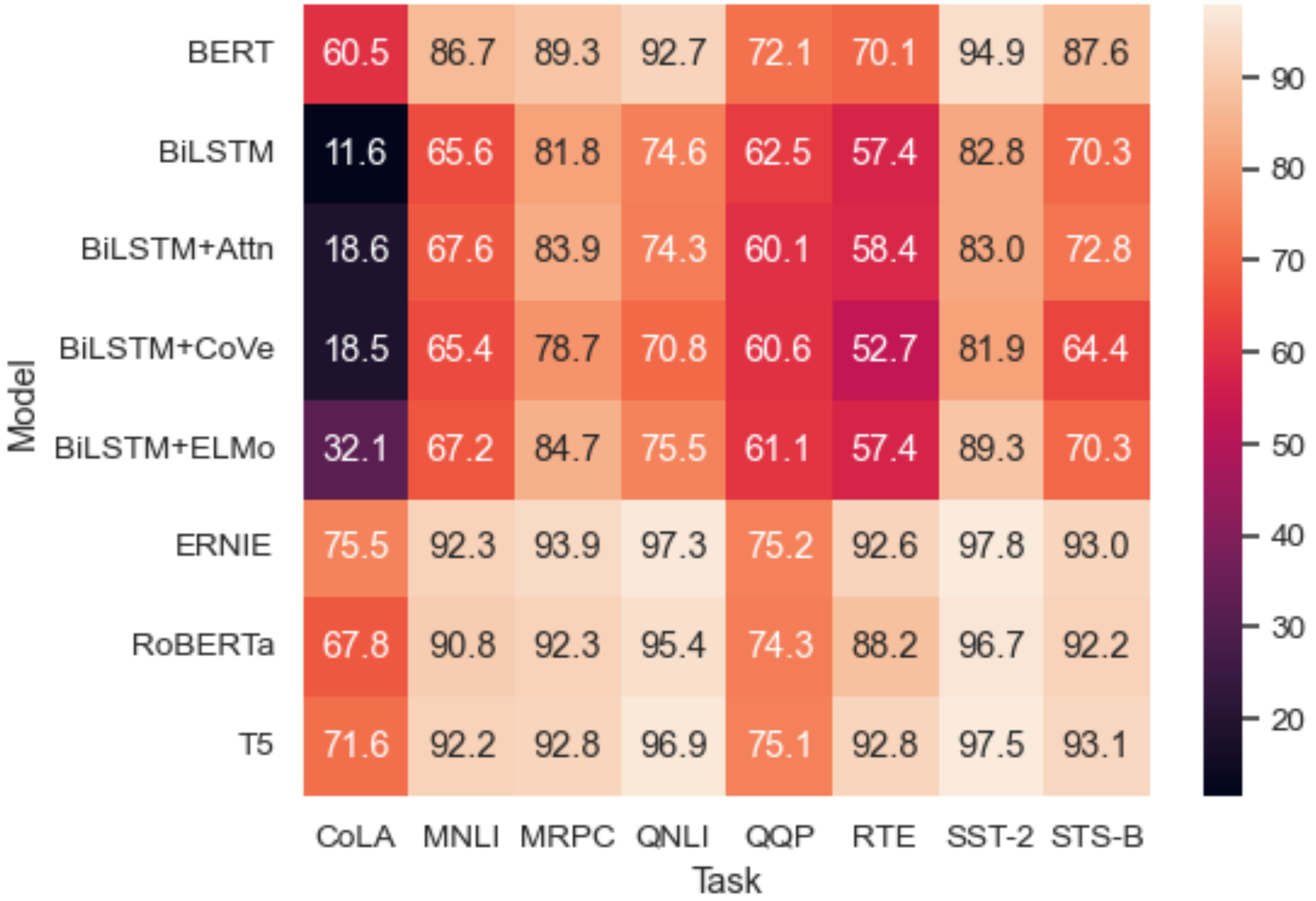
- **Pandas**: `.describe()`, `.info()`
- **Matplotlib / Seaborn**: Histograms, box plots, correlation heatmaps
- **Plotly**: Interactive dashboards
- **Google Colab**: Free cloud platform for coding and visualizing

Example Visuals for Student Projects



Courtesy: https://obinnanweke.com/game-of-thrones-book-analysis/


Sales Across Different Products
Sales ($100K)
2500
2250
2000
1750
1500
1250
1000
750
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Months
Product A
Product B
Product C



Courtesy: https://seaborn.pydata.org/generated/seaborn.heatmap.html

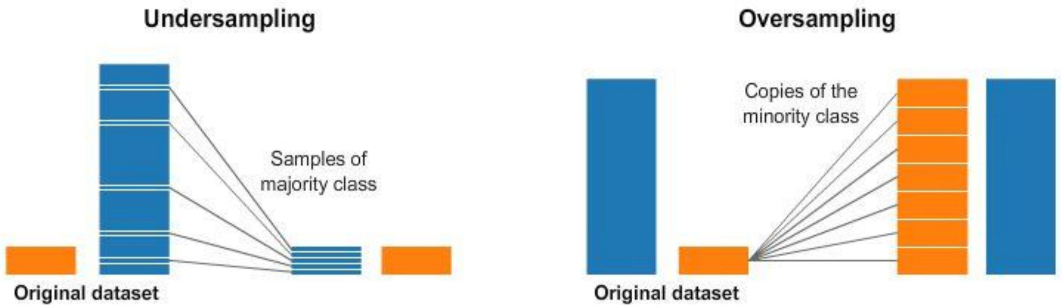
16. Dealing with Bias & Imbalanced Data

Bias in Data:

- Example: 90% of health data is from male patients — results may not generalize to women.
- Use techniques like:
  - Re-sampling (oversample minority class or undersample majority)
  - Fairness-aware algorithms (e.g., IBM AI Fairness 360)

Imbalanced Classes:

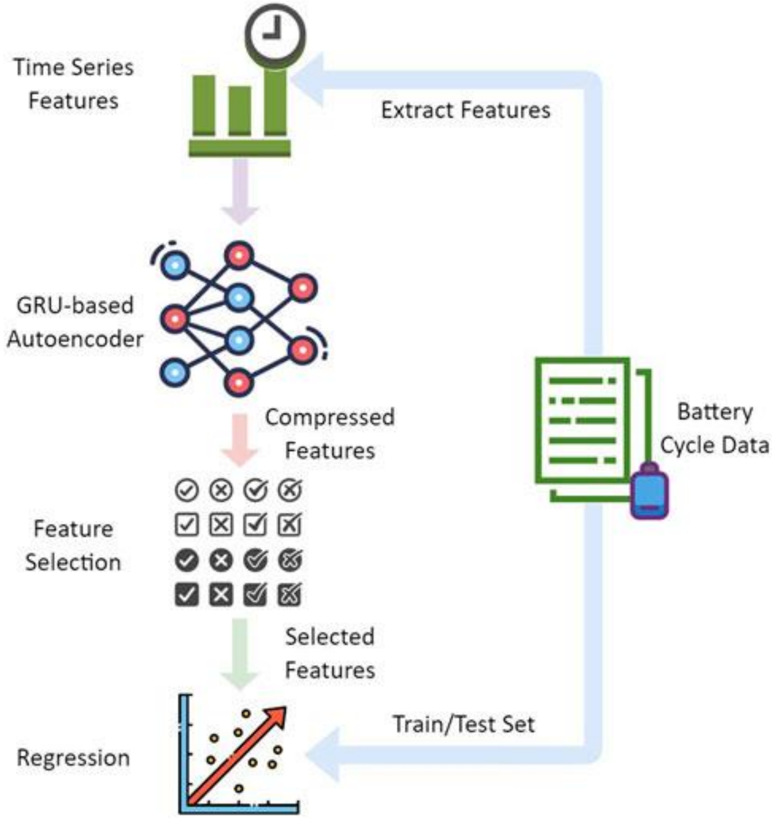
- One class dominates — e.g., 95% of users are active, 5% inactive
- Metrics like **accuracy** become misleading — use **F1-score** or **AUC**



Balanced data = fairer, more accurate models

17. Feature Engineering

- Feature engineering in machine learning is the process of transforming raw data into a format that is more suitable for machine learning algorithms to learn from and make accurate predictions. It involves selecting, modifying, and creating new variables (features) from the existing dataset to enhance the model's ability to capture underlying patterns



Key aspects of feature engineering include:

1. Feature Selection
2. Feature Transformation
   - Handling Missing Values
   - Encoding
   - Scaling
3. Feature Creation

18. Choosing the Right Model

Model	Best For	Notes
Decision Trees	Simple classification tasks	Easy to interpret
Random Forests	Tabular data, low overfitting	Ensemble of trees is better accuracy
Gradient Boosting (e.g., XGBoost, LightGBM)	Complex structured data	Used in many ML competitions
Neural Networks	Text, image, audio, sensor data	Requires more data & compute
Rule-Based Systems	Low-data scenarios	Useful in early prototypes (e.g., if-else logic)

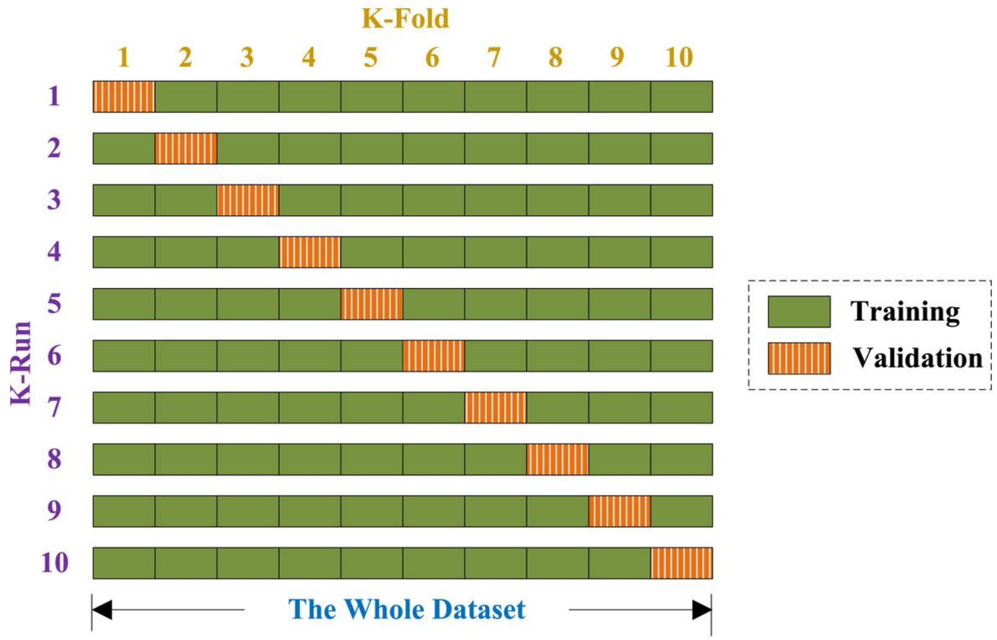
19. Training Best Practices

Always Split Your Data

- **Train Set** – Used to fit the model
- **Validation Set** – Used to tune hyperparameters
- **Test Set** – Final performance check (never touch during training!)

Cross-Validation (k-fold)

- Breaks your data into *k* chunks, rotates training/testing
- Helps improve model stability
- Essential in smaller datasets (common in student projects)



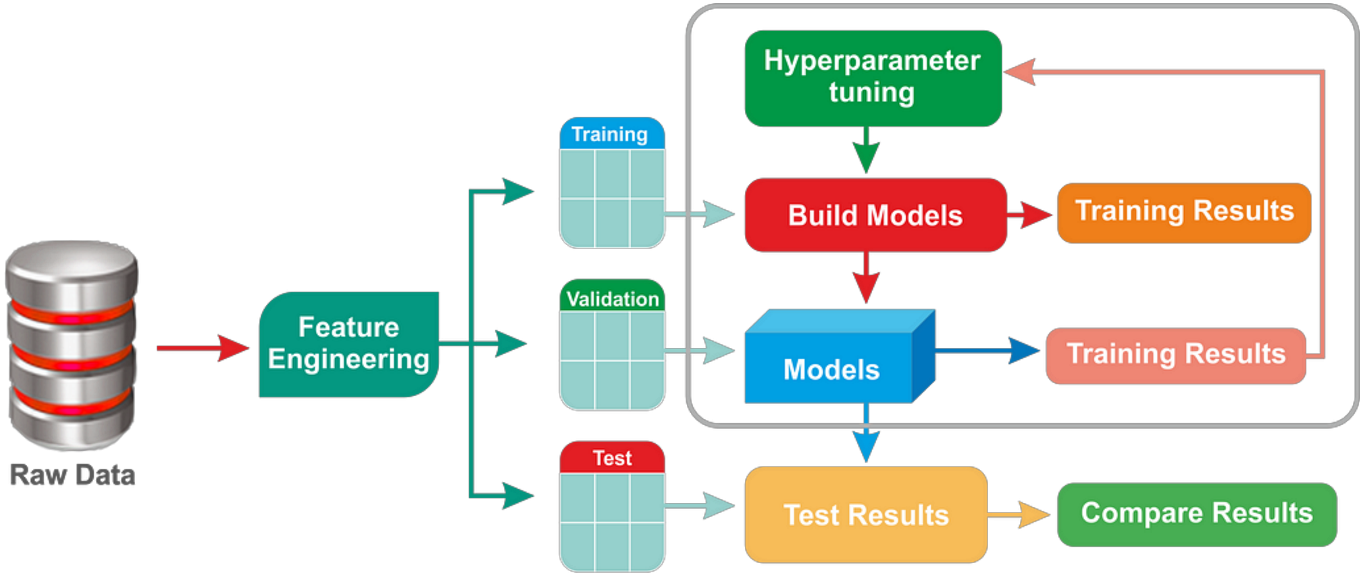
20. Hyperparameter Tuning

Common Techniques:

- **Grid Search** – Try all combinations
- **Random Search** – Try random combinations (faster)
- **Bayesian Optimization** – Smart search using probability (e.g., Optuna)

What Can Be Tuned?

- Learning rate
- Tree depth (for decision trees)
- Number of neurons/layers (in neural nets)
- Regularization terms



Note: Use **validation data**, not test data, when tuning!

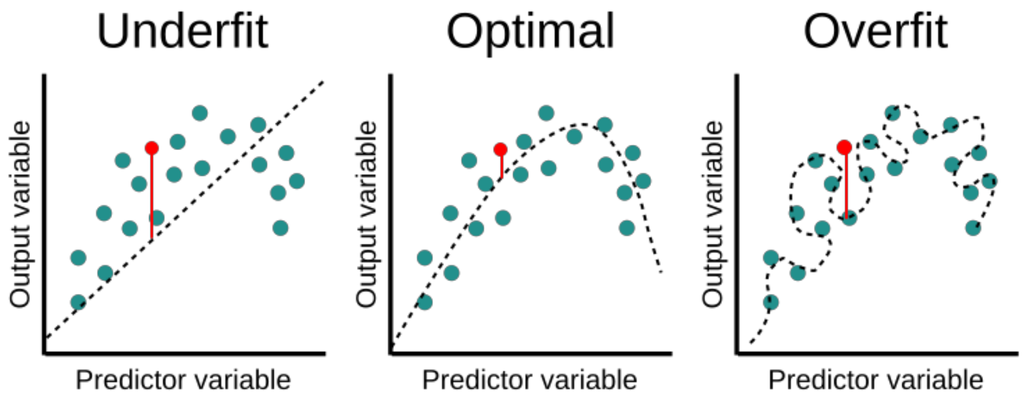
Image Courtesy: https://blog.devops.dev/hyperparameter-tuning-62bdf135248f

Overfitting & Underfitting

Underfitting in machine learning occurs when a model is too simple to capture the underlying patterns in the training data, leading to poor performance on both the training and new data

Causes:

1. Model is too simple
2. Insufficient training
3. Poor Feature Engineering
4. Limited Training Data

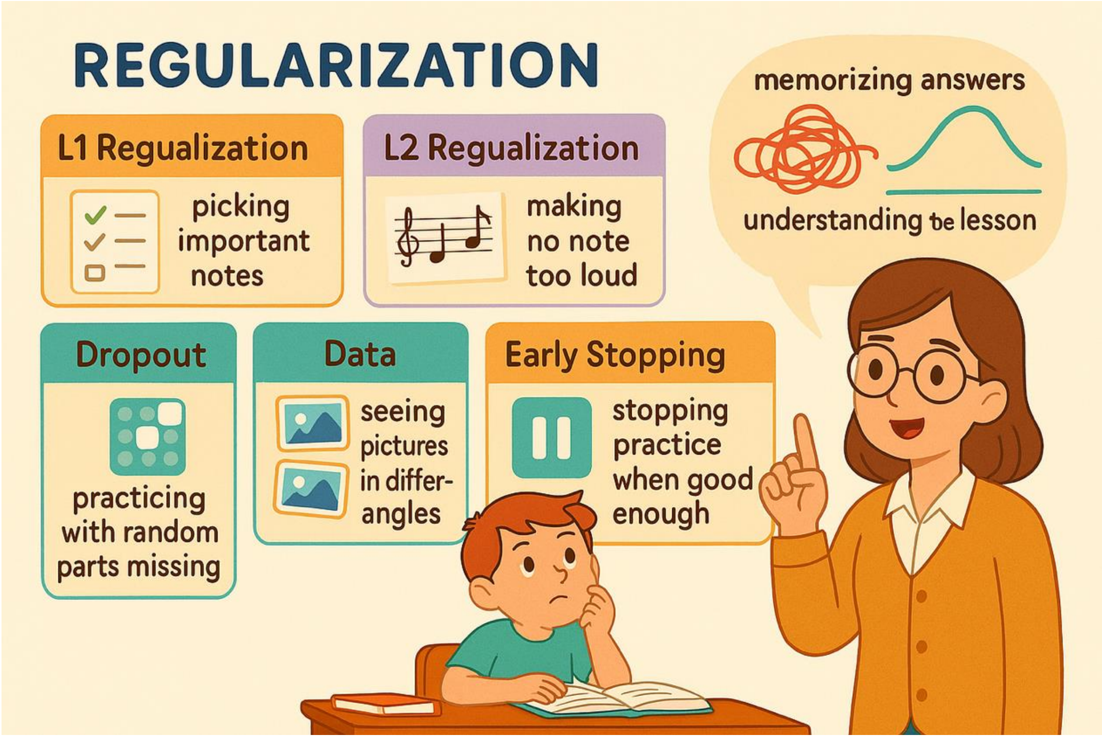


Overfitting in machine learning occurs when a model becomes too complex, fitting the training data's noise and anomalies so closely that it loses its ability to generalize and make accurate predictions on new, unseen data

Causes:

1. Model is complex
2. Long training

To Reduce Overfitting:	To Reduce Underfitting:
• **Regularization** (L1, L2) • **Dropout** in neural networks • **Early stopping** – stop training when validation score worsens • **Simplify the model** (reduce depth/neurons)	• Use **more complex models** • Add **better features** • Increase **training time**



Source: https://medium.com/@raed.da3k_87566/understanding-regularization-techniques-l1-l2-dropout-data-augmentation-and-early-stopping-8425f9e7d3c5

21. Interpretability & Explainability

In sectors like **finance** and **healthcare**, it's not just about *what* the model predicts—but *why*.

Tools for Explainability:

- **SHAP (SHapley Additive Explanations)** – Visualizes feature impact
- **LIME (Local Interpretable Model-agnostic Explanations)** – Explains individual predictions
- **Feature Importance** – Shows which input variables matter most

Examples:

- **RBI compliance**: Fintechs need to explain credit decisions
- **Medical diagnostics**: Doctors need to verify AI decisions (e.g., cancer detection apps)

22. Ethical & Compliance Considerations

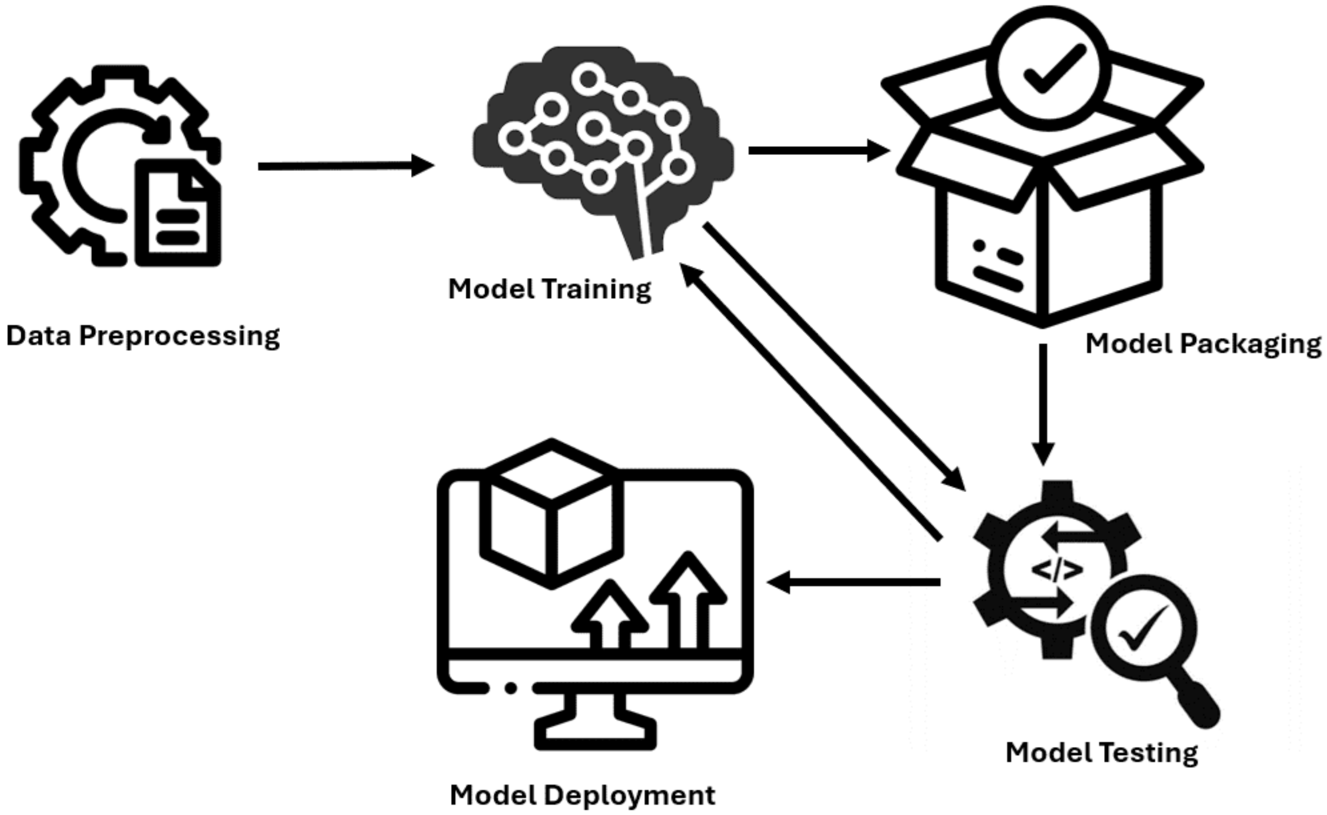
- **Data Privacy**
  - India's **Digital Personal Data Protection Act (DPDPA)**
  - Don't use personal data (like Aadhaar, phone numbers) without permission
- **Bias & Fairness**
  - Avoid models that discriminate (e.g., loan approvals biased by gender or caste)
- **Transparency**
  - Users should know how decisions are made (especially in healthcare/finance)
- **Draft AI Policy (IndiaAI Mission)**
  - Promotes responsible, inclusive, and explainable AI

Note:

- Always anonymize data
- Document decisions (what data you used, why)
- Respect user consent, even in projects

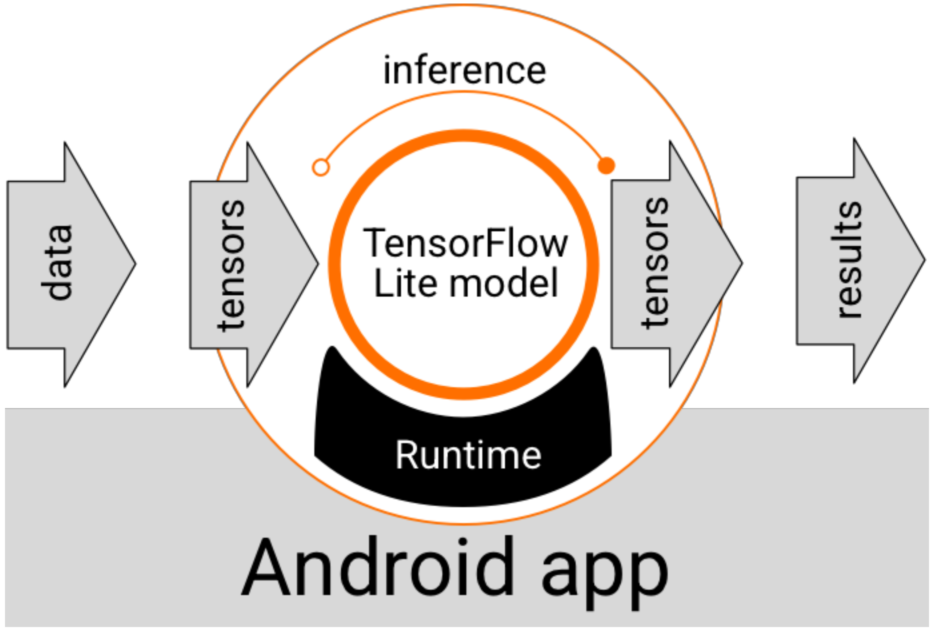
23. Model Deployment

Model deployment is the process of integrating a trained machine learning model into a production environment so it can be used by real-world applications to make predictions



Common Deployment Methods:

1. **Hosted APIs**
    1. Host your model as a REST API using FastAPI or Flask
    2. Use in web/mobile apps
    3. Ideal for chatbot, image recognition projects
2. **Mobile App Integration**
    1. Deploy models directly on Android using **TensorFlow Lite**
    2. Example: On-device crop disease detection for farmers
3. **Edge Deployment**
    1. Run models on low-resource devices (e.g., Raspberry Pi, basic smartphones)
    2. Useful in **rural areas** where internet is weak or absent



24. Infrastructure & Tooling

Cloud Platforms:

- **AWS**, **Google Cloud (GCP)**, **Microsoft Azure** – Provide compute, storage, ML services
- Use **free tiers** for student projects

Local Infra:

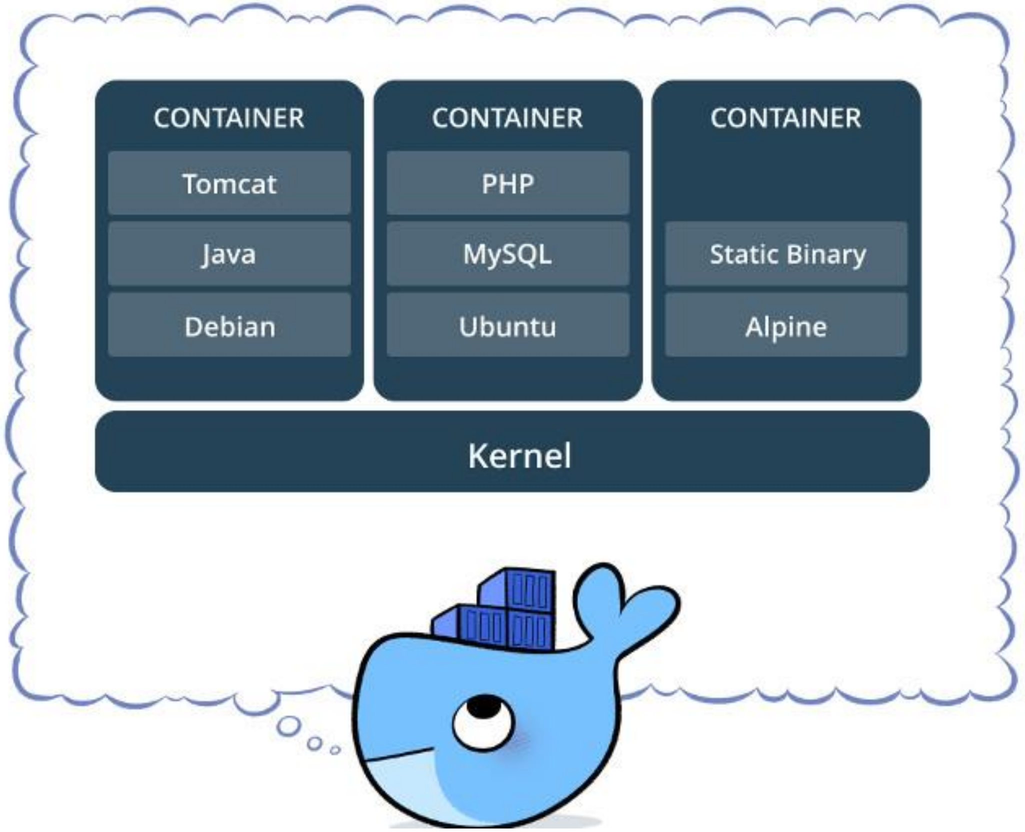
- Use college servers or affordable VPS hosting
- Ideal for early testing or low-scale deployments

Tools to Know:

- **FastAPI** – Lightweight API serving
- **TensorFlow Lite** – ML for mobile/edge
- **Streamlit/Gradio** – Quick web-based model demos
- **Hugging Face Spaces** – Deploy NLP models easily (supports regional languages)

25. Containerization & CI/CD

Containerization is the packaging of software code with all its dependencies, like libraries, frameworks, and configuration files into a single, standardized, and executable unit called a container



Docker

- Package your model + code + dependencies
- Run it anywhere without setup hassles

Kubernetes (K8s)

- Orchestrate multiple containers
- Used for production-scale deployments

CI/CD (Continuous Integration/Deployment):

- Tools: **GitHub Actions**, **GitLab CI**, **CircleCI**
- Automate testing, building, and updating models when code changes

Monitoring Deployed Models

What to Monitor:

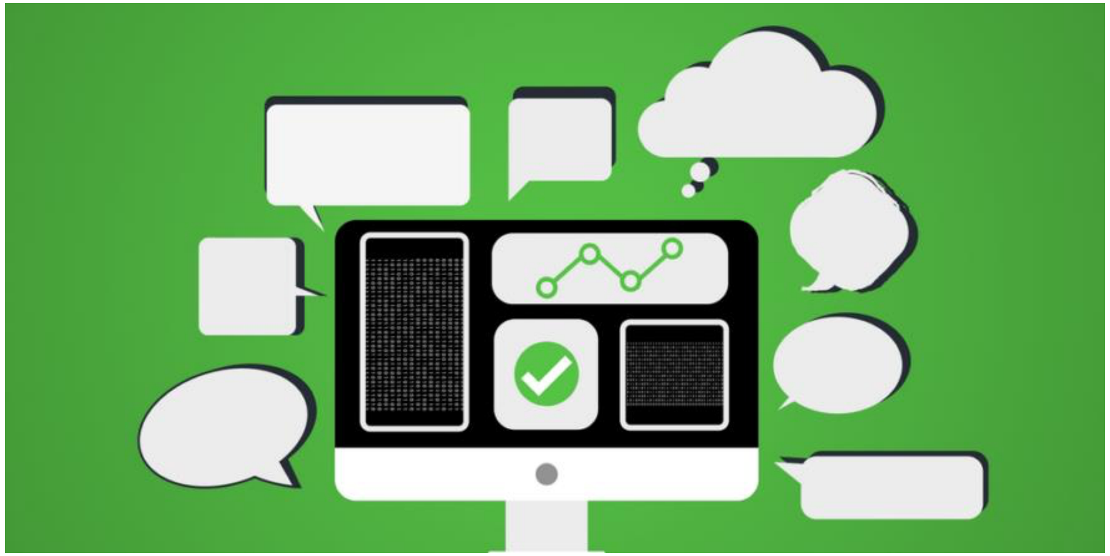
- **Prediction Accuracy** – Is it dropping on real-world data?
- **Latency** – Is the model fast enough?
- **User Feedback** – Are users satisfied with results?

Detecting:

- **Data Drift**: Real-world data distribution changes

  E.g., new slang in social media sentiment detection
- **Concept Drift**: The meaning behind data changes

  E.g., seasonal trends in shopping behavior



https://stackify.com/monitor-software-deployment/

Tools:

- Prometheus + Grafana
- MLflow
- Custom logging in FastAPI / Streamlit

Maintenance & Retraining Strategies

When to Retrain:

- Periodic (e.g., monthly, quarterly)
- When performance drops
- When new data types are detected

Strategies:

- **Scheduled Retraining** – Add new data and retrain offline
- **Online Learning** – Model adapts on-the-fly (e.g., for news trends)
- **Model Versioning** – Track changes with tools like **DVC**, **MLflow**, or **Weights & Biases**

AI is not one-time coding, it's ongoing learning

https://www.evidentlyai.com/blog/retrain-or-not-retrain

Key Takeaways

Recap: The AI/ML Lifecycle

1. Frame the Problem
2. Collect & Clean Data
3. Build & Train Model
4. Evaluate & Test
5. Deploy & Monitor

https://www.activtrak.com/blog/workplace-productivity-studies/

Conclusion

- AI/ML projects require a structured, step-by-step approach—from problem identification to deployment and monitoring.
- Understanding data quality, model selection, and evaluation metrics is key to building reliable solutions.
- Ethical, responsible AI practices—especially in the Indian context—must guide every stage of development.
- Deployment and real-world feedback help refine models and drive actual impact.
- With the right mindset and tools, students can build AI solutions that solve meaningful, local problems.

Source : www.freepik.com/

References

1. NITI Aayog – National Strategy for Artificial Intelligence (IndiaAI)
Source: Government of India
URL: https://indiaai.gov.in
Details AI use cases across agriculture, healthcare, and education in India.
Outlines policy support, ethics, and India's AI roadmap.

2. Google Machine Learning Crash Course
Source: Google Developers
URL: https://developers.google.com/machine-learning/crash-course

3. Covers AI/ML lifecycle, model evaluation, hyperparameter tuning, etc.
Student-friendly and interactive.
Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media, 2022.

4. Google Developers. Machine Learning Crash Course
https://developers.google.com/machine-learning/crash-course

5. Microsoft Learn. AI/ML Fundamentals
https://learn.microsoft.com/en-us/training/paths/ml-introduction/
IBM. **AI Engineering Professional Certificate** (Coursera)
https://www.coursera.org/professional-certificates/ai-engineer

Let’s Start

Quiz

1: Which of the following represents the correct order of the AI/ML project lifecycle?

A. Data collection → Modeling → Deployment → Ideation → Monitoring

B. Modeling → Ideation → Monitoring → Deployment → Data Collection

C. Ideation → Data Collection → Modeling → Deployment → Monitoring

D. Monitoring → Modeling → Deployment → Ideation → Data Cleaning

Quiz

1: Which of the following represents the correct order of the AI/ML project lifecycle?

A. Data collection → Modeling → Deployment → Ideation → Monitoring

B. Modeling → Ideation → Monitoring → Deployment → Data Collection

C. Ideation → Data Collection → Modeling → Deployment → Monitoring

D. Monitoring → Modeling → Deployment → Ideation → Data Cleaning

Answer: C

Ideation → Data Collection → Modeling → Deployment → Monitoring

Quiz

2. In a classification task, which metric is most useful when the dataset is highly imbalanced (e.g., 95% negative and 5% positive)?

A. Accuracy

B. Precision

C. Recall

D. F1-Score

Quiz

2. In a classification task, which metric is most useful when the dataset is highly imbalanced (e.g., 95% negative and 5% positive)?

A. Accuracy

B. Precision

C. Recall

D. F1-Score

Answer: D

F1-Score

Quiz

3. Which of the following is a common method for hyperparameter tuning in machine learning?

A. Cross-entropy loss

B. Grid Search

C. Feature Scaling

D. Data Augmentation

Quiz

3. Which of the following is a common method for hyperparameter tuning in machine learning?

A. Cross-entropy loss

B. Grid Search

C. Feature Scaling

D. Data Augmentation

Answer: B

Grid Search

Quiz

4. What is the primary purpose of using Docker in an AI/ML deployment pipeline?

A. To monitor model drift

B. To perform hyperparameter tuning

C. To create a mobile app interface

D. To containerize and run the model consistently

Quiz

4. What is the primary purpose of using Docker in an AI/ML deployment pipeline?

A. To monitor model drift

B. To perform hyperparameter tuning

C. To create a mobile app interface

D. To containerize and run the model consistently

Answer: D

To containerize and run the model consistently

Quiz

5. Which of the following is the most suitable option for deploying AI models on smartphones in rural areas with limited internet connectivity?

A. Google Cloud AutoML

B. Streamlit Web App

C. TensorFlow Lite

D. Docker on AWS

Quiz

5. Which of the following is the most suitable option for deploying AI models on smartphones in rural areas with limited internet connectivity?

A. Google Cloud AutoML

B. Streamlit Web App

C. TensorFlow Lite

D. Docker on AWS

Answer: C

TensorFlow Lite

Thank You