LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

INT395- SUPERVISED ML

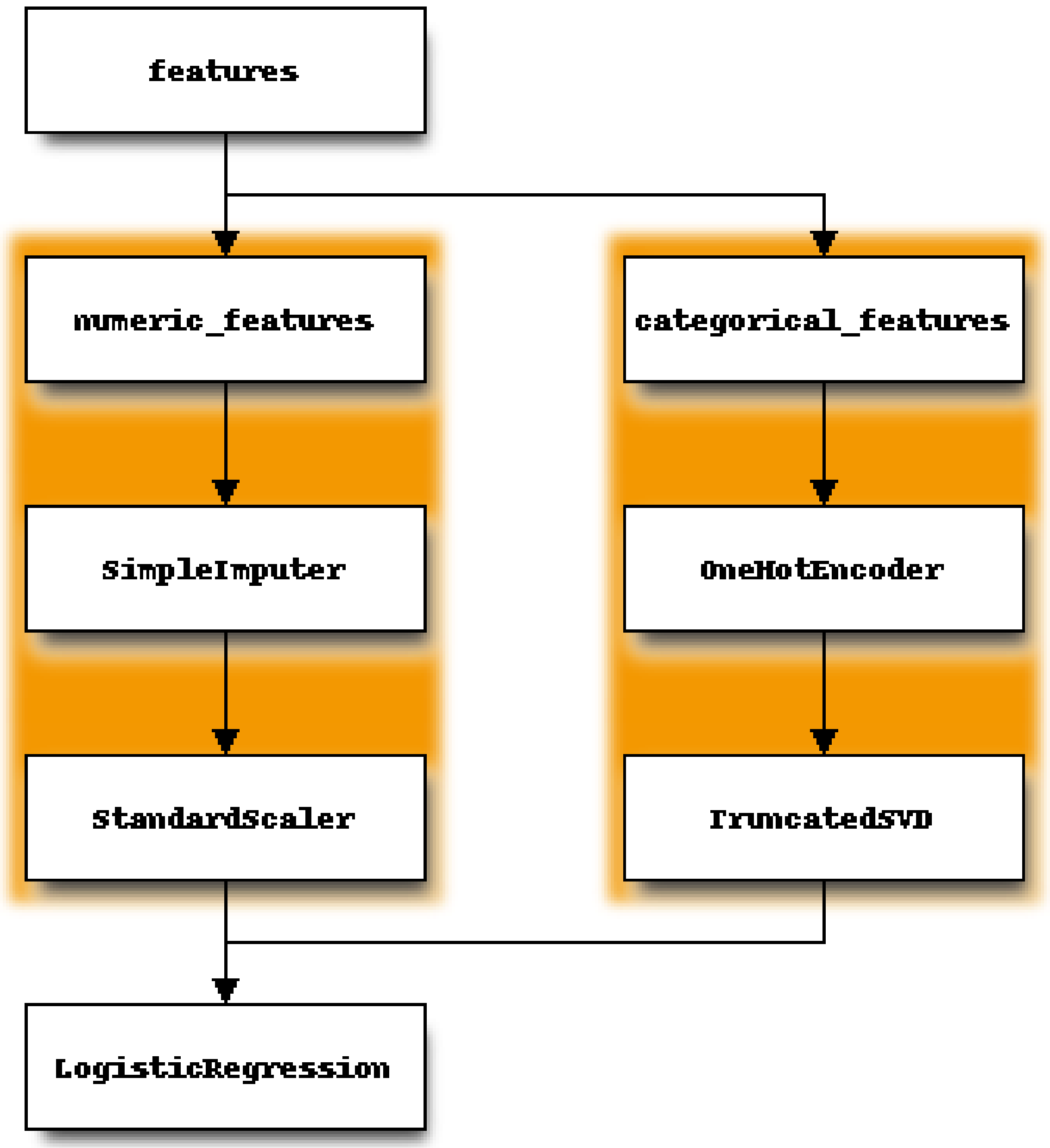
Unit 6:

- Pipelines and Model Evaluation
- Model Deployment

Presented By: Blossom Kaler
Assistant Professor
SCAI, LPU

ML PIPELINES

- Machine Learning involves multiple sequential steps
- Preprocessing, feature engineering and modeling must follow the same order
- Managing steps separately can cause errors and data leakage
- **Pipelines combine all steps into a single workflow**
- **A pipeline is a sequence of data processing steps**
- Each step has a transformer or estimator
- Data flows through all steps automatically
- **Typical flow: Scaling → Feature Selection → Model**
- Pipeline acts as a single object with fit and predict methods



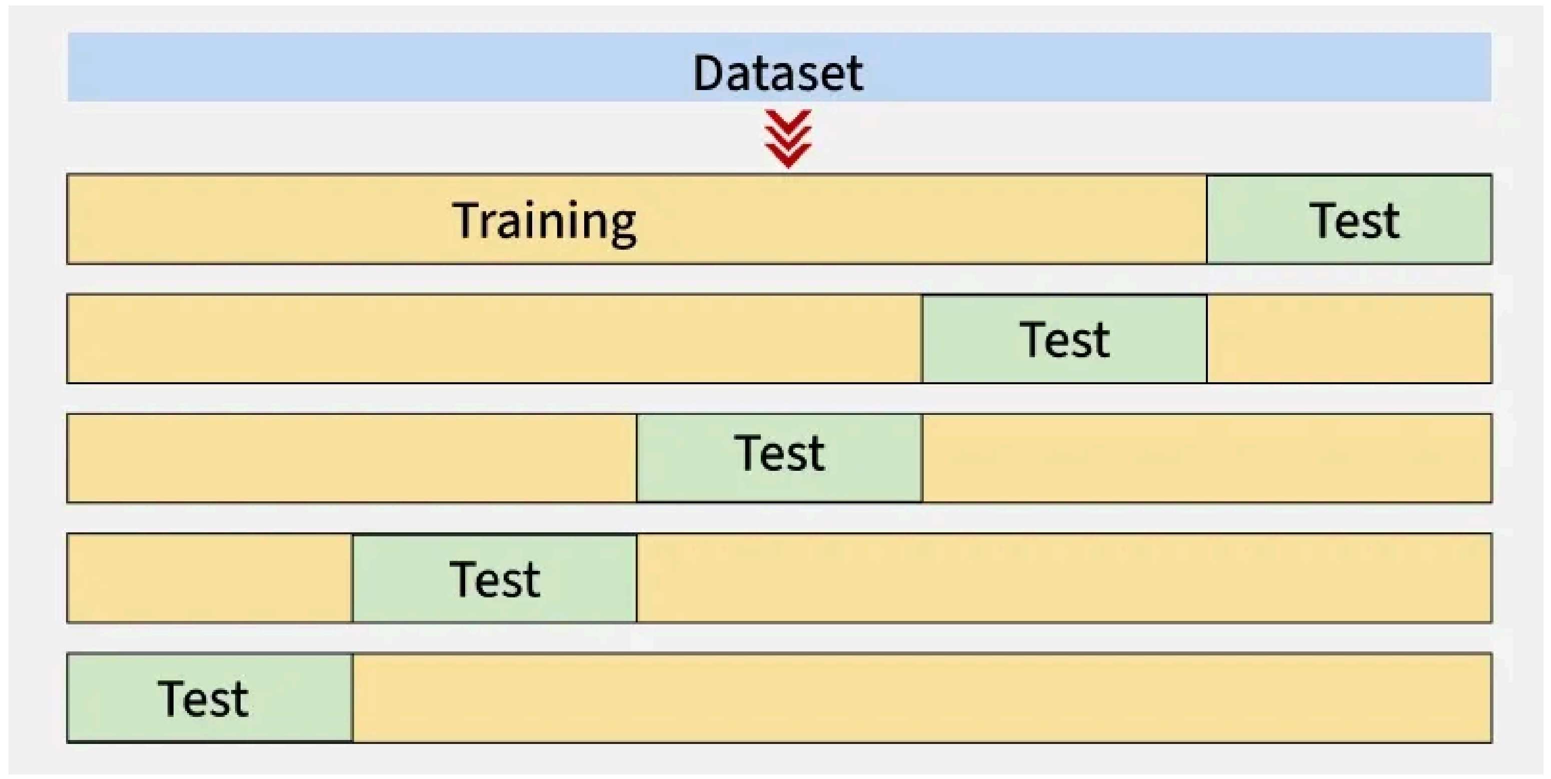
Refer the code shared with this PPT!

LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

WHAT IS CROSS VALIDATION?

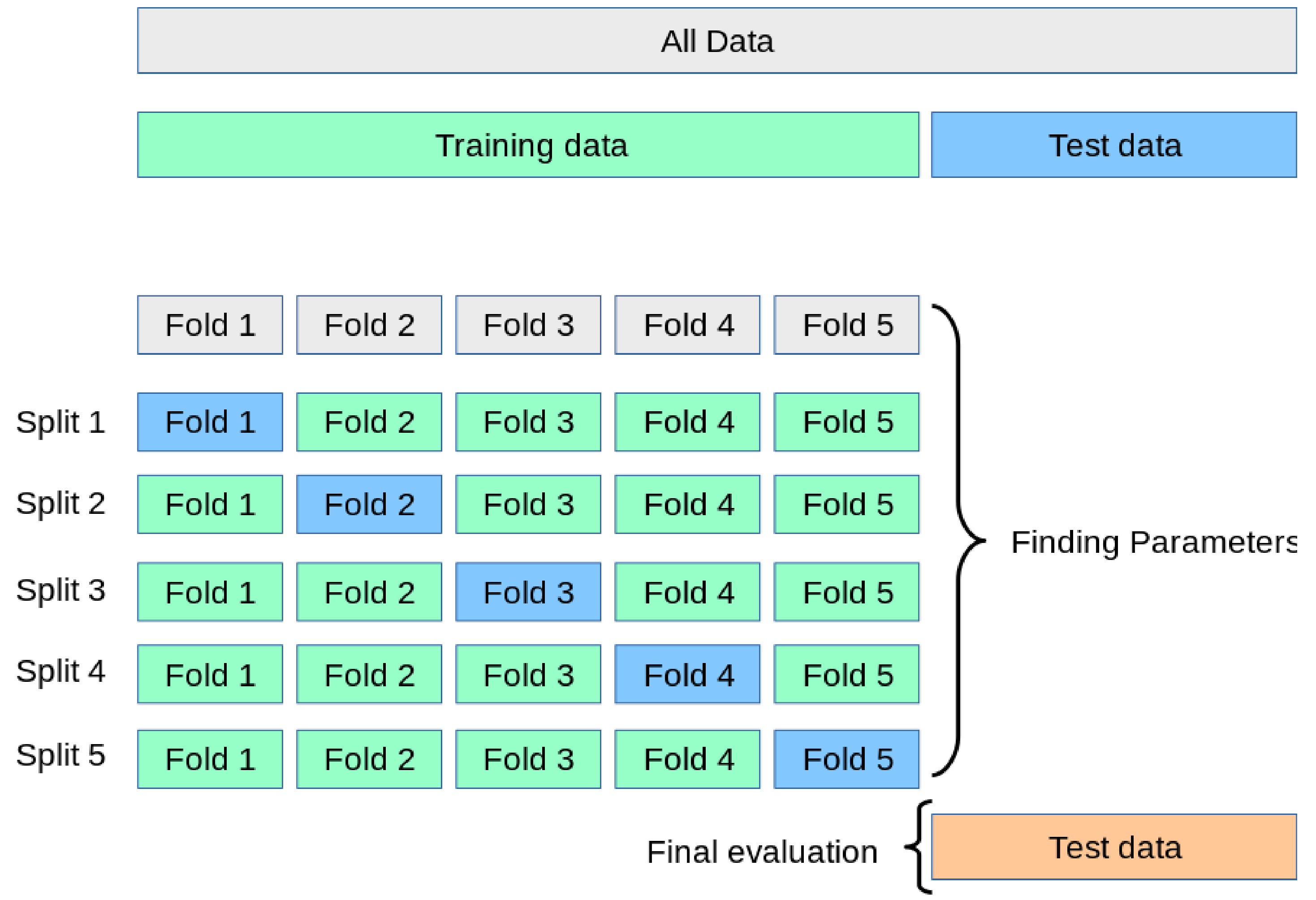
- Cross validation evaluates model performance reliably
- Dataset is split into multiple training and testing parts
- Model is trained and tested multiple times
- Performance is averaged across all splits
- Reduces dependency on a single train-test split
- Provides better estimate of model generalization



K-FOLD CROSS VALIDATION

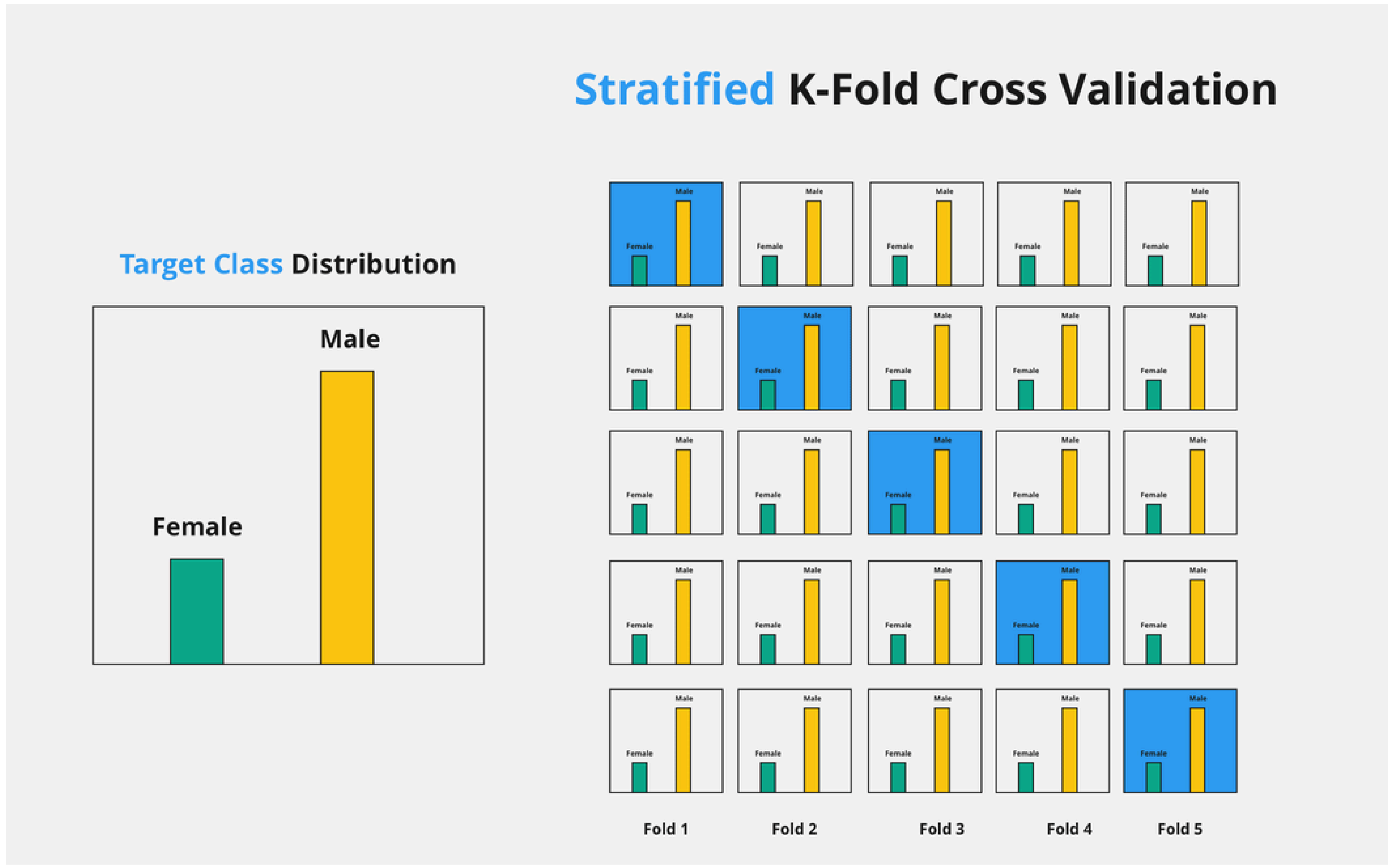
- To evaluate model performance more reliably using all available data.

  Process:
- Split dataset into k equal folds.
- For each iteration (total k):
- Train model on (k-1) folds
- Validate on the remaining 1 fold
- Repeat this k times (each fold acts as validation once).
- Average the validation results → gives final performance estimation



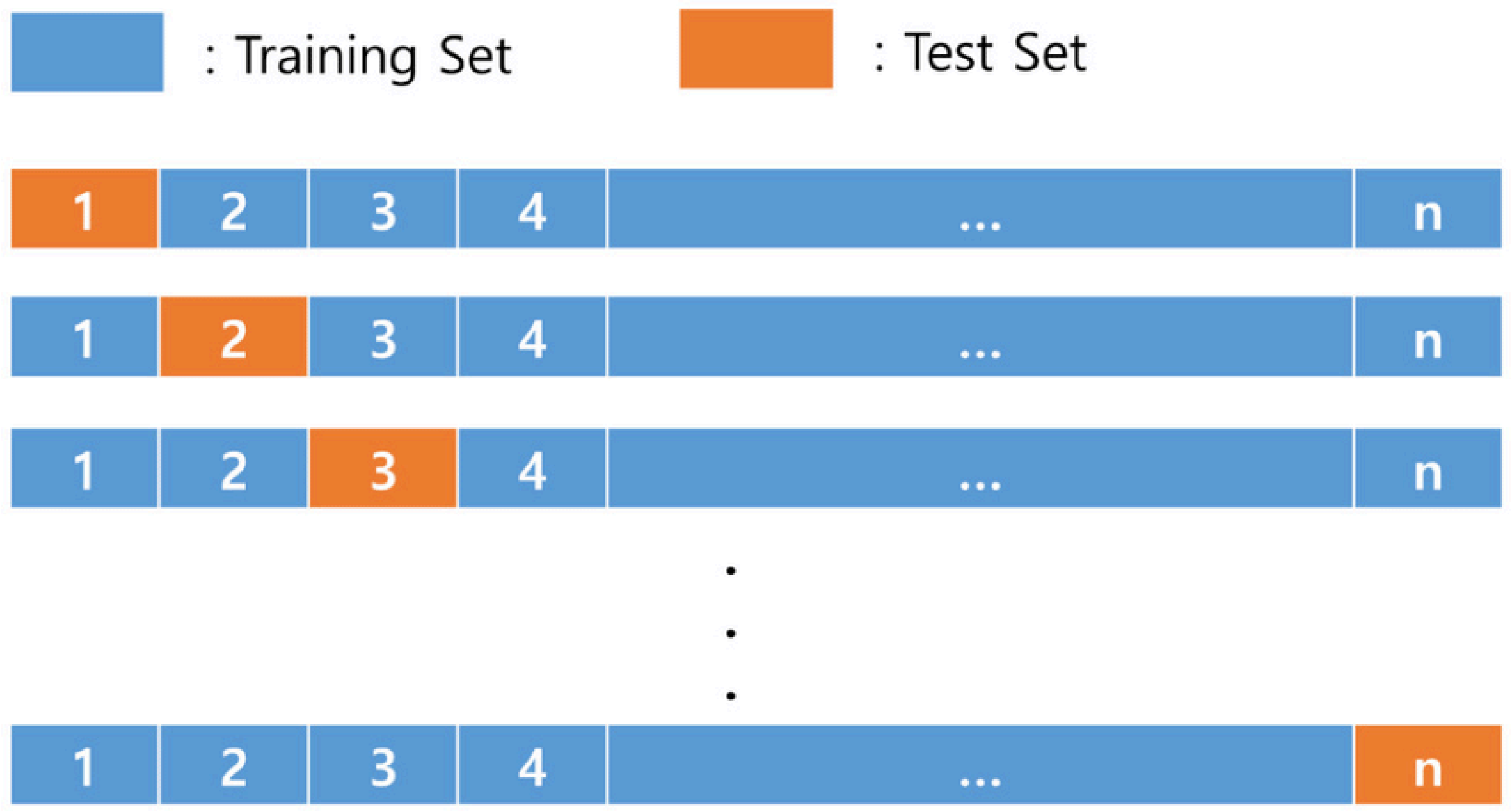
STRATIFIED K-FOLD CROSS VALIDATION

- Variation of K-Fold CV; **preserves class proportions in each fold**
- Useful for imbalanced classification
- Data → split into K folds maintaining same label distribution
- Train on K-1 folds; test on remaining fold; repeat K times
- Reduces bias due to uneven class splits
- More reliable evaluation than normal K-Fold for classification problems



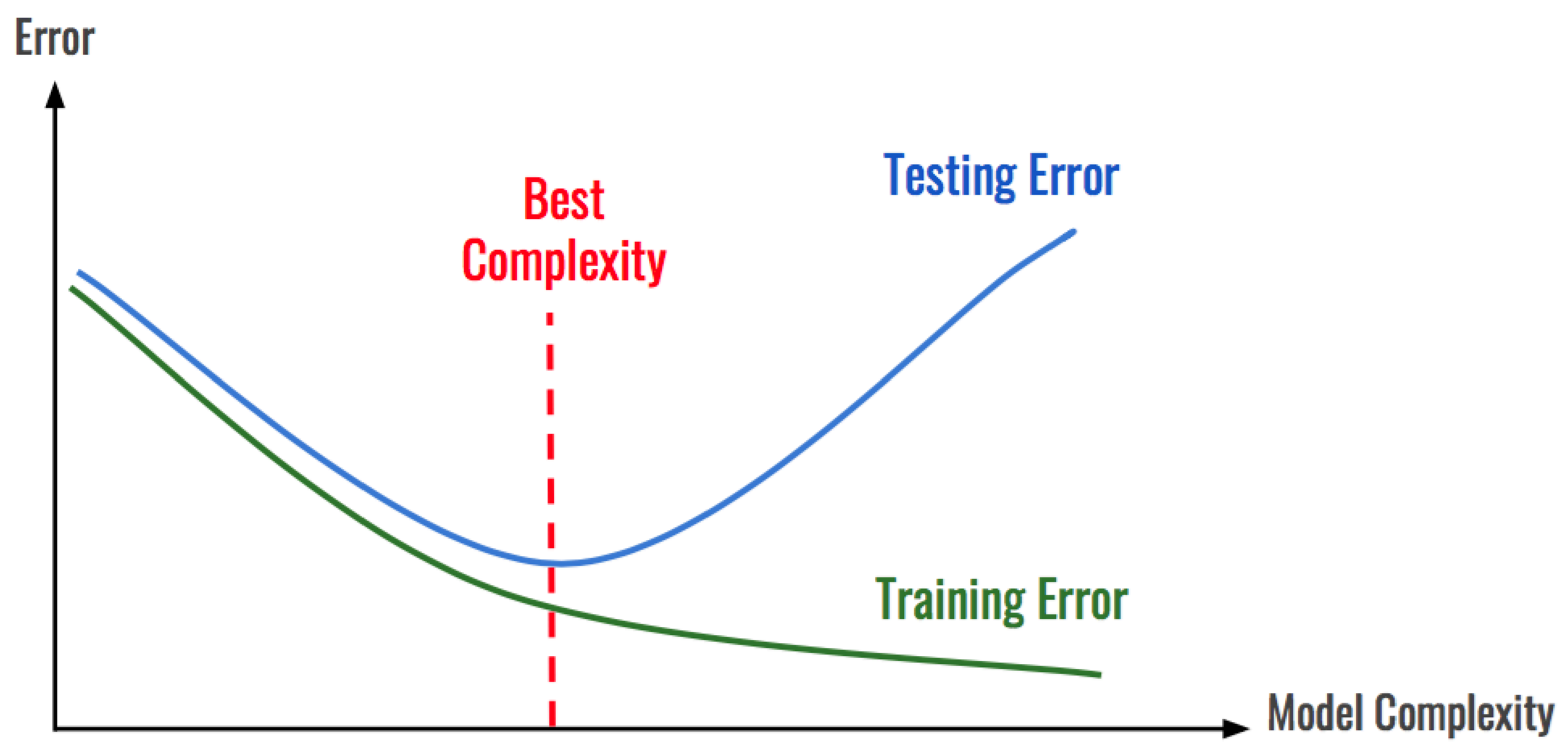
LEAVE ONE OUT CROSS VALIDATION (LOOCV)

- **Special case of K-Fold where K = number of samples**
- Each iteration: train on all samples except one, test on that one
- Very reliable for small datasets
- Low bias but high variance & computational cost
- Gives nearly unbiased model performance estimate



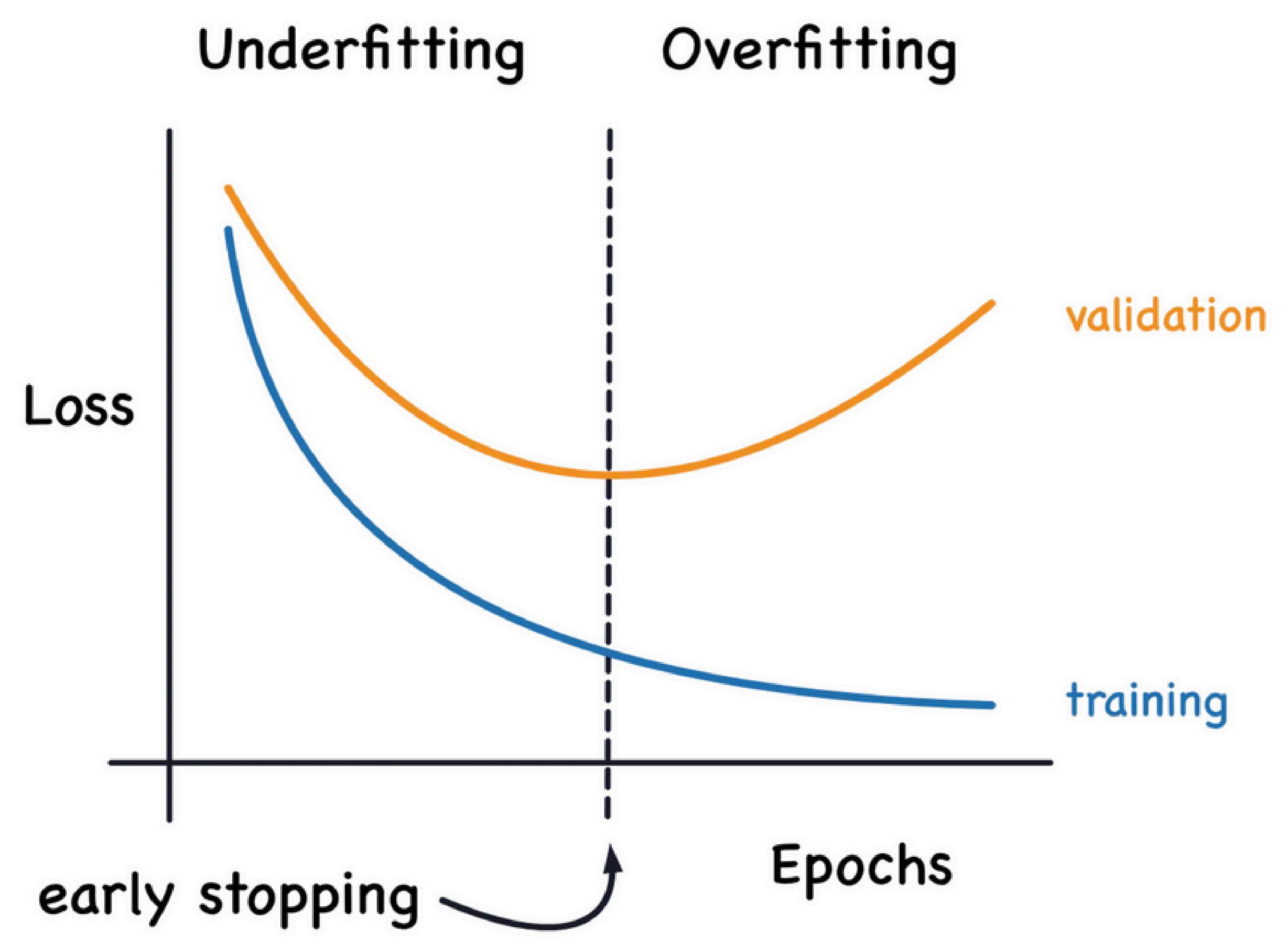
NEED FOR MODEL DEBUGGING

- Low accuracy does not explain model behavior
- Models may suffer from underfitting or overfitting
- Debugging helps understand learning patterns
- **Learning curves and validation curves are diagnostic tools**
- Helps decide model complexity and data requirements
- Improves model performance systematically



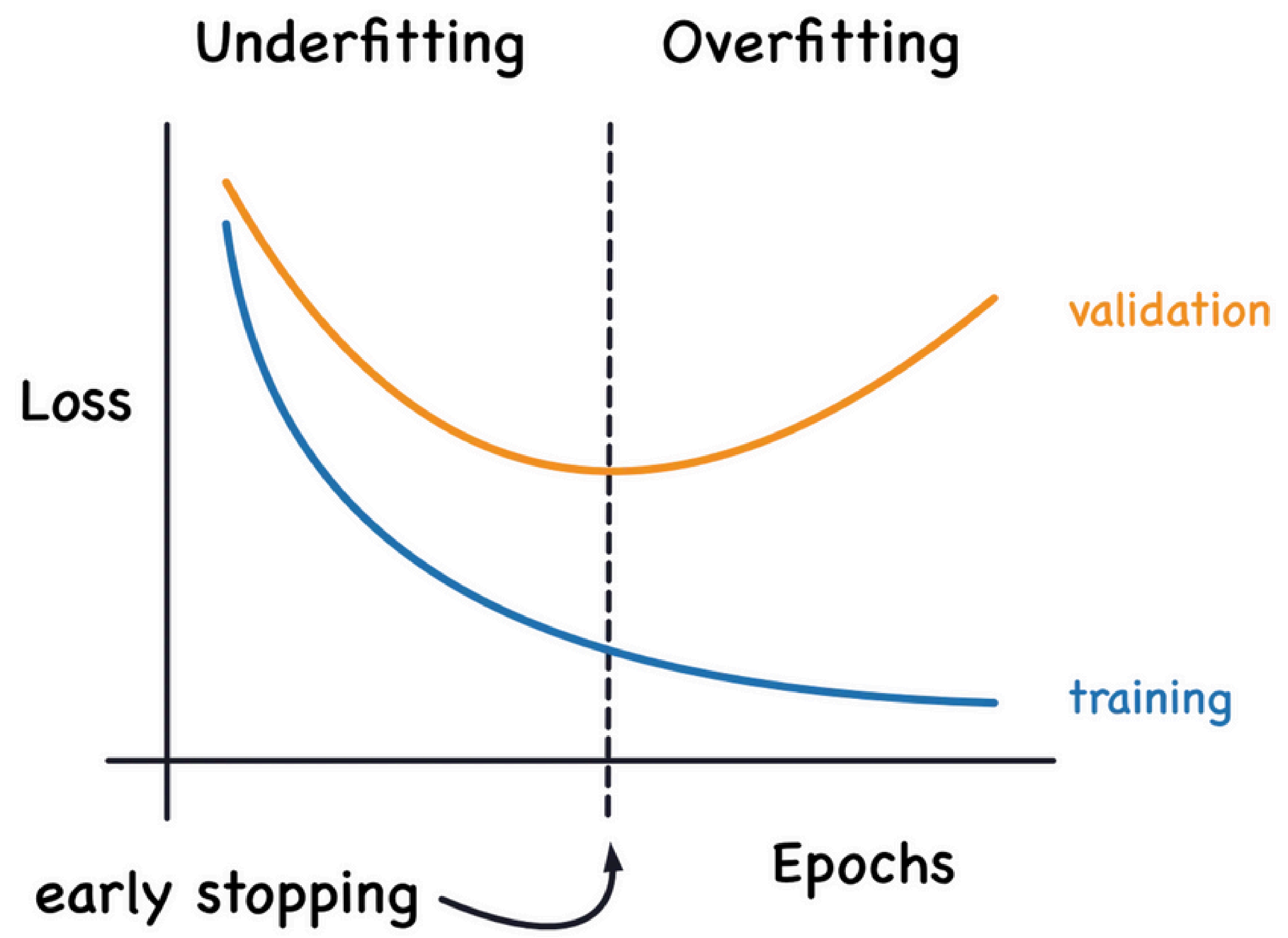
LEARNING CURVES

- Plot between training size and model performance
- **X-axis: Training Time/ Epochs**
- **Y-axis: Training and validation score**
- Underfitting: Both scores are low and close
- Overfitting: Large gap between training and validation score
- Helps determine if more data will improve performance



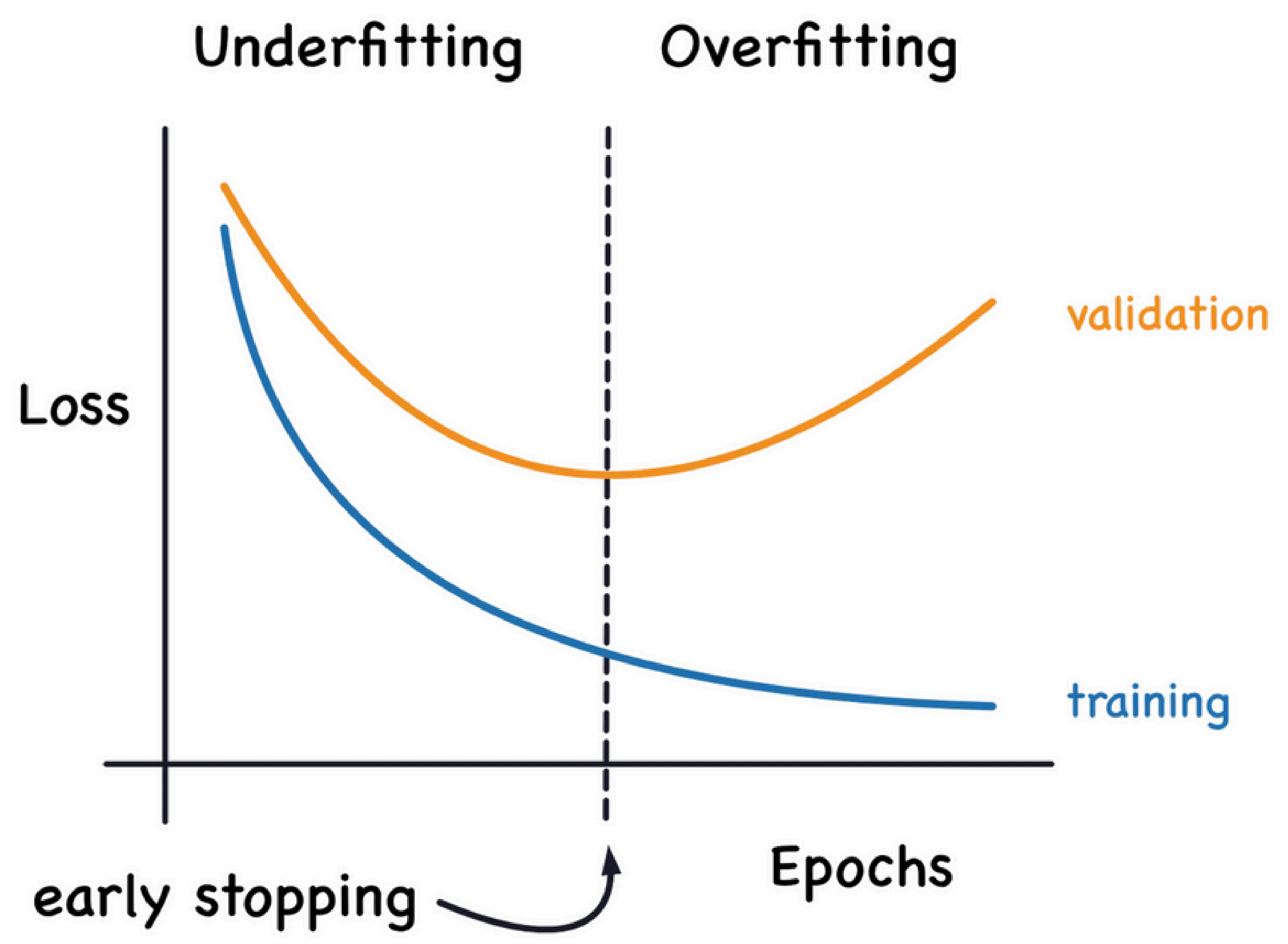
UNDERFITTING IN LEARNING CURVES

- Model is too simple to learn patterns in data
- Training score is low
- Validation score is also low
- Gap between training and validation curves is small
- Both curves converge at a poor performance level



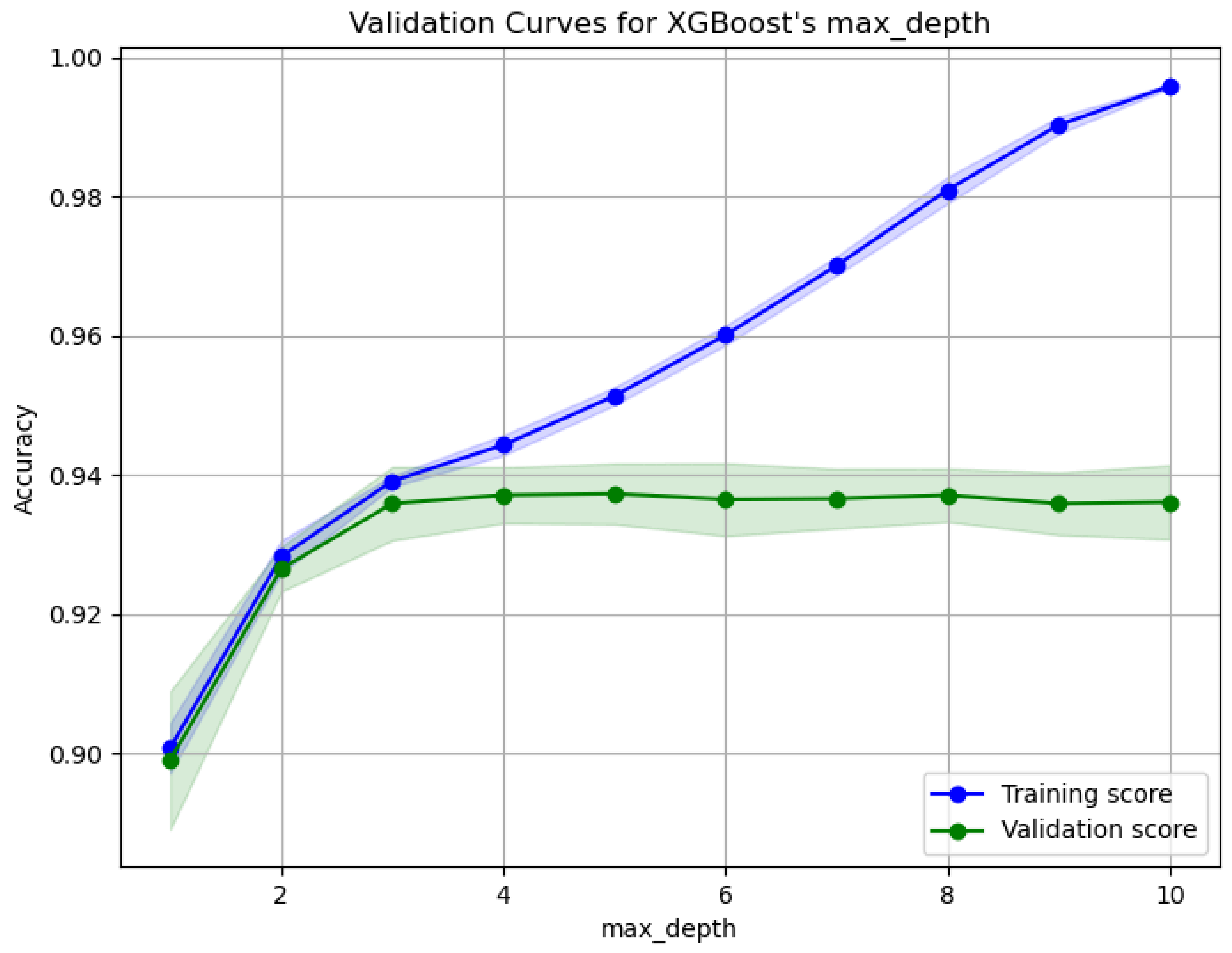
OVERFITTING IN LEARNING CURVES

- Model learns noise instead of general patterns
- Training score is very high
- Validation score is significantly lower
- Large gap between training and validation curves
- Gap does not reduce even with more data



VALIDATION CURVES

- Plot between hyperparameter values and performance
- **X-axis: Hyperparameter value (e.g., C, max depth)**
- **Y-axis: Training and validation score**
- Low parameter value → Underfitting
- High parameter value → Overfitting
- Helps choose optimal hyperparameter value

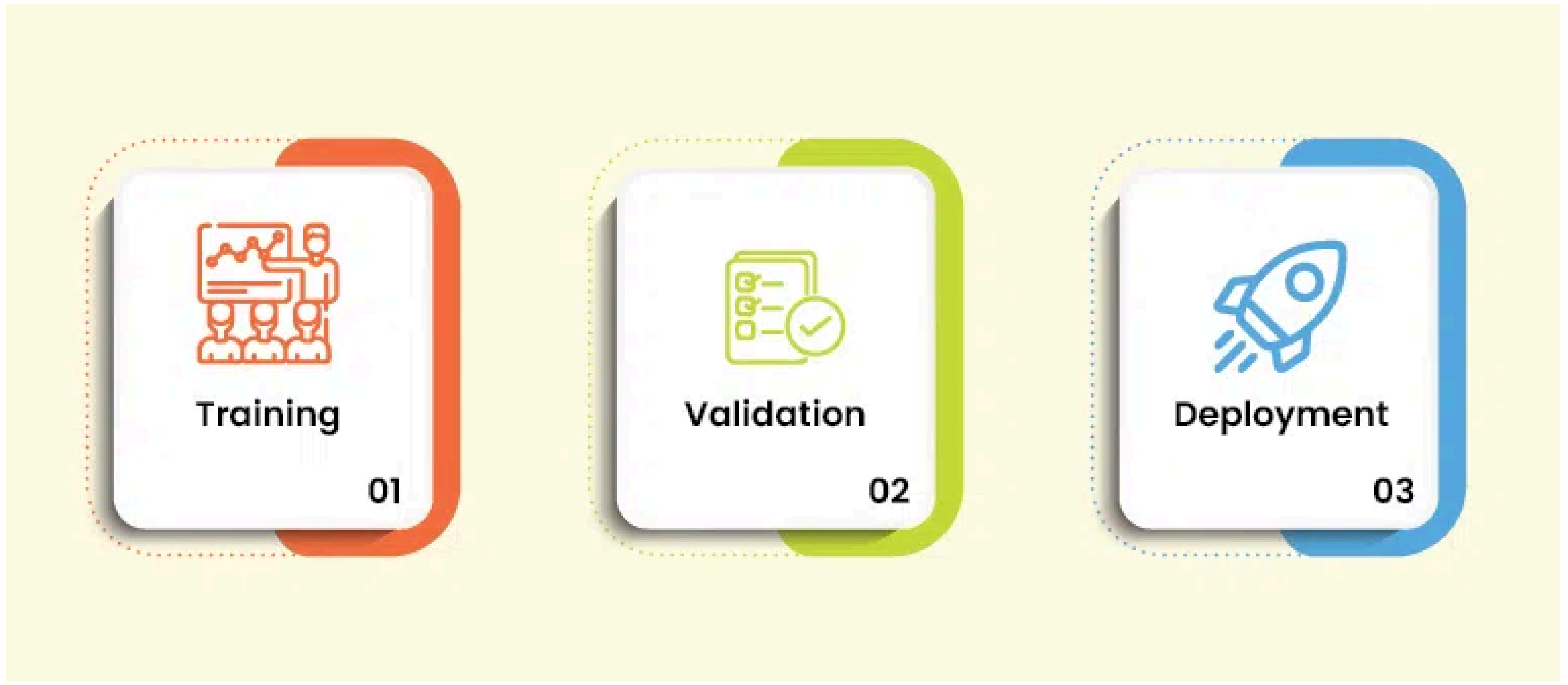


LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

MODEL DEPLOYMENT – IMPORTANCE

- **Model deployment is the process of integrating a trained machine learning model into a live, operational environment so it can make predictions or decisions on new data, effectively turning a prototype into a usable, real-world tool for users, applications, or systems**
- Deployment makes the model usable in real applications
- Enables predictions on real-time or new data
- Adds practical and business value to ML solutions
- A model has impact only when deployed

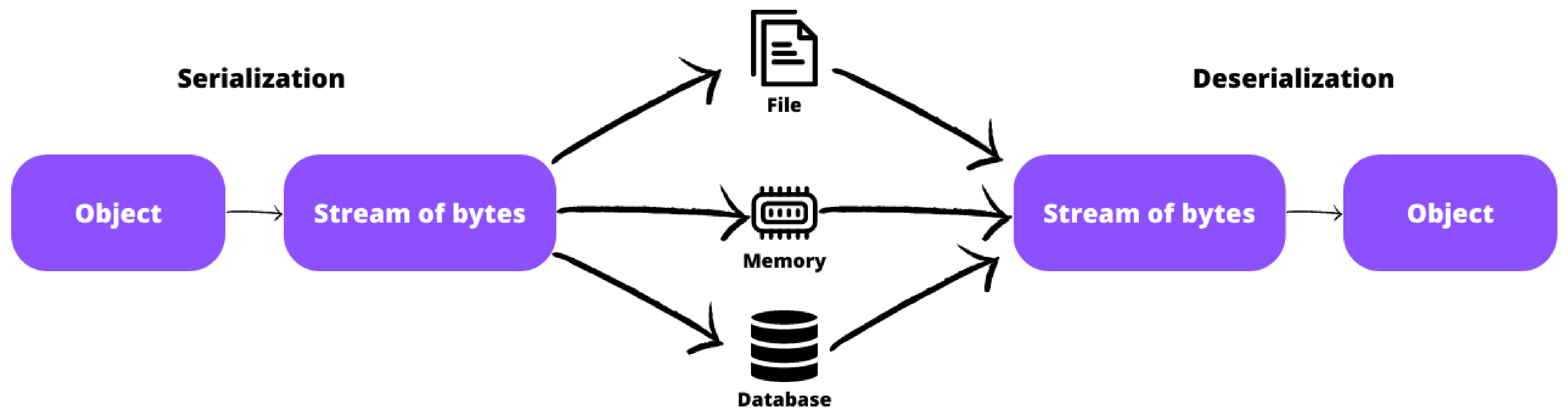


LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

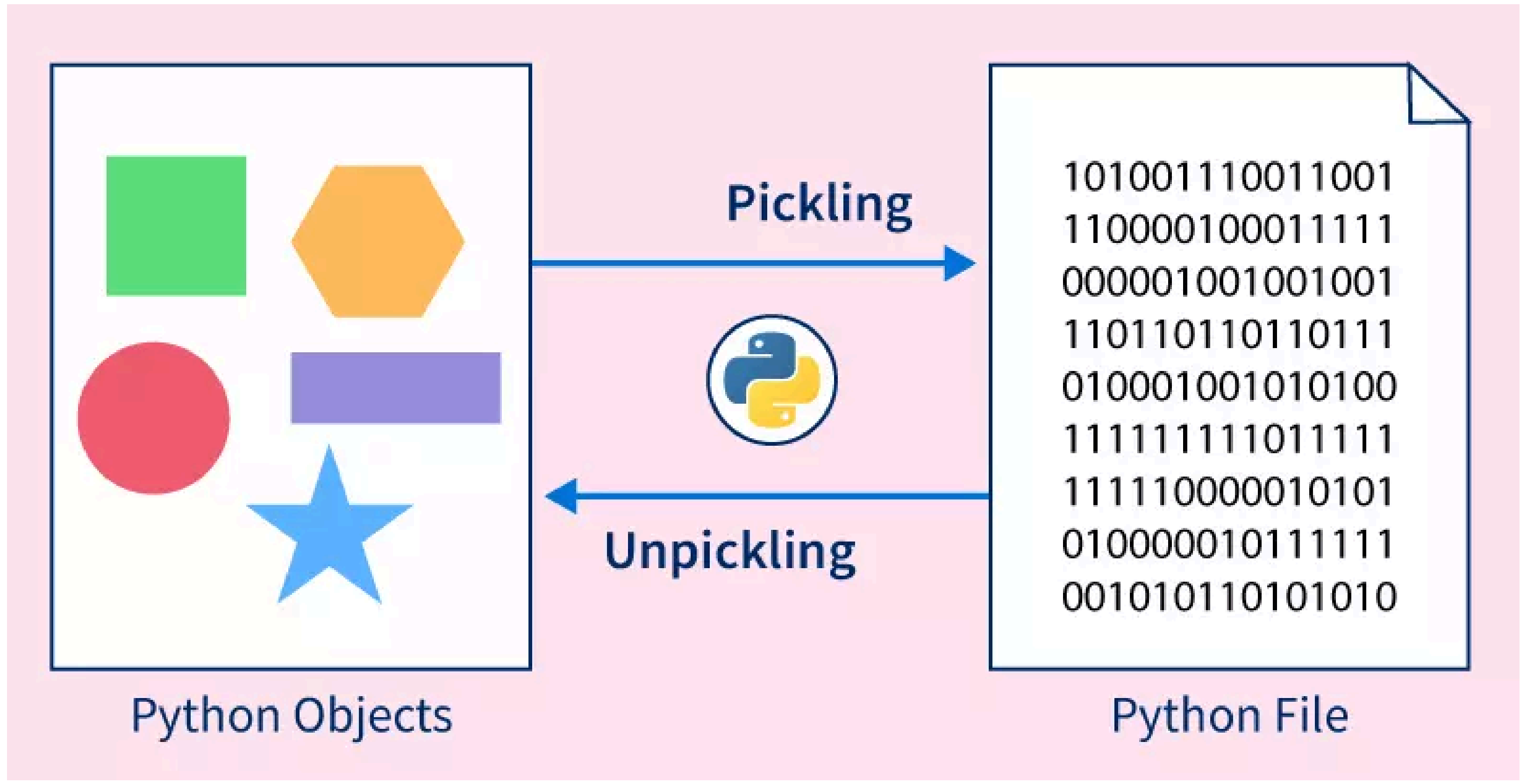
MODEL SERIALIZATION AND DESERIALIZATION

- **Serialization means saving a trained model to disk**
- **Deserialization means loading the saved model**
- Common tools: Pickle and Joblib
- Avoids retraining the model every time
- Enables portability and reuse of trained models
- Essential step before deployment



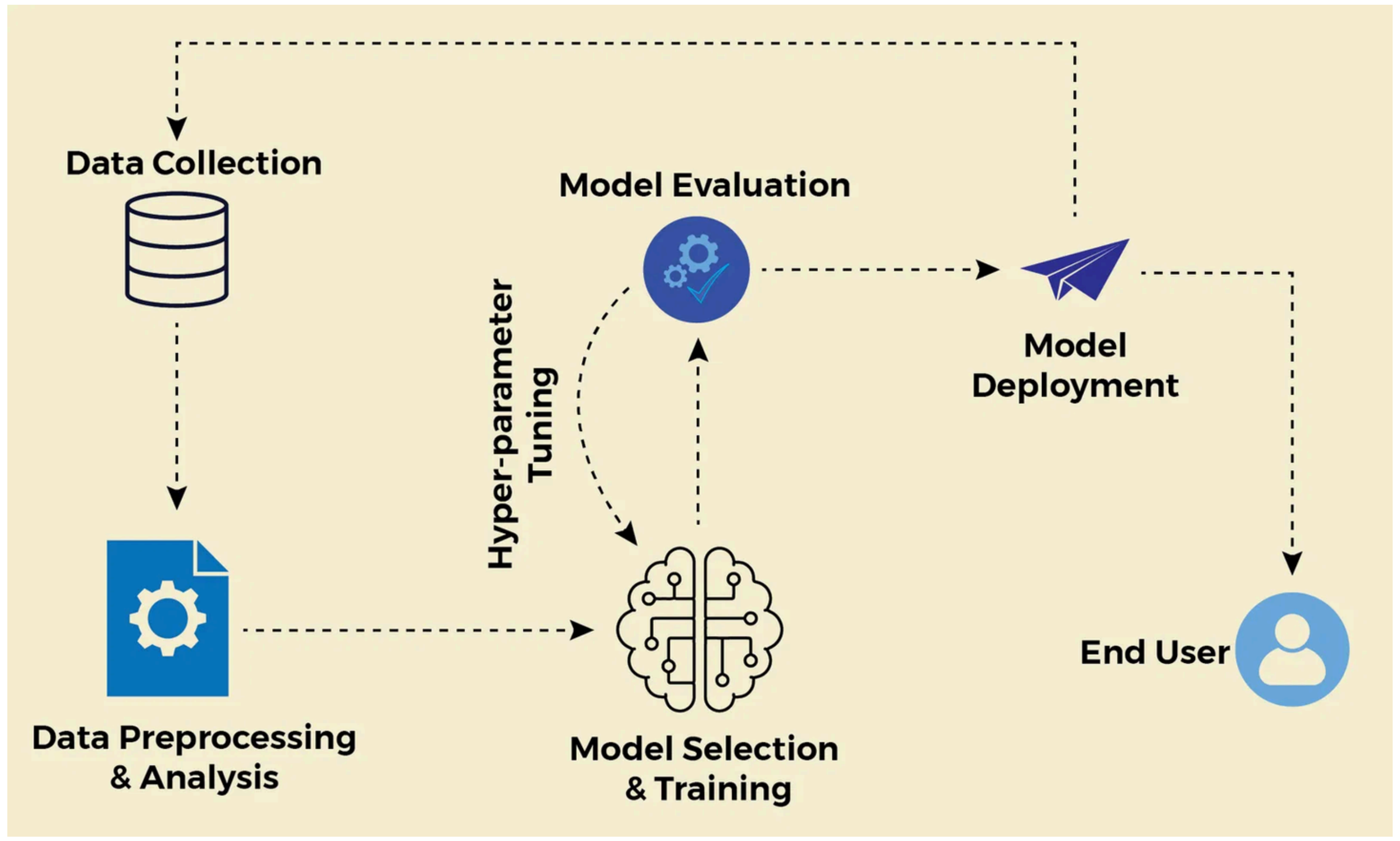
PICKLING

- Pickling is the process of serializing Python objects.
- The pickle module converts objects into a byte stream.
- Useful for saving ML models, preprocessed data, or dictionaries.
- **pickle.dump(obj, file)** saves an object; **pickle.load(file)** restores it.
- Helps avoid retraining models every time.
- Be cautious: pickle files are Python-specific and not secure against untrusted sources.



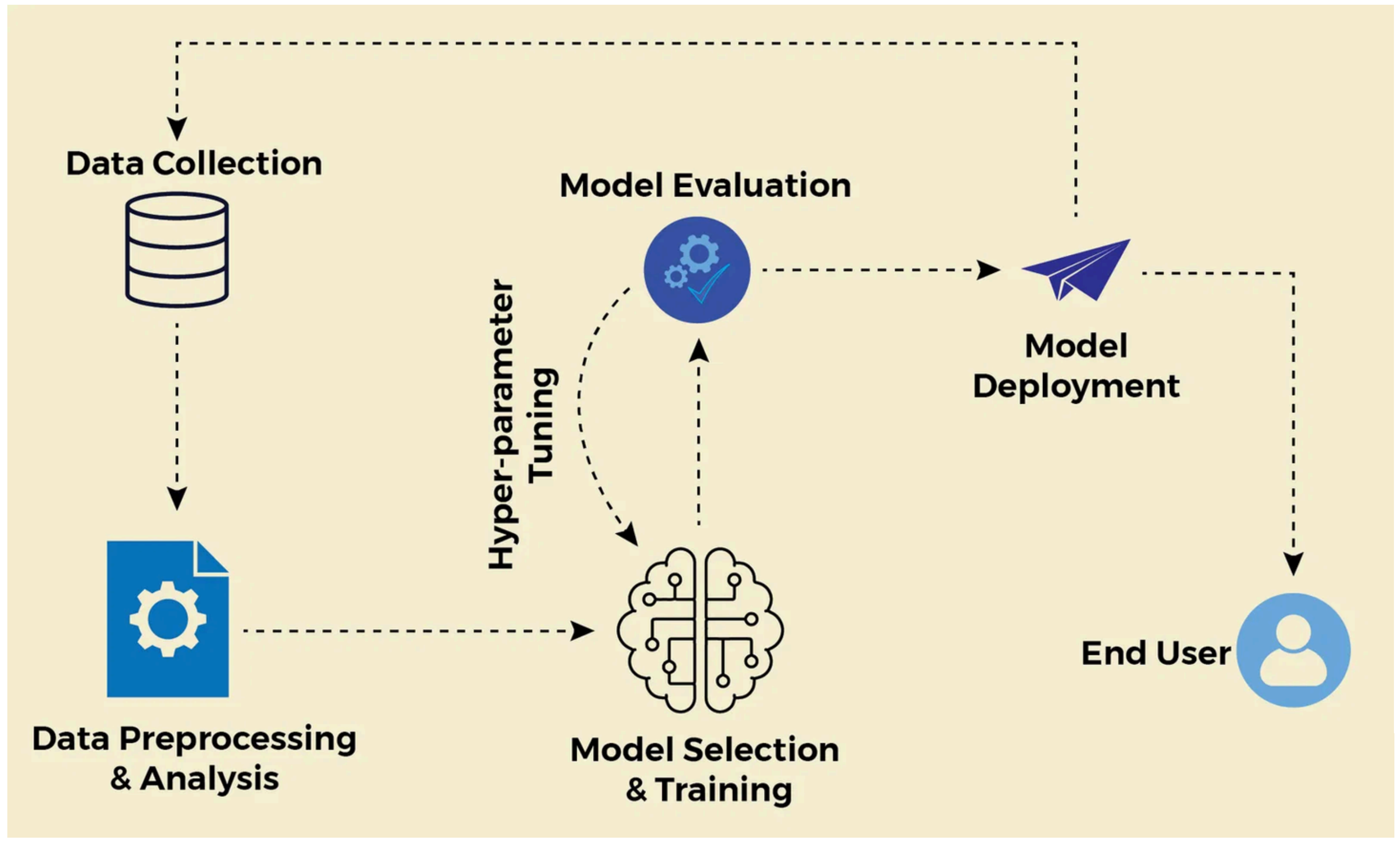
LOCAL MODEL DEPLOYMENT

- Model is deployed and executed on a local machine
- Used mainly for testing and experimentation
- No internet connection required
- Simple deployment using Python scripts
- Suitable for small-scale or offline applications
- Limited scalability and accessibility



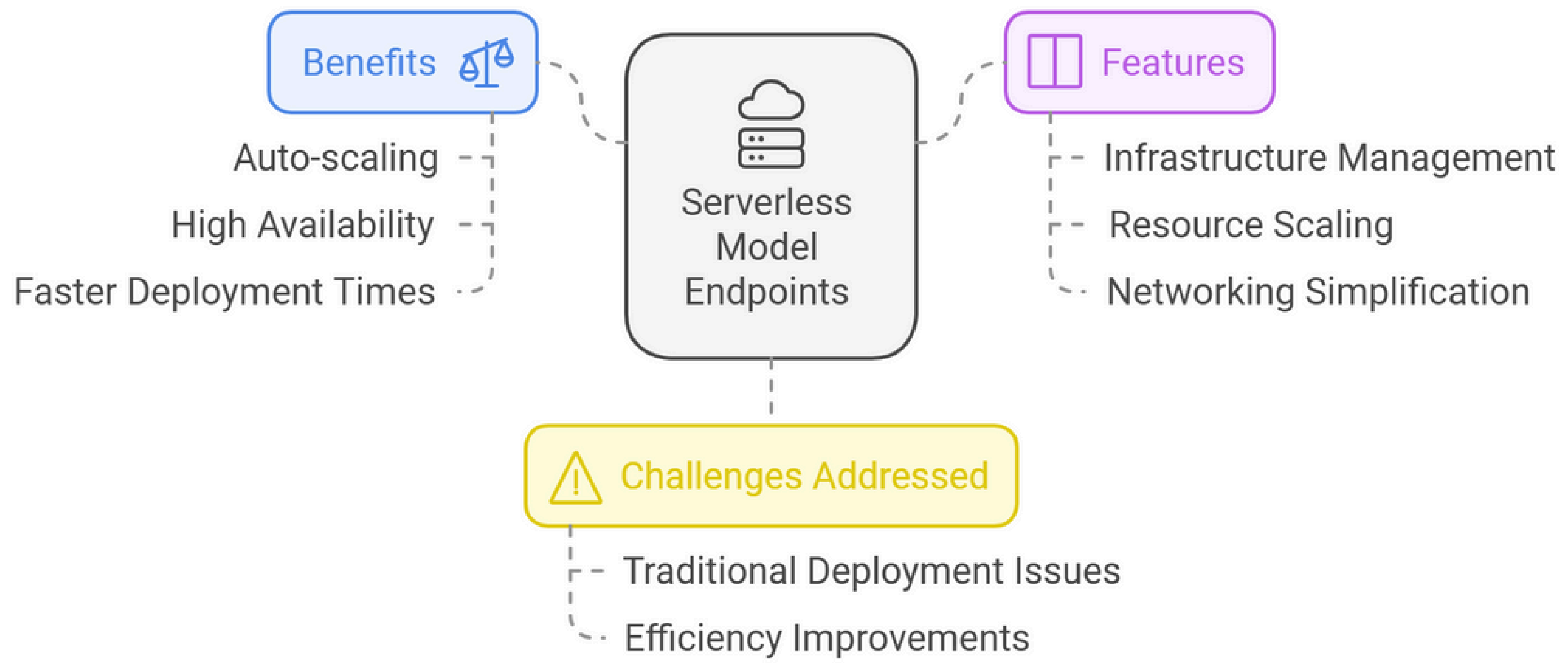
WEB SERVICE MODEL DEPLOYMENT

- Model is exposed as a web API
- Input data sent via HTTP requests
- Predictions returned as responses
- Allows integration with web and mobile applications
- Common frameworks: Flask, FastAPI
- Supports real-time prediction systems



SERVERLESS MODEL DEPLOYMENT

- No need to manage servers manually
- Automatically scales based on incoming requests
- Pay only for actual usage
- Common platforms: AWS Lambda, Google Cloud Functions
- Suitable for event-driven applications
- Low maintenance and fast deployment



LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

THANK YOU