

DATA SCIENCE & MACHINE LEARNING

From Raw Data to Deployed Models

by Delina Gholamipour

A practical course covering the full data science lifecycle — from exploratory analysis and preprocessing through classical models, ensembles, deep learning, evaluation, and transfer learning.

COURSE OVERVIEW • 11 TOPICS • AFTERNOON SESSION

Agenda — 11 Topics

1

Data Science Project Life Cycle

2

Exploratory Data Analysis (EDA)

3

Data Preprocessing

4

Data Modeling

5

Baseline Models (Linear & Logistic Regression)

6

Decision Trees

7

Ensemble Methods

8

MLP & Deep Learning — A Brief Introduction

9

Model Evaluation

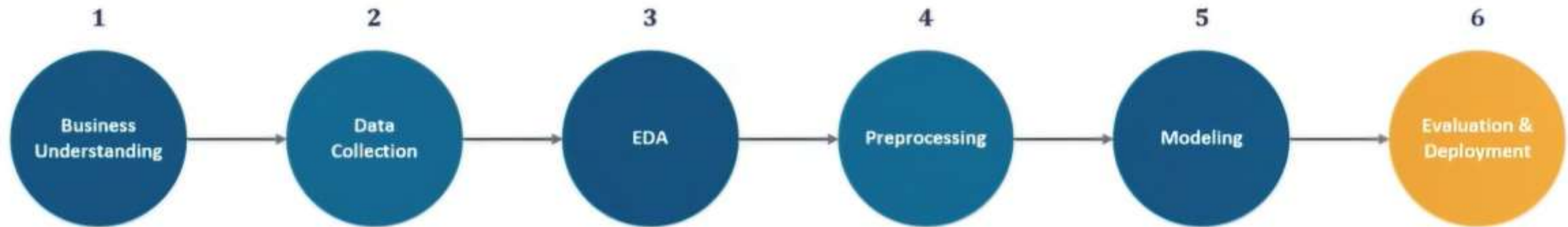
10

Cross Validation

11

Transfer Learning & Fine-Tuning

Data Science Project Life Cycle



The lifecycle is iterative, not linear — insights from evaluation routinely send teams back to preprocessing or even to reframing the business question.

Define the problem, success metrics, and constraints

Gather data from databases, APIs, sensors, or files

Explore & Prepare understand, clean, and transform the data

Model & Deploy train, validate, and ship into production

Exploratory Data Analysis (EDA)

1

Descriptive Statistics

Mean, median, variance, and distribution shape for every feature

2

Visualization

Histograms, boxplots, scatterplots, and correlation heatmaps

3

Missing Value Analysis

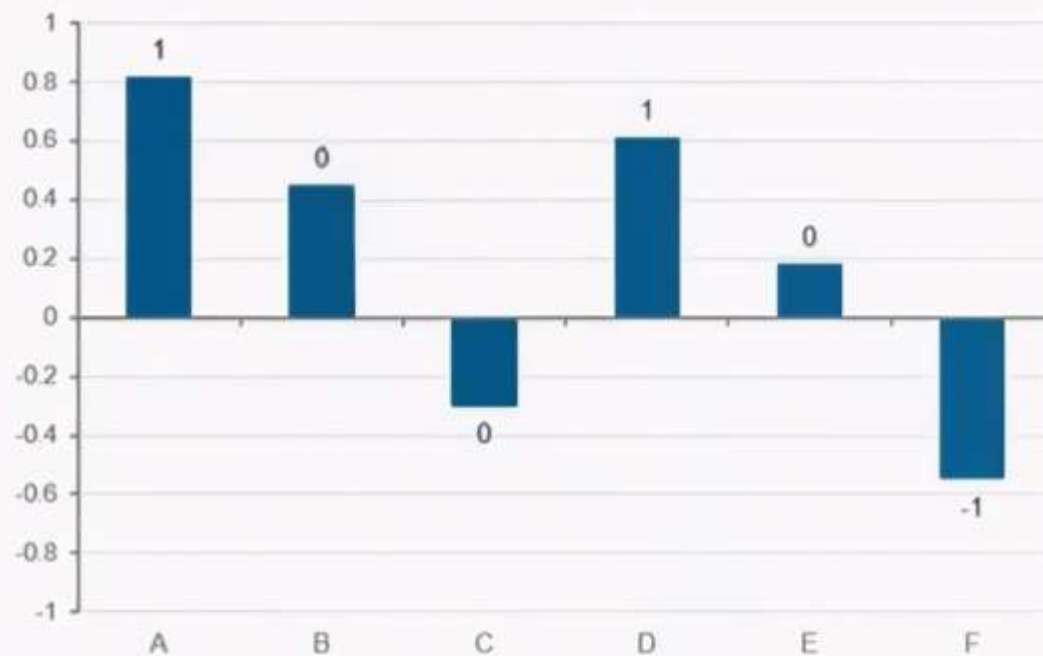
Identify patterns and mechanisms behind missing data

4

Outlier Detection

Flag anomalies that may distort model training

Correlation Snapshot



Data Preprocessing

MD

Missing Data

Imputation (mean/median/mode) or removal

FS

Feature Scaling

Standardization or min-max normalization

CE

Categorical Encoding

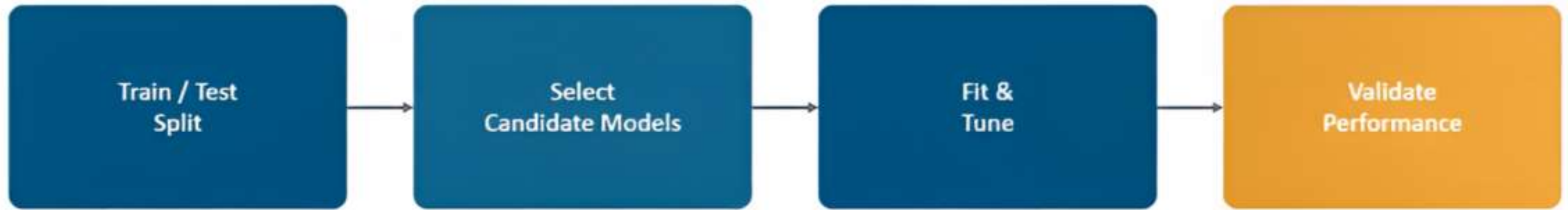
One-hot, label, or target encoding

FE

Feature Engineering

Creating, transforming, and selecting features

Data Modeling



Key Considerations

- Choose models appropriate to the problem type (regression, classification)
- Guard against data leakage between train and test sets
- Use hyperparameter search (grid / random / Bayesian) to tune models
- Iterate: modeling is rarely a single pass

Baseline Models

Linear Regression

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

Predicts a continuous target as a weighted sum of features.

Use when: the outcome is numeric (price, temperature, demand) and relationships are roughly linear.

Evaluated with: RMSE, MAE, R^2

Logistic Regression

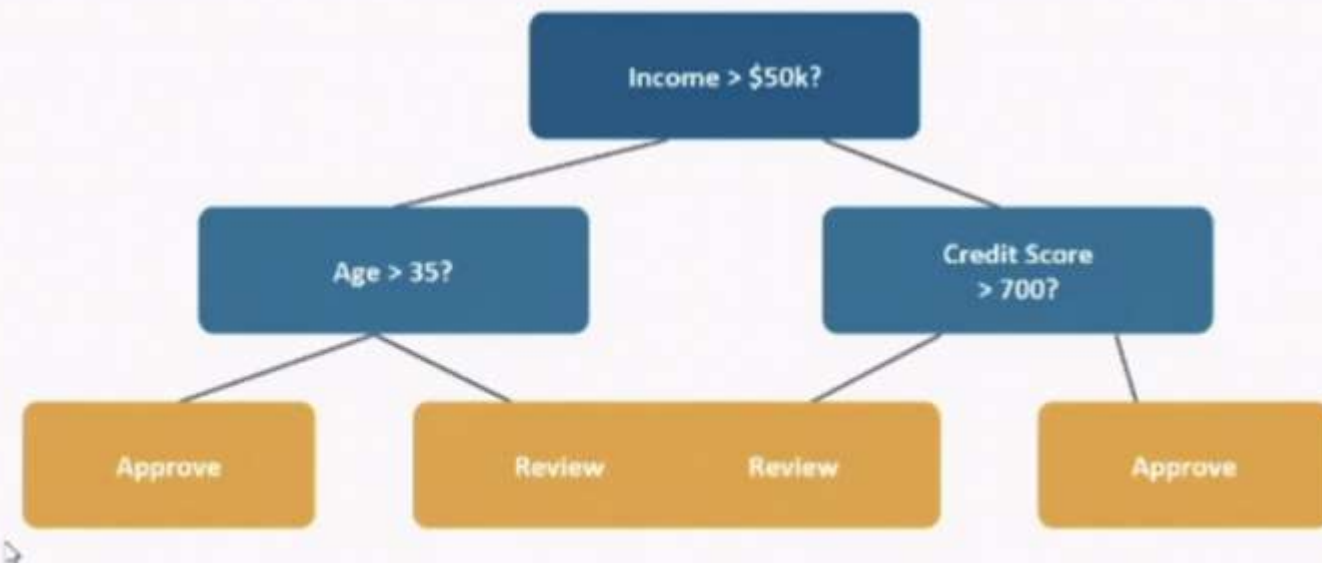
$$p = 1 / (1 + e^{-(\beta_0 + \beta_1 x_1 + \dots)})$$

Predicts the probability of a binary (or multi-class) outcome using a sigmoid transform.

Use when: the outcome is a class label (churn / no churn, fraud / not fraud).

Evaluated with: Accuracy, Precision, Recall, AUC

Decision Trees



Strengths

- Easy to interpret & visualize
- Handles non-linear relationships
- No feature scaling required

Watch Out For

- Prone to overfitting
- Unstable — small data changes reshape the tree
- Biased toward dominant classes

Splits are chosen to maximize information gain (or minimize Gini impurity) at every node.

Ensemble Methods

Bagging

Bootstrap Aggregating — parallel, variance reduction

Trains many models independently on random subsets of data, then averages (or votes on) their predictions to reduce variance and stabilize results.

Examples

Random Forest

Extra Trees

Boosting

Sequential — bias reduction

Trains models sequentially, where each new model focuses on correcting the errors of the previous ones — reducing bias and often achieving top accuracy.

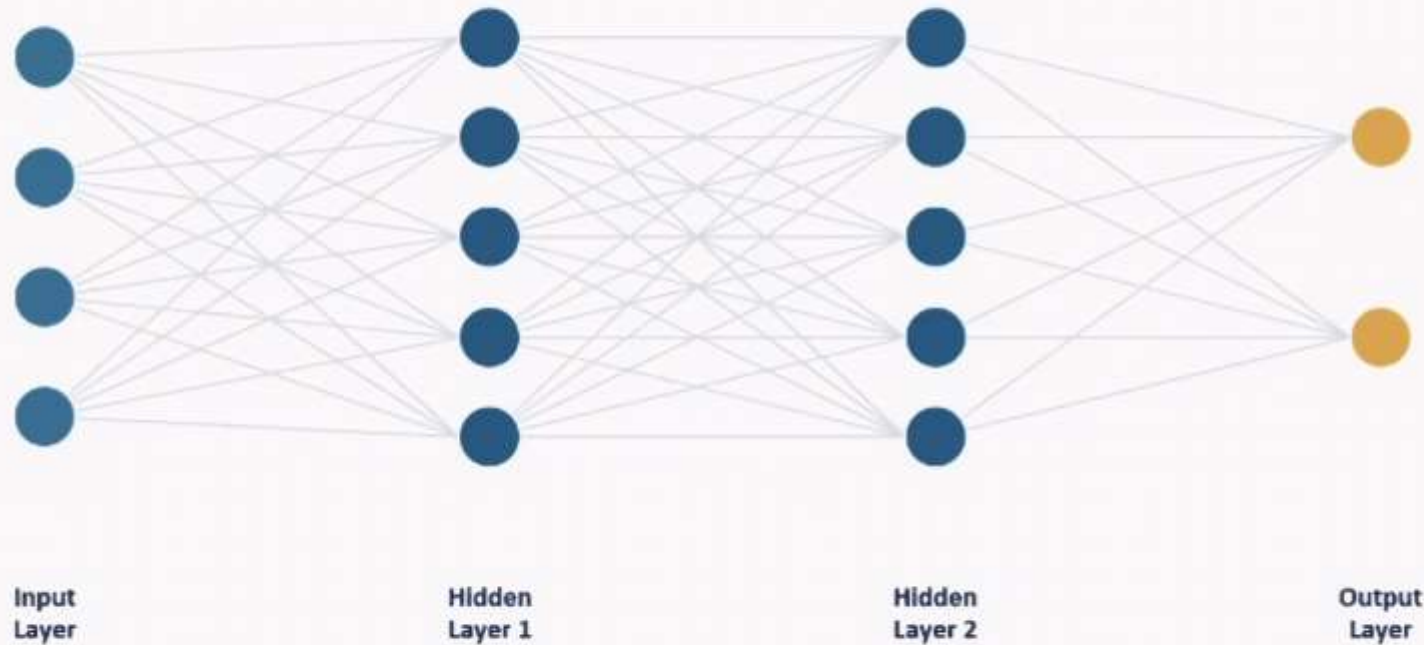
Examples

AdaBoost

Gradient Boost

XGBoost

MLP & Deep Learning — A Brief Introduction



Multilayer Perceptrons (MLPs) stack layers of neurons with non-linear activations, letting the network learn complex, non-linear patterns beyond what linear models capture.

Deep Learning extends this idea to many layers and specialized architectures:

- CNNs — images
- RNNs / LSTMs — sequences
- Transformers — language & beyond

Trained via forward pass → loss calculation → backpropagation → gradient-based weight updates (e.g., Adam, SGD), repeated over many epochs.

Model Evaluation

Classification

Accuracy

Overall correctness

Precision

Correct positive predictions

Recall

Positives correctly found

F1-Score

Balance of precision & recall

ROC-AUC

Separability across thresholds

Regression

MAE

Average absolute error

MSE / RMSE

Penalizes larger errors more

 R^2

Variance explained by the model

Always match the metric to the business question — e.g., recall matters more than precision when missing a positive case is costly (fraud, disease detection).

Cross Validation

K-Fold Cross Validation ($k = 5$)

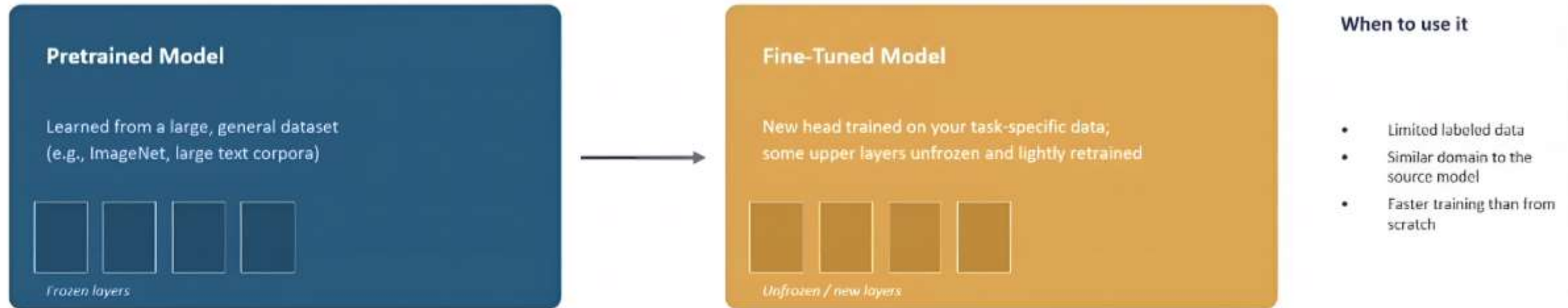
Run 1



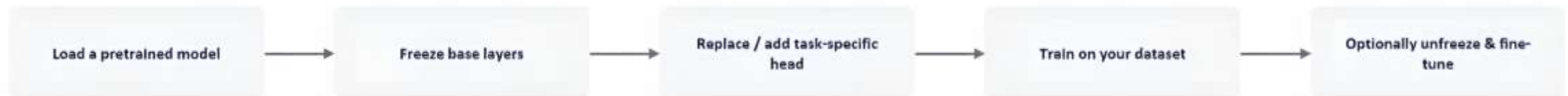
Each of the k folds takes a turn as the held-out test set while the model trains on the rest — every data point is used for both training and testing exactly once for evaluation.

Why it matters: gives a more reliable estimate of model performance than a single train/test split, and helps detect overfitting.

Transfer Learning & Fine-Tuning



Typical Workflow



Questions & Discussion

From lifecycle to fine-tuning — thank you for joining today's session.

DATA SCIENCE & MACHINE LEARNING COURSE