

Lecture 8:

ML Fundamentals for Biomedical Data

ML meets medicine - Success stories & Challenges

Introduction to Biomedical Data Science

Lecture Contents

Part 1: Biomedical Data Challenges

Part 2: ML Methods for Biomedical Applications

Part 3: Clinical ML and Validation

Part 1/3:

Biomedical Data Challenges

- Unique characteristics of biomedical data
- Statistical considerations
- Preprocessing strategies

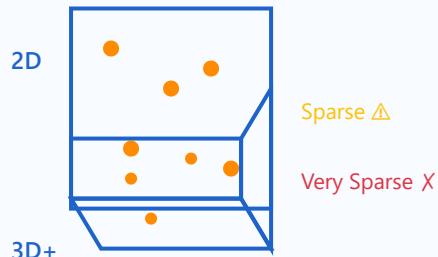
High Dimensionality

⚠ The Curse of Dimensionality

When features (P) greatly exceed samples (N): $P \gg N$

Data Sparsity in High-D

1D  Dense ✓



Overfitting Risks

Models memorize training data rather than learning generalizable patterns

Distance Metrics Fail

All points appear equidistant in very high dimensions

Sparse Data Space

Data points become increasingly sparse in high-dimensional space

Computational Cost

Training time and memory requirements grow exponentially

✓ Solutions & Strategies

Regularization (L1/L2)

Feature Selection

Dimensionality Reduction

Cross-validation

Domain Knowledge

Small Sample Sizes

Statistical Power Challenge

Limited samples reduce ability to detect true effects and increase risk of false discoveries



Cross-validation

K-fold, stratified, leave-one-out strategies for robust evaluation



Bootstrap Methods

Resampling techniques to estimate uncertainty and confidence intervals



Data Augmentation

Generate synthetic samples while preserving statistical properties



Transfer Learning

Leverage pre-trained models from larger datasets or related domains



Regularization

Penalize model complexity to prevent overfitting on small datasets

Domain Priors

Incorporate biological knowledge to guide model learning

Class Imbalance

⚠ The Accuracy Paradox

99% accuracy is useless if 99% of samples are negative!

Model predicting all negatives achieves high accuracy but zero clinical utility

Sampling Strategies

- Random oversampling
- Random undersampling
- SMOTE (Synthetic Minority)
- ADASYN (Adaptive Synthetic)

Cost-sensitive Learning

- Weighted loss functions
- Class weights in sklearn
- Focal loss for deep learning
- Penalize misclassifications differently

✓ Proper Evaluation Metrics

Precision

Recall

F1-score

PR-AUC

Balanced Accuracy

Matthews CC

Cohen's Kappa

G-mean

Missing Data

MCAR

Missing Completely At Random -
missingness unrelated to data



Random pattern

MAR

Missing At Random - related to observed
variables

Group A:

Group B:

Depends on group

MNAR

Missing Not At Random - related to
unobserved values

Low:

High:

Missing if high value

Imputation Strategies

Mean/Median

Simple but biased

K-NN

Uses similar samples

MICE

Multiple Imputation

MissForest

Random Forest based

EM Algorithm

Maximum Likelihood

Deep Learning

Autoencoders, GANs

Batch Effects

⚠ Technical Variation Problem

Non-biological variations from different labs, instruments, or time periods can overwhelm true biological signals

Correction Methods

- ComBat (most popular)
- Limma removeBatchEffect
- Harmony (single-cell)
- Seurat CCA integration
- Deep learning approaches

Best Practices

- Randomize across batches
- Include batch in study design
- Balance classes per batch
- Use supervised correction
- Validate on independent data

Feature Selection



Filter Methods

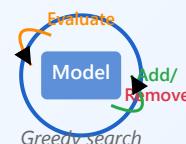


Independent of model

- Correlation analysis
- Chi-square test
- ANOVA F-test
- Mutual information



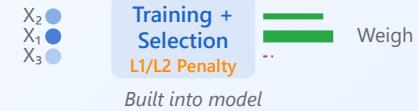
Wrapper Methods



- Forward selection
- Backward elimination
- Recursive Feature Elimination
- Genetic algorithms



Embedded Methods



Built into model

- Lasso (L1)
- Ridge (L2)
- Elastic Net
- Tree importance

✓ Advanced & Clinical Considerations

Stability Selection

Permutation Importance

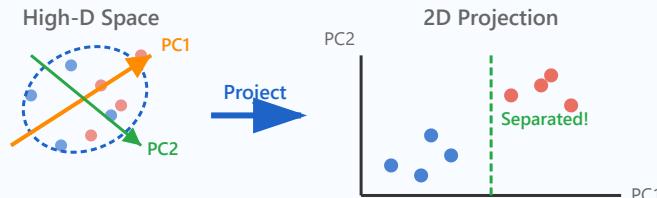
Clinical Interpretability

Domain Knowledge Integration

Dimensionality Reduction

PCA

Principal Component Analysis - linear transformation preserving maximum variance



Clinical Applications:

- Gene expression clustering
- Quality control visualization
- Batch effect detection

t-SNE

t-Distributed Stochastic Neighbor Embedding - nonlinear visualization method



Clinical Applications:

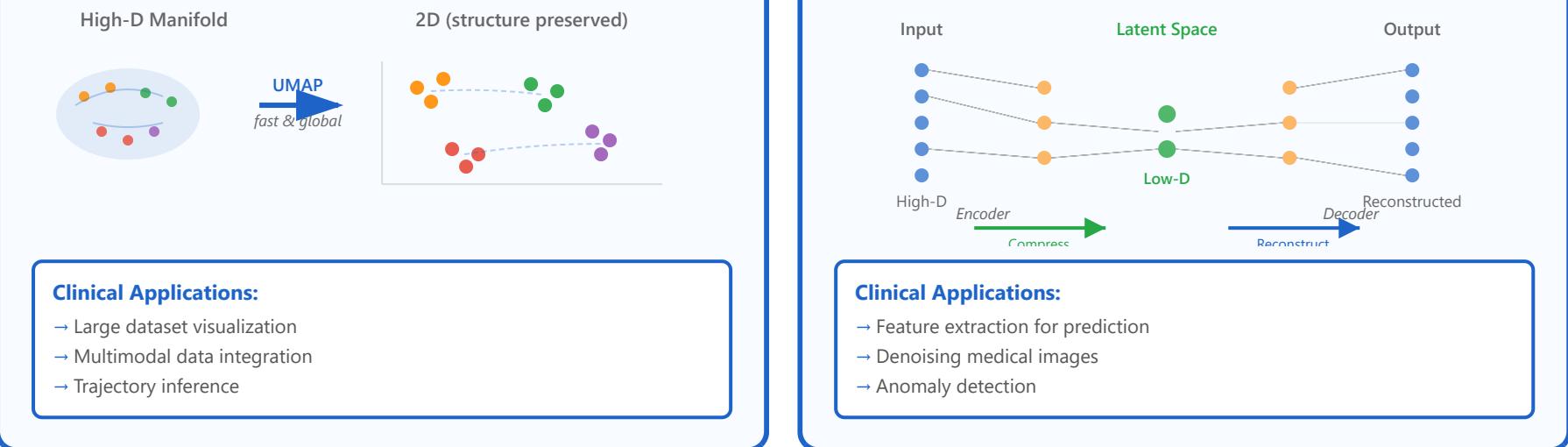
- Single-cell RNA-seq visualization
- Patient subtype discovery
- Exploratory data analysis

UMAP

Uniform Manifold Approximation - faster than t-SNE, preserves global structure

Autoencoders

Deep learning compression - learns nonlinear representations



Part 2/3:

ML Methods for Biomedical Applications

- Algorithm taxonomy
- Model selection strategies
- Performance evaluation
- Interpretability requirements

Supervised vs Unsupervised Learning



Supervised

Learn from labeled data - input-output pairs with known targets

Clinical Examples:

- Disease diagnosis (healthy/sick)
- Drug response prediction
- Survival time estimation
- Risk score calculation



Unsupervised

Discover patterns in unlabeled data - no predefined targets

Clinical Examples:

- Patient subtype discovery
- Gene expression clustering
- Anomaly detection
- Feature extraction



Hybrid Approaches

Semi-supervised Learning

Self-supervised Learning

Weakly-supervised Learning

Classification in Medicine

Common algorithms for diagnosis and prediction tasks

Logistic Regression

Linear model for binary/multi-class classification with probability outputs

- | | |
|------------------------|----------------------|
| ✓ Highly interpretable | ✗ Assumes linearity |
| ✓ Fast training | ✗ Limited complexity |

Random Forests

Ensemble of decision trees for robust, non-linear classification

- | | |
|-------------------------|----------------------|
| ✓ Handles non-linearity | ✗ Less interpretable |
| ✓ Feature importance | ✗ Can overfit |

Support Vector Machines

Maximum margin classifier with kernel tricks for non-linear boundaries

- | | |
|-------------------------|----------------------|
| ✓ Effective in high-dim | ✗ Slow on large data |
| ✓ Versatile kernels | ✗ Hard to interpret |

Neural Networks

Deep learning models for complex pattern recognition

- | | |
|-----------------------|--------------------|
| ✓ Highest performance | ✗ Black box |
| ✓ Automatic features | ✗ Needs large data |

Regression for Biomarkers

Predicting continuous outcomes - lab values, disease progression, dosages

Linear Regression

Simple, interpretable baseline model

Use case: Predicting HbA1c levels from patient features

Ridge / Lasso / Elastic Net

Regularized regression preventing overfitting

Use case: Gene expression → biomarker prediction

Random Forest Regression

Non-linear relationships, feature importance

Use case: ICU length of stay prediction

Gradient Boosting (XGBoost)

State-of-the-art performance on tabular data

Use case: Drug dosage optimization



Critical: Prediction Intervals

Clinical decisions require not just point estimates but confidence intervals - quantify uncertainty!

Clustering for Disease Subtypes

Discover hidden patient subgroups with distinct characteristics

K-means

Fast, scalable clustering with predefined number of clusters

Cancer subtypes from gene expression

Hierarchical Clustering

Dendrogram-based, no need to specify K upfront

Patient stratification visualization

DBSCAN

Density-based, finds arbitrary shapes, handles outliers

Anomaly detection in clinical data

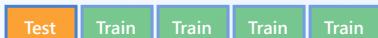
Consensus Clustering

Robust clustering through multiple runs and voting

Stable disease subtype identification

Cross-validation Strategies

K-Fold CV

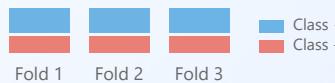


Split data into K folds, train on K-1, test on 1, repeat K times

Standard: K=5 or 10

Stratified CV

Class Balance Preserved



Preserve class distribution in each fold - crucial for imbalanced data

Use for classification

Leave-One-Out



Train on N-1 samples, test on 1, repeat N times

For very small N

Nested CV

Outer: Evaluation

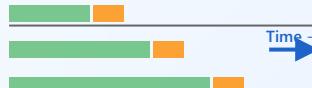


Unbiased performance estimate

Outer loop for evaluation, inner loop for hyperparameter tuning

Unbiased estimates

Time Series Split



Train on past, test on future - respects temporal order

For longitudinal data

Group CV



Keep all samples from same patient/site together

Prevent data leakage

⚠ Biomedical Pitfall

Multiple samples from same patient must stay in same fold to avoid optimistic bias!

Performance Metrics

Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)
		Correct	Errors

Sensitivity (Recall)

$$\text{TP} / (\text{TP} + \text{FN})$$

How many actual positives detected

Specificity

$$\text{TN} / (\text{TN} + \text{FP})$$

How many actual negatives identified

PPV (Precision)

$$\text{TP} / (\text{TP} + \text{FP})$$

Positive predictive value

NPV

$$\text{TN} / (\text{TN} + \text{FN})$$

Negative predictive value

F1 Score

$$2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$$

Harmonic mean of Precision & Recall

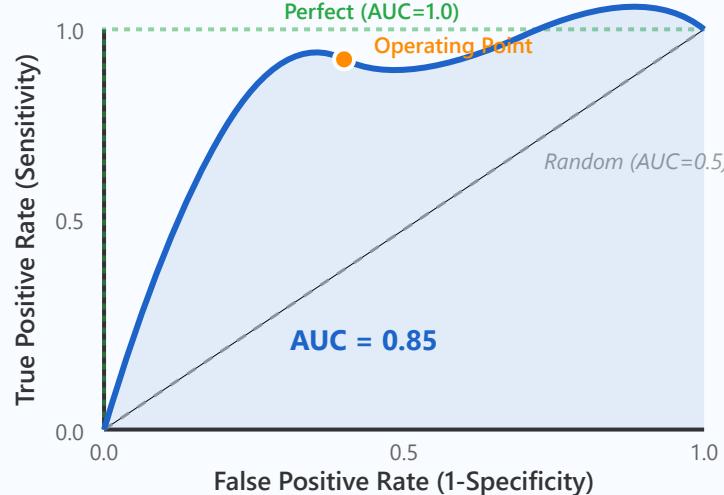
Matthews CC

Balanced measure even for imbalanced data

Range: -1 to +1, 0 = random

ROC and PR Curves

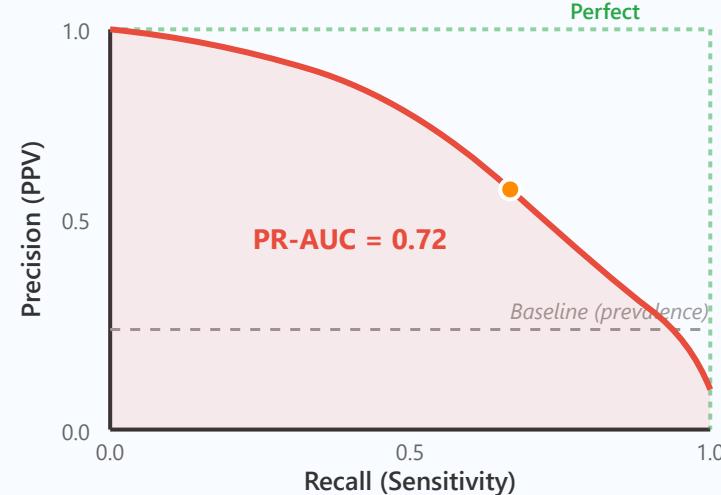
ROC Curve



Plots TPR vs FPR at various thresholds

- AUC: 0.5 = random, 1.0 = perfect
- Good for balanced datasets
- Threshold-independent metric

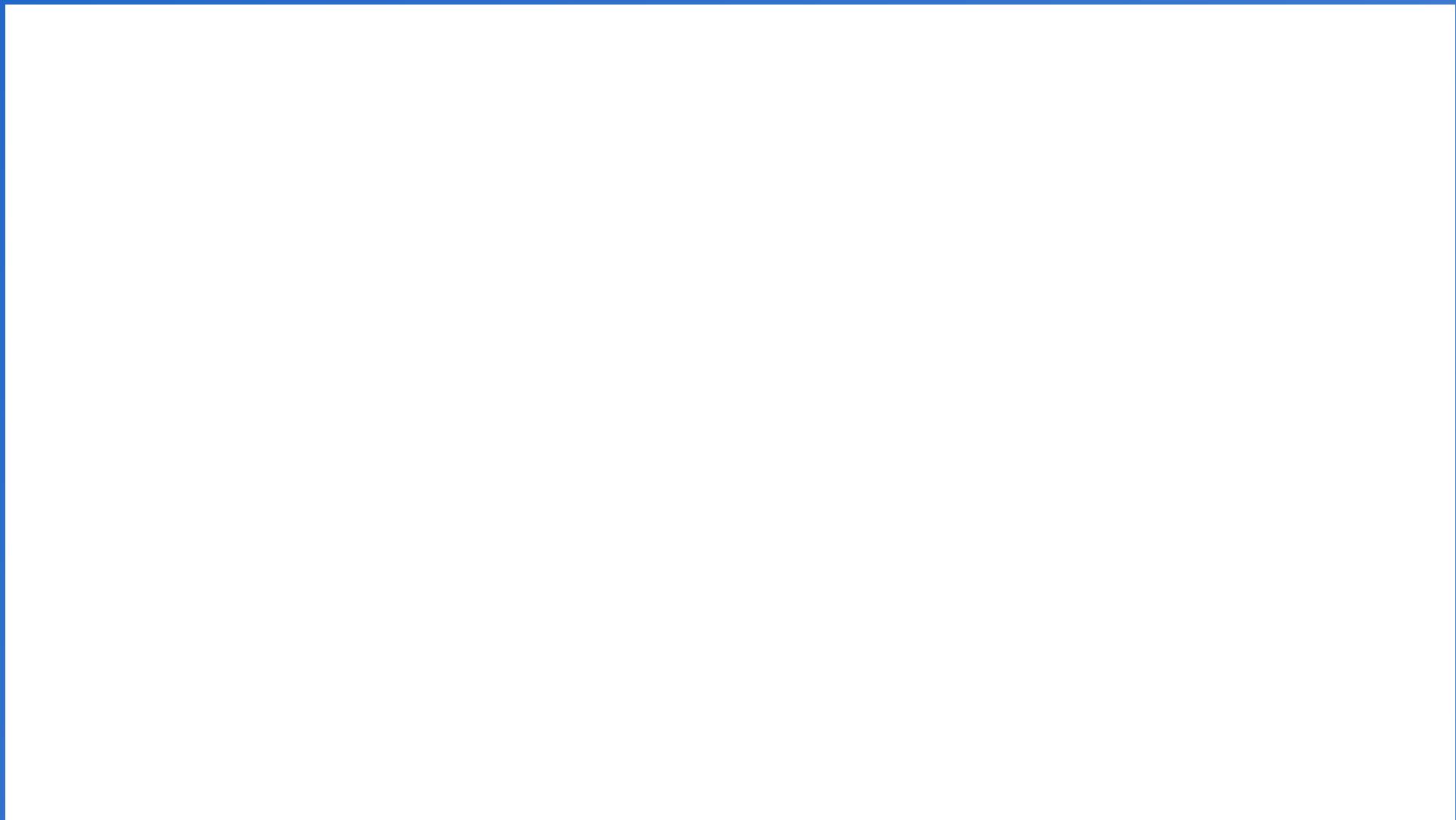
PR Curve



Plots Precision vs Recall

- Better for imbalanced data
- Focus on positive class
- More informative for rare diseases

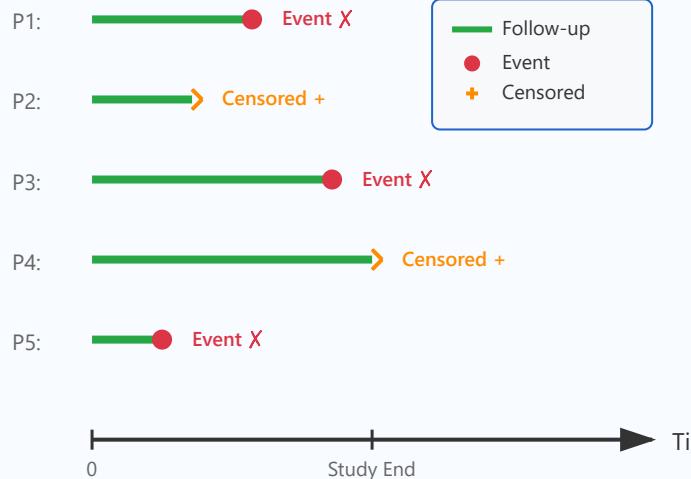
Clinical Decision: Choose operating point based on cost of FP vs FN



Survival Analysis

Time-to-event analysis: Death, disease recurrence, hospital readmission

Censoring Visualization



Censoring Types

- Right censoring (most common)
- Left censoring
- Interval censoring

Key Functions

- Survival function $S(t)$
- Hazard function $h(t)$
- Cumulative hazard $H(t)$

Advanced Topics

- Competing risks
- Recurrent events
- Time-varying covariates

Clinical Applications

- Overall survival (OS)
- Progression-free survival (PFS)
- Time to treatment failure

Cox Proportional Hazards Model

Gold standard for survival analysis in clinical research

Model Formula

$$h(t|X) = h_0(t) \cdot \exp(\beta_1 X_1 + \dots + \beta_p X_p)$$

Hazard ratio = $\exp(\beta)$

Hazard Rate Over Time



Key Assumptions

- **Proportional hazards:** HR constant over time
- **Linear relationship:** with log-hazard
- **Independent censoring:** uninformative

Testing PH Assumption



Time-varying Covariates

Handle changing predictors

Stratification

When PH assumption violated

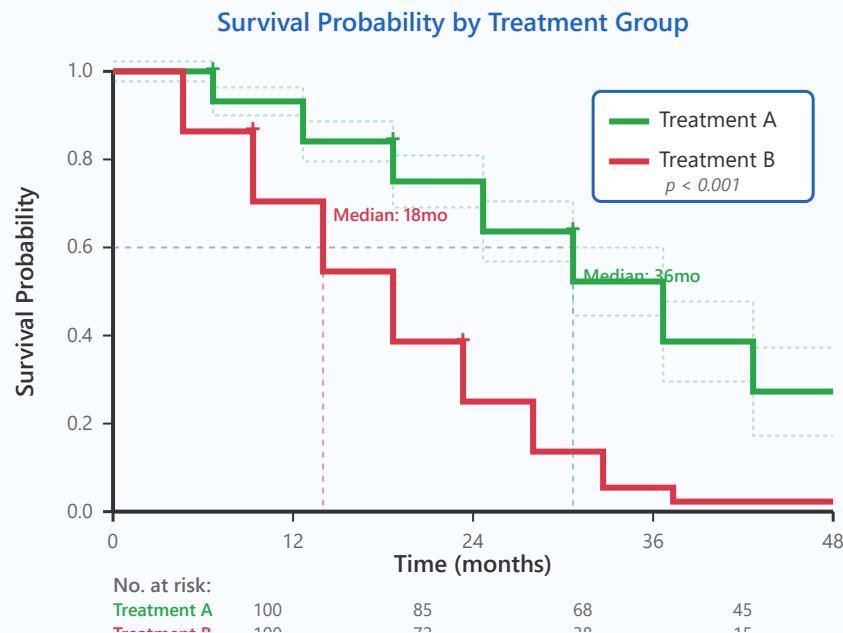
Penalized Cox

High-dimensional data

Kaplan-Meier Survival Curves

Non-parametric estimator of survival function

KM Curve Example



Key Components

- Step function visualization
- Confidence intervals (95% CI)
- Number at risk table
- Censoring markers (+)
- Median survival time

Log-rank Test

- Compare survival curves
- Null: no difference in survival
- P-value < 0.05 = significant
- Non-parametric test
- Assumption: PH holds

Clinical Interpretation: Always report median survival, 95% CI, and number at risk at key timepoints

Time-to-Event Prediction with ML



Random Survival Forests

Ensemble method, handles non-linearity, feature importance



DeepSurv

Neural networks for survival, learns complex patterns



Discrete Time Models

Logistic regression per time interval

Evaluation Metrics

C-index (Concordance)

Time-dependent AUC

Calibration plots

Brier score

Clinical Risk Scores

Translate complex models into simple, actionable scoring systems

Development Steps

1. Select predictors (clinical + statistical)
2. Fit regression model
3. Convert coefficients to points
4. Create nomogram or web calculator
5. Validate externally

Validation Requirements

- Calibration plots (predicted vs observed)
- Decision curve analysis
- External validation cohort
- Sensitivity/Specificity at cutoffs
- Net benefit assessment

✓ Clinical Examples

APACHE II (ICU mortality) • Framingham Risk Score (CVD) • MELD Score (liver transplant) • GRACE Score (ACS)

Interpretable ML for Clinical Adoption



Black Box Problem: Clinicians won't trust models they can't understand

Glass Box Models

- Linear/Logistic Regression
- Decision Trees
- GAMs (Generalized Additive Models)
- Rule-based systems

Inherently interpretable

Post-hoc Explanations

- Feature Importance (RF, XGBoost)
- Partial Dependence Plots
- LIME (Local Interpretable)
- SHAP Values

Explain black boxes

Clinical Acceptance Requires: Model explanation + Clinical validation + Physician trust

SHAP Values for Model Interpretation

SHapley Additive exPlanations - unified framework for interpretability

Shapley Theory

- From game theory (Nobel Prize)
- Fair contribution of each feature
- Considers all feature combinations
- Unique solution with nice properties

SHAP Algorithms

- TreeSHAP (fast for trees)
- DeepSHAP (for neural nets)
- KernelSHAP (model-agnostic)
- LinearSHAP (for linear models)

Visualization Types

Waterfall Plots

Single prediction

Summary Plots

Global importance

Dependence Plots

Feature effects

Interaction Plots

Feature interactions

Clinical Validation Framework



Internal Validation

Cross-validation
Bootstrap
Same institution data



External Validation

Different hospitals
Different populations
Geographic diversity



Prospective Studies

Real-time predictions
Clinical workflow
RCT if possible



Real-World Performance

Models often degrade when deployed - monitor performance continuously!

FDA Requirements for Medical AI: Multi-site validation, diverse populations, clinical outcomes

Hands-on: scikit-learn for Biomedical Data

Pipeline Creation

```
from sklearn.pipeline import Pipeline
pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('selector', SelectKBest()),
    ('clf', LogisticRegression())
])
```

Model Selection

```
from sklearn.model_selection import
    GridSearchCV
gs = GridSearchCV(pipe, params,
    cv=StratifiedKFold(5),
    scoring='roc_auc')
```

Practice Tasks

- Load biomedical dataset
- Handle missing values
- Scale features appropriately
- Deal with class imbalance

- Build classification pipeline
- Perform nested CV
- Generate evaluation report
- Plot ROC and PR curves

Hands-on: lifelines for Survival Analysis

Python library for survival analysis in clinical research

Kaplan-Meier

```
from lifelines import KaplanMeierFitter  
kmf = KaplanMeierFitter()  
kmf.fit(T, E, label='Group A')  
kmf.plot_survival_function()
```

Cox Regression

```
from lifelines import CoxPHFitter  
cph = CoxPHFitter()  
cph.fit(df, 'T', 'E')  
cph.print_summary()
```

Practice Tasks

Load clinical trial data

Plot KM curves by treatment

Fit Cox model

Test PH assumption

Calculate C-index

Make predictions

Thank You!

Questions? Let's discuss!

Next: Practice and case studies

Office Hours: By appointment