Machine Learning for survival analysis

M2 Données massives en santé

Juliette Murris

Januray 2025

Outline

Introduction

Introduction

- Statistics and Machine Learning (ML) are distinct but complementary disciplines.
- **Statistics:** Focuses on modeling relationships between explanatory variables and the outcome variable, often relying on strong assumptions about data distributions.
- **Machine Learning:** Emphasizes accurate prediction of a target value, often relaxing distributional assumptions.
- Breiman (2001) advocated for integrating the two approaches to leverage their respective strengths for optimal data analysis and prediction.

Machine Learning for prediction

- A machine learning approach automatically learns patterns from training data to predict outcomes.
- Example: Predicting post-operative complications based on historical data using ML.
- Advantages:
  - Shorter and more maintainable models.
  - Ability to adapt to new data or changing risk factors.
- ML excels where traditional rule-based approaches fail:
  - Tasks too complex for manual algorithms.
  - Scenarios without predefined algorithms.

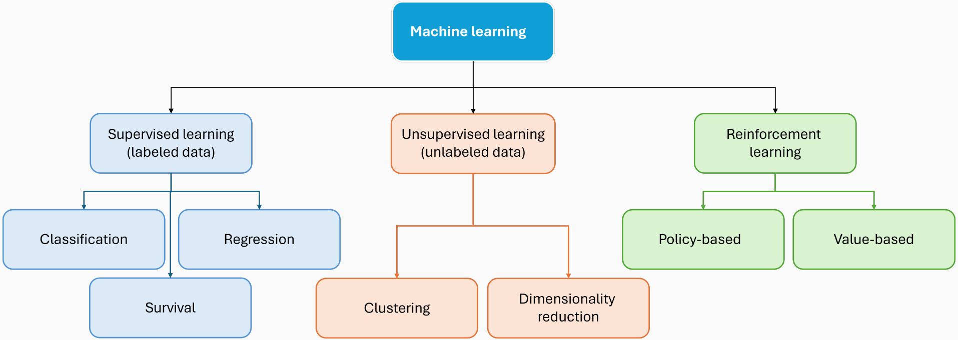
Types of learning in Machine Learning

Definition: An algorithm is a set of rules to achieve an objective function f. A machine learning model $\hat{f}$ associates input data (X) with predictions $\hat{y}$ (Mitchell 1997).

Four types of learning (common in medical research, Hastie 2009):

- **Supervised Learning:** Learn from labeled data (*e.g.*, cancer diagnosis prediction, Yaqoob 2023)
- **Unsupervised Learning:** Discover patterns in unlabeled data (*e.g.*, symptom clustering, Xu 2023)
- **Semi-supervised Learning:** Combine labeled and unlabeled data
- **Reinforcement Learning:** Sequential decision-making from interactions (*e.g.*, therapy optimization, Padmanabhan 2017)

Types of learning in Machine Learning



Why Machine Learning for survival data?

- Clinical datasets are increasingly large and complex.
- ML is adept at identifying patterns and relationships, such as:
  - Discovering biomarkers or genetic signatures.
  - Profiling subgroups of patients for personalized medicine.
- Example: Predicting survival risk or recurrence in cancer treatment.

Why Machine Learning for survival data?

- Traditional Cox proportional hazard (CPH) models:
  - **Strengths**: Easy to implement, interpretable, fast computation.
  - **Limitations**:
    - Assumes proportional hazards.
    - Ineffective for non-linear and interaction effects.
    - Assumes no correlation among explanatory variables.
- Machine Learning (ML) models address these limitations:
  - Capture non-linearities and interactions.
  - Handle high-dimensional and censored data.

Regularization

Penalized Regressions for Survival Analysis

Why Penalized Regressions?

- Reduces overfitting in high-dimensional datasets.
- Selects important variables and handles multicollinearity.

Key Penalization Techniques:

Model	Penalty Function
LASSO-Cox	$\lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$
Ridge-Cox	$\frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2$
Elastic-Net (EN)-Cox	$\lambda \left(\alpha \sum_{j=1}^{p} \lvert\beta_j\rvert + \frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right)$

Coding regularized Cox models

In Python:

- `lifelines`:
  - Provides implementation for Cox proportional hazards model.
  - Supports LASSO, Ridge, and Elastic-Net regularization via `CoxPHFitter`.
- `scikit-survival`:
  - Extends `scikit-learn` for survival analysis.
  - Offers support for regularized Cox models using `CoxnetSurvivalAnalysis`.

In R:

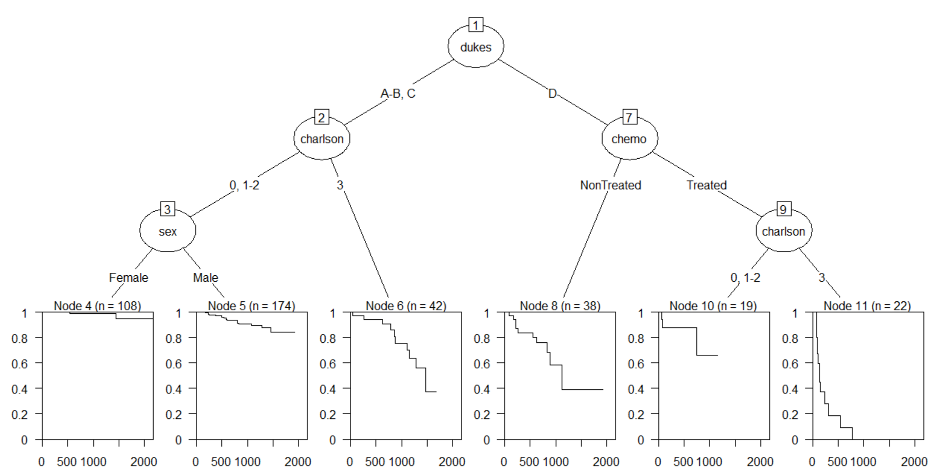
- `glmnet`:
  - Provides elastic-net regularization for Cox proportional hazards models.
  - Supports both LASSO and Ridge penalties through parameter tuning.
- `survival`:
  - A foundational package for survival analysis in R.
  - Can be paired with `glmnet` for penalized regression.

TP2

TP2

Survival decision trees

Decision trees for survival analysis



Decision trees for survival analysis

Structure:

- Partition data into homogeneous groups based on explanatory variables.
- Nodes represent decisions; terminal nodes provide survival estimates.

Advantages:

- Easy to interpret and visualize.
- Captures non-linear relationships.

Limitations:

- Deep trees can overfit the data.
- Limited generalization ability.

Survival ensembles

Ensemble methods for survival data

Why Ensembles?

- Overcome instability of single decision trees.
- Combine multiple models to improve accuracy and generalization.

Key Methods:

- **Bagging:**
  - Use bootstrap samples to create multiple trees.
  - Aggregate predictions using averages or voting.
- **Random Survival Forests (RSF):**
  - Use random subsets of variables at each split to reduce correlation.
  - Combine predictions from survival trees.

Boosting for survival data

What is Boosting?

- Combine weak learners sequentially to correct errors iteratively.
- Final model is a weighted sum of individual learners.

Key Models:

- **CoxBoost:**
  - Boosting based on residuals of the Cox model.
  - Incorporates regularization to prevent overfitting.
- **Gradient Boosted Trees:**
  - Sequentially adjust survival trees to minimize prediction errors.

Coding survival ensemble methods

In Python:

- `scikit-survival`:
  - Implements ensemble methods like Random Survival Forests (RSF) and survival boosting.
- `xgboost`:
  - Provides gradient boosting for survival analysis with custom objective functions for censored data.
  - Often used for survival boosting tasks.

In R:

- `randomForestSRC`:
  - Implements Random Survival Forests for survival analysis.
  - Allows for high-dimensional survival prediction with variable importance estimation.
- `caret`:
  - Used for cross-validation and model tuning when applying ensemble methods.

Survival SVM

Survival Support Vector Machines (SSVMs)

- Construct hyperplanes to separate data in high-dimensional spaces.
- Maximize the margin between classes or predicted survival times.

Approaches for survival data:

- **Regression SVMs:** Predict survival times directly but ignore censored data.
- **Ranking SVMs:** Rank patients by survival times, accounting for censored data.
- **Survival SVMs:** Combine regression and ranking for censored data with custom loss functions.

Coding Survival SVM

In Python:

- `scikit-survival`:
  - Implements survival regression models using SVMs.
  - Includes `CoxPH` and other survival methods, with support for SVM-based models for survival prediction.

In R:

- `survivalsvm`:
  - A package for advanced kernel methods, including support vector machines.

Deep learning for survival data

What are Deep Survival Models?

- Leverage deep learning techniques to model survival data.
- Overcome limitations of traditional survival models by capturing:
  - Non-linear relationships.
  - Complex interactions among features.

Deep learning for survival data

DeepSurv:

- A deep neural network generalization of the Cox proportional hazards model.
- Predicts risk scores based on input features.

DeepHit:

- Models the joint distribution of survival times and events.
- Uses a neural network to predict probabilities for multiple competing risks.

Deep learning for survival data

Advantages:

- Handles high-dimensional and complex data effectively.
- Adapts to various types of censoring and competing risks.

Challenges:

- Requires large datasets for training.
- Less interpretable than traditional models.

Feature importance

Feature importance

- Supervised learning: **Predicting** a known outcome
- In healthcare, the question is often broader than simple prediction. We want to understand **why** the model produces a certain result.
  We aim to identify:
    - Risk factors
    - Prognostic factors
    - Predictive factors

Permutation Feature Importance (PFI): Principle

Steps to evaluate the importance of variable X_i:

1. Implement the predictive model
2. Calculate the model's error on the original dataset err_{ref}

Permutation Feature Importance (PFI): Principle

Steps to evaluate the importance of variable X_i:

1. Implement the predictive model
2. Calculate the model's error on the original dataset err_{ref}
3. Create a new dataset by permuting the data of variable X_i

Permutation Feature Importance (PFI): Principle

Steps to evaluate the importance of variable X_i:

1. Implement the predictive model
2. Calculate the model's error on the original dataset err_{ref}
3. Create a new dataset by permuting the data of variable X_i $\rightarrow$ This breaks any potential relationship between variable X_i and the target variable

Permutation Feature Importance (PFI): Principle

Steps to evaluate the importance of variable X_i:

1. Implement the predictive model
2. Calculate the model's error on the original dataset err_{ref}
3. Create a new dataset by permuting the data of variable X_i $\rightarrow$ This breaks any potential relationship between variable X_i and the target variable
4. Calculate the model's error on the new dataset err

Permutation Feature Importance (PFI): Principle

Steps to evaluate the importance of variable X_i:

1. Implement the predictive model
2. Calculate the model's error on the original dataset err_{ref}
3. Create a new dataset by permuting the data of variable X_i $\rightarrow$ This breaks any potential relationship between variable X_i and the target variable
4. Calculate the model's error on the new dataset err
5. Calculate the difference of the two performances $err_{ref} - err$

Repeat steps 3 to 5

Permutation Feature Importance (PFI): Principle

- It is advisable to repeat steps 3 to 5 **multiple times** to obtain an average effect and a confidence interval;
- Computationally inexpensive: the model is only built once, and predictions are repeated on the different datasets;
- Variable importance can be assessed on the training set or on the test set;
- Easy to implement and available in many programming languages;
- Note that interactions between variables are not taken into account;

Conclusion

Summary

Key Takeaways:

- Machine learning methods address limitations of traditional survival models.
- Penalized regressions, SVMs, decision trees, and ensemble methods are effective for survival data.

TP3

TP3