1 Regression

1.1 Linear Model (OLS)

- **Core concept**
  Additive linear relationship between the response and covariates, estimated by minimizing the mean squared error.
- **Mathematical idea**: $\hat{\beta} = \arg\min_\beta \sum_{i=1}^n (y_i - X_i\beta)^2$

Pros	Cons
– Maximum interpretability – Closed-form solution (fast) – Strong baseline – Statistical inference available (p-values, CI)	– Multicollinearity – Overfitting when $p \approx n$ or $p > n$ – Cannot capture non-linear relationships

- **When to use**
  - Small to medium-sized datasets
  - Explanatory objectives
  - Approximately linear relationships
- **Note**. Mandatory baseline: if a complex model does not outperform OLS, it is not justified.

1.2 Ridge Regression

- **Core concept**
  OLS with L2 regularization to reduce variance.
- **Mathematical idea**: $\min_\beta \sum (y_i - X_i\beta)^2 + \lambda \sum \beta_j^2$

Pros	Cons
– Handles multicollinearity – Reduces overfitting – Performs well with many correlated predictors	– No variable selection – Reduced interpretability

- **When to use**
  - Many correlated predictors
  - Large p with all predictors potentially relevant
- **Note**. Ridge shrinks, it does not select.

1.3 Lasso

- **Core concept**
  OLS with L1 regularization: automatic variable selection.
- **Mathematical idea**: $\min_\beta \sum (y_i - X_i\beta)^2 + \lambda \sum |\beta_j|$

Pros	Cons
– Sparsity – Automatic feature selection – More interpretable models	– Unstable with correlated predictors – Bias on large coefficients

- **When to use**

- $p \gg n$
- Goal: identification of key variables

- **Note**. Lasso tends to select one among correlated predictors, often arbitrarily.

1.4 Adaptive Lasso

- **Core concept**
  Lasso with adaptive weights: important variables are penalized less.
- **Mathematical idea**: $\min_\beta \sum(y_i - X_i\beta)^2 + \lambda \sum w_j|\beta_j|$ (with w_j estimated from an initial model)

Pros	Cons
– Oracle properties – More stable selection – Improves over standard Lasso	– More complex – Depends on the initial model

- **When to use**
  - Statistically consistent variable selection
  - High dimensionality with inference
- **Note**. This is Lasso "done properly".

1.5 Elastic Net

- **Core concept**
  Compromise between Ridge and Lasso.
- **Mathematical idea**: $\min_\beta \sum(y_i - X_i\beta)^2 + \lambda \left[(1-\alpha)\sum \beta_j^2 + \alpha \sum |\beta_j|\right]$

Pros	Cons
– Group selection of correlated variables – Very stable – Strong predictive performance	– Two hyperparameters – Reduced interpretability

- **When to use**
  - Many correlated predictors
  - Goal: performance and parsimony
- **Note**. Modern default for penalized regression.

1.6 Regression Tree

- **Core concept**
  Recursive partitioning of the covariate space.
- **Mathematical idea**: Minimizes within-node variance $\sum_m \sum_{i \in R_m} (y_i - \bar{y}_{R_m})^2$

Pros	Cons
– Extremely interpretable – Captures non-linearities and interactions – No distributional assumptions	– High variance – Overfitting – Poor accuracy for a single tree

- **When to use**
  - Exploratory analysis
  - Communication with non-technical audiences
- **Note**. A single tree is pedagogical, not competitive.

1.7 MARS (Multivariate Adaptive Regression Splines)

- **Core concept**
  Piecewise regression with adaptive spline basis functions.
- **Mathematical idea**: $f(x) = \beta_0 + \sum \beta_m h_m(x)$ where h_m are hinge functions.

Pros	Cons
– Captures non-linearities – Interpretable – Automatic model construction	– Limited scalability – Less powerful than modern boosting methods

- **When to use**
  - Moderate non-linearities
  - Goal: functional interpretability
- **Note**. Bridge between parametric models and machine learning.

1.8 PPR (Projection Pursuit Regression)

- **Core concept**
  Sum of univariate functions applied to linear projections of the data.
- **Mathematical idea**: $y = \sum_{k=1}^{K} g_k(a_k^T x)$

Pros	Cons
– Captures complex structures – Implicit dimensionality reduction	– Poor interpretability – Complex optimization

- **When to use**
  - Complex non-linear relationships
  - Medium-sized datasets
- **Note**. Historically important, rarely used today.

1.9 Random Forest

- **Core concept**
  Ensemble of independent trees trained on bootstrap samples.
- **Mathematical idea**: Averaging $\hat{f}(x) = \frac{1}{B} \sum f_b(x)$

Pros	Cons
– Very robust – Captures interactions – Strong out-of-the-box performance	– Black-box model – Computationally heavy

- **When to use**
  - General-purpose problems
  - No extensive feature engineering required
- **Note**. The workhorse of tabular machine learning.

1.10 Ranger RF

- **Core concept**
  Optimized Random Forest implementation (C++).
- **Mathematical idea**: Same as RF, more efficient implementation.

Pros	Cons
– Very fast – Scalable – Supports weights and survival analysis	– Same interpretability limitations as RF

- **When to use**
  - Large datasets
  - Production environments
- **Note**. "Industrial-grade" Random Forest.

1.11 XGBoost

- **Core concept**
  Gradient Boosting with decision trees.
- **Mathematical idea**: Minimizes $\sum l(y_i, \hat{y}_i) + \sum \Omega(f_k)$

Pros	Cons
– Very high accuracy – Native handling of missing values – Built-in regularization	– Many hyperparameters – Sensitive to tuning

- **When to use**
  - Competitions
  - Complex predictive tasks
- **Note**. Performance prioritized over interpretability.

1.12 LightGBM

- **Core concept**
  Gradient Boosting with leaf-wise tree growth.
- **Mathematical idea**: Same objective as XGBoost, different tree growth strategy.

Pros	Cons
– Extremely fast – Highly scalable – Excellent for large datasets	– More unstable – Risk of overfitting if not properly controlled

- **When to use**
  - Large datasets
  - High-dimensional feature spaces
- **Note**. XGBoost optimized for speed.

1.13 Summary

- Explanation: OLS / Lasso / Elastic Net / MARS
- Feature selection: Lasso / Adaptive Lasso
- Moderate non-linearity: MARS / Tree
- Maximum tabular performance: RF / XGBoost / LightGBM
- Production: Ranger / LightGBM

1.14 Comparison

Model	**Linearity**	**Feature selection**	**Interpretability**	**Robustness**	**Performance**	**Scalability**
Linear Model	⋆⋆⋆	×	⋆⋆⋆	⋆	⋆	⋆⋆⋆
Ridge	⋆⋆⋆	×	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆
Lasso	⋆⋆⋆	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆
Adaptive Lasso	⋆⋆⋆	⋆⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆
Elastic Net	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
Regression Tree	×	⋆⋆	⋆⋆⋆	⋆	⋆	⋆⋆
MARS	×	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆
PPR	×	×	⋆	⋆	⋆⋆	⋆
Random Forest	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆
Ranger RF	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
XGBoost	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆
LightGBM	×	⋆	⋆	⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆

1.15 Context-based model selection

Main requirement	**Recommended model**	**Motivation**
Explanatory baseline	Linear Model	Full transparency
Multicollinearity	Ridge	Stabilizes coefficients
Variable selection	Lasso	Automatic sparsity
Consistent selection	Adaptive Lasso	Oracle properties
Correlated predictors + selection	Elastic Net	Group selection
Clear decision rules	Regression Tree	Interpretability
Interpretable non-linearity	MARS	Piecewise functions
Complex structures	PPR	Projections
"Safe" model	Random Forest	Robust out-of-the-box
Fast RF / big data	Ranger RF	C++ implementation
Top performance	XGBoost	Boosting + regularization
Big data / speed	LightGBM	Leaf-wise growth

1.16 Bias–Variance trade-off

Model	**Bias**	**Variance**
Linear Model	High	Low
Ridge	Medium	Low
Lasso	Medium	Medium
Elastic Net	Medium	Medium
Tree	Low	High
Random Forest	Low	Low
Boosting	Very low	Medium

2 Classification

2.1 Multinomial Logistic Regression

- **Basic concept**
  Extension of binary logistic regression to multiple classes, with linear decision boundaries in the covariate space.
- **Mathematical idea**: $P(Y = k \mid x) = \frac{e^{\beta_k^T x}}{\sum_{j=1}^K e^{\beta_j^T x}}$ estimated via maximum likelihood.

Pros	Cons
– Interpretable – Calibrated probabilities – Solid theoretical foundation	– Linear decision boundaries – Sensitive to multicollinearity – Does not scale well with large p

- **When to use it**
  - Multiple classes
  - Explanatory objective
  - Features already well selected
- **Note**. Natural baseline for multiclass classification.

2.2 Ridge Logistic

- **Basic concept**
  Logistic regression with L2 penalization to stabilize estimates.
- **Mathematical idea**: $-\ell(\beta) + \lambda \sum \beta_j^2$

Pros	Cons
– Handles multicollinearity – More stable than plain logistic regression – Good generalization	– No variable selection – Reduced interpretability

- **When to use it**
  - Many correlated predictors
  - All potentially informative
- **Note**. Ridge shrinks, but does not eliminate.

2.3 Lasso Logistic

- **Basic concept**
  Logistic regression with L1 penalization: sparse classifier.
- **Mathematical idea**: $-\ell(\beta) + \lambda \sum |\beta_j|$

Pros	Cons
– Automatic feature selection – Parsimonious models – Useful when $p \gg n$	– Unstable with correlated variables – Bias on large coefficients

- **When to use it**
  - High dimensionality
  - Objective: identify which features matter
- **Note**. Lasso selects, sometimes aggressively.

2.4 Elastic Net (Logistic)

- **Basic concept**
  Compromise between Ridge and Lasso in a logistic setting.
- **Mathematical idea**: $-\ell(\beta) + \lambda[(1-\alpha)\sum \beta_j^2 + \alpha \sum |\beta_j|]$

Pros	Cons
– Selection of groups of correlated variables – Very stable – Excellent predictive performance	– More hyperparameters – Interpretation not immediate

- **When to use it**
  - Many correlated predictors
  - Objective: performance + parsimony
- **Note**. Modern default for penalized logistic regression.

2.5 Regression Tree

- **Basic concept**
  Recursive partitioning of the feature space to maximize node purity.
- **Mathematical idea**: Minimizes impurity (Gini / Entropy).

Pros	Cons
– Extremely interpretable – Captures interactions – No assumptions	– High variance – Overfitting – Low single-tree accuracy

- **When to use it**
  - Explanation
  - Rule-based decision making
- **Note**. A single tree is clear but fragile.

2.6 Random Forest

- **Basic concept**
  Ensemble of independent trees with bootstrap sampling and random feature selection.
- **Mathematical idea**: Majority vote $\hat{y} = mode\{f_b(x)\}$

Pros	Cons
– Very robust – Captures non-linearities and interactions – Minimal tuning	– Black box – Heavy models

- **When to use it**
  - Generic problems
  - "Safe" model
- **Note**. Workhorse of tabular classification.

2.7 Ranger RF

- **Basic concept**
  Optimized Random Forest (C++, memory-efficient).
- **Mathematical idea**: Same as RF, with efficient implementation.

Pros	Cons
– Extremely fast – Scalable – Support for weights / probabilities	– Same interpretation as RF

- **When to use it**
  - Large datasets
  - Production
- **Note**. "Industrial" Random Forest.

2.8 XGBoost

- **Basic concept**
  Tree-based Gradient Boosting with explicit regularization.
- **Mathematical idea**: Minimizes $\sum l(y_i, \hat{y}_i) + \sum \Omega(f_k)$

Pros	Cons
– Very high accuracy – Missing value handling – Regularization	– Many hyperparameters – Sensitive to tuning

- **When to use it**
  - Performance-first
  - Complex problems
- **Note**. Dominant in ML competitions.

2.9 Naive Bayes

- **Basic concept**
  Probabilistic classification with conditional independence.
- **Mathematical idea**: $P(y \mid x) \propto P(y) \prod P(x_j \mid y)$

Pros	Cons
– Extremely fast – Works with little data – Excellent baseline	– Unrealistic assumption – Simple decision boundaries

- **When to use it**
  - Text / NLP
  - Small datasets
- **Note**. When it works, it works well.

2.10 SVM (Linear)

- **Basic concept**
  Finds the hyperplane that maximizes the margin.
- **Mathematical idea**: $\min \frac{1}{2}||w||^2 + C\sum \xi_i$

Pros	Cons
– Excellent in high dimensions – Maximum margin – Good generalization	– No natural probabilities – Limited interpretability

- **When to use it**
  - $p \gg n$
  - Almost linearly separable data
- **Note**. Classic choice for genomics, text, raw images.

2.11 SVM (RBF)

- **Basic concept**
  Non-linear SVM with Gaussian kernel.
- **Mathematical idea**: $K(x, x') = \exp(-\gamma||x - x'||^2)$

Pros	Cons
– Highly flexible decision boundaries – Excellent accuracy	– Poor scalability – Delicate tuning – Black box

- **When to use it**
  - Small to medium datasets
  - Complex non-linearities
- **Note**. Powerful but expensive.

2.12 Comparison

Model	**Linearity**	**Feature selection**	**Interpretability**	**Robustness**	**Performance**	**Scalability**
Multin. Logistic	⋆⋆⋆	×	⋆⋆⋆	⋆	⋆	⋆⋆⋆
Ridge Logistic	⋆⋆⋆	×	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆
Lasso Logistic	⋆⋆⋆	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆
Elastic Net	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
Classification Tree	×	⋆⋆	⋆⋆⋆	⋆	⋆	⋆⋆
Random Forest	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆
Ranger RF	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆
XGBoost	×	⋆	⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆
Naive Bayes	⋆⋆	×	⋆⋆	⋆	⋆	⋆⋆⋆⋆
SVM (Linear)	⋆⋆⋆	×	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆
SVM (RBF)	×	×	⋆	⋆⋆	⋆⋆⋆	⋆

2.13 Context-based choice

Main requirement	**Recommended model**
Interpretable baseline	Multinomial Logistic
Multicollinearity	Ridge / Elastic Net
Feature selection	Lasso / Elastic Net
Decision rules	Classification Tree
Robust model	Random Forest
Fast RF / big data	Ranger RF
Top accuracy	XGBoost
Text / NLP	Naive Bayes
High dimensionality	Linear SVM
Strong non-linearities	SVM RBF

2.14 Bias-Variance trade-off

Model	**Bias**	**Variance**
Logistic	High	Low
Ridge / Lasso	Medium	Medium
Tree	Low	High
Random Forest	Low	Low
Boosting	Very low	Medium
SVM RBF	Very low	Medium-high