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

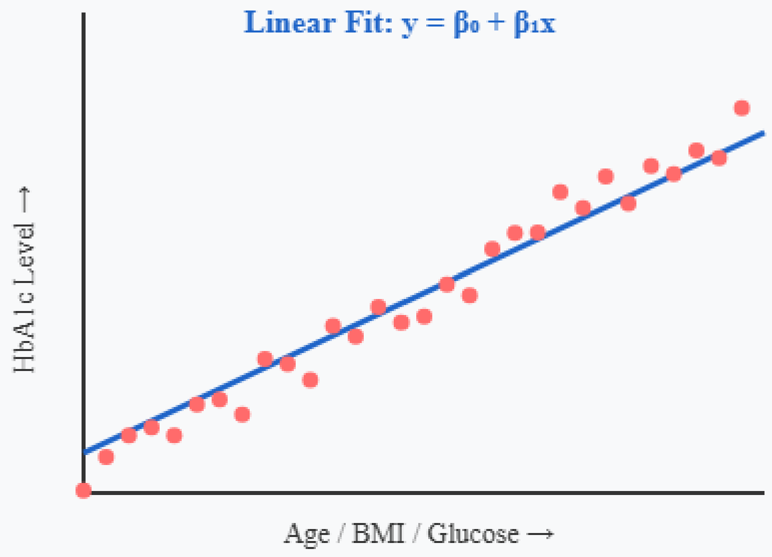
Use case: Drug dosage optimization

⚠ Critical: Prediction Intervals

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

Detailed Methods & Applications

1. Linear Regression



What is Linear Regression?

Linear regression models the relationship between input variables (features) and a continuous output by fitting a straight line (or hyperplane in multiple dimensions) through the data points.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$

Key Characteristics

- **Interpretability:** Coefficients (β) show the direct effect of each feature
- **Assumptions:** Linearity, independence, homoscedasticity, normality
- **Training:** Minimizes Mean Squared Error (MSE)

✓ Advantages

✗ Limitations

- Highly interpretable coefficients
- Fast training and prediction
- Works well with limited data
- Provides confidence intervals easily

- Assumes linear relationships
- Sensitive to outliers
- Cannot capture complex interactions
- Struggles with high-dimensional data

📊 Clinical Example: HbA1c Prediction

Scenario: Predicting HbA1c levels based on patient age, BMI, fasting glucose, and exercise frequency.

Model: HbA1c = 4.2 + 0.03(Age) + 0.08(BMI) + 0.02(Fasting_Glucose) - 0.15(Exercise_Hours)

Interpretation: Each 1 kg/m^2 increase in BMI is associated with a 0.08% increase in HbA1c, while each additional hour of weekly exercise decreases HbA1c by 0.15%.

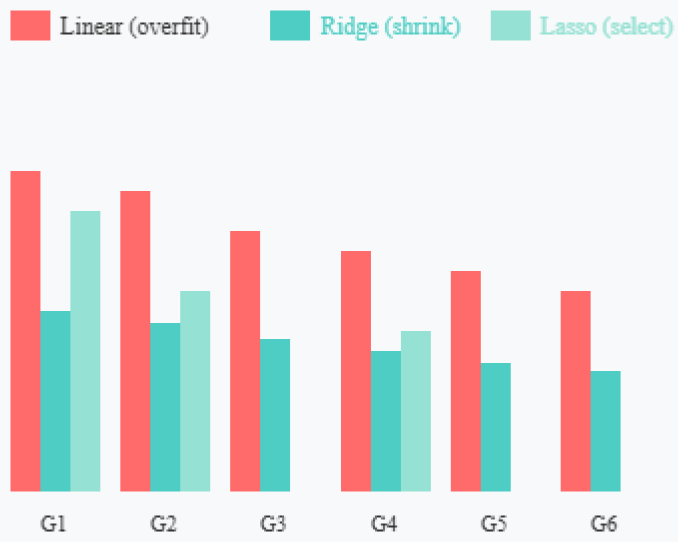
2. Ridge / Lasso / Elastic Net Regression

What is Regularized Regression?

Regularization adds a penalty term to the loss function to prevent overfitting by constraining coefficient magnitudes. This is crucial when dealing with many features or correlated predictors.

$$\text{Ridge (L2): Loss} + \lambda\sum\beta_i^2$$
$$\text{Lasso (L1): Loss} + \lambda\sum|\beta_i|$$
$$\text{Elastic Net: Loss} + \lambda_1\sum|\beta_i| + \lambda_2\sum\beta_i^2$$

Key Differences



- **Ridge:** Shrinks coefficients but keeps all features
- **Lasso:** Can reduce coefficients to exactly zero (feature selection)
- **Elastic Net:** Combines both L1 and L2 penalties

✓ Advantages

- Handles multicollinearity effectively
- Prevents overfitting with many features
- Lasso provides automatic feature selection
- Elastic Net balances both approaches

✗ Limitations

- Requires tuning regularization parameter (λ)
- Still assumes linear relationships
- Feature scaling is critical
- Interpretation becomes more complex

📊 Clinical Example: Gene Expression Biomarker Prediction

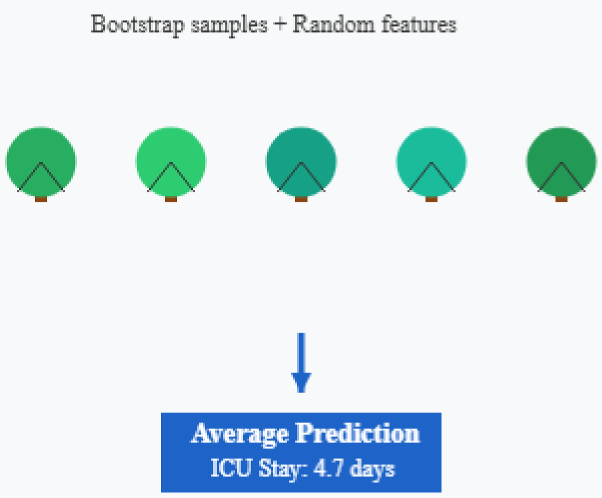
Scenario: Predicting tumor size from 5,000 gene expression levels with only 200 patient samples.

Challenge: Many more features than samples (p >> n problem) causes overfitting in standard linear regression.

Solution: Lasso regression identifies 47 genes with non-zero coefficients, providing both prediction and biological insight into which genes drive tumor growth.

Result: Test R^2 improved from 0.23 (linear) to 0.68 (Lasso) by preventing overfitting.

3. Random Forest Regression



What is Random Forest Regression?

Random Forest builds multiple decision trees on random subsets of data and features, then averages their predictions. This ensemble approach captures non-linear relationships and complex interactions.

How It Works

- **Bootstrap Sampling:** Each tree trains on a random sample with replacement
- **Random Features:** At each split, only a random subset of features is considered
- **Aggregation:** Final prediction is the average of all tree predictions
- **Feature Importance:** Calculated by measuring prediction accuracy decrease when a feature is permuted

✓ Advantages

- Captures non-linear relationships automatically
- Handles missing data well
- Provides feature importance rankings
- Robust to outliers
- No feature scaling required

✗ Limitations

- Less interpretable than linear models
- Can overfit with too many/deep trees
- Larger memory footprint
- Slower prediction than linear models

📊 Clinical Example: ICU Length of Stay Prediction

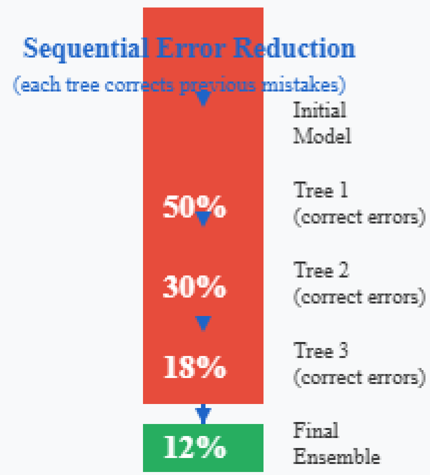
Scenario: Predicting ICU stay duration using admission vitals, lab results, comorbidities, and treatment interventions.

Complexity: Relationships are highly non-linear (e.g., U-shaped relationship between blood pressure and LOS; interactions between age and organ failure).

Model Performance: Random Forest achieved RMSE of 2.3 days vs 3.8 days for linear regression.

Feature Insights: Top predictors were APACHE score (importance: 0.24), mechanical ventilation (0.18), and sepsis presence (0.15), guiding resource allocation.

4. Gradient Boosting (XGBoost)



What is Gradient Boosting?

Gradient boosting builds trees sequentially, where each new tree corrects the errors of the previous ensemble. XGBoost is an optimized implementation with regularization and efficient algorithms.

Key Concepts

- **Sequential Learning:** Each tree focuses on the residual errors of previous trees
- **Gradient Descent:** Uses gradients to minimize loss function
- **Regularization:** L1/L2 penalties on leaf weights prevent overfitting

- **Learning Rate:** Controls contribution of each tree (shrinkage)

✓ Advantages

- State-of-the-art accuracy on tabular data
- Handles mixed data types seamlessly
- Built-in feature importance
- Efficient with missing values
- Highly customizable with many hyperparameters

✗ Limitations

- Prone to overfitting without proper tuning
- Longer training time than Random Forest
- Many hyperparameters to optimize
- Less interpretable than linear models

📊 Clinical Example: Personalized Drug Dosage Optimization

Scenario: Predicting optimal warfarin dosage based on genetic variants (CYP2C9, VKORC1), demographics, medications, and clinical factors.

Challenge: Complex gene-drug-disease interactions with non-linear dose-response curves.

XGBoost Results: Mean absolute error of 0.73 mg/day vs 1.42 mg/day for clinical algorithms, reducing bleeding/clotting events by 28%.

Model Insights: SHAP values revealed that VKORC1 genotype had the largest individual impact, but age-BMI interactions were critical for elderly patients.

🎯 Model Selection Guidelines

Start Simple: Begin with Linear Regression for interpretability.

Add Regularization: Use Ridge/Lasso/Elastic Net when you have many features or multicollinearity.

Go Non-linear: Use Random Forest when relationships are complex but interpretability is still important.

Maximize Performance: Use XGBoost for best predictive accuracy on structured data.

Always: Validate on held-out test data, calculate confidence intervals, and consider clinical interpretability alongside statistical performance.