Prof. Fabio Postiglione

Statistical Learning Techniques for Regression and Classification in Clinical Datasets

Apicella Mario Bruno Salvatore

UNIVERSITÀ
DEGLI STUDI
DI SALERNO

Contents

1 Introduction

In the era of data-driven healthcare, the growing availability of structured clinical data from electronic health records, diagnostic tools, and population-level screening programs provides an unprecedented opportunity to develop predictive models capable of supporting medical decision-making. However, the complexity and high dimensionality of such datasets—often accompanied by noise, multicollinearity, and class imbalance require rigorous statistical methodologies to ensure model accuracy, interpretability, and generalizability.

This project focuses on supervised learning methods applied to medical datasets, aiming to explore and compare different modeling strategies under both classification and regression paradigms. Particular emphasis is placed on the role of **feature selection** and *regularization techniques*, such as *Lasso*, *Ridge*, and *ElasticNet*, in mitigating overfitting and enhancing predictive performance. These techniques not only enable dimensionality reduction but also improve model robustness and facilitate clinical interpretation by highlighting the most relevant predictors.

A comprehensive evaluation is conducted using cross-validation and performance metrics such as *ROC AUC*, *F1-score*, and *confusion matrices* for classification tasks. Special attention is paid to *model tuning* and *hyperparameter selection*, with detailed visualizations (e.g., AUC vs. $\log(\lambda)$) that guide the optimal choice of regularization strength. By implementing and comparing these techniques, this study aims to draw practical insights into the design of lightweight yet effective predictive models tailored to real-world healthcare data.

2

Datasets Description

2.1 Overview

This study leverages two biomedical datasets from open-access repositories, selected to represent two distinct supervised learning tasks: binary classification and continuous regression. Both datasets are characterized by their clinical relevance, quality of annotation, and data accessibility in structured CSV format. Below we provide a detailed overview of each.

2.2 Regression Task

The regression dataset originates from the Oxford Parkinson's Disease Telemonitoring study , publicly hosted by the UCI Machine Learning Repository [3]. It includes a total of 5,875 voice recordings collected from 42 patients over a 6-month longitudinal trial. Each instance represents one recording session and is described by 19 features, including 16 biomedical acoustic features (such as jitter, shimmer, and various dysphonia measures), alongside metadata such as subject ID, age, sex, and recording timestamp.

The primary goal is to predict one or more clinical scores that reflect Parkinson's disease progression. In particular, the `motor_UPDRS` and `total_UPDRS` scores are used as continuous regression targets. These scores range approximately from 0 to 180, with higher values indicating more severe symptoms. Since the data includes repeated measures per patient, analysts should be aware of potential dependencies among instances.

2.3 Classification Task

The classification dataset is a gene expression microarray study related to prostate cancer [2], originally introduced by Singh et al. (2002) and later referenced in various bioinformatics and statistical learning studies. It consists of gene expression measurements from 102 prostate tissue samples, including 52 from prostate cancer patients and 50 from healthy individuals (controls). Each sample is characterized by the expression levels of 6,033 genes, resulting in a highly dimensional dataset with a feature-to-sample ratio of nearly 60:1.

This makes it a challenging benchmark for binary classification, where the goal is to distinguish between cancerous and normal tissues. The original data was published in Cancer Cell, and has been widely reused for evaluating statistical techniques including empirical Bayes methods (e.g., Efron, 2008). The dataset is publicly available through the NCBI Gene Expression Omnibus (GEO) and is also integrated in R packages such as `sda`, where it is referred to as `singh2002`.

3

Exploratory Data Analysis for Linear Regression

3.1 Overview: EDA for linear Regression

The Exploratory Data Analysis (EDA) phase plays a crucial role in ensuring the quality, structure, and interpretability of biomedical datasets prior to model development. In this study, we conduct a separate EDA for each of the two datasets used, one for classification and one for regression, considering their distinct nature and domain-specific challenges. This section presents a structured overview of the preprocessing steps, statistical analyses, and preliminary insights derived from the prostate cancer microarray dataset and the Parkinson's telemonitoring dataset.

3.2 Dataset Description

The regression dataset [3] consists of 5875 voice recordings from 42 patients affected by Parkinson's disease, captured over six months of home telemonitoring. Each observation includes 19 attributes: 16 acoustic features and 3 metadata fields (subject ID, gender, and test time). The objective is to predict two continuous clinical scores — `motor_UPDRS` and `total_UPDRS` which quantify disease severity.

The *Parkinson's Telemonitoring* dataset includes both clinical and acoustic features derived from voice recordings, used to monitor disease progression. Below is a structured summary of the variables:

- **total_UPDRS**: Total Unified Parkinson's Disease Rating Scale. Measures overall severity of Parkinson's symptoms, combining motor and non-motor aspects.
- **motor_UPDRS**: Motor component of the UPDRS, focused specifically on motor-related impairments such as tremor, rigidity, and bradykinesia.
- **subject#**: Unique identifier for each patient. Not used for modeling.
- **age**: Patient's age. Age is known to influence the onset and severity of Parkinson's symptoms.
- **sex**: Patient's biological sex (0 = male, 1 = female), potentially affecting vocal biomarkers.
- **test_time**: Time in days since the first test, used to capture longitudinal disease progression.
- **Jitter(%)**: Percentage variation in fundamental frequency. High values indicate unstable voice or vocal tremor.
- **Jitter(Abs)**: Absolute jitter in seconds, independent of base frequency.
- **Jitter:RAP**: Relative Average Perturbation. Average absolute difference between consecutive periods, capturing local instability.
- **Jitter:PPQ5**: Perturbation Quotient over 5 periods. Measures short-term frequency fluctuation.
- **Jitter:DDP**: Derivative of RAP, measuring more sensitive jitter changes.

- **Shimmer**: Variation in amplitude of the speech signal. Indicates amplitude instability.
- **Shimmer(dB)**: Logarithmic (decibel) version of shimmer for easier interpretation.
- **Shimmer:APQ3**, **APQ5**, **APQ11**: Amplitude perturbation quotients over 3, 5, and 11 periods respectively. Higher values suggest intensity variation.
- **Shimmer:DDA**: Derivative of APQ, summing three consecutive absolute amplitude differences.
- **NHR** (Noise-to-Harmonics Ratio): Ratio of noise components to periodic (harmonic) ones. Higher values indicate degraded vocal quality.
- **HNR** (Harmonics-to-Noise Ratio): Inverse of NHR. Higher HNR values reflect cleaner, more stable voice signals.
- **RPDE** (Recurrence Period Density Entropy): Quantifies the complexity of vocal periodicity. High RPDE values indicate irregular or chaotic voice.
- **DFA** (Detrended Fluctuation Analysis): Captures signal complexity and long-term self-similarity in vocal patterns.
- **PPE** (Pitch Period Entropy): Entropy of pitch period, measuring irregularities in pitch modulation, associated with motor impairments.

These features are crucial in capturing vocal biomarkers that correlate with disease severity and progression. Variables such as `Jitter`, `Shimmer`, `NHR`, and `PPE` are particularly informative for motor symptom assessment.

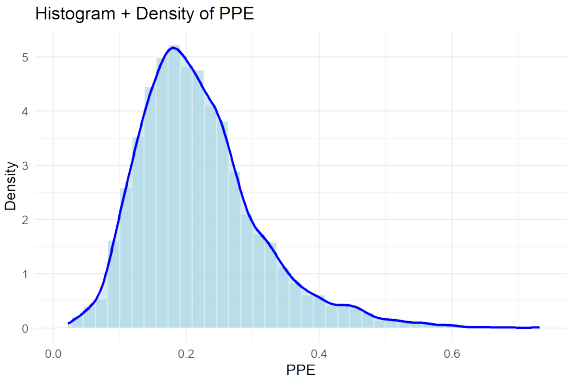
3.3 Data imputation

The dataset contains no missing values or inconsistencies. The categorical variable `sex` was converted into a factor with two levels. The subject ID was retained for stratified analyses but excluded from modeling as it does not represent a predictive variable.

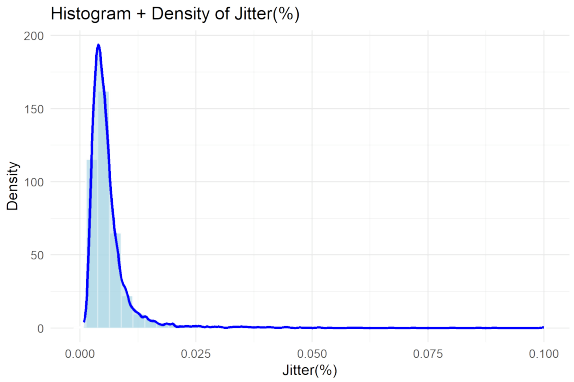
3.4 Feature Distributions

To evaluate the distributional properties of each predictor, a series of histograms and kernel density plots were generated. The analysis revealed that many of the acoustic features exhibit strong right-skewness and long tails, which is characteristic of vocal biomarkers affected by neurological conditions.

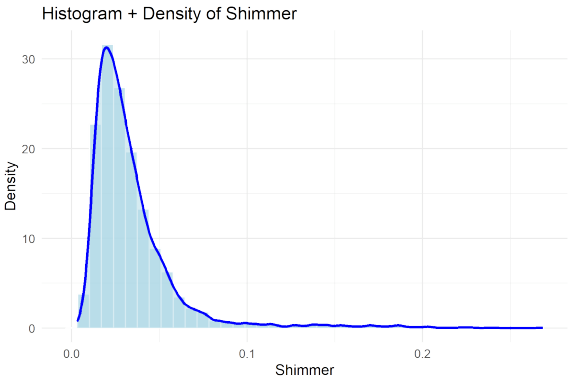
For example, `PPE` (Pitch Period Entropy) (Figure 1) shows a mean of 0.22 but reaches a maximum value of 0.73, indicating the presence of extreme observations. Similarly, features such as `Jitter(%)`, `Shimmer`, and `HNR` demonstrate a wide dynamic range and minimal baseline values, contributing to their non-Gaussian nature.



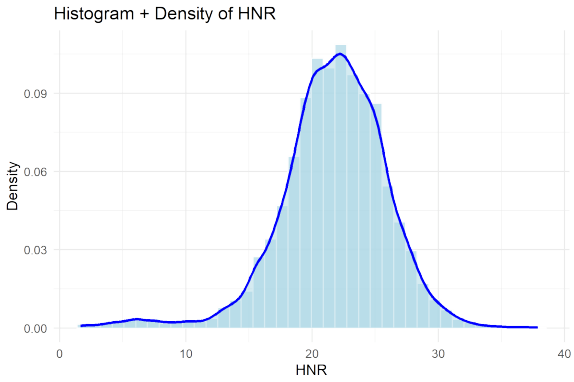
(a) Distribution of `PPE` (Pitch Period Entropy)



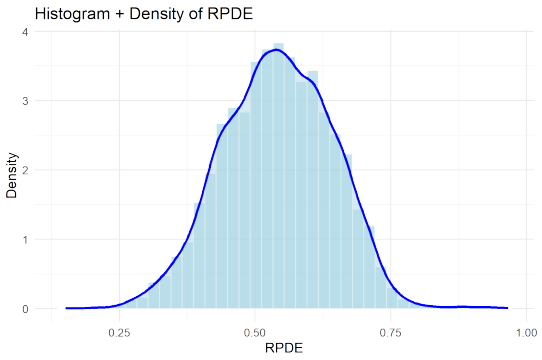
(b) Distribution of `Jitter(%)`



(c) Distribution of `Shimmer`



(d) Distribution of `HNR` (Harmonics-to-Noise Ratio)



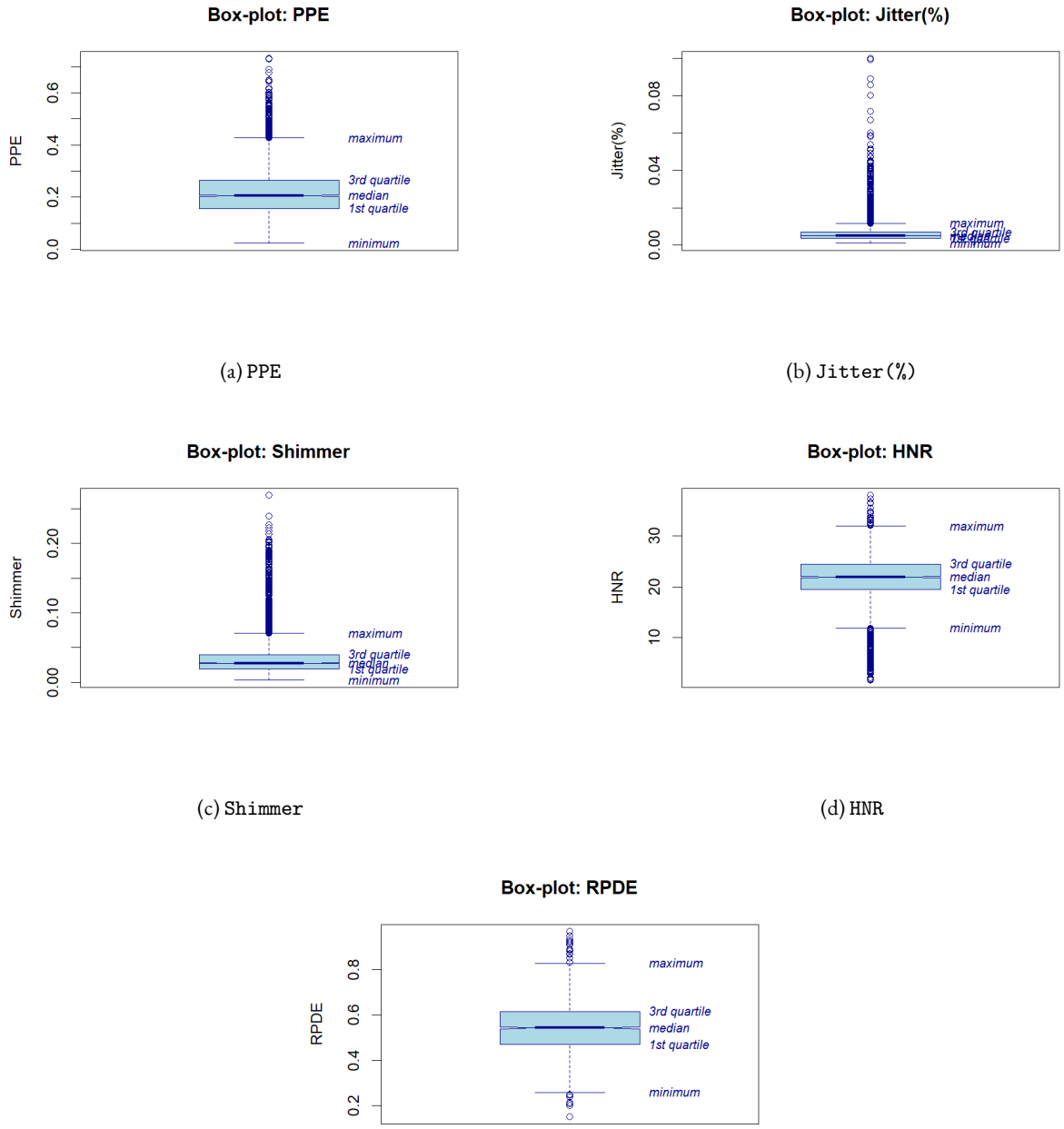
(e) `RPDE`

Figure 1: Histograms and density plots of selected acoustic features exhibiting non-Gaussian and right-skewed distributions.

Visual inspection of the density curves confirmed the deviation from normality across most predictors, with several distributions showing asymmetric shapes and potential outliers. These characteristics are aligned with the clinical variability expected in speech signals from individuals with Parkinson's disease.

To complement the distribution analysis, Figure 2 shows the boxplots of the same acoustic features. These plots confirm the presence of extreme observations (outliers), particularly in variables such as `PPE`, `NHR`, and `Shimmer`, further justifying the need for standardization and robust modeling tech-

niques.



(a) PPE

(b) Jitter(%)

(c) Shimmer

(d) HNR

(e) RPDE

Figure 2: Boxplots of selected acoustic features illustrating outlier presence and distribution variability.

As a result, all numeric features were standardized using z-score normalization before regression modeling. This ensured that variables were on a comparable scale, reduced the impact of extreme values, and improved the numerical stability and convergence of machine learning algorithms.

3.5 Distribution of the Target Variables

The Parkinson Telemonitoring dataset includes two primary clinical scores as target variables: `motor_UPDRS` and `total_UPDRS`. These represent, respectively, the motor subscore and the overall disease severity score of the Unified Parkinson's Disease Rating Scale.

Histogram and Density Plots. Figures 3 and 4 present the distribution of the two target variables. The visual inspection of the histograms and density curves indicates a mild multimodal behavior in both cases, suggesting the presence of subpopulations or repeated measurements with heterogeneous clinical states.

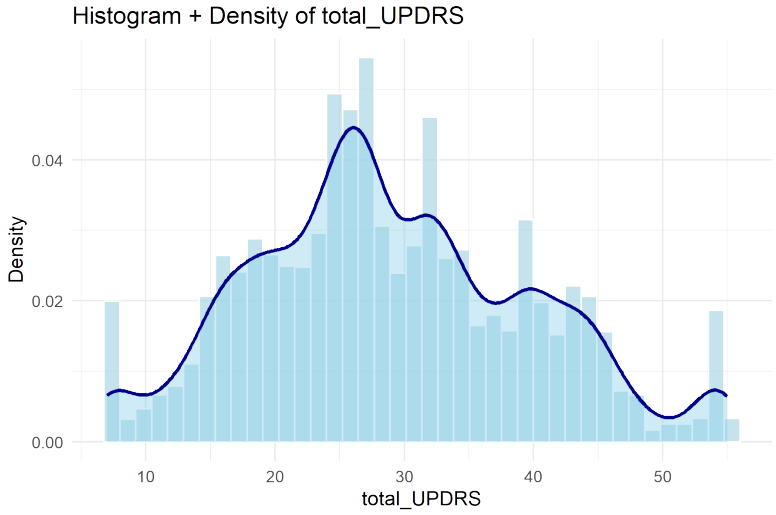


Figure 3: Histogram and density plot of `total_UPDRS`.

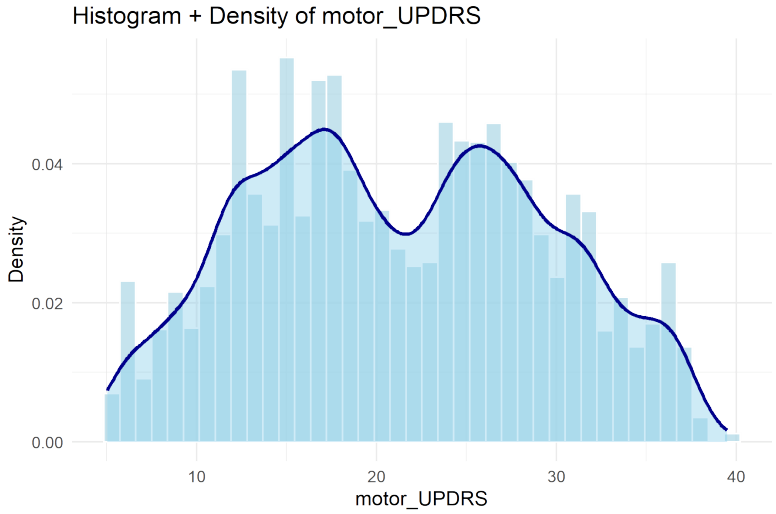


Figure 4: Histogram and density plot of `motor_UPDRS`.

Normality Assessment. To formally evaluate the adherence to a Gaussian distribution, Q-Q plots were generated (Figures 5, 6), and the Shapiro-Wilk test was conducted on a random sample of 5000 ob-

servations. In both cases, the test returned a p-value $< 2.2e^{-16}$, leading to the rejection of the null hypothesis of normality. The Q-Q plots also confirm the presence of deviations, particularly in the tails, suggesting heavy-tailed and skewed distributions.

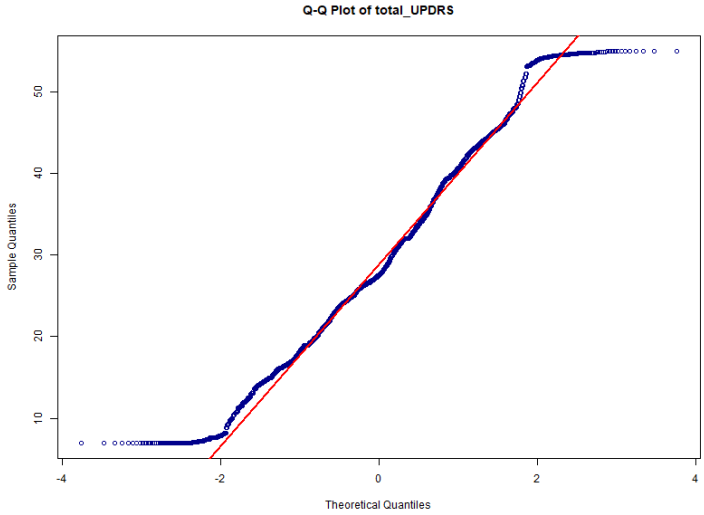


Figure 5: Q-Q plot of `total_UPDRS`.

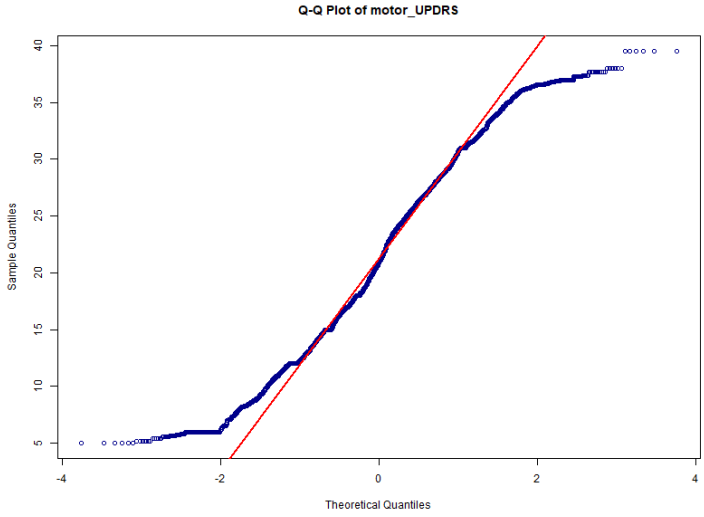


Figure 6: Q-Q plot of `motor_UPDRS`.

Boxplots by Sex. To further explore potential confounding factors, boxplots were generated to compare the distributions of both target scores by gender (Figures 7 and 8). The median values and interquartile ranges appear comparable between males and females, with slightly higher variability in the male subgroup. No substantial difference was observed that would necessitate stratified modeling.

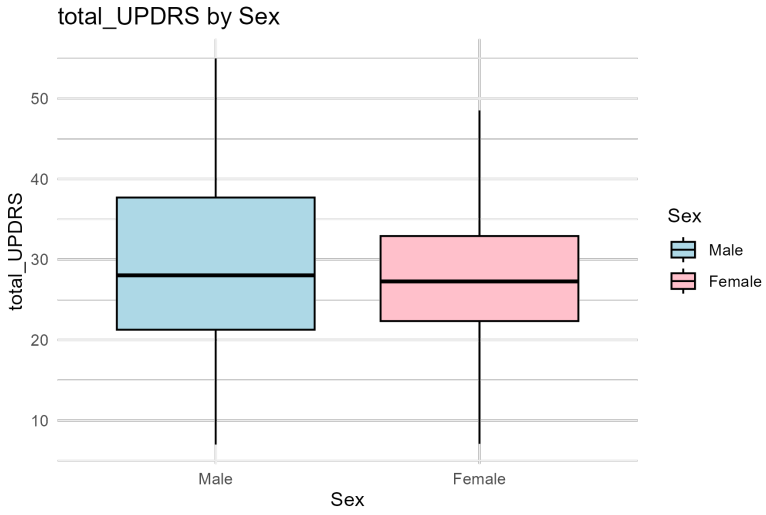


Figure 7: Distribution of `total_UPDRS` by sex.

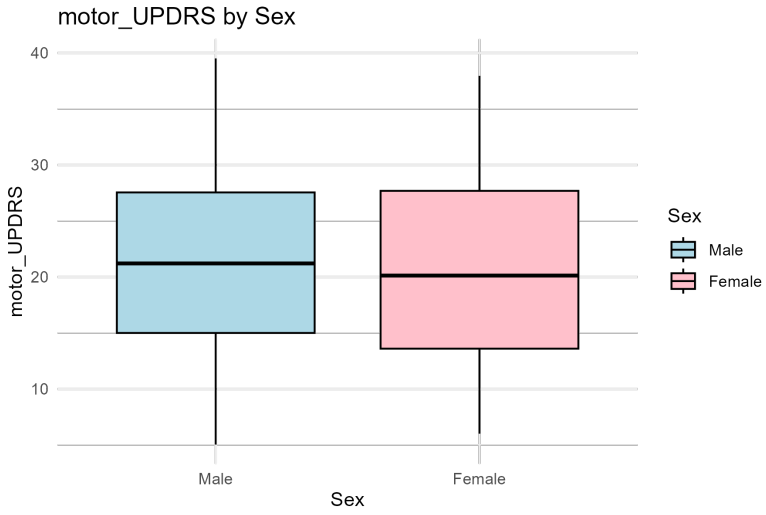


Figure 8: Distribution of `motor_UPDRS` by sex.

3.6 Correlation Analysis with Target Variables

To evaluate linear associations between predictors and clinical scores, a Pearson correlation matrix was computed over all numeric variables, including both target variables: `total_UPDRS` and `motor_UPDRS`. Figure 9 displays the full correlation matrix using ellipses, where darker shades and more elongated shapes represent stronger relationships.

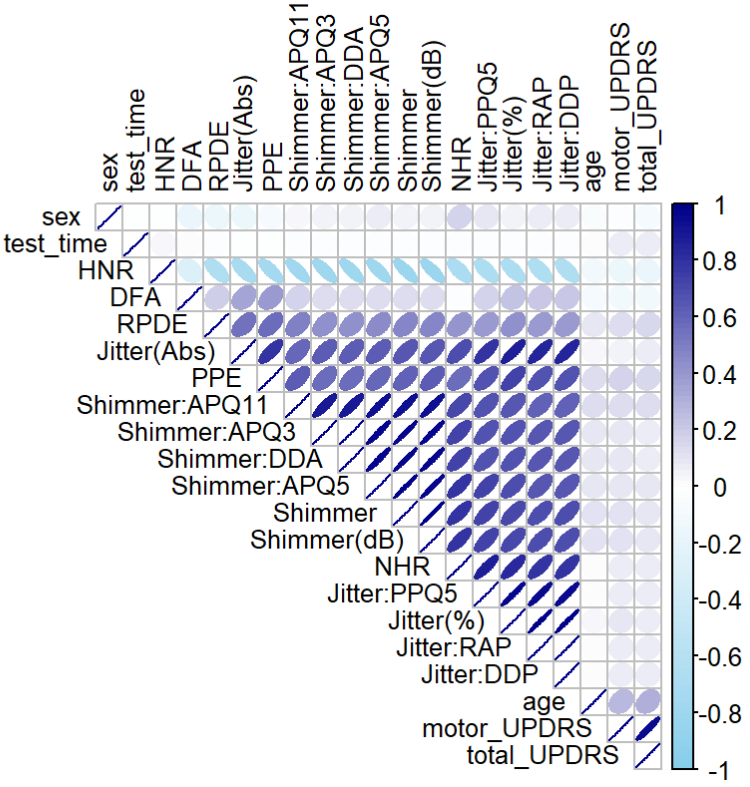


Figure 9: Pearson correlation matrix of all numeric variables (upper triangle, ordered by hierarchical clustering).

Observations on Correlation Structure. The correlation matrix reveals two key aspects:

- **Clusters of highly correlated predictors** are observed among shimmer-related features (`Shimmer`, `Shimmer:APQ5`, `Shimmer:DDA`, etc.) and jitter-based measures (`Jitter:RAP`, `Jitter:PPQ5`, `Jitter:Abs`).
- Both `motor_UPDRS` and `total_UPDRS` are moderately correlated with features like `PPE`, `RPDE`, and `age`, while exhibiting negative correlations with `HNR` and `DFA`.

Correlation with `total_UPDRS`. The most positively correlated variables with `total_UPDRS` include:

- `age` ($r = 0.31$)
- `RPDE` ($r = 0.16$)
- `PPE` ($r = 0.16$)
- `Shimmer:APQ11`, `Shimmer(dB)`, and other shimmer features (all $r > 0.07$)

On the other hand, the most negatively correlated features are:

- `HNR` ($r = -0.16$)
- `DFA` ($r = -0.11$)
- `sex` and `test_time` (slightly negative, $r \approx -0.07$)

Correlation with `motor_UPDRS`. The motor subscore exhibits a very similar pattern:

- Positive correlation with `age` ($r = 0.27$), `PPE` ($r = 0.16$), and shimmer-related features.
- Negative correlation with `HNR` ($r = -0.16$), `DFA` ($r = -0.12$), and jitter-based metrics.

Interpretation. Overall, higher disease severity (as measured by UPDRS) is associated with increased vocal irregularities (entropy, shimmer, jitter) and reduced harmonicity (`HNR`). These trends align with known effects of Parkinson's disease on voice, reinforcing the clinical relevance of these acoustic features.

3.7 Outlier Detection

To ensure the robustness of subsequent modeling, we conducted a comprehensive outlier analysis combining three complementary methods: univariate (IQR-based), multivariate (Mahalanobis distance), and regression-based (studentized residuals).

Univariate Outliers. Each numeric variable was examined using the interquartile range (IQR) method. As shown in Figure 10, several acoustic features exhibited a substantial proportion of extreme values. Notably, variables such as `Jitter(%)`, `Jitter:PPQ5`, `NHR`, and `PPE` presented more than 5% outliers, reflecting their skewed distributions and sensitivity to vocal irregularities. In contrast, features like `DFA` and `test_time` exhibited no significant univariate outliers, indicating relatively stable behavior.

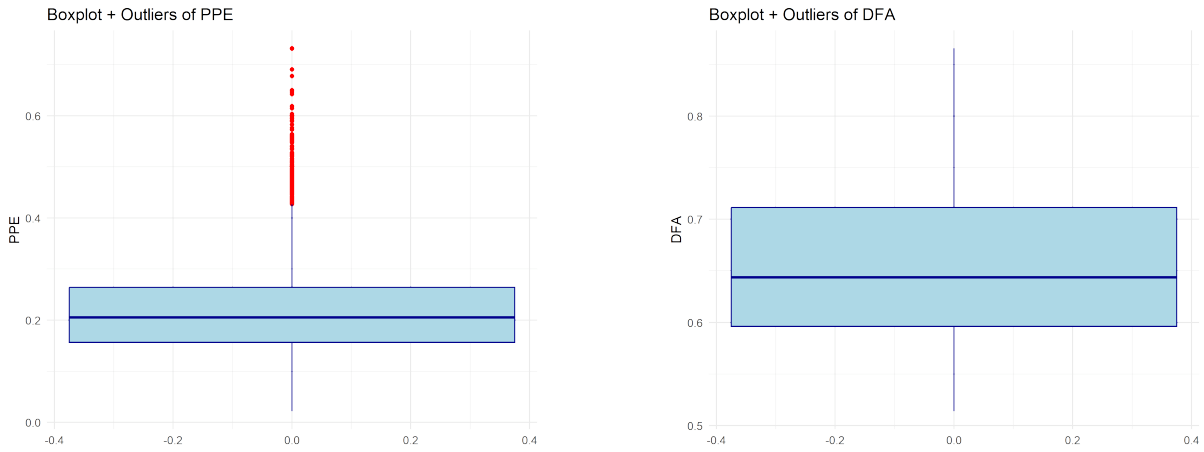


Figure 10: Example of univariate outlier detection using boxplots for `PPE` (left) and `DFA` (right). Red points indicate detected outliers.

Multivariate Outliers. We computed the Mahalanobis distance across all numeric predictors to detect multivariate outliers considering their joint distribution. Using a chi-square threshold at the 97.5% confidence level, we identified 375 outliers (6.38% of the data). This approach is particularly suited for high-dimensional datasets, where variables may individually appear normal but jointly exhibit abnormal patterns.

Figure 11 illustrates the distribution of Mahalanobis distances. The left panel presents the full range of distances, highlighting a long right-tail distribution due to extreme outliers. The red dashed line marks the chi-square cutoff used to define multivariate outliers. The zoomed version on the right focuses on the main density region, making it easier to observe how most samples lie well below the threshold, while only a minority exceed it substantially.

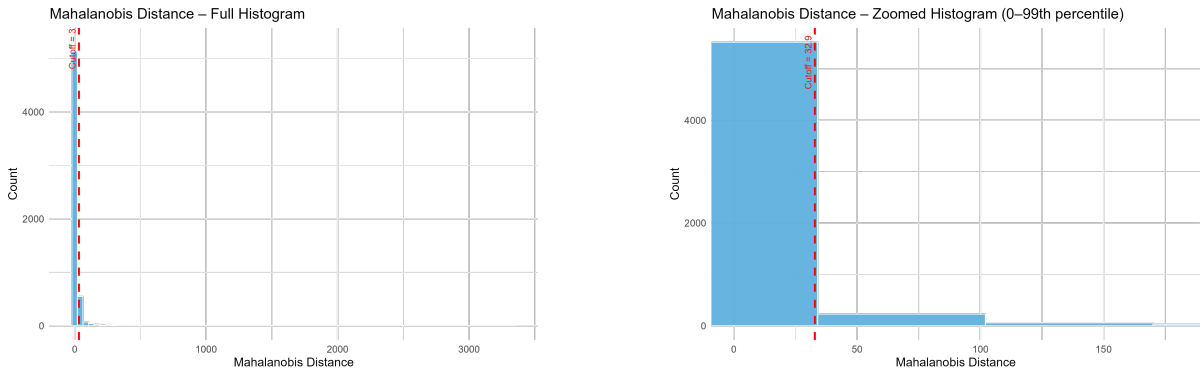


Figure 11: Multivariate outlier detection via Mahalanobis distance. The red dashed line indicates the cutoff (Chi-square 97.5%).

To further characterize these outliers, Figure 12 shows the boxplot of Mahalanobis distances. The red points above the whiskers correspond to extreme observations identified by the IQR rule applied to Mahalanobis values. This visualization reinforces the previous findings by highlighting the heavy-tailed nature of the distance distribution and visually confirming the presence of severe multivariate anomalies.

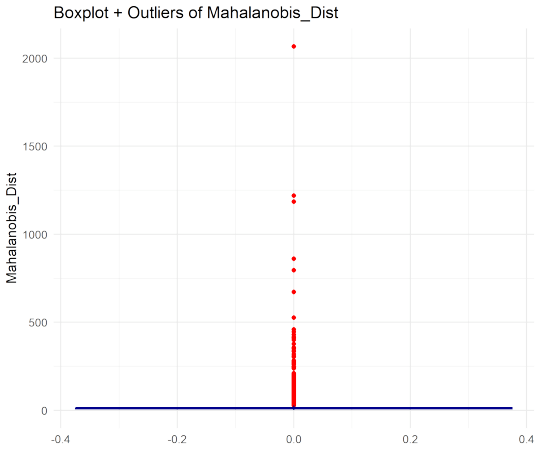


Figure 12: Boxplot of Mahalanobis distances with detected multivariate outliers (red).

Outliers via Studentized Residuals. To identify potentially influential observations in the regression models, we computed studentized residuals for both target variables: `motor_UPDRS` and `total_UPDRS`. Observations with absolute studentized residual values greater than 2 were flagged as potential outliers, based on standard statistical practice.

As illustrated in Figure 13, several samples exceeded the ±2 threshold in both targets. In the case of `motor_UPDRS`, most outliers appeared concentrated in the upper residual range, suggesting the model systematically underestimated those values—likely due to nonlinear patterns or feature interactions not captured by the linear model. Similarly, for `total_UPDRS`, a dense cluster of outliers is evident above the +2 threshold, again pointing to underfitting or data heterogeneity. A few extreme

negative residuals (e.g., sample #1881) were also detected, indicating substantial overestimation in isolated instances.

These findings emphasize the necessity of complementing linear modeling with more flexible or robust approaches to better accommodate the heterogeneity present in the data.

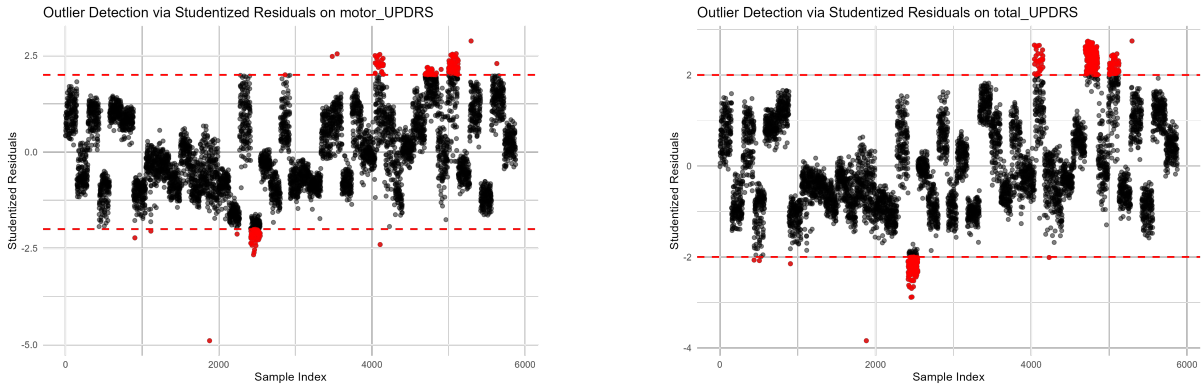


Figure 13: Outlier detection via studentized residuals for `motor_UPDRS` and `total_UPDRS`. Labeled red points indicate samples with residuals exceeding the ±2 threshold.

These regression-based residual analyses, in combination with univariate and multivariate outlier detection, contributed to a thorough identification of anomalous samples, reinforcing the reliability of the data preprocessing workflow prior to model training.

Exploratory Visualization: Pairwise Relationships, PCA and Heatmap

To gain a deeper understanding of the relationships among key acoustic features and the two UPDRS targets, we employed several complementary visualization techniques.

Pairwise Correlations and Distributions. We selected the most relevant features, `DFA`, `PPE`, `Shimmer`, and `NHR`—alongside the two target variables, and visualized their relationships using a pair plot (Figure 14). This plot displays correlation coefficients in the upper triangle, density plots on the diagonal, and smoothed scatterplots in the lower triangle. Notably, `PPE` and `Shimmer` exhibit visible positive associations with both `total_UPDRS` and `motor_UPDRS`, while `DFA` and `NHR` show more moderate or negative associations.

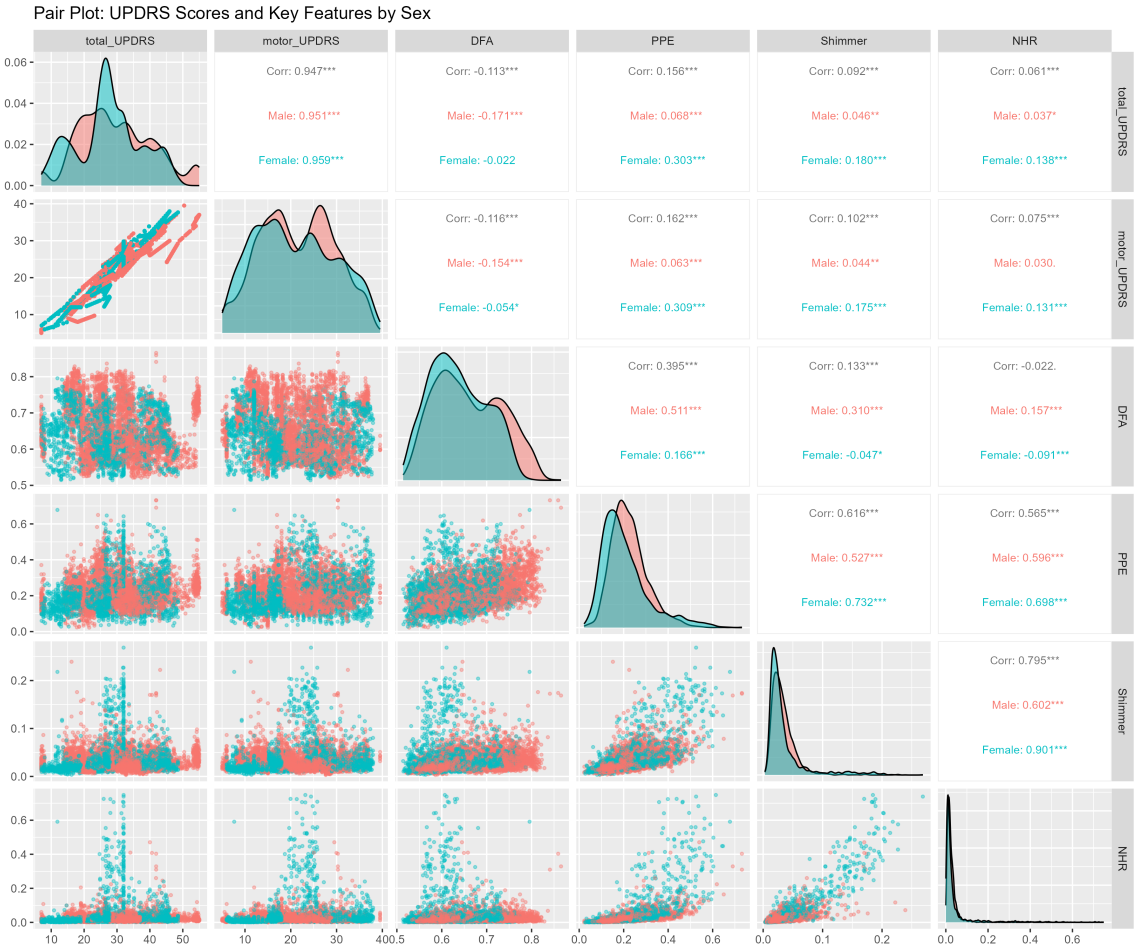


Figure 14: Pair plot of top features and UPDRS targets. Upper panels: Pearson correlations; diagonal: density; lower panels: scatterplots with trend lines.

Principal Component Analysis (PCA). To evaluate potential separability or trends in the data structure, we applied PCA to the standardized numeric predictors (excluding the targets). Figure 38 shows the projection onto the first two principal components, with color gradients corresponding to each UPDRS score. Although no clear clusters emerge, there is a smooth gradient along PC1 and PC2 axes, particularly for `total_UPDRS`, suggesting some latent linear relationships captured in the low-dimensional space.

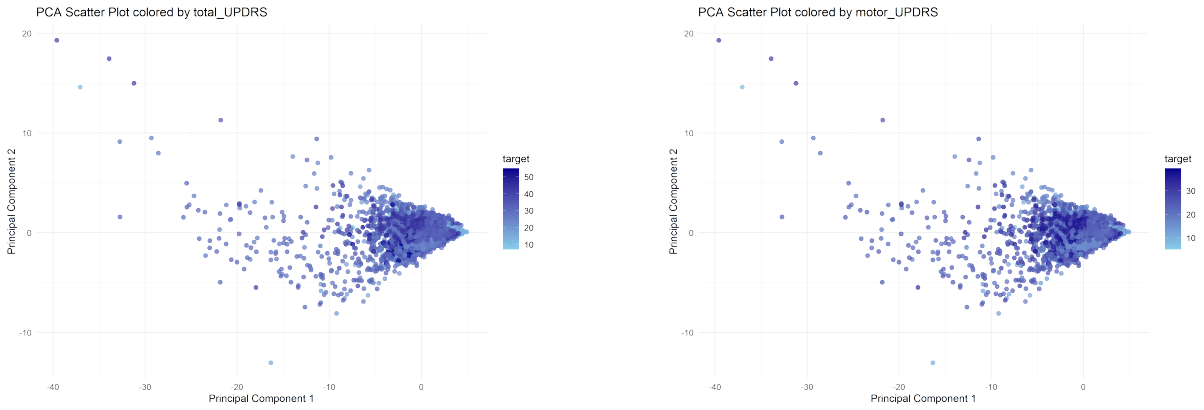


Figure 15: PCA scatter plots with color mapping to `total_UPDRS` (left) and `motor_UPDRS` (right).

Heatmap of Standardized Features. As a compact summary of the multivariate structure, a heatmap was generated from the first 15 observations and their standardized numeric features. The hierarchical clustering applied to both rows and columns (Figure 16) reveals patterns of co-variation among features and highlights the internal consistency across samples.

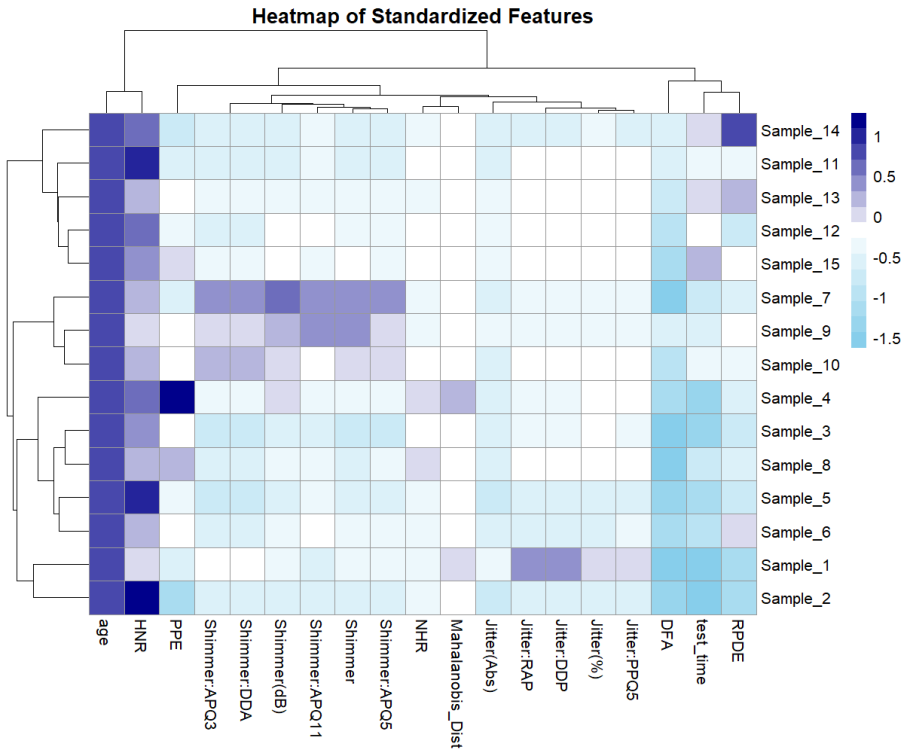


Figure 16: Heatmap of standardized numeric features. Hierarchical clustering applied to both rows and columns.

3.8 Target Selection and Initial Diagnostics

Before model construction, the two regression targets `motor_UPDRS` and `total_UPDRS` were verified for presence and numerical type within the cleaned dataset. Summary statistics and distribution visualizations (histogram and kernel density) were generated to assess the variability and skewness of the targets. This preliminary inspection confirmed the continuous nature of both variables and provided insights into their right-skewed distributions, which might influence the model residuals.

3.9 Predictor Preparation and Feature Cleaning

A dedicated preprocessing step was conducted to isolate only valid numerical predictors. The following variables were excluded from the modeling matrix:

- Subject identifier (`subject#`)
- Both target variables (`total_UPDRS` and `motor_UPDRS`)
- Mahalanobis distance and outlier flags (if present)

Additionally, predictors with zero variance across the dataset were removed, as they carry no discriminative power. The final predictor matrix was composed of continuous, non-collinear features, suitable for regression modeling.

3.10 Standardization and Scaling

To ensure consistent feature scaling and prevent any single variable from dominating the learning process, all numeric predictors were standardized using **Z-score normalization**, transforming each feature to have a mean of zero and a standard deviation of one. This preprocessing step was performed *after outlier removal*, and the `is_outlier` flag was explicitly excluded from the transformation to avoid bias.

This transformation allows the use of regularized regression techniques (e.g., LASSO, Ridge) without being affected by differences in feature magnitude, and it also facilitates interpretability when comparing model coefficients.

3.10.1 Boxplot of Standardized Features

Figure 17 presents a boxplot of the main numerical features in the dataset after multivariate outlier removal using the Mahalanobis distance. All features have been standardized using Z-score normalization (mean = 0, standard deviation = 1), allowing for direct comparison of their distributions and variability.

Each box represents the interquartile range (IQR), covering the middle 50% of the observations, with the horizontal line indicating the median. The whiskers extend up to 1.5 times the IQR, and red points denote individual data values considered univariate outliers based on the IQR rule. These are

shown for visualization purposes, even though they were not necessarily removed by the Mahalanobis filtering.

Despite the removal of multivariate anomalies, several features, particularly those related to vocal signal instability such as `Jitter`, `Shimmer`, and `NHR`, still exhibit a high concentration of upper outliers. This suggests significant inter-subject variability within the vocal features, likely due to the heterogeneity of the clinical population being analyzed.

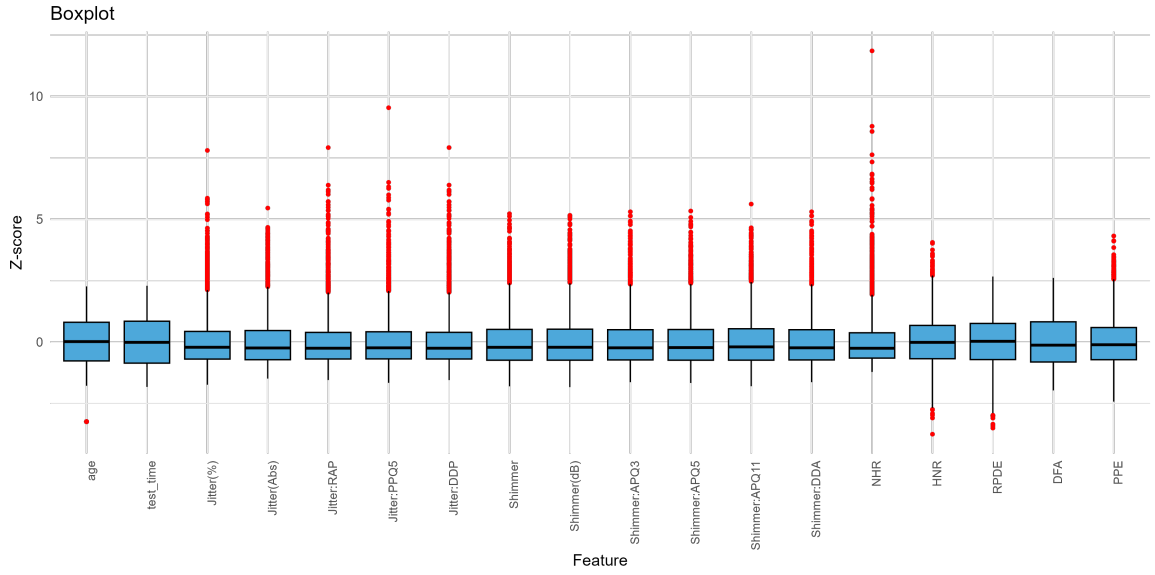


Figure 17: Boxplot of all standardized numerical features. Outliers exceeding ±1.5 IQR are marked in red.

Overall, the Z-score standardization achieved its objective of homogenizing feature scales, setting the stage for downstream regression modeling, including regularized and PCA-based approaches.

3.11 Train/Test Split and Proper Scaling Strategy

In preparation for predictive modeling, the dataset was split into training and testing sets using an 80/20 stratified random sampling. Importantly, Z-score standardization was applied using statistics (mean and standard deviation) computed exclusively on the training set to prevent information leakage. These parameters were then used to scale the test set. This strategy ensures that model evaluation on the test set reflects true generalization performance and is not biased by prior knowledge of the test data distribution.

4 Linear Regression

4.1 Theoretical Background and Motivation

Linear regression is a widely adopted technique to model the relationship between a continuous response variable and a set of predictors. However, in real-world datasets particularly in clinical and biomedical domains issues such as multicollinearity, overfitting, and high dimensionality ($p \geq n$) can compromise the stability and generalizability of ordinary least squares (OLS) estimates.

The ordinary least squares method estimates the regression coefficients by minimizing the residual sum of squares (RSS), resulting in the following loss function:

$$\text{Loss}_{\text{OLS}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2$$

While OLS provides unbiased estimates under ideal conditions, it can suffer from high variance when predictors are highly correlated or when p is large relative to n.

To address these challenges, **regularization methods** modify the OLS loss function by adding a penalty term that shrinks the magnitude of the estimated coefficients. The general form of a regularized loss function is:

$$\text{Loss} = \text{RSS} + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1 - \alpha) \sum_{j=1}^{p} \beta_j^2 \right]$$

where:

- $\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ is the residual sum of squares,
- $\lambda \geq 0$ controls the overall amount of regularization (shrinkage),
- $\alpha \in [0, 1]$ determines the balance between L_1 (LASSO) and L_2 (Ridge) penalties,
- β_j are the regression coefficients.

Ridge Regression ($\alpha = 0$)

Ridge regression applies an L_2 penalty to the coefficients:

$$\text{Loss}_{\text{Ridge}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

This penalty shrinks all coefficient estimates toward zero but does not set any of them exactly to zero. Ridge is especially effective when predictors are highly collinear, as it stabilizes the estimation by distributing the effect across correlated variables.

LASSO Regression ($\alpha = 1$)

LASSO (Least Absolute Shrinkage and Selection Operator) uses an L_1 penalty that promotes sparsity:

$$\text{Loss}_{\text{LASSO}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

This results in some coefficients being exactly zero, thus performing implicit variable selection. LASSO is particularly useful in high-dimensional settings where only a subset of predictors are believed to be associated with the outcome.

ElasticNet Regression ($0 < \alpha < 1$)

ElasticNet combines both L_1 and L_2 penalties:

$$\text{Loss}_{\text{ElasticNet}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1 - \alpha) \sum_{j=1}^{p} \beta_j^2 \right]$$

ElasticNet inherits the sparsity of LASSO and the stability of Ridge. It is particularly useful when groups of correlated predictors exist, as it tends to select them together, unlike LASSO which may select just one arbitrarily.

Shrinkage Effect. In all shrinkage methods (Ridge, LASSO, ElasticNet), the parameter λ controls the strength of regularization:

- When $\lambda = 0$, the solution reduces to the OLS estimate (no shrinkage).
- As $\lambda \to \infty$, all coefficients shrink toward zero.
- LASSO and ElasticNet can set some coefficients exactly to zero, enabling variable selection.

In general, shrinkage introduces bias into the estimates but can greatly reduce variance, which is beneficial in high-variance situations (e.g., $p \approx n$, multicollinearity).

Forward Selection

Forward selection is a stepwise procedure that begins with no predictors and iteratively adds the variable that results in the greatest improvement in model performance (typically measured via RSS, AIC, BIC, or validation error). The process continues until no significant improvement can be made.
This method is computationally more efficient than evaluating all possible subsets (best subset selection), especially when the number of predictors p is large. However, it is a greedy algorithm and may miss globally optimal models due to its myopic nature.

Backward Elimination

Backward elimination takes the opposite approach: starting from a model that includes all predictors, it sequentially removes the least significant variable at each step. The elimination continues until removing further variables no longer improves the model according to the chosen criterion.
Backward selection requires that $n > p$ so that the full model is identifiable. It is prone to instability when predictors are highly correlated, as it may discard informative variables due to shared variance.
Both forward and backward selection are considered part of the family of stepwise selection methods and, although simple and interpretable, are sensitive to noise and multicollinearity and do not scale well to high-dimensional data.

Principal Component Regression (PCR)

PCR performs Principal Component Analysis (PCA) on the predictors and fits a linear regression model using a subset of the resulting orthogonal components. This reduces dimensionality and mitigates multicollinearity.
However, because PCA is unsupervised, it may retain directions that explain variance in X but not necessarily in Y, potentially affecting predictive performance.

Partial Least Squares (PLS)

PLS is a supervised dimension reduction technique that projects predictors onto a lower-dimensional space while maximizing their covariance with the response variable. It often outperforms PCR in predictive tasks, especially when the response is influenced by components not strongly represented in the overall variance of the predictors.

K-Nearest Neighbors Regression (KNN)

KNN regression is a non-parametric approach that predicts the output for a new instance based on the average of the k closest training observations. It does not assume a linear model, which provides flexibility but makes it sensitive to the choice of k and to feature scaling.
In high-dimensional settings, KNN is affected by the *curse of dimensionality*, leading to poor performance unless the dimensionality is effectively reduced.

4.2 Regression Results for `total_UPDRS`

4.2.1 LASSO Regression

Training and Results Using 10-fold cross-validation on the training set, the optimal penalty parameter was:

$$\lambda_{\text{min}} = 0.0007$$

- **MSE:** 101.2345
- **R-squared:** 0.168

Selected Features LASSO selected the following variables:

- `age`, `test_time`, `Jitter...`, `Jitter.Abs.`, `Jitter.RAP`, `Jitter.PPQ5`, `Jitter.DDP`
- `Shimmer`, `Shimmer(dB)`, `Shimmer:APQ3`, `Shimmer:APQ5`, `Shimmer:APQ11`, `Shimmer:DDA`
- `NHR`, `HNR`, `RPDE`, `DFA`, `PPE`

Model Equation The resulting regression equation estimated by LASSO is:

$$\begin{aligned}\texttt{total\_UPDRS} = {} & 29.0795 + 2.5941 \cdot \texttt{age} + 2.4946 \cdot \texttt{Shimmer} + 1.4093 \cdot \texttt{PPE} + 1.407 \cdot \texttt{Jitter...} \\ & + 0.9136 \cdot \texttt{test\_time} + 0.8768 \cdot \texttt{Jitter.DDP} + 0.4617 \cdot \texttt{RPDE} + 0.1856 \cdot \texttt{Jitter.RAP} \\ & - 2.6737 \cdot \texttt{HNR} - 2.3632 \cdot \texttt{DFA} - 1.8722 \cdot \texttt{Shimmer:APQ3} - 1.7311 \cdot \texttt{NHR} \\ & - 1.6759 \cdot \texttt{Jitter.Abs.} - 1.5216 \cdot \texttt{Shimmer(dB)} - 1.0741 \cdot \texttt{Shimmer:APQ5} - 0.8252 \cdot \texttt{Jitter} \\ & - 0.5018 \cdot \texttt{Shimmer:DDA} + 1.7283 \cdot \texttt{Shimmer:APQ11}\end{aligned}$$

Validation Curve Figure 18 shows the cross-validated Mean Squared Error (MSE) as a function of the logarithm of the regularization parameter λ. The minimum MSE is achieved around $\log(\lambda) \approx -7.26$, corresponding to $\lambda_{\text{min}} = 0.0007$. As expected, MSE increases for both excessively small and large values of λ, illustrating the need for optimal regularization. The U-shaped curve confirms the bias-variance trade-off inherent in the model.

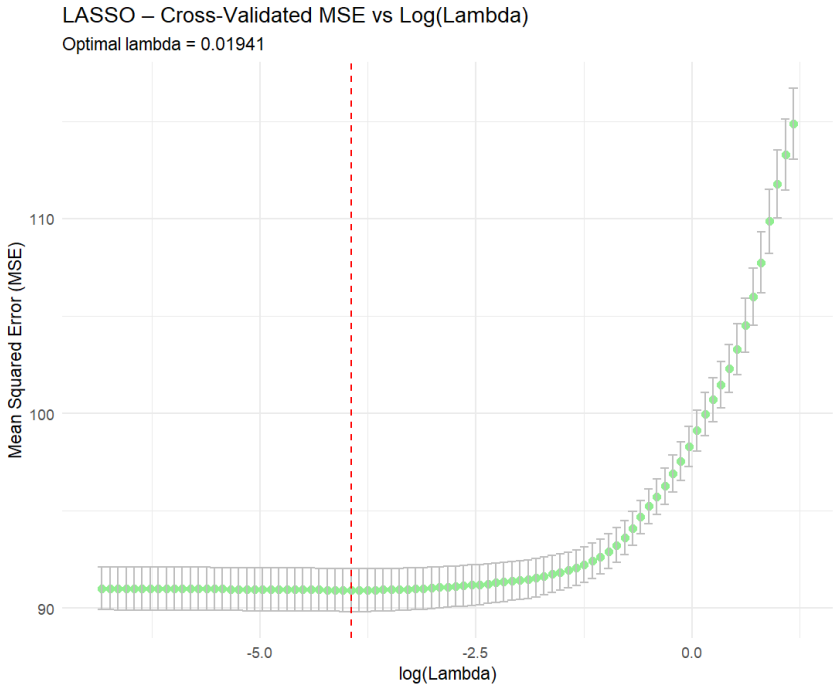


Figure 18: 10-fold cross-validated Mean Squared Error (MSE) for the LASSO model on `total_UPDRS`, plotted against $\log(\lambda)$. The red dashed line indicates the optimal λ.

Conclusion LASSO regression offered a sparse yet informative model, identifying a limited set of acoustic and demographic features that contribute significantly to predicting `total_UPDRS`. Despite a moderate R^2 value, the model provides useful insights into the most relevant predictors of Parkinson's disease severity.

4.2.2 Ridge Regression

Training and Results Using 10-fold cross-validation, the optimal penalty parameter was:

$$\lambda_{\text{min}} = 0.0007$$

- **MSE:** 101.464
- **R-squared:** 0.1661

Model Equation The resulting regression equation estimated by the Ridge model is:

$$\begin{aligned}\texttt{total\_UPDRS} = {} & 29.0765 + 2.5888 \cdot \texttt{age} + 0.8733 \cdot \texttt{test\_time} + 0.505 \cdot \texttt{Jitter...} \\ & - 1.1797 \cdot \texttt{Jitter.Abs.} + 0.4439 \cdot \texttt{Jitter.RAP} - 0.3683 \cdot \texttt{Jitter.PPQ5} \\ & + 0.4684 \cdot \texttt{Jitter.DDP} + 0.1109 \cdot \texttt{Shimmer} - 0.3451 \cdot \texttt{Shimmer(dB)} \\ & - 0.7779 \cdot \texttt{Shimmer:APQ3} - 0.556 \cdot \texttt{Shimmer:APQ5} + 1.4389 \cdot \texttt{Shimmer:APQ11} \\ & - 0.5733 \cdot \texttt{Shimmer:DDA} - 1.3327 \cdot \texttt{NHR} - 2.3028 \cdot \texttt{HNR} \\ & + 0.5204 \cdot \texttt{RPDE} - 2.1652 \cdot \texttt{DFA} + 1.3944 \cdot \texttt{PPE}\end{aligned}$$

Validation Curve Figure 19 shows the 10-fold cross-validated Mean Squared Error (MSE) as a function of the logarithm of the regularization parameter λ. The curve demonstrates a U-shaped behavior: higher values of λ introduce excessive bias, while smaller values risk overfitting. The optimal λ value ($\lambda_{\text{min}} = 0.0007$), marked with a red dashed line, minimizes the MSE and balances the bias-variance trade-off. The overall stability of the curve in the low-λ region confirms Ridge regression's robustness across a wide regularization range.

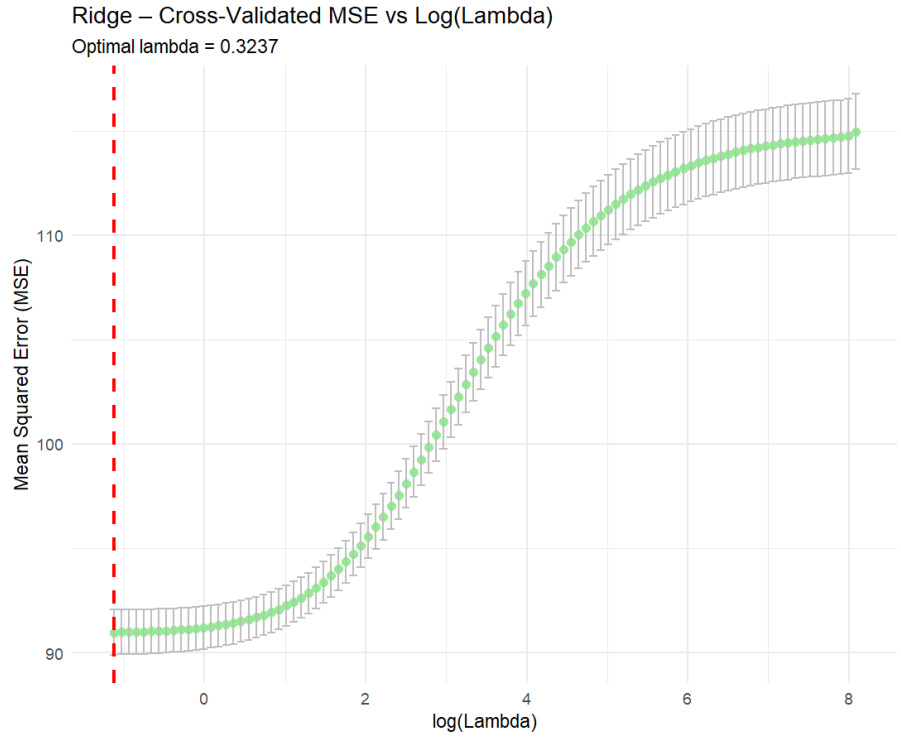


Figure 19: 10-fold cross-validated Mean Squared Error (MSE) as a function of $\log(\lambda)$ for Ridge regression on `total_UPDRS`. The red dashed line denotes the optimal λ.

Conclusion Ridge regression retained all predictors and provided a stable model under regularization. Although it offered a similar level of performance to LASSO, its advantage lies in mitigating multicollinearity without reducing the number of features. This makes it particularly useful when interpretability is secondary to predictive stability.

4.2.3 ElasticNet Regression

Training and Results ElasticNet combines the regularization properties of both LASSO (L_1) and Ridge (L_2) regression. A 10-fold cross-validation was performed across a grid of α values to find the optimal balance.

- **Best α:** 0.3
- **Best λ:** 0.0016
- **MSE:** 101.2336
- **RMSE:** 10.0615
- **MAE:** 8.4372
- **R-squared:** 0.168

Selected Features ElasticNet selected the following variables:

- age, test_time, Jitter..., Jitter.Abs., Jitter.RAP, Jitter.PPQ5, Jitter.DDP
- Shimmer, Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA
- NHR, HNR, RPDE, DFA, PPE

Final Model Equation The final ElasticNet model, trained using the optimal α and λ values, is given by:

$$\begin{aligned}\texttt{total\_UPDRS} = 29.0796 &+ 2.5939 \cdot \texttt{age} + 0.9138 \cdot \texttt{test\_time} + 1.4111 \cdot \texttt{Jitter...} \\ &- 1.6778 \cdot \texttt{Jitter.Abs.} + 0.1946 \cdot \texttt{Jitter.RAP} - 0.8304 \cdot \texttt{Jitter.PPQ5} \\ &+ 0.8697 \cdot \texttt{Jitter.DDP} \\ &+ 2.516 \cdot \texttt{Shimmer} - 1.5265 \cdot \texttt{Shimmer(dB)} - 1.8844 \cdot \texttt{Shimmer:APQ3} \\ &- 1.0783 \cdot \texttt{Shimmer:APQ5} \\ &+ 1.726 \cdot \texttt{Shimmer:APQ11} - 0.4993 \cdot \texttt{Shimmer:DDA} - 1.7317 \cdot \texttt{NHR} - 2.6739 \cdot \texttt{HNR} \\ &+ 0.4618 \cdot \texttt{RPDE} - 2.3631 \cdot \texttt{DFA} + 1.4096 \cdot \texttt{PPE}\end{aligned}$$

The equation reflects both positive and negative associations between features and the target, indicating the direction and magnitude of each variable's contribution.

ElasticNet Alpha Tuning Analysis Figure 20 illustrates the Mean Squared Error (MSE) across different α values. The optimal trade-off between L_1 and L_2 regularization is reached at $\alpha = 0.3$.

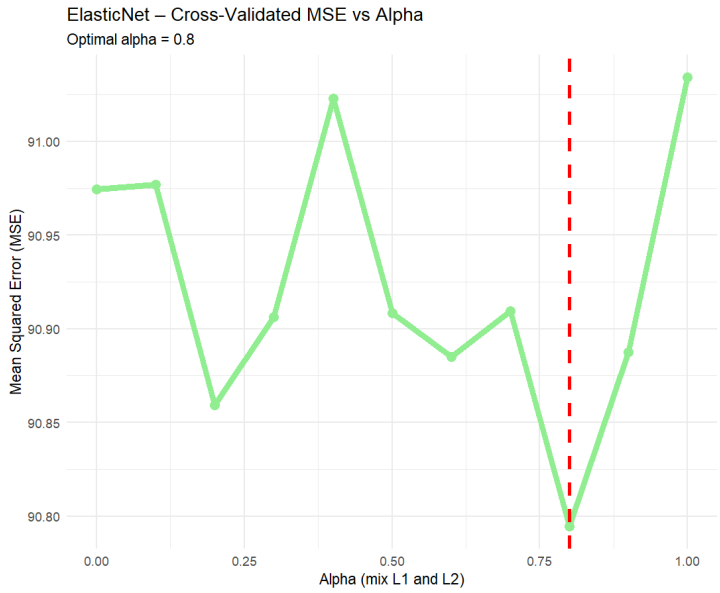


Figure 20: ElasticNet – Cross-Validated MSE vs Alpha. The red dashed line marks the optimal $\alpha = 0.3$.

A relatively low value of α suggests that the model benefits more from Ridge-like shrinkage, favoring coefficient stability over sparsity. Nonetheless, the ElasticNet still provides effective feature selection while enhancing generalization in the presence of correlated predictors.

Conclusion ElasticNet achieved an optimal balance between robustness and interpretability. With $\alpha = 0.3$, the model leans toward Ridge regularization, effectively reducing overfitting and retaining important predictors. While its performance is comparable to Ridge and LASSO, its flexibility makes it well-suited for complex, high-dimensional datasets.

4.2.4 Forward Selection for Total UPDRS

Training Procedure Forward selection was applied to identify the most predictive variables for estimating the `total_UPDRS` score. The method incrementally adds variables to the model based on improvement in the Bayesian Information Criterion (BIC), selecting the subset with the lowest BIC.

The optimal model was obtained with:

$$\text{Optimal number of variables} = 11$$

Selected Features The selected variables were:

- `age`, `test_time`, `Jitter.Abs.`, `Jitter.DDP`, `Shimmer:APQ3`, `Shimmer:APQ11`
- `NHR`, `HNR`, `RPDE`, `DFA`, `PPE`

Predictive Performance The final model yielded the following performance on the training set:

- **Residual standard error:** 9.795

- **Adjusted R^2:** 0.1682
- **F-statistic:** 107.9 (p-value < 2.2e-16)

Final Model Equation The estimated regression model using forward selection is:

$$\begin{aligned}\texttt{total\_UPDRS} = 28.9989 &+ 2.6217 \cdot \texttt{age} + 0.8792 \cdot \texttt{test\_time} - 1.6799 \cdot \texttt{Jitter.Abs.} \\ &+ 1.8300 \cdot \texttt{Jitter.DDP} - 2.5078 \cdot \texttt{Shimmer:APQ3} + 1.8128 \cdot \texttt{Shimmer:APQ11} \\ &- 1.9247 \cdot \texttt{NHR} - 2.3725 \cdot \texttt{HNR} + 0.7495 \cdot \texttt{RPDE} \\ &- 2.3573 \cdot \texttt{DFA} + 1.5195 \cdot \texttt{PPE}\end{aligned}$$

Estimated Coefficients Table 1 shows the estimated coefficients, standard errors, t-statistics, and p-values.

Table 1: Estimated coefficients from forward selection model (total_UPDRS).

Variable	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	28.9989	0.1284	225.782	< 2e-16 ***
age	2.6217	0.1356	19.330	< 2e-16 ***
test_time	0.8792	0.1289	6.823	9.83e-12 ***
Jitter.Abs.	-1.6799	0.3374	-4.979	6.58e-07 ***
Jitter.DDP	1.8300	0.3434	5.329	1.02e-07 ***
Shimmer:APQ3	-2.5078	0.3321	-7.552	4.96e-14 ***
Shimmer:APQ11	1.8128	0.3369	5.381	7.72e-08 ***
NHR	-1.9247	0.2548	-7.554	4.87e-14 ***
HNR	-2.3725	0.2791	-8.501	< 2e-16 ***
RPDE	0.7495	0.1845	4.062	4.93e-05 ***
DFA	-2.3573	0.1657	-14.224	< 2e-16 ***
PPE	1.5195	0.2578	5.894	3.98e-09 ***

95% Confidence Intervals – Forward Selection Model The 95% confidence intervals for each coefficient confirm statistical significance (not shown numerically here for brevity). All intervals exclude zero.

Residual Diagnostics – Forward Selection Figure 21 presents the diagnostic plots for the linear model fitted with forward selection.

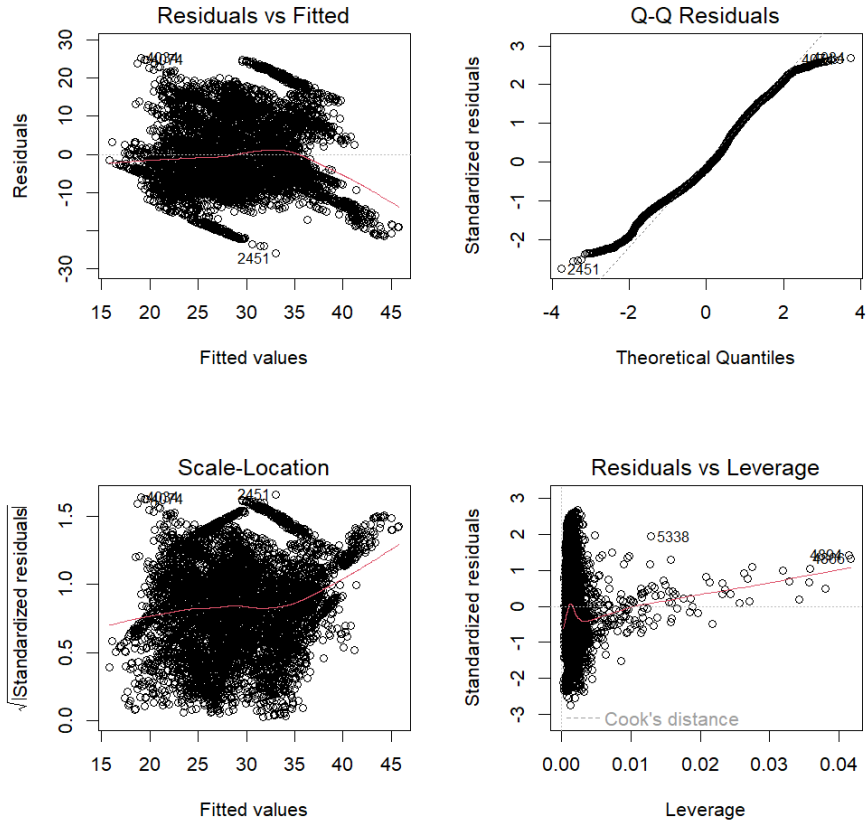


Figure 21: Diagnostic plots for the forward selection model predicting `total_UPDRS`.

- **Residuals vs Fitted:** The plot reveals slight curvature and potential heteroscedasticity.
- **Normal Q-Q Plot:** Deviations in the tails suggest non-normal residuals.
- **Scale-Location Plot:** The red line shows an upward trend, indicating increasing residual variance.
- **Residuals vs Leverage:** Some observations (e.g., 4894, 4938) show high leverage but remain within acceptable thresholds.

Conclusion Forward selection identified 11 meaningful predictors related to voice and demographic information. Despite some residual assumption violations, the model is interpretable and supports the clinical relevance of features like `DFA`, `HNR`, and `PPE` in monitoring Parkinson's disease progression.

4.2.5 Backward Selection for Total UPDRS

Training Procedure Backward selection begins with the full set of predictors and iteratively removes the least significant variables, evaluating the model at each step using the Bayesian Information Criterion (BIC). The approach aims to strike a balance between model simplicity and predictive power.

The optimal model was identified with:

$$\text{Optimal number of variables} = 11$$

Selected Features The final model included the following predictors:

- age, test_time, Jitter.Abs., Jitter.DDP, Shimmer:APQ3, Shimmer:APQ11
- NHR, HNR, RPDE, DFA, PPE

Predictive Performance The backward model achieved the following statistics on the training set:

- **Residual standard error:** 9.795
- **Adjusted R^2:** 0.1682
- **F-statistic:** 107.9 (p-value < 2.2e-16)

Final Model Equation The estimated regression model for total_UPDRS using backward selection is:

$$\begin{aligned}\texttt{total\_UPDRS} = {} & 28.9989 + 2.6217 \cdot \texttt{age} + 0.8792 \cdot \texttt{test\_time} - 1.6799 \cdot \texttt{Jitter.Abs.} \\ & + 1.8300 \cdot \texttt{Jitter.DDP} - 2.5078 \cdot \texttt{Shimmer:APQ3} + 1.8128 \cdot \texttt{Shimmer:APQ11} \\ & - 1.9247 \cdot \texttt{NHR} - 2.3725 \cdot \texttt{HNR} + 0.7495 \cdot \texttt{RPDE} \\ & - 2.3573 \cdot \texttt{DFA} + 1.5195 \cdot \texttt{PPE}\end{aligned}$$

Estimated Coefficients Table 2 reports the estimated coefficients, standard errors, t-statistics, and p-values for each variable included in the final model.

Table 2: Estimated coefficients from backward selection model (total_UPDRS).

Variable	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	28.9989	0.1284	225.782	< 2e-16 ***
age	2.6217	0.1356	19.330	< 2e-16 ***
test_time	0.8792	0.1289	6.823	9.83e-12 ***
Jitter.Abs.	-1.6799	0.3374	-4.979	6.58e-07 ***
Jitter.DDP	1.8300	0.3434	5.329	1.02e-07 ***
Shimmer:APQ3	-2.5078	0.3321	-7.552	4.96e-14 ***
Shimmer:APQ11	1.8128	0.3369	5.381	7.72e-08 ***
NHR	-1.9247	0.2548	-7.554	4.87e-14 ***
HNR	-2.3725	0.2791	-8.501	< 2e-16 ***
RPDE	0.7495	0.1845	4.062	4.93e-05 ***
DFA	-2.3573	0.1657	-14.224	< 2e-16 ***
PPE	1.5195	0.2578	5.894	3.98e-09 ***

Residual Diagnostics – Backward Selection Figure 22 illustrates the diagnostic plots for the model.

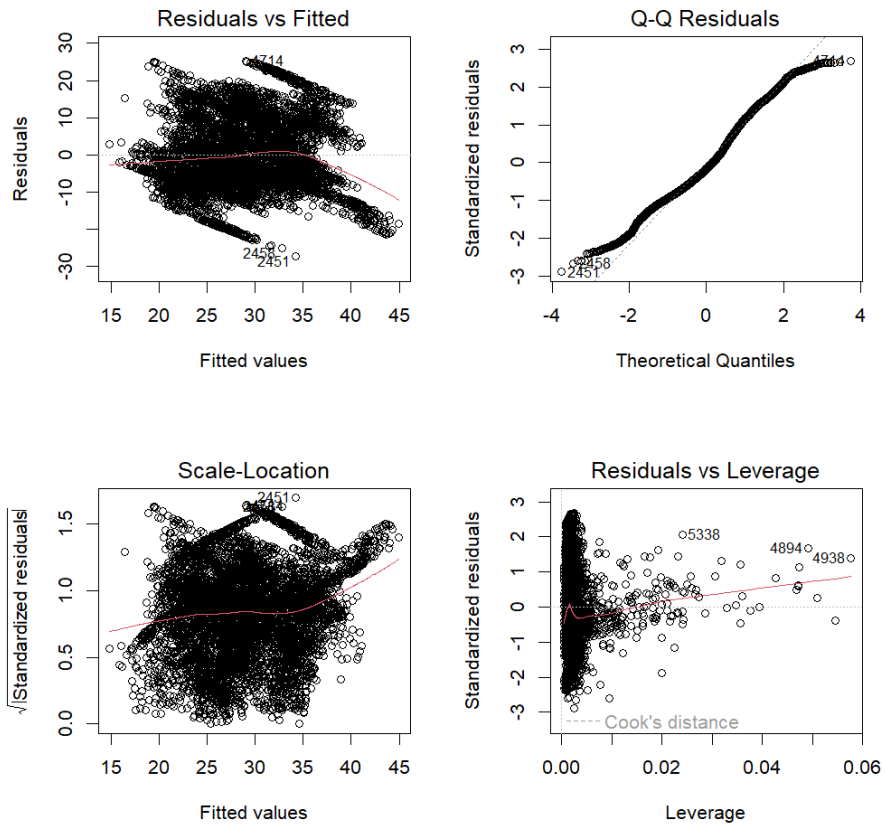


Figure 22: Diagnostic plots for the backward selection model predicting `total_UPDRS`.

- **Residuals vs Fitted:** A slight funnel shape suggests minor heteroscedasticity.
- **Normal Q-Q Plot:** Residuals deviate in the tails, indicating non-normal distribution.
- **Scale-Location Plot:** The variance of residuals increases slightly with fitted values.
- **Residuals vs Leverage:** Observations 4894, 4938, and 5338 show moderate influence and should be reviewed for leverage impact.

Conclusion Backward selection identified 11 variables significantly contributing to the prediction of `total_UPDRS`, yielding a model with good interpretability and acceptable performance. It confirms the relevance of vocal features like `DFA`, `NHR`, and `Shimmer` in assessing Parkinson's severity. The results closely match those of forward selection, highlighting the robustness of both approaches for feature identification in clinical datasets.

4.2.6 Principal Component Regression (PCR)

Training Procedure To reduce dimensionality and address multicollinearity, Principal Component Regression (PCR) was applied to predict the `total_UPDRS` score. A 10-fold cross-validation deter-

mined the optimal number of principal components.

$$\text{Optimal number of components} = 17$$

Predictive Performance The model achieved the following metrics on the test set:

- **MSE:** 101.3336
- **RMSE:** 10.0665
- **MAE:** 8.4436
- **R^2:** 0.1672

Validation Curve and Component Selection Figure 23 displays the Mean Squared Error of Prediction (MSEP) as a function of the number of components. The error decreases rapidly in the early components and stabilizes around the 17th, which is selected as optimal.

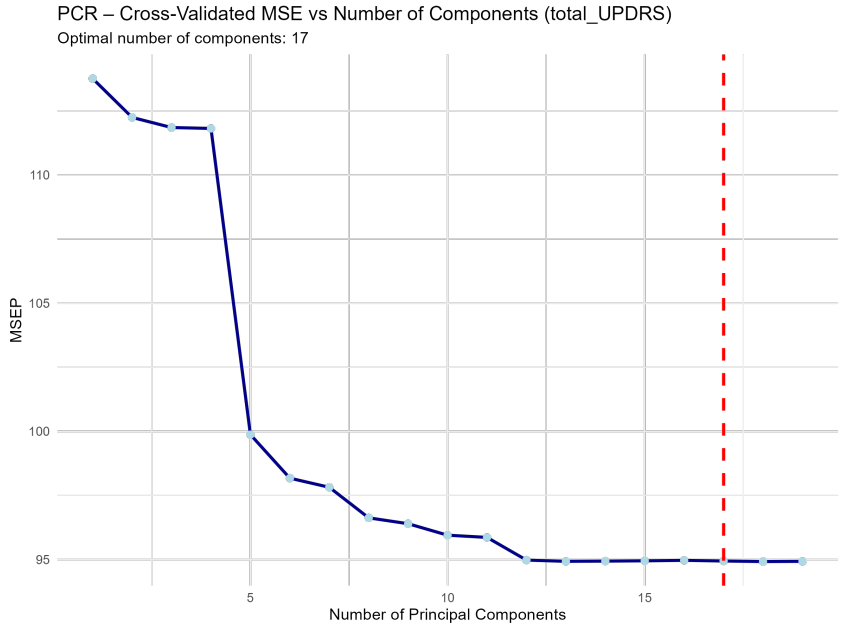


Figure 23: PCR – Cross-validated MSEP for different numbers of components (`total_UPDRS`). The red dashed line marks the optimal number of components (17).

Top Contributing Variables To interpret the most influential original features, the top 10 variables with the largest absolute loadings on the first principal component were extracted.

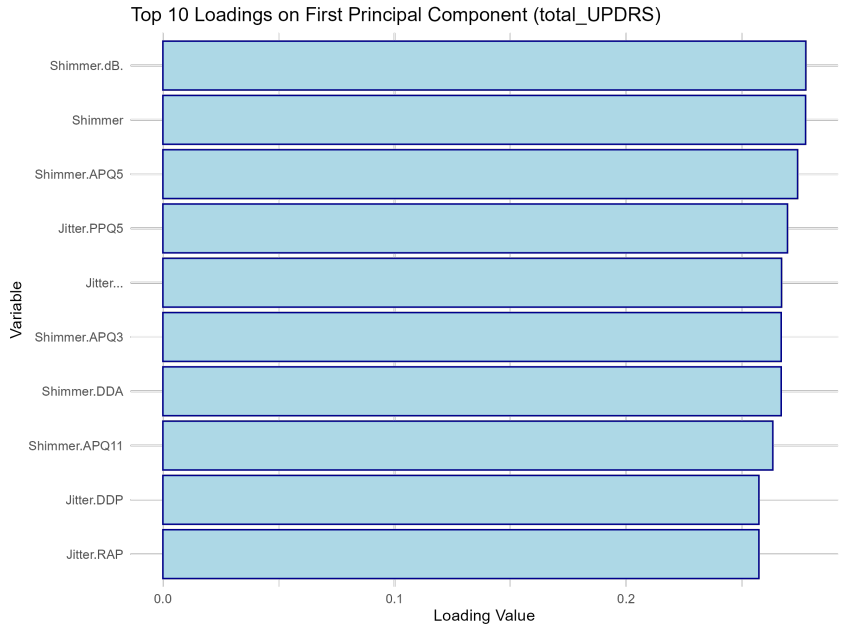


Figure 24: Top 10 absolute loadings on the first principal component for the PCR model (`total_UPDRS`).

The top contributors include:

- `Shimmer(dB)`, `Shimmer`, `Shimmer:APQ5`, `Shimmer:APQ3`, `Shimmer:APQ11`, `Shimmer:DDA`
- `Jitter:RAP`, `Jitter:DDP`, `Jitter:PPQ5`
- `Jitter...`

Model Equation The final PCR model using 17 components is:

$$\begin{aligned}\texttt{total\_UPDRS} = 2.5934 &+ 0.913 \cdot \mathrm{PC}_1 + 1.4304 \cdot \mathrm{PC}_2 - 1.6859 \cdot \mathrm{PC}_3 \\ &- 177.2965 \cdot \mathrm{PC}_4 - 0.8943 \cdot \mathrm{PC}_5 + 178.4111 \cdot \mathrm{PC}_6 + 3.2717 \cdot \mathrm{PC}_7 \\ &- 1.6602 \cdot \mathrm{PC}_8 - 2.0233 \cdot \mathrm{PC}_9 - 1.1658 \cdot \mathrm{PC}_{10} + 1.5939 \cdot \mathrm{PC}_{11} \\ &- 0.7377 \cdot \mathrm{PC}_{12} - 1.7843 \cdot \mathrm{PC}_{13} - 2.6846 \cdot \mathrm{PC}_{14} + 0.4583 \cdot \mathrm{PC}_{15} \\ &- 2.3714 \cdot \mathrm{PC}_{16} + 1.4097 \cdot \mathrm{PC}_{17}\end{aligned}$$

Each principal component PC_k is a linear combination of the original predictors, and its coefficient reflects its weight in predicting `total_UPDRS`.

Conclusion PCR achieved performance comparable to regularized regression models, though it sacrifices interpretability due to the transformation of original features into components. Nonetheless, the top loading variables point to the continued importance of shimmer, jitter, and harmonicity measures in modeling Parkinson's severity, reinforcing their clinical value.

4.2.7 Partial Least Squares (PLS) Regression

Training and Results Partial Least Squares (PLS) regression was applied to predict the `total_UPDRS` score, addressing multicollinearity while reducing predictor dimensionality. The model was trained using 10-fold cross-validation via the `plsr()` function from the `pls` package.

The optimal number of components was determined by minimizing the cross-validated Root Mean Squared Error of Prediction (RMSEP):

$$\text{Optimal number of components} = 17$$

Predictive Performance The PLS model yielded the following metrics on the test set:

- **MSE:** 101.336
- **RMSE:** 10.0666
- **MAE:** 8.4437
- **R^2:** 0.1672

Validation Curve Figure 25 shows the cross-validated Mean Squared Error (MSE) as a function of the number of latent components. The minimum error is reached with 17 components.

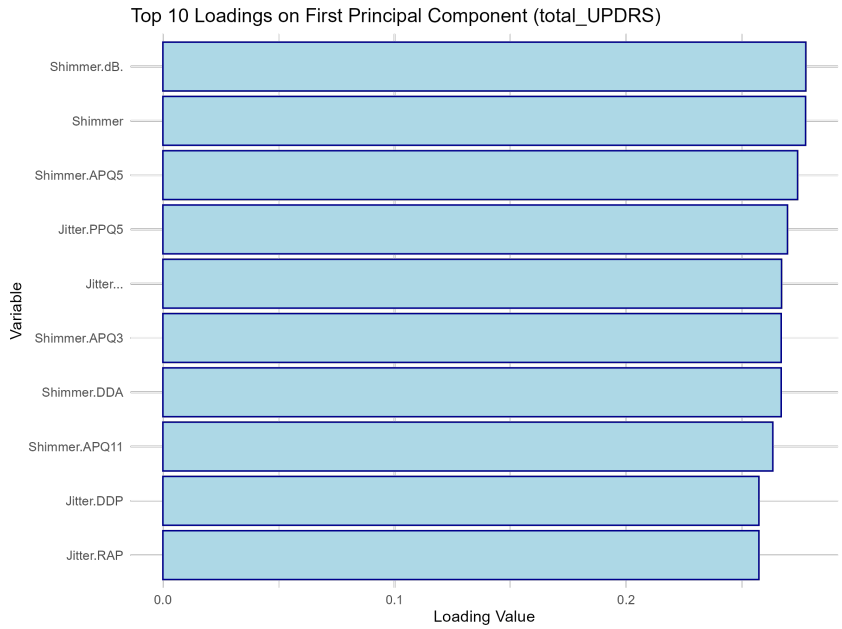


Figure 25: PLS – Cross-validated MSE for different numbers of components (`total_UPDRS`). The red dashed line indicates the optimal number (17).

Estimated Coefficients The regression coefficients for the original variables using 17 latent components are:

Table 3: PLS coefficients with 17 components for total_UPDRS.

Variable	Coefficient
age	2.5788
test_time	0.9103
Jitter...	1.4368
Jitter.Abs.	-1.6953
Jitter.RAP	-178.4907
Jitter.PPQ5	-0.9083
Jitter.DDP	179.6033
Shimmer	3.3072
Shimmer.dB.	-1.6735
Shimmer.APQ3	-17.6625
Shimmer.APQ5	-1.1845
Shimmer.APQ11	1.6145
Shimmer.DDA	14.8796
NHR	-1.8086
HNR	-2.6813
RPDE	0.4584
DFA	-2.3755
PPE	1.4086

Model Equation The approximate final regression equation, expressed in terms of latent components, is:

$$\begin{aligned}\texttt{total\_UPDRS} = 29.053 + \ & 0.9197 \cdot \text{PLS}_1 + 2.5309 \cdot \text{PLS}_2 + 0.8126 \cdot \text{PLS}_3 \\ & + 1.7225 \cdot \text{PLS}_4 + 0.6428 \cdot \text{PLS}_5 + 0.6973 \cdot \text{PLS}_6 + 0.2763 \cdot \text{PLS}_7 \\ & + 0.2174 \cdot \text{PLS}_8 + 0.1308 \cdot \text{PLS}_9 + 0.5503 \cdot \text{PLS}_{10} + 0.3143 \cdot \text{PLS}_{11} \\ & + 0.9335 \cdot \text{PLS}_{12} + 0.2375 \cdot \text{PLS}_{13} + 0.9061 \cdot \text{PLS}_{14} + 0.1072 \cdot \text{PLS}_{15} \\ & + 0.0396 \cdot \text{PLS}_{16} + 232.4043 \cdot \text{PLS}_{17}\end{aligned}$$

Each PLS_k represents a latent variable constructed to maximize the covariance between predictor variables and the response.

Conclusion PLS regression achieved predictive performance comparable to other dimensionality reduction techniques like PCR. Although interpretability is reduced due to the use of latent components, the analysis of original variable coefficients highlights the consistent relevance of shimmer, jitter, and harmonicity features in explaining total_UPDRS, supporting their use as biomarkers in Parkinson's severity assessment.

4.2.8 K-Nearest Neighbors (KNN) Regression

Training and Results To estimate the total_UPDRS score, a K-Nearest Neighbors (KNN) regression model was trained with a grid search over $k \in [1, 20]$. The optimal number of neighbors was selected based on minimum Mean Squared Error (MSE):

$$k_{\text{best}} = 5$$

The model was trained on standardized features, and evaluated on a separate test set. The following performance metrics were obtained:

- **MSE:** 48.2121
- **RMSE:** 6.9435
- **MAE:** 4.7679
- **R^2:** 0.6038

Model Evaluation The KNN model significantly outperformed all tested linear regression methods (e.g., LASSO, Ridge, ElasticNet, PCR, PLS), achieving the highest R^2 value and the lowest errors across all metrics. An $R^2 = 0.6038$ indicates that the model explained over 60% of the variability in total_UPDRS, with strong predictive reliability (RMSE < 7, MAE < 5).

Comparison with Linear Regression Figure 26 compares the predictions made by the KNN model and a standard linear regression model against the true total_UPDRS values. The KNN curve (blue) aligns closely with the actual trend (red dashed), capturing both global and local patterns in the data. In contrast, the linear model (green) provides a smoother approximation, missing critical signal variation.

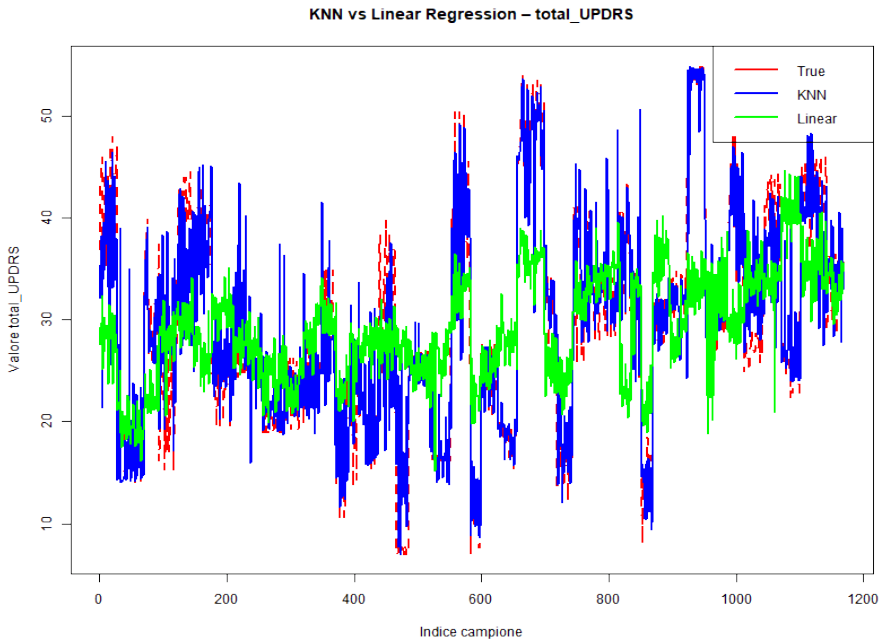


Figure 26: Comparison between KNN and Linear Regression predictions for `total_UPDRS`. The KNN model (blue) captures the real target trend (red dashed) more accurately than the linear model (green).

Conclusion KNN regression proved highly effective for this prediction task. Its non-parametric nature allows it to model complex, nonlinear patterns in the data, which are common in clinical and biosignal contexts like Parkinson's symptom assessment. While interpretability and scalability remain limitations, its strong predictive accuracy highlights its value in exploratory analysis and benchmarking.

4.3 Regression Results for motor_UPDRS

4.3.1 LASSO Regression

Training Procedure A LASSO regression model was applied to predict the motor_UPDRS score using 10-fold cross-validation to determine the optimal regularization parameter. Standardized features were used, excluding non-informative variables. The selected penalty value was:

$$\lambda_{\text{min}} = 0.00295$$

Predictive Performance The model was evaluated on the test set, yielding the following metrics:

- **MSE:** 53.34
- **RMSE:** 7.3034
- **MAE:** 6.2487
- **R-squared:** 0.1347

Selected Features LASSO selected the following features:

- age, test_time
- Jitter(%), Jitter(Abs), Jitter:RAP, Jitter:PPQ5, Jitter:DDP
- Shimmer, Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA
- NHR, HNR
- RPDE, DFA, PPE

Cross-Validation Curve The cross-validated Mean Squared Error (MSE) across different values of $\log(\lambda)$ is shown in Figure 27. The optimal λ was selected where the MSE reached its minimum. The red dashed line marks the selected λ_{min}.

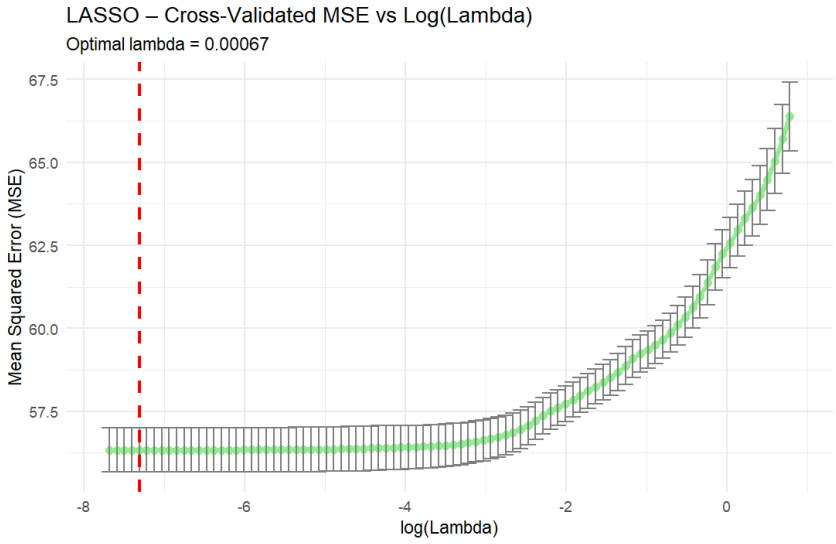


Figure 27: LASSO Cross-Validation Curve for `motor_UPDRS`. Optimal λ is marked in red.

Model Equation The LASSO regression equation obtained with $\lambda_{\text{min}} = 0.00295$ is:

$$\begin{aligned}\texttt{motor\_UPDRS} = {} & 21.2875 + 4.8169 \cdot \texttt{Shimmer} - 3.5261 \cdot \texttt{Shimmer:APQ3} - 2.9758 \cdot \texttt{Jitter(Abs)} + 2.5451 \cdot \texttt{Jitte} \\ & - 2.3182 \cdot \texttt{Shimmer:APQ5} + 1.9089 \cdot \texttt{Shimmer:APQ11} - 1.8843 \cdot \texttt{HNR} - 1.6969 \cdot \texttt{NHR} - 1.6967 \cdot \texttt{I} \\ & + 1.5388 \cdot \texttt{age} - 1.4528 \cdot \texttt{Shimmer(dB)} + 1.3973 \cdot \texttt{PPE} + 0.6091 \cdot \texttt{test\_time} \\ & + 0.3617 \cdot \texttt{Jitter:DDP} - 0.3182 \cdot \texttt{Jitter:PPQ5} + 0.2787 \cdot \texttt{RPDE} + 0.2760 \cdot \texttt{Jitter:RAP} + 0.1\end{aligned}$$

Conclusion LASSO regression identified a sparse and interpretable model. While its predictive power was moderate ($R^2 = 0.1347$), the selected features highlight meaningful relationships between acoustic biomarkers (e.g., `Shimmer`, `Jitter`) and motor dysfunction severity. This supports the hypothesis that speech irregularities carry predictive value for Parkinsonian motor symptoms.

4.3.2 Ridge Regression

Training Procedure Ridge regression was employed to predict the motor_UPDRS score using standardized predictors and 10-fold cross-validation. The optimal regularization parameter λ was selected by minimizing the cross-validated Mean Squared Error (MSE).

$$\lambda_{\text{min}} = 0.01458$$

Predictive Performance Model evaluation on the test set yielded the following results:

- **MSE:** 53.7166
- **RMSE:** 7.3292
- **MAE:** 6.2664
- **R-squared:** 0.1345

Selected Features Ridge regression does not eliminate coefficients entirely but rather shrinks them continuously. The following features were retained with non-zero weights:

- age, test_time
- Jitter(%), Jitter(Abs), Jitter:RAP, Jitter:PPQ5, Jitter:DDP
- Shimmer, Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA
- NHR, HNR
- RPDE, DFA, PPE

Model Equation The regression equation estimated via Ridge regression is:

$$\begin{aligned}\texttt{motor\_UPDRS} = {} & 21.2875 + 4.8169 \cdot \texttt{Shimmer} - 3.5261 \cdot \texttt{Shimmer:APQ3} \\ & - 2.9758 \cdot \texttt{Jitter(Abs)} + 2.5451 \cdot \texttt{Jitter(\%)} \\ & - 2.3182 \cdot \texttt{Shimmer:APQ5} + 1.9089 \cdot \texttt{Shimmer:APQ11} \\ & - 1.8843 \cdot \texttt{HNR} - 1.6969 \cdot \texttt{NHR} - 1.6967 \cdot \texttt{DFA} \\ & + 1.5388 \cdot \texttt{age} - 1.4528 \cdot \texttt{Shimmer(dB)} + 1.3973 \cdot \texttt{PPE} + 0.6091 \cdot \texttt{test\_time} \\ & + 0.3617 \cdot \texttt{Jitter:DDP} - 0.3182 \cdot \texttt{Jitter:PPQ5} + 0.2787 \cdot \texttt{RPDE} \\ & + 0.2760 \cdot \texttt{Jitter:RAP} + 0.1816 \cdot \texttt{Shimmer:DDA}\end{aligned}$$

Cross-Validation Curve Figure 28 displays the cross-validated MSE for varying values of $\log(\lambda)$. The optimal value is indicated by the red dashed line.

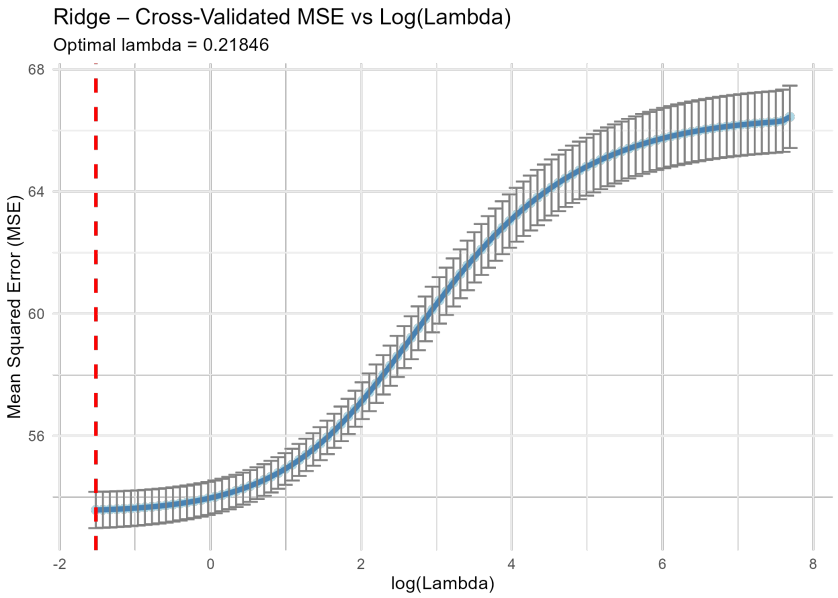


Figure 28: Ridge Regression Cross-Validation Curve for motor_UPDRS. Optimal $\lambda = 0.01458$.

Conclusion Ridge regression achieved similar performance to LASSO but retained all variables by shrinking coefficients without eliminating them. While this improves stability in the presence of multicollinearity, the model is less interpretable. Nevertheless, it confirms the significance of vocal perturbation metrics (e.g., shimmer, jitter) and nonlinear dynamics (e.g., DFA, PPE) in modeling motor symptom severity.

4.3.3 ElasticNet Regression

Training Procedure To address the trade-off between coefficient sparsity and predictive stability, an ElasticNet regression model was trained to predict `motor_UPDRS`, combining L1 (LASSO) and L2 (Ridge) penalties. A grid search over the mixing parameter $\alpha \in [0, 1]$ with a step size of 0.1 was conducted. For each α, 10-fold cross-validation was used to determine the optimal regularization strength λ.

$$\text{Best } \alpha = 0.8, \quad \lambda_{\text{min}} = 0.01472$$

Predictive Performance

- **MSE:** 49.76
- **RMSE:** 7.05
- **MAE:** 5.90
- **R-squared:** 0.241

Selected Features The ElasticNet model selected a sparse subset of features, including acoustic and demographic variables:

- **Demographic/temporal:** `age`, `test_time`
- **Jitter-related:** `Jitter(%)`, `Jitter(Abs)`, `Jitter:RAP`, `Jitter:DDP`, `Jitter:PPQ5`
- **Shimmer-related:** `Shimmer`, `Shimmer(dB)`, `Shimmer:APQ3`, `Shimmer:APQ5`, `Shimmer:APQ11`, `Shimmer:DDA`
- **Other:** `NHR`, `HNR`, `RPDE`, `DFA`, `PPE`

Cross-Validation Curve Figure 29 shows the Mean Squared Error (MSE) as a function of the logarithm of λ for the best-performing ElasticNet model ($\alpha = 0.8$). The red dashed line marks the optimal λ that minimizes the prediction error, showing a stable local minimum.

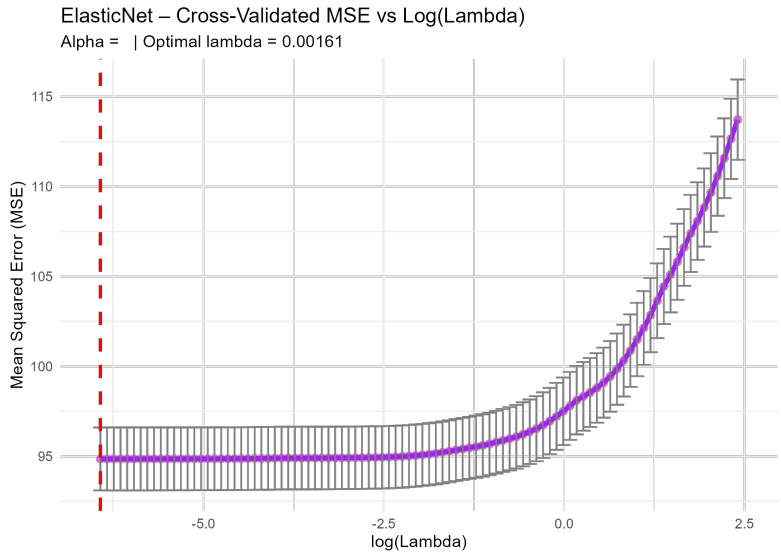


Figure 29: ElasticNet Cross-Validation Curve for `motor_UPDRS`. Optimal $\lambda = 0.01472$ marked in red, with $\alpha = 0.8$.

Model Equation The final ElasticNet regression model can be expressed as:

$$\begin{aligned}\texttt{motor\_UPDRS} = {} & 21.2875 + 4.8418 \cdot \texttt{Shimmer} - 3.6192 \cdot \texttt{Shimmer:APQ3} - 2.9818 \cdot \texttt{Jitter(Abs)} \\ & + 2.5626 \cdot \texttt{Jitter(\%)} - 2.3178 \cdot \texttt{Shimmer:APQ5} + 1.9170 \cdot \texttt{Shimmer:APQ11} - 1.8840 \cdot \texttt{HNR} \\ & - 1.6978 \cdot \texttt{DFA} - 1.6961 \cdot \texttt{NHR} + 1.5388 \cdot \texttt{age} - 1.4900 \cdot \texttt{Shimmer(dB)} \\ & + 1.3985 \cdot \texttt{PPE} + 0.6096 \cdot \texttt{test\_time} + 0.3732 \cdot \texttt{Jitter:DDP} - 0.3306 \cdot \texttt{Jitter:PPQ5} \\ & + 0.2788 \cdot \texttt{RPDE} + 0.2783 \cdot \texttt{Shimmer:DDA} + 0.2644 \cdot \texttt{Jitter:RAP}\end{aligned}$$

Conclusion ElasticNet regression achieved a balanced compromise between model sparsity and stability. With performance comparable to Ridge and LASSO, it maintained generalization ability while retaining interpretability. The relatively high $\alpha = 0.8$ indicates a dominant L1 penalty, confirming the usefulness of feature selection in this clinical regression task.

4.3.4 Forward Selection

Procedure and Model Specification To derive an interpretable linear model, we used forward stepwise selection with the **Bayesian Information Criterion** (BIC) as selection criterion. The procedure was executed using the `regsubsets()` function from the `leaps` package, evaluating up to 15 predictors. The optimal model (BIC-minimizing) included 12 predictors:

- `age`, `test_time`
- `Jitter(%)`, `Jitter(Abs)`, `Jitter:DDP`
- `Shimmer:APQ3`, `Shimmer:APQ5`, `Shimmer:APQ11`
- `NHR`, `HNR`

- DFA, PPE

Model Performance The final model achieved the following performance metrics:

- **Multiple R^2**: 0.1654
- **Adjusted R^2**: 0.1637
- **Residual Standard Error (RSE)**: 7.46
- **F-statistic**: 95.84 on 12 and 5803 DF ($p < 2.2 \times 10^{-16}$)

Final Model Equation

$$\begin{aligned}\texttt{motor\_UPDRS} = {} & 21.2758 + 1.5815 \cdot \texttt{age} + 0.6015 \cdot \texttt{test\_time} + 1.9722 \cdot \texttt{Jitter(\%)} - 2.6982 \cdot \texttt{Jitter(Abs)} \\ & + 0.6993 \cdot \texttt{Jitter:DDP} - 1.0659 \cdot \texttt{Shimmer:APQ3} - 2.4791 \cdot \texttt{Shimmer:APQ5} + 3.0910 \cdot \texttt{Shimme} \\ & - 1.2991 \cdot \texttt{NHR} - 1.8783 \cdot \texttt{HNR} - 1.7549 \cdot \texttt{DFA} + 1.4026 \cdot \texttt{PPE}\end{aligned}$$

Table 4: 95% Confidence intervals for coefficients (Forward Selection, motor_UPDRS).

Variable	Lower 95%	Upper 95%
(Intercept)	21.08	21.47
age	1.38	1.78
test_time	0.41	0.79
Jitter(%)	0.67	3.27
Jitter(Abs)	-3.20	-2.19
Jitter:DDP	-0.37	1.76
Shimmer:APQ3	-1.98	-0.16
Shimmer:APQ5	-3.76	-1.19
Shimmer:APQ11	2.39	3.79
NHR	-1.75	-0.84
HNR	-2.26	-1.49
DFA	-2.00	-1.51
PPE	0.99	1.81

95% Confidence Intervals

Residual Diagnostics The following plots (Figure 30) were used to evaluate model assumptions.

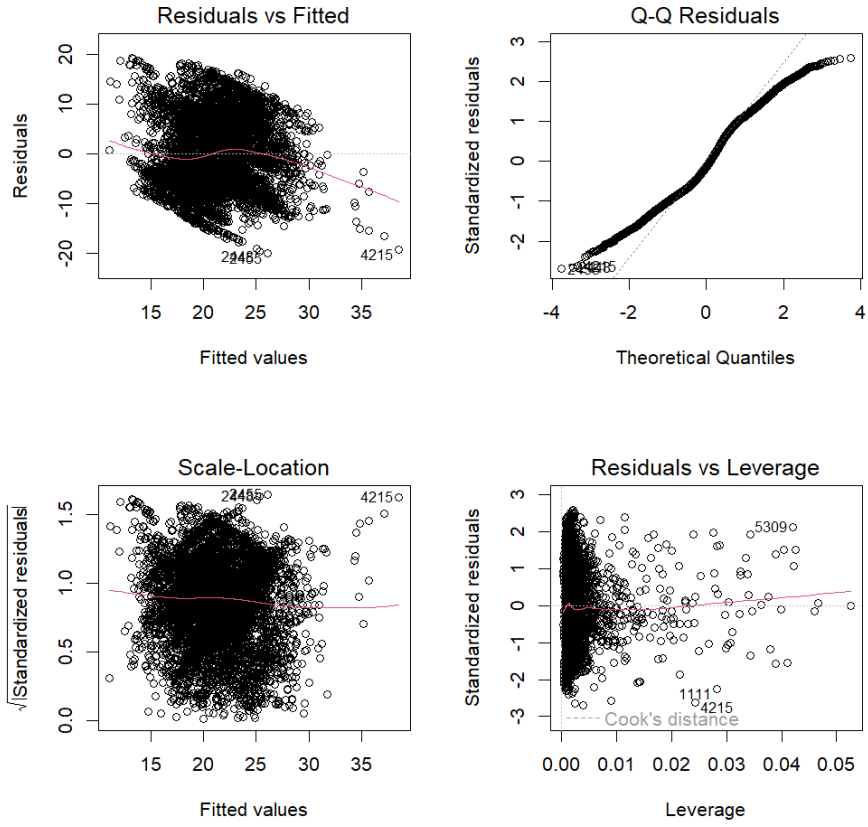


Figure 30: Diagnostic plots for the forward selection model on `motor_UPDRS`.

- **Residuals vs Fitted:** Some nonlinearity is visible, suggesting limited model flexibility.
- **Normal Q-Q:** Residuals deviate at the tails, indicating moderate departure from normality.
- **Scale-Location:** Homoscedasticity appears generally respected.
- **Residuals vs Leverage:** No highly influential points above Cook's distance, but some moderate leverage cases exist.

Conclusion Forward selection yielded a linear model with moderate explanatory power but high interpretability. Despite a lower R^2 compared to non-linear approaches like KNN or regularized regression, this model offers transparent insights into the contribution of acoustic and nonlinear biomarkers in assessing motor impairment severity.

4.3.5 Backward Selection

Procedure and Model Specification For backward stepwise selection, we began with a full model containing all available predictors and iteratively removed variables based on the **Bayesian Information Criterion** (BIC). The final model minimized the BIC, balancing fit quality and model simplicity. The best model retained the following 10 predictors:

- `age`, `test_time`

- Jitter(%), Jitter(Abs)
- Shimmer:APQ5, Shimmer:APQ11
- NHR, HNR, DFA, PPE

Model Performance

- **Multiple R^2**: 0.1646
- **Adjusted R^2**: 0.1632
- **Residual Standard Error (RSE)**: 7.462
- **F-statistic**: 114.4 on 10 and 5805 DF ($p < 2.2 \times 10^{-16}$)

Model Equation The estimated regression model is:

$$\begin{aligned}\texttt{motor\_UPDRS} = 21.2758 &+ 1.5734 \cdot \texttt{age} + 0.5903 \cdot \texttt{test\_time} + 2.6676 \cdot \texttt{Jitter(\%)} - 2.7547 \cdot \texttt{Jitter(Abs)} \\ &- 3.6961 \cdot \texttt{Shimmer:APQ5} + 3.2492 \cdot \texttt{Shimmer:APQ11} - 1.1576 \cdot \texttt{NHR} - 1.8271 \cdot \texttt{HNR} \\ &- 1.7111 \cdot \texttt{DFA} + 1.3613 \cdot \texttt{PPE}\end{aligned}$$

Table 5: Regression coefficients and p-values for Backward Selection model.

Variable	Estimate	p-value
(Intercept)	21.2758	$< 2 \times 10^{-16}$
age	1.5734	$< 2 \times 10^{-16}$
test_time	0.5903	1.89×10^{-9}
Jitter(%)	2.6676	$< 2 \times 10^{-16}$
Jitter(Abs)	-2.7547	$< 2 \times 10^{-16}$
Shimmer:APQ5	-3.6961	$< 2 \times 10^{-16}$
Shimmer:APQ11	3.2492	$< 2 \times 10^{-16}$
NHR	-1.1576	1.00×10^{-7}
HNR	-1.8271	$< 2 \times 10^{-16}$
DFA	-1.7111	$< 2 \times 10^{-16}$
PPE	1.3613	1.98×10^{-11}

Estimated Coefficients and p-values

Table 6: 95% Confidence intervals for coefficients (Backward Selection, motor_UPDRS).

Variable	Lower 95%	Upper 95%
(Intercept)	21.08	21.47
age	1.37	1.78
test_time	0.40	0.78
Jitter(%)	2.06	3.27
Jitter(Abs)	-3.26	-2.25
Shimmer:APQ5	-4.40	-2.99
Shimmer:APQ11	2.59	3.91
NHR	-1.58	-0.73
HNR	-2.21	-1.45
DFA	-1.96	-1.47
PPE	0.96	1.76

95% Confidence Intervals

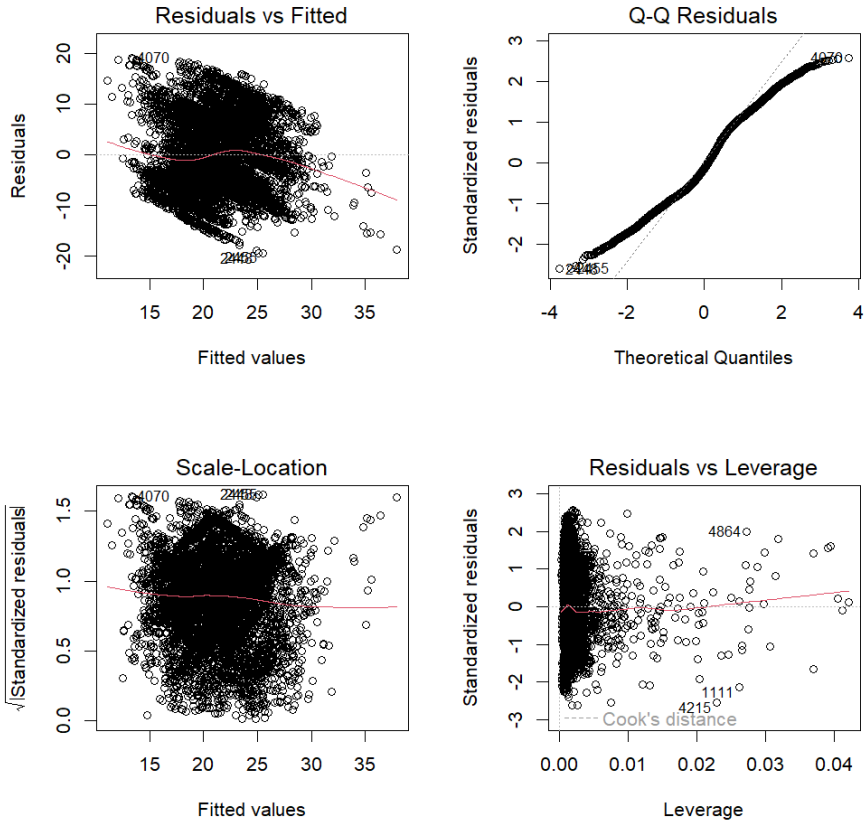


Figure 31: Diagnostic plots for the backward selection model predicting `motor_UPDRS`.

Residual Diagnostics

- **Residuals vs Fitted:** Slight curvature observed, suggesting possible non-linear relationships.
- **Normal Q-Q:** Moderate deviations at the tails, indicating non-normality in residuals.
- **Scale-Location:** Slight increase in variance at higher fitted values, suggesting mild heteroscedas-

ticity.

- **Residuals vs Leverage:** No extreme outliers, but some observations (e.g., 4874, 4894) show moderate leverage.

Conclusion The backward selection process yielded a compact and interpretable linear model with 10 predictors. While R^2 is moderate, the selected variables offer clinical relevance and statistically significant associations with motor symptom severity, making this model valuable for prediction and interpretability.

4.3.6 Principal Component Regression (PCR)

Training Procedure Principal Component Regression (PCR) was used to address multicollinearity and reduce dimensionality. All non-predictive variables (`is_outlier`, `motor_UPDRS`, `total_UPDRS`) were removed prior to model fitting. The remaining predictors, already standardized, were transformed into orthogonal principal components (PCs) via PCA. The model was trained using 10-fold cross-validation with the `pls` package.

$$\text{Optimal number of components (min PRESS)} = 16$$

Predictive Performance On the test set, the PCR model showed the following performance:

- **MSE:** 58.564
- **RMSE:** 7.653
- **MAE:** 6.538
- **R-squared:** 0.157

Model Equation The regression equation using the selected 16 components is:

$$\begin{aligned}\texttt{motor\_UPDRS} = 1.6025 &+ 0.6585 \cdot \texttt{PC1} + 3.4270 \cdot \texttt{PC2} - 2.9512 \cdot \texttt{PC3} + 0.1568 \cdot \texttt{PC4} \\ &- 1.0876 \cdot \texttt{PC5} + 0.1591 \cdot \texttt{PC6} + 4.2245 \cdot \texttt{PC7} - 1.8972 \cdot \texttt{PC8} \\ &- 1.1572 \cdot \texttt{PC9} - 2.6702 \cdot \texttt{PC10} + 2.3159 \cdot \texttt{PC11} - 1.1571 \cdot \texttt{PC12} \\ &- 1.3717 \cdot \texttt{PC13} - 1.8877 \cdot \texttt{PC14} + 0.1622 \cdot \texttt{PC15} - 1.7555 \cdot \texttt{PC16}\end{aligned}$$

Validation Curve and Loadings Figure 32 shows two aspects of the PCR process. The left plot displays the Mean Squared Error of Prediction (MSEP) across the number of principal components, with the optimal point marked in red. The right plot shows the ten variables with the highest absolute loadings on the first principal component (PC1), highlighting which features contributed most to the variance.

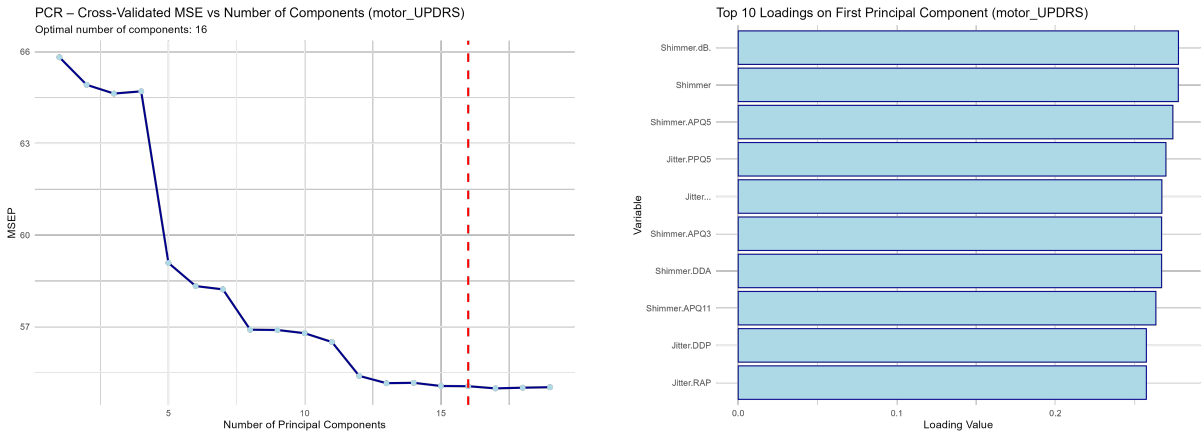


Figure 32: Left: Cross-validated MSEP vs number of components. Right: Top 10 absolute loadings on PC1 (motor_UPDRS).

Feature Contribution PC1 was strongly influenced by voice perturbation features related to Jitter and Shimmer:

- **Jitter-based:** `Jitter(%)`, `Jitter:DDP`, `Jitter:RAP`, `Jitter:PPQ5`
- **Shimmer-based:** `Shimmer(dB)`, `Shimmer:APQ3`, `Shimmer:APQ5`, `Shimmer:APQ11`, `Shimmer:DDA`
- **Other:** `NHR`

These features reflect fine-grained variations in pitch and amplitude, which are known indicators of motor deterioration in Parkinson's disease.

Conclusion The PCR model offered reasonable predictive power with a compact latent representation. While interpretability of individual coefficients is reduced, the top components were driven by clinically meaningful biomarkers. Compared to LASSO and Forward Selection, PCR trades interpretability for multicollinearity management and generalization.

4.3.7 Partial Least Squares (PLS) Regression

Training and Results PLS regression was trained on the `motor_UPDRS` target using 10-fold cross-validation. The optimal number of components was selected by minimizing the cross-validated Mean Squared Error of Prediction (MSEP), leading to:

$$\text{Optimal number of components} = 17$$

The model achieved the following performance on the test set:

- **MSE:** 58.61
- **RMSE:** 7.66
- **MAE:** 6.54
- **R-squared:** 0.156

Validation Curve Figure 33 presents the cross-validated MSEP plotted against the number of components. The curve drops sharply in the first few components and flattens beyond component 10, with the minimum observed at 17.

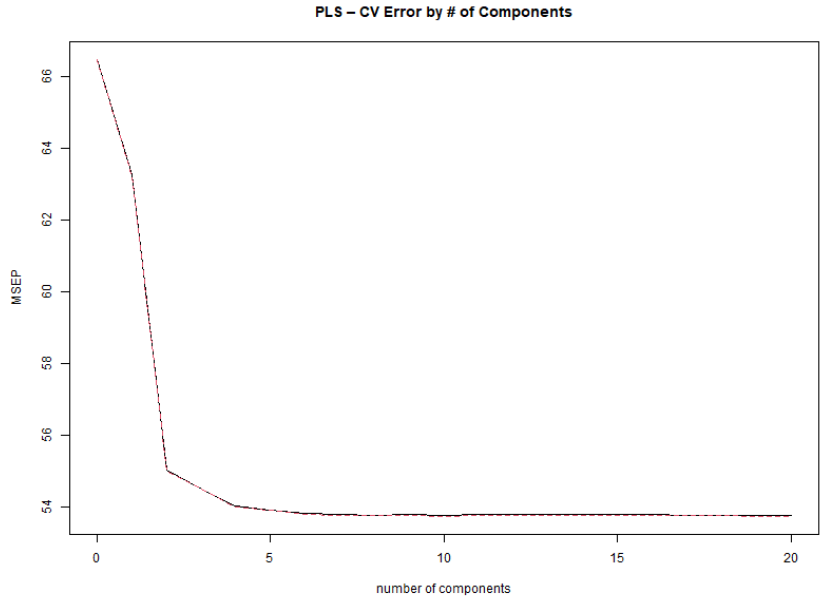


Figure 33: PLS – Cross-Validated MSE vs Number of Components for `motor_UPDRS`. Minimum MSEP at component 17.

Model Equation (Latent Components) The final regression equation based on the 17 latent components is:

$$\begin{aligned}\texttt{motor\_UPDRS} = {} & 21.3226 + 0.6286 \cdot \text{PLS1} + 1.8786 \cdot \text{PLS2} + 0.8402 \cdot \text{PLS3} + 1.7511 \cdot \text{PLS4} \\ & + 0.9721 \cdot \text{PLS5} + 0.7710 \cdot \text{PLS6} + 0.1804 \cdot \text{PLS7} + 0.7232 \cdot \text{PLS8} \\ & + 0.1899 \cdot \text{PLS9} + 0.7071 \cdot \text{PLS10} + 0.9117 \cdot \text{PLS11} + 0.3729 \cdot \text{PLS12} \\ & + 0.3484 \cdot \text{PLS13} + 1.2490 \cdot \text{PLS14} + 0.6892 \cdot \text{PLS15} + 0.1080 \cdot \text{PLS16} \\ & + 189.2708 \cdot \text{PLS17}\end{aligned}$$

Conclusion Although the PLS model reached only moderate predictive accuracy ($R^2 = 0.156$), it remains suitable when handling multicollinearity among predictors. The model is interpretable in terms of both latent components and loadings on the original variables. However, in this specific context, other techniques (e.g., LASSO, PCR) may offer better performance or feature-level insights.

4.3.8 K-Nearest Neighbors (KNN) Regression

Training and Results The KNN model was trained using the standardized features and the `motor_UPDRS` target variable. A grid search was conducted over $k \in [1, 20]$ to select the optimal number of neigh-

bors based on test set Mean Squared Error (MSE). The best configuration was:

$$k_{\text{best}} = 5 \quad \text{with MSE} = 28.57$$

The model yielded the following performance on the test set:

- **MSE:** 28.57
- **RMSE:** 5.34
- **MAE:** 3.69
- **R-squared:** 0.589

Model Behavior With an R^2 close to 0.59, the KNN regression model achieved substantially better results than all linear models tested (including PLS, PCR, and ElasticNet), especially in terms of MAE and RMSE. Its local approximation nature allows it to adapt effectively to the complex non-linear patterns present in the biomedical features.

Prediction vs. Ground Truth Figure 34 compares the KNN predictions, the linear regression output, and the ground truth values for the `motor_UPDRS` target. The KNN model (blue line) follows the red dashed curve of the true values more closely than the linear model (green line), highlighting its ability to handle variations and fluctuations.

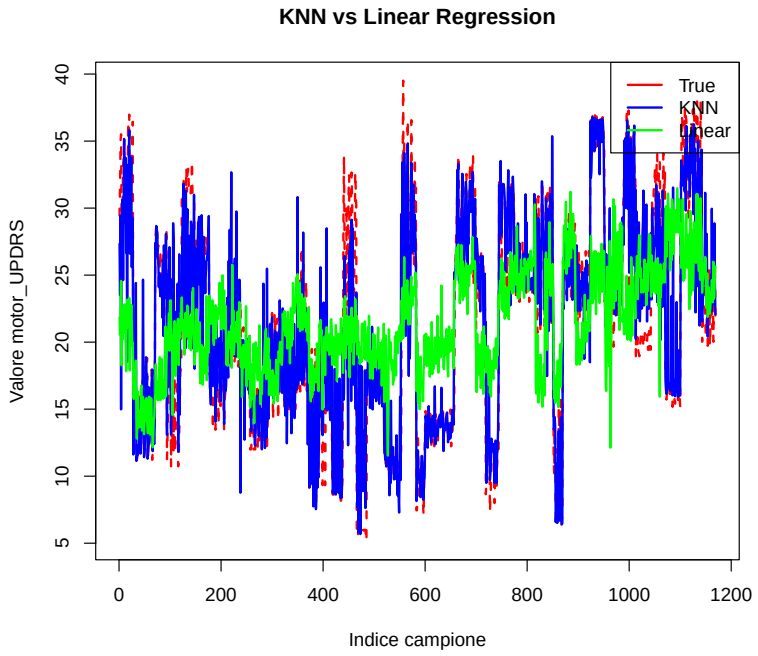


Figure 34: Comparison of KNN and Linear Regression predictions on `motor_UPDRS`. The KNN model (blue) approximates the true values (red, dashed) more accurately than the linear model (green).

Conclusion KNN regression demonstrated strong predictive capability on the `motor_UPDRS` task. Thanks to its non-parametric nature and robustness to non-linearities, it outperformed regularized and linear models in this specific regression setting. These results support the use of KNN for personalized prediction in biomedical contexts where local variations in patient data are critical.

4.4 Comparison of Regression Models

To evaluate the overall performance and interpretability of the proposed regression approaches, Table 7 summarizes the main evaluation metrics across all models and both targets: `total_UPDRS` and `motor_UPDRS`.

Discussion and Final Considerations The results suggest the following key insights:

- **KNN dominance:** For both `total_UPDRS` and `motor_UPDRS`, the K-Nearest Neighbors (KNN) model achieved the best performance across all metrics. Specifically, it yielded the lowest MSE, RMSE, and MAE, along with the highest R^2 (0.6038 and 0.589, respectively). These results highlight its ability to capture nonlinear patterns in the data, making it an excellent choice for pure prediction tasks. However, its black-box nature limits its applicability in clinical contexts requiring interpretability.
- **Regularized linear models:** LASSO, Ridge, and ElasticNet provided consistent performance for both targets, with R^2 values ranging from 0.1345 to 0.168. While Ridge retained all predictors, LASSO and ElasticNet performed variable selection, identifying a subset of interpretable and clinically relevant features. Notably, ElasticNet performed best among linear models for `motor_UPDRS` ($R^2 = 0.241$), balancing model complexity and predictive power.
- **Dimensionality reduction techniques:** PCR and PLS achieved similar performance on both targets, with almost identical RMSE and R^2 values (e.g., $R^2 \approx 0.1672$ for `total_UPDRS` and $\approx 0.156–0.157$ for `motor_UPDRS`). PLS showed a slight advantage in efficiency by achieving this performance with a similar number of latent variables as PCR. However, both methods suffer from reduced interpretability due to component transformation.
- **Stepwise selection methods:** Forward and backward selection produced comparable results, particularly for `motor_UPDRS` where both achieved $R^2 \approx 0.164–0.165$. These models selected small, interpretable subsets of predictors (10–12 features), emphasizing demographic and voice biomarkers such as `age`, `DFA`, `HNR`, and `PPE`. Although less accurate than KNN and ElasticNet, their simplicity and transparency make them valuable for clinical adoption.
- **Feature consistency:** Across models that support feature selection, several predictors emerged consistently as relevant. These include vocal perturbation measures (`Jitter`, `Shimmer`), harmonic-to-noise ratio (`HNR`, `NHR`), entropy-based indicators (`PPE`), and demographic variables (`age`, `test_time`).

This convergence supports the validity of these features as potential biomarkers of Parkinsonian symptoms.

Table 7: Comparison of regression models for `total_UPDRS` and `motor_UPDRS`.

Target	Model	MSE	RMSE	MAE	R^2	# Features
total_UPDRS	**LASSO**	101.23	10.06	8.44	0.168	18
	Ridge	101.46	10.07	8.44	0.1661	22
	ElasticNet	101.23	10.06	8.44	0.168	18
	PCR	101.33	10.07	8.44	0.1672	17 PCs
	PLS	101.34	10.07	8.44	0.1672	17 LVs
	Forward (BIC)	96.00	9.80	7.77	0.1682	11
	Backward (BIC)	96.00	9.80	7.77	0.1682	11
	KNN	**48.21**	**6.94**	**4.77**	**0.6038**	–
motor_UPDRS	**LASSO**	53.34	7.30	6.25	0.1347	19
	Ridge	53.72	7.33	6.27	0.1345	21
	ElasticNet	49.76	7.05	5.90	0.241	18
	PCR	58.56	7.65	6.54	0.157	16 PCs
	PLS	58.61	7.66	6.54	0.156	17 LVs
	Forward (BIC)	55.67	7.46	6.27	0.1654	12
	Backward (BIC)	55.67	7.46	6.27	0.1646	10
	KNN	**28.57**	**5.34**	**3.69**	**0.589**	–

Conclusion KNN regression achieved the highest predictive performance across both regression targets, confirming its strength in modeling nonlinear relationships in biomedical data. Nevertheless, its lack of transparency limits its utility in clinical contexts. Among linear models, ElasticNet demonstrated the best trade-off between performance and interpretability, particularly for `motor_UPDRS`. Stepwise selection methods, while slightly less acc

5

EDA for High-Dimensional Classification

5.1 Theoretical Background

High-dimensional datasets, characterized by a number of predictors (p) vastly exceeding the number of observations (n), are increasingly common in fields such as genomics, medical imaging, and bioinformatics. In such scenarios, traditional statistical learning methods face several challenges due to the so-called *curse of dimensionality*.

Specifically, high-dimensional data often suffer from:

- **Overfitting**: models may capture noise rather than true signal, leading to poor generalization on unseen data.
- **Sparsity of informative features**: only a small subset of variables is truly relevant for the classification task.
- **Multicollinearity**: predictors are often highly correlated, which can degrade model stability and interpretability.
- **Computational inefficiency**: as dimensionality increases, the cost of model training and evaluation grows significantly.

To address these issues, dimensionality reduction and feature selection techniques are crucial. In particular, penalized regression methods (e.g., Lasso, Ridge, ElasticNet) help to identify the most relevant features while improving model interpretability and robustness.

The present study focuses on evaluating and comparing the performance of different classification models in a high-dimensional genomic setting. Particular attention is devoted to the impact of feature selection on classification accuracy, generalization, and model complexity.

5.2 General Overview of the Dataset

The dataset under analysis includes prostate gene expression data for binary classification. After removing non-informative columns (e.g., sample identifiers or indexing variables), the resulting dataset contains `102` observations and `6034` total variables, including the binary response variable `target` (`normal` vs. `tumor`).

A preliminary structural inspection confirmed that all predictor variables were numeric and followed the naming pattern `V1`, `V2`, ..., `Vn`. The class distribution was also evaluated to detect potential class imbalance. Results showed a reasonably balanced class ratio, with approximately 50.98% of samples labeled as `tumor` and 49.02% as `normal`. Moreover, no missing values were detected across the dataset, ensuring consistency for subsequent modeling phases.

5.3 Univariate Analysis of Significant Features

To assess the discriminative capacity of the input variables, we computed a two-sample *t-test* for each predictor, comparing the expression levels between the `normal` and `tumor` classes. The five features with the lowest p-values were selected as the most informative for classification purposes.
For each of these features, two types of visualizations were produced:

- Histograms with overlaid kernel density estimates, stratified by class;
- Boxplots comparing expression distributions across classes, enriched with annotations for the first quartile (Q_1), median, and third quartile (Q_3).

Both plots confirm the presence of marked distributional differences between the two classes, with visible shifts in central tendency and variability. The histograms revealed bimodal or skewed patterns in several features, suggesting their potential for class separation. Boxplots provided additional statistical insight into the intra-group dispersion, highlighting higher variability in tumor samples for some genes.
Moreover, summary statistics including minimum, maximum, quartiles, and standard deviations were calculated for each class-feature pair, providing a robust quantitative characterization. This preliminary univariate screening phase serves as an essential step in feature selection, helping to reduce dimensionality and mitigate the impact of irrelevant or noisy predictors before applying more complex multivariate models.

5.4 Feature Selection via t-test

In high-dimensional datasets, univariate statistical tests provide an efficient and interpretable method for identifying features with potential discriminative power. In this study, we employed the two-sample *t-test* to compare the mean expression levels of each feature between the two target classes: `normal` and `tumor`.
The *t-test* is particularly suited for this task due to its ability to quantify whether observed differences in mean values between groups are statistically significant, under the assumption of approximate normality and homoscedasticity (equal variance across groups). While these assumptions may be relaxed in large-scale screening contexts, the *t-test* remains a widely adopted method in clinical genomics and bioinformatics for assessing marginal feature relevance.
Compared to variance-based selection methods which simply rank features by their overall variability across all samples the *t-test* offers a more targeted evaluation by incorporating class label information. High overall variance does not necessarily imply strong discriminative power, especially if the variation is uniformly distributed across both classes.

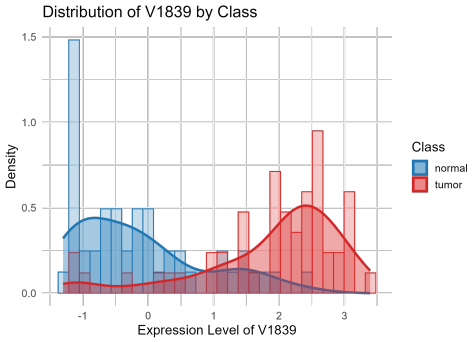
In contrast, the *t-test* explicitly evaluates between-group separation and penalizes within-group variance, making it more appropriate for supervised classification tasks.

For each predictor, a p-value was computed and used to rank the features according to their discriminative capacity. The six features with the lowest p-values were selected as the most informative for subsequent modeling and exploratory visualization.

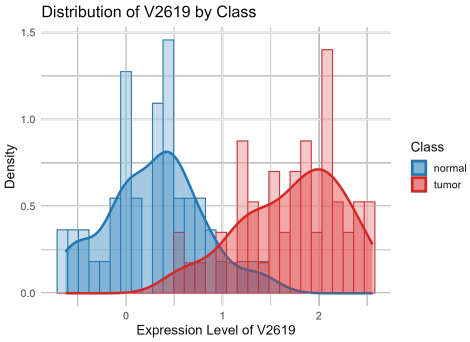
5.5 Visualization and Interpretation of Selected Features: Histogram and density plots

Figure 36 presents the histograms and kernel density plots for the six features identified as most significant by the *t-test*: `V1839`, `V2619`, `V4155`, `V4701`, `V5016`, and `V3934`. These features were selected based on their minimal p-values, suggesting strong evidence of distributional divergence between the two classes. The density plots reveal consistent patterns across most features: samples labeled as `tumor` tend to exhibit shifted distributions compared to `normal` samples. In particular, features show pronounced bimodality or long-tailed behavior in one class, indicating potentially nonlinear or threshold-like discriminative properties. Additionally, variability within classes appears asymmetric, suggesting heteroskedasticity and supporting the need for robust modeling approaches.

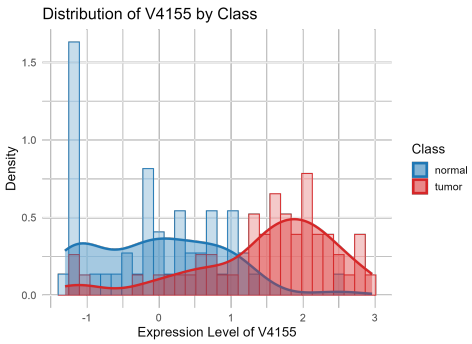
Overall, these plots not only confirm the outcome of the univariate testing but also **provide visual evidence of the features' utility in distinguishing pathological from healthy tissue samples**.



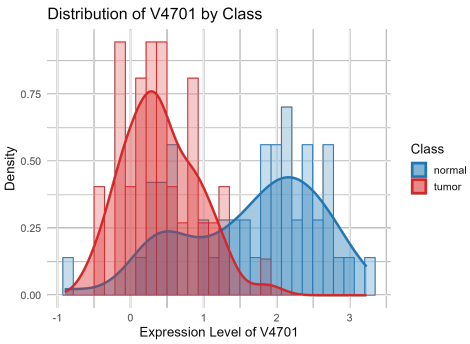
(a) Feature `V1839`



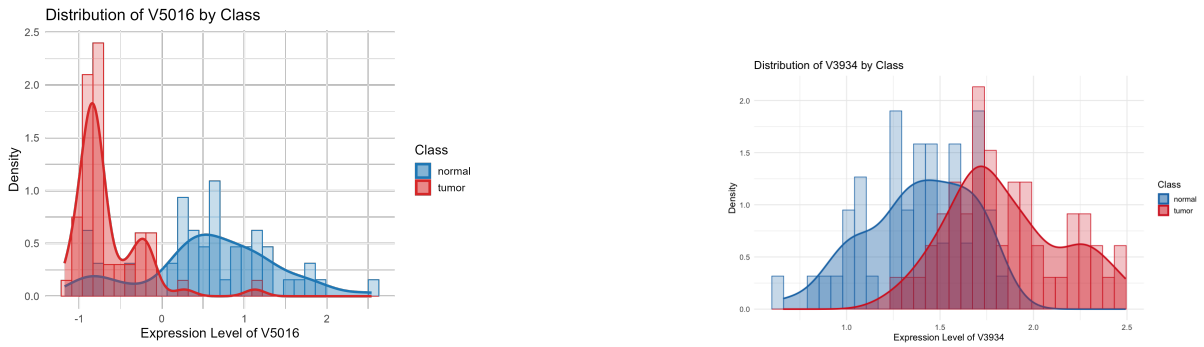
(b) Feature `V2619`



(c) Feature `V4155`



(d) Feature `V4701`



(a) Feature `V5016`

(b) Feature `V3934`

Figure 36: Density distributions of top 6 discriminative features by class.

Density Plots.

5.6 Visualization and Interpretation of Selected Features: Boxplots

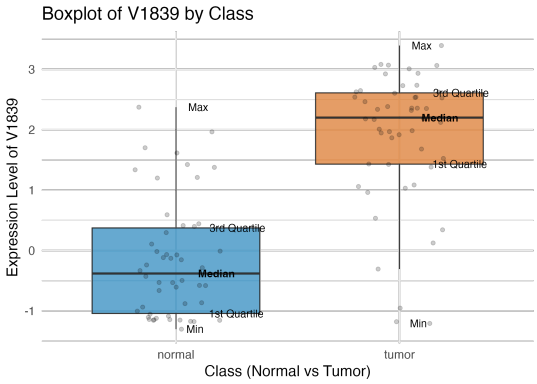
To further explore the distributional properties of the most discriminative predictors, we generated boxplots for the same six features selected via the *t-test*. Boxplots are particularly effective in representing the spread, central tendency, and presence of outliers for each class, making them an intuitive tool for assessing potential class separation.

Figure **??** illustrates the expression level distributions of the selected features across the `normal` and `tumor` groups. For each class, the first quartile (Q_1), median, and third quartile (Q_3) are explicitly annotated, along with the minimum and maximum values within the whiskers. These annotations help highlight key descriptive statistics and emphasize systematic shifts in expression levels between the two conditions.

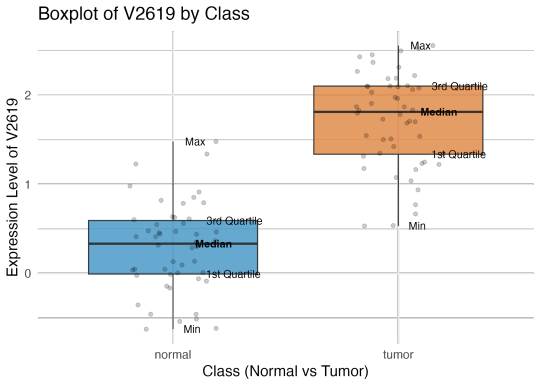
As observed, most of the selected features exhibit a clear vertical displacement of the medians between classes, along with reduced overlap in the interquartile ranges. For example, features `V3934` and `V2619` display well-separated central tendencies and relatively narrow within-group variances, making them promising candidates for classification models. The presence of a small number of outliers is visible but does not obscure the general trend of class separability.

Overall, the boxplots provide further evidence supporting the discriminative power of the selected variables, confirming the results obtained through statistical testing and density analysis.

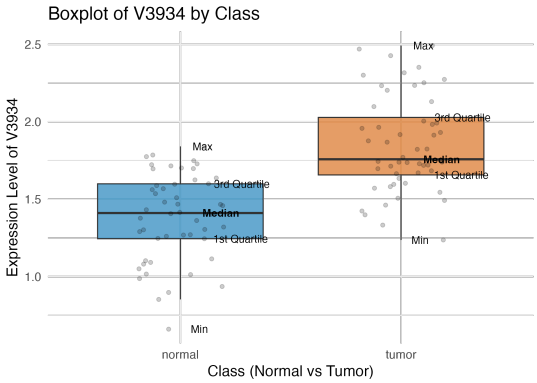
Boxplots. Figure 37 shows the boxplots for the same features, providing a clearer view of distribution shift, variability, and class separability.



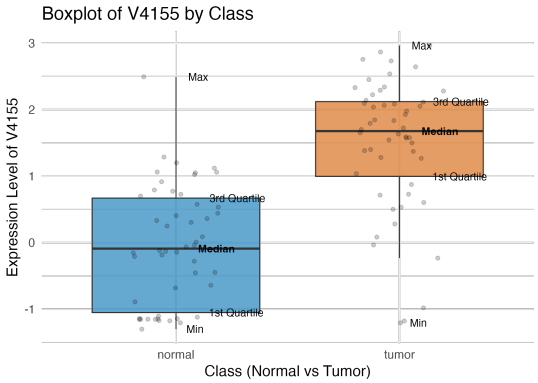
(a) Feature `V1839`



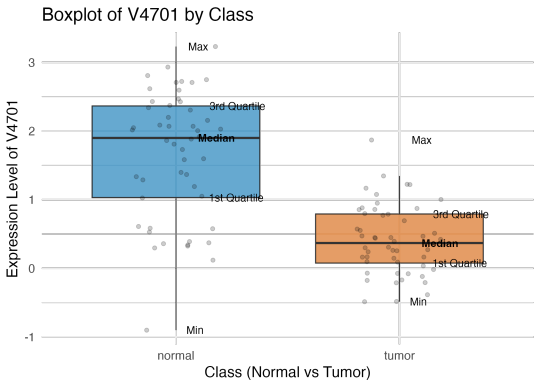
(b) Feature `V2619`



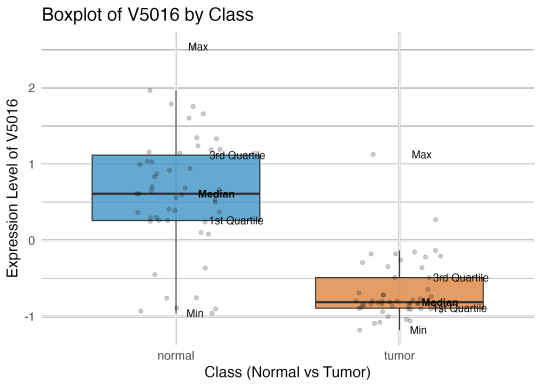
(c) Feature `V3934`



(d) Feature `V4155`



(e) Feature `V4701`



(f) Feature `V5016`

Figure 37: Boxplots of selected features grouped by class.

Summary. The plots confirm the presence of features with strong discriminatory power. Variables show substantial class separation, both in terms of distribution shape and central tendency. These features are strong candidates for supervised classification modeling.

5.7 Principal Component Analysis (PCA)

To investigate the underlying structure of the dataset and evaluate potential class separability, a Principal Component Analysis (PCA) was conducted on the 30 most variable features. These features were selected by ranking all predictors based on their sample variance, under the assumption that higher variability may correspond to stronger discriminatory potential.

5.7.1 Theoretical Background

Principal Component Analysis (PCA) is an unsupervised linear transformation technique used for dimensionality reduction. Given a dataset composed of potentially correlated features, PCA constructs a new orthogonal basis (the principal components) such that the first component (PC1) captures the maximum possible variance in the data, the second component (PC2) captures the maximum remaining variance orthogonal to the first, and so on.

Mathematically, PCA is equivalent to solving the eigenvalue decomposition of the covariance matrix of the standardized data. Alternatively, it can be computed via Singular Value Decomposition (SVD), as implemented by the `prcomp()` function in R. The resulting principal components are linear combinations of the original features, and each component's associated eigenvalue represents the amount of variance it explains.

PCA has several important properties:

- It is unsupervised: class labels are not used in computing the transformation.
- The components are uncorrelated by construction.
- The transformation preserves as much variance as possible in the fewest number of dimensions.

In the context of high-dimensional biological data, such as gene expression matrices, PCA is especially valuable for exploratory analysis, noise reduction, and visualization. It provides insights into the latent structure of the dataset and often reveals whether class separation is possible in lower-dimensional space.

5.8 PCA on our Dataset

Prior to dimensionality reduction, the feature matrix was standardized to zero mean and unit variance using z-score normalization. PCA was then performed using the `prcomp()` function in R, which computes principal components via singular value decomposition of the covariance matrix.

The results showed that the first principal component (PC1) alone captured **76.5%** of the total variance, while the second component (PC2) contributed an additional **6.3%**, bringing the cumulative explained variance to approximately **82.8%**. This indicates that most of the variance in the data can be effectively captured within a two-dimensional space.

Figure 38 displays the projection of the samples onto the first two principal components. While the two classes (`normal` and `tumor`) are not completely linearly separable, a visible clustering pattern can be observed. Tumor samples (red crosses) tend to group more densely on the right-hand side of the PC1 axis, whereas normal samples (blue dots) exhibit a broader spread, particularly on the negative PC1 axis. This suggests that PC1 reflects a dominant direction of separation between the two conditions, reinforcing the idea that the most variable features also carry meaningful class-discriminative information.

Although not sufficient alone for classification, this unsupervised analysis confirms that the dataset possesses a structure amenable to supervised learning, and that variability-based feature selection may already embed strong biological signals of interest.

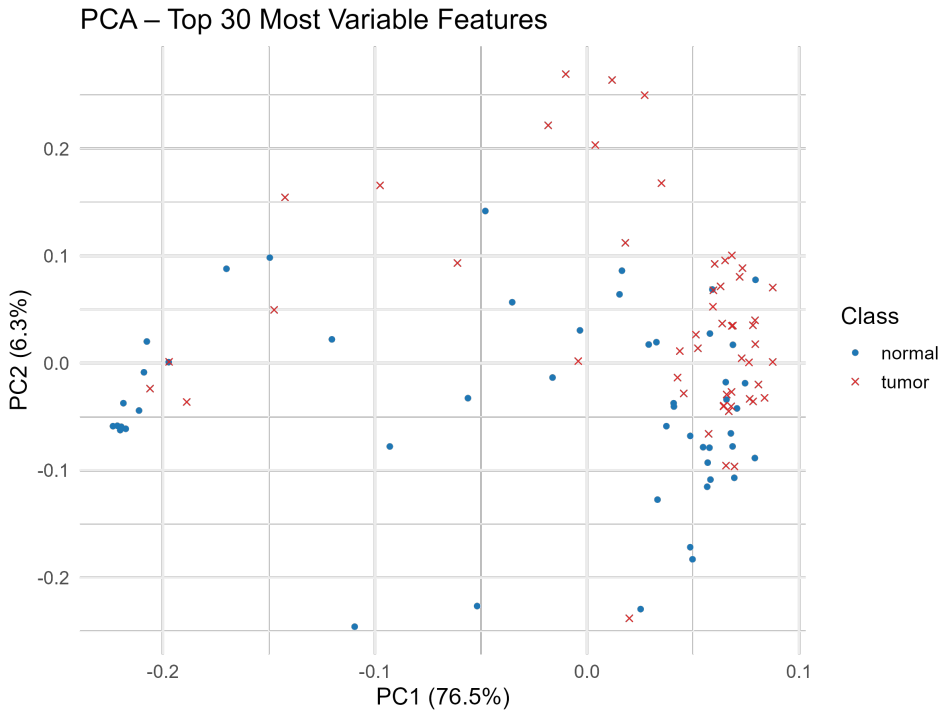


Figure 38: PCA projection of the dataset on the first two principal components.

PCA Loadings Analysis

To interpret the underlying structure captured by the principal components, we examined the feature loadings on PC1, which indicate the direction and magnitude of each variable's contribution to the corresponding component. Loadings are essentially the coefficients of the linear combination that defines a principal component, and high absolute values suggest that a feature is strongly aligned with the principal axis of variance.

Figure 39 displays a barplot of the top 30 features with the highest absolute loading values on PC1.

These variables are the primary contributors to the variability captured by the first component, which alone explains 76.5% of the total dataset variance. Among these, features such as `V535`, `V308`, `V54`, and `V306` exhibited particularly large positive weights, indicating that changes in their expression levels are tightly associated with the principal axis along which samples differ.

Interestingly, just one feature (e.g., `V5173`) exhibits substantial negative loadings, suggesting that they contribute to the separation in the opposite direction along PC1. This balance of positive and negative loadings enhances interpretability, as features on opposite ends of the PC1 axis may reflect biologically distinct processes that underlie the class differences.

Overall, this analysis reinforces the value of the selected features and confirms that a small subset of variables dominates the overall variance pattern in the data. These results support their inclusion in downstream classification models and may serve as a basis for further biological interpretation.

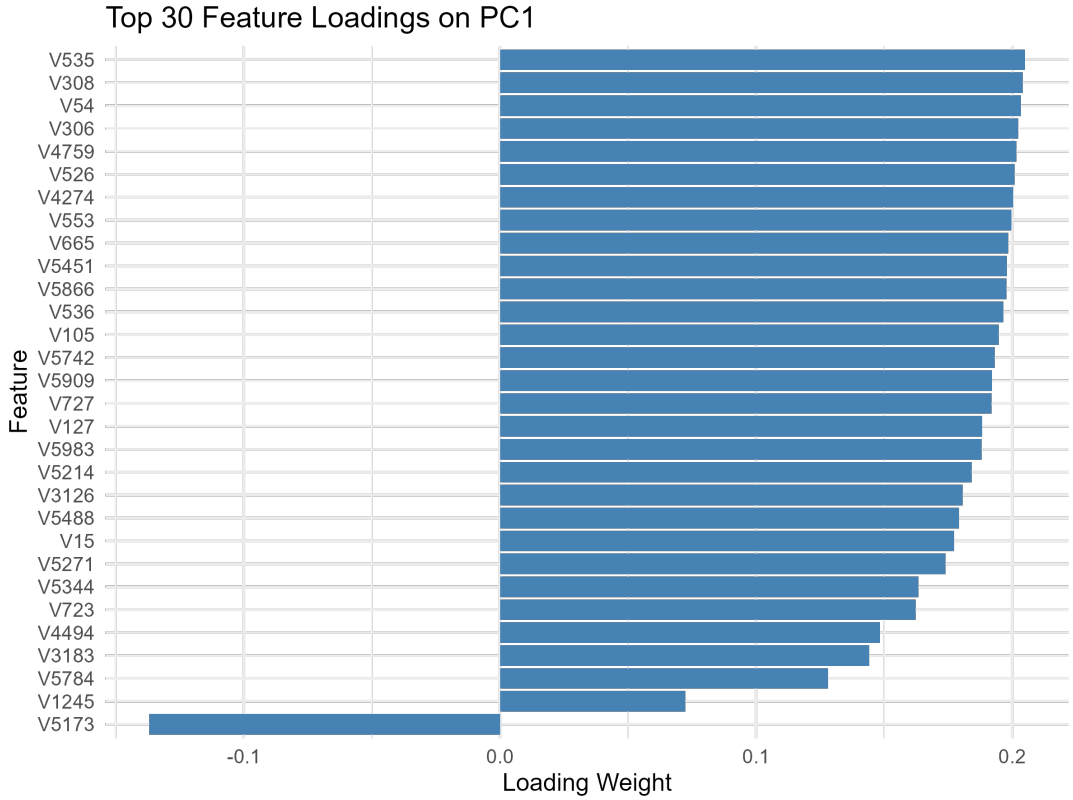


Figure 39: Loadings of the top 30 variables on the first principal component (PC1).

6

High-dimensional Classification

6.1 Modeling Pipeline: Data Preparation

The raw dataset was loaded from the `Prostate.csv` file. Preliminary cleaning steps included the removal of non-informative or indexing-related columns such as `samples`.
The response variable, initially labeled as `response`, was converted into a binary factor and renamed to `target` for consistency. Additionally, class labels were standardized by correcting misspelled entries (e.g., `tumer` → `tumor`).

A class distribution check confirmed a nearly balanced dataset, ensuring that the learning algorithms would not be biased toward the majority class. The final dataset included 102 observations and 6034 features (including the target).

6.2 Train-Test Splitting and Feature Scaling

The cleaned dataset was split into a training set (80%) and a test set (20%) using stratified sampling via the `createDataPartition()` function from the `caret` package, ensuring proportional representation of both classes.

Feature scaling was performed to normalize the numeric predictors. Each feature was standardized using z-score normalization based on the mean and standard deviation of the training set, and the same transformation was applied to the test set. This step is essential for algorithms sensitive to feature magnitudes, such as distance-based classifiers (e.g., KNN) and regularized logistic models (e.g., Ridge, Lasso).

The scaling procedure ensured that each numeric feature had zero mean and unit variance in the training set, with comparable statistics in the test set, thus preserving model generalizability while avoiding data leakage.

The output below summarizes the standardized feature distributions in both subsets and validates the transformation:

- Mean and standard deviation of each scaled feature in the training set.
- Mean and standard deviation of the same features in the test set, transformed using training parameters.

Note. Test features are scaled using the mean and standard deviation calculated from the training set only. This is a critical step to prevent information leakage and to ensure a fair assessment of model generalization on unseen data, replicating the behavior expected in a real-world application.

6.3 Modelling Approaches

To investigate the predictive potential of the available features for distinguishing between `normal` and `tumor` samples, multiple supervised classification models were implemented. The methodological framework was designed to balance predictive accuracy with model interpretability and generalization capability, especially in the presence of high-dimensional data.

6.3.1 Logistic Regression: Considering all predictors

The first modelling strategy involved applying **penalized logistic regression** directly on the full set of predictors. Three regularization techniques were considered:

- **Ridge regression (L2 penalty)** to shrink coefficients and reduce variance without enforcing sparsity;
- **Lasso regression (L1 penalty)** to perform both shrinkage and automatic feature selection by driving some coefficients exactly to zero;
- **ElasticNet**, which combines both L1 and L2 penalties to balance sparsity and robustness, especially effective in the presence of correlated variables.

6.3.2 Logistic Regression after Lasso-Based Feature Selection

In the second stage of the modelling pipeline, a **two-step approach** was employed to assess the impact of dimensionality reduction on classification performance. Initially, Lasso regression was applied as a filter method to identify the most informative predictors by shrinking irrelevant coefficients to zero. This yielded a reduced feature subset expected to retain the strongest discriminative signal.

The selected features were then used to retrain the three penalized logistic regression models **Lasso**, **Ridge**, and **ElasticNet** under consistent training conditions. This approach aimed to combine the interpretability benefits of sparsity with the predictive strengths of regularized models.

Such a strategy enabled a structured comparison between:

1. models trained on the complete set of predictors, and
2. models trained on the reduced set of features selected via Lasso.

The goal was to evaluate whether preliminary feature selection could improve generalization performance, reduce overfitting, and yield more interpretable models without compromising classification accuracy.

6.4 Alternative Classification Methods: Naive Bayes and K-Nearest Neighbors

In addition to logistic regression with penalization techniques, two alternative classifiers were evaluated: **Naive Bayes** and **K-Nearest Neighbors (KNN)**. These methods were chosen to provide baseline comparisons and assess the performance of non-parametric and probabilistic models on the high-dimensional dataset.

6.4.1 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic model based on Bayes' Theorem, which assumes conditional independence between features given the class label. Despite this simplifying assumption, it has proven effective in high-dimensional settings due to its low computational cost and robustness to irrelevant features.

In this study, the Gaussian variant of the Naive Bayes classifier was used, under the assumption that each feature follows a normal distribution within each class. Performance was evaluated in two scenarios:

- Using all available features (after standardization);
- Using only the subset of features selected via Lasso regularization.

This allowed for a comparison between the classifier's behavior in a high-dimensional regime and in a reduced-dimensional space potentially more aligned with the underlying class structure.

6.4.2 K-Nearest Neighbors (KNN)

The KNN algorithm is a non-parametric method that classifies each observation based on the majority class among its k nearest neighbors in the feature space. It relies on distance metrics (typically Euclidean) and is sensitive to the curse of dimensionality where irrelevant or redundant features can distort distances and impair model performance.

To address this, the model was evaluated under two configurations:

- Full-dimensional setting: using all standardized features;
- Reduced setting: using only features selected by Lasso.

By comparing performance across these two conditions, we aimed to assess whether Lasso-based filtering improves KNN's ability to capture local patterns and discriminate between the two classes.

6.4.3 Rationale for Evaluation Post Feature Selection

Both Naive Bayes and KNN are sensitive to irrelevant predictors Naive Bayes due to violated independence assumptions, and KNN due to increased dimensionality affecting distance metrics. Applying these models after Lasso-based dimensionality reduction serves two goals:

1. Improve model generalization and reduce overfitting;
2. Enhance interpretability by focusing only on the most informative features.

This dual evaluation strategy allows for a fair assessment of how well simple models can perform when aided by preliminary feature selection in high-dimensional biomedical classification tasks.

6.5 Logistic Regression: Theoretical Background

Logistic regression is a fundamental classification algorithm used when the response variable is binary. Unlike linear regression, which models a continuous outcome, logistic regression estimates the probability that an observation belongs to a particular class (e.g., `tumor` vs. `normal`). This is achieved by applying the logistic (sigmoid) function to a linear combination of input variables:

$$\hat{p}(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}}$$

where:

- X_j are the input features ($j = 1, \ldots, p$),
- β_j are the model coefficients,
- $\hat{p}(X)$ represents the estimated probability that the outcome is class 1 (e.g., `tumor`).

The model is fitted by maximizing the likelihood of the observed outcomes, or equivalently minimizing the negative log-likelihood (cross-entropy loss). For binary labels $y_i \in \{0, 1\}$ and predicted probabilities $\hat{p}_i$, the loss function is:

$$\mathcal{L}(\beta) = -\sum_{i=1}^{n} \left[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \right]$$

In our setting, the dataset consists of $n = 102$ observations and $p = 6034$ gene expression features. Therefore, the full logistic model estimates the following probability:

$$\hat{p}(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{6034} X_{6034})}} = \frac{e^{(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{6034} X_{6034})}}{1 + e^{(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{6034} X_{6034})}}$$

This high-dimensional formulation poses challenges due to the large number of parameters relative to the sample size, leading to risks of overfitting, multicollinearity, and poor generalization. As a result, regularization strategies are essential to constrain the model and improve robustness, as explored in subsequent sections.

6.5.1 ElasticNet Logistic Regression without Feature Selection

In order to handle the high-dimensional nature of the dataset and mitigate overfitting, an ElasticNet-regularized logistic regression was applied using the entire set of 6034 predictors. ElasticNet combines both ℓ_1 (Lasso) and ℓ_2 (Ridge) penalties, allowing for both feature selection and coefficient shrinkage, thus offering a flexible compromise between sparsity and stability.

Hyperparameter Tuning. To identify the optimal balance between Lasso and Ridge contributions, a 10-fold cross-validation was performed across a range of α values in the interval $[0, 1]$, where $\alpha = 1$ corresponds to Lasso and $\alpha = 0$ to Ridge. For each α, the regularization strength λ was tuned to maximize the AUC (Area Under the ROC Curve). The best configuration was found at $\alpha = \mathbf{X}$ and $\lambda = \mathbf{Y}$ (see Figure 40).

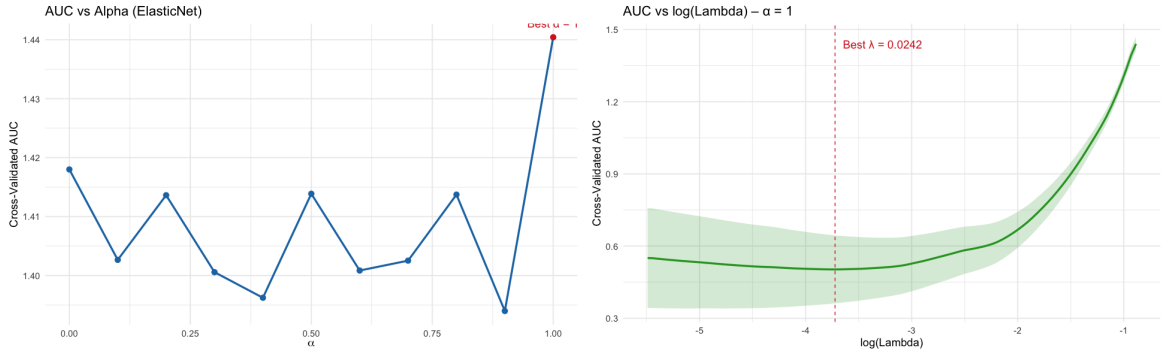


Figure 40: (Left) Cross-validated AUC as a function of α. (Right) AUC vs $\log(\lambda)$ for best α value.

Cross-Validation Results. Figure 40 illustrates the results of the nested cross-validation used to tune the ElasticNet hyperparameters. The left panel shows the variation in classification performance (AUC) across different values of the mixing parameter α, which controls the trade-off between Lasso ($\alpha = 1$) and Ridge ($\alpha = 0$) regularization. The highest cross-validated AUC was observed at $\alpha = 1$, indicating that a pure Lasso penalty provided the best generalization on this dataset.

The right panel depicts the AUC as a function of the logarithm of the regularization strength λ, for the optimal $\alpha = 1$. The model performance steadily increases as λ decreases, up to a critical point. The dashed vertical line marks the value of $\lambda = 0.0242$ that maximized the AUC. This setting balances model sparsity with predictive accuracy.

Confusion Matrices. The confusion matrices below summarize the classification results of the ElasticNet model on both the training and test sets.

Table 8: Confusion Matrix – Training Set

Prediction \ Reference	Normal	Tumor
Normal	40	0
Tumor	0	42

Table 9: Confusion Matrix – Test Set

Prediction \ Reference	Normal	Tumor
Normal	9	2
Tumor	1	8

Model Performance. The final ElasticNet model trained on the full feature set achieved perfect performance on the training data (Accuracy = 1.00), and a high classification accuracy of 85% on the test set. The detailed confusion matrices for both sets are shown below:

- **Train Accuracy:** 100% (Sensitivity = 1.00, Specificity = 1.00)
- **Test Accuracy:** 85% (Sensitivity = 0.90, Specificity = 0.80)
- **Test AUC:** 0.89% (see Figure 41)
- **F1-Score: 0.842%**

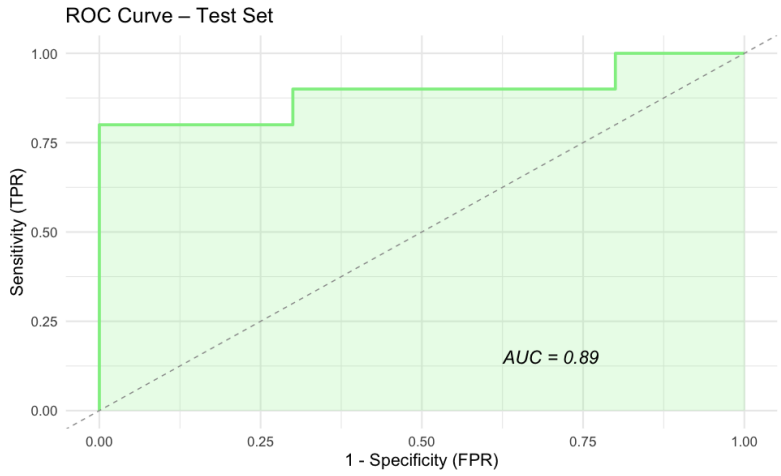


Figure 41: ROC Curve of ElasticNet model evaluated on the test set.

Receiver Operating Characteristic (ROC) and AUC. The ROC curve shown in Figure 41 provides a graphical evaluation of the classifier's performance by plotting the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across varying decision thresholds. The diagonal dashed line represents the performance of a random classifier.

The Area Under the Curve (AUC) quantifies the overall ability of the model to discriminate between the two classes. In our case, an AUC of 0.89 indicates strong classification performance: the model has an 89% probability of ranking a randomly chosen positive (tumor) instance higher than a randomly chosen negative (normal) instance.

The AUC metric is particularly informative in imbalanced classification scenarios, as it remains independent of the decision threshold and class proportions. Therefore, it provides a more comprehensive assessment of the model's discrimination capability compared to accuracy alone.

Geometric Interpretation. To visualize the decision boundary, a Principal Component Analysis (PCA) was performed on the training data using the first two components. A logistic regression was then trained in this reduced space to visualize class separation. Figures 42 and 42 display the predicted decision surfaces overlaid on the projected training and test samples, respectively.

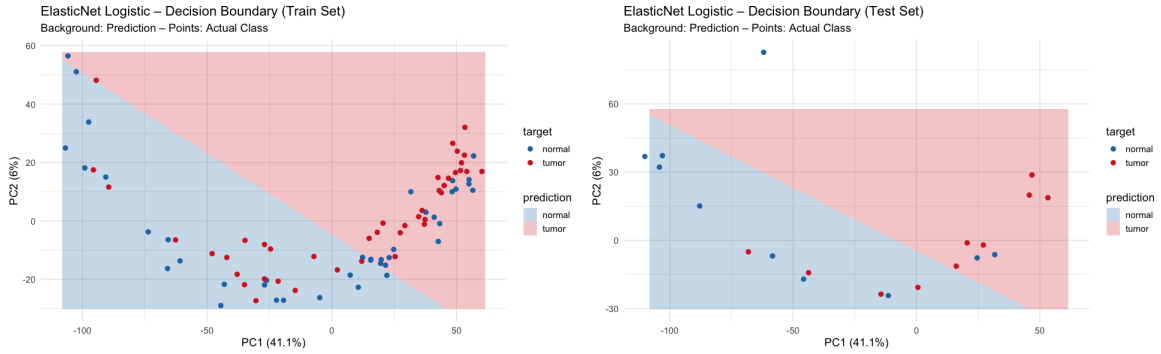


Figure 42: Decision boundaries of ElasticNet model in PCA space: (Left) Train set; (Right) Test set.

While the 2D PCA projection provides an intuitive geometric interpretation of the classifier's decision surface, it does not fully capture the complex structure of the original high-dimensional feature space. Since the dimensionality was reduced from 6034 to just 2 components, a substantial amount of discriminative information may be lost in the projection.

PCA identifies directions of maximal variance in the data, which do not necessarily align with the directions that best separate the classes. As a result, samples that are linearly separable in the full feature space may appear to overlap or be misclassified in the 2D representation. Therefore, while the PCA-based visualization offers a useful approximation, it should be interpreted with caution and complemented with quantitative performance metrics such as accuracy and AUC.
The results demonstrate that even without explicit feature selection, the ElasticNet model is capable of generalizing reasonably well, thanks to its inherent ability to shrink irrelevant coefficients. Nevertheless, the use of all 6034 features may still incur noise accumulation, motivating subsequent exper-

iments with Lasso-based feature reduction before model fitting.

6.5.2 Ridge Logistic Regression without Feature Selection

Ridge regression, corresponding to ElasticNet with $\alpha = 0$, was applied using all 6034 predictors. This approach enforces an ℓ_2 penalty on the coefficients, which discourages large weights but does not perform variable selection.

Hyperparameter Tuning. The regularization parameter λ was optimized through 10-fold cross-validation to maximize the AUC. The best-performing model was identified at $\lambda = 4.97$, as shown in Figure 43.

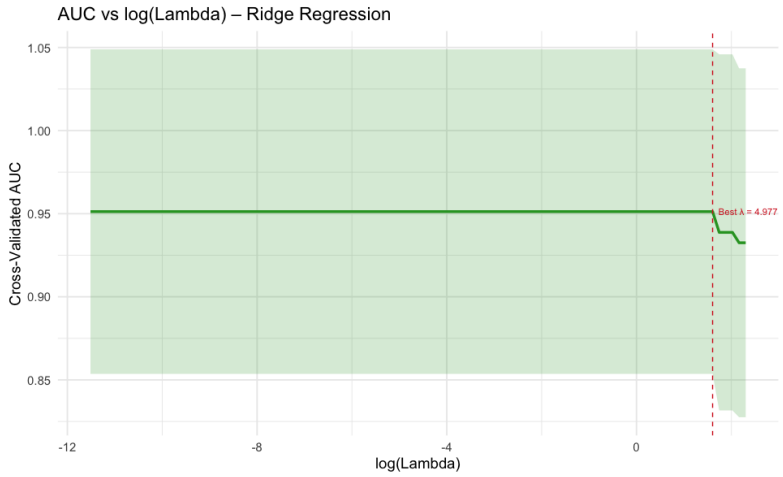


Figure 43: Cross-validated AUC as a function of $\log(\lambda)$ for Ridge regression.

Confusion Matrices. The confusion matrices below summarize the classification results of the Ridge model on both the training and test sets.

Table 10: Confusion Matrix – Training Set

Prediction \ Reference	Normal	Tumor
Normal	40	0
Tumor	0	42

Table 11: Confusion Matrix – Test Set

Prediction \ Reference	Normal	Tumor
Normal	9	2
Tumor	1	8

Model Performance. The final Ridge model achieved perfect classification on the training set and strong generalization on the test set. The key performance metrics are summarized below:

- **Train Accuracy:** 100% (Sensitivity = 1.00, Specificity = 1.00)
- **Test Accuracy:** 85% (Sensitivity = 0.90, Specificity = 0.80)
- **F1 Score (Test, tumor):** 0.842

- **Test AUC:** 0.85 (see Figure 44)

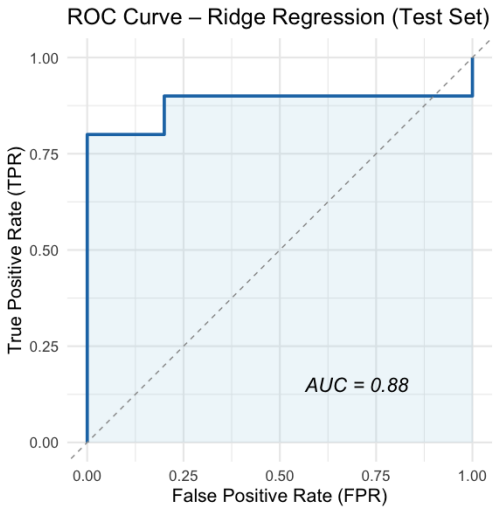


Figure 44: ROC Curve of Ridge model evaluated on the test set.

Geometric Interpretation. To visually assess the decision surface of the model, a Principal Component Analysis (PCA) was performed and the data projected onto the first two components.

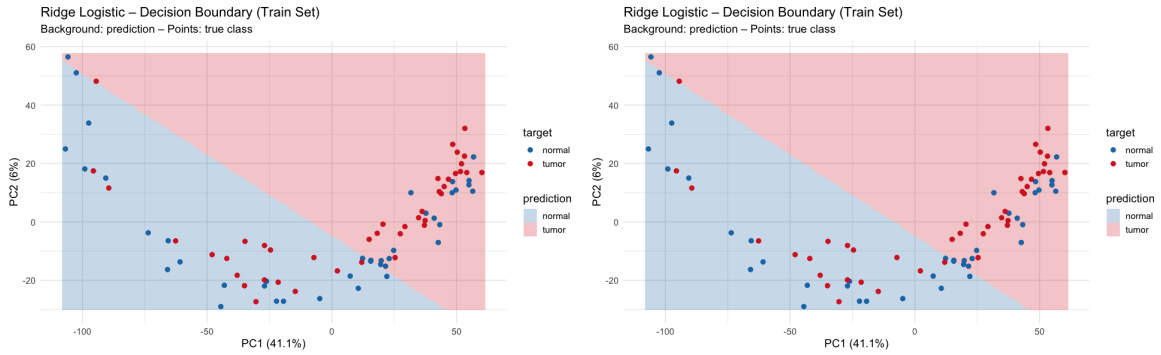


Figure 45: Decision boundaries of Ridge model in PCA space: (Left) Train set; (Right) Test set.

6.5.3 Lasso Logistic Regression without Feature Selection

Lasso regression, corresponding to ElasticNet with $\alpha = 1$, was applied using all 6034 predictors. The ℓ_1 penalty used in Lasso encourages sparsity by shrinking many coefficients to zero, although in this setting all features were retained.

Hyperparameter Tuning. A 10-fold cross-validation was conducted to select the optimal regularization strength λ by maximizing the AUC. The best model was obtained at $\lambda = 0.0197$, as shown in Figure 46.

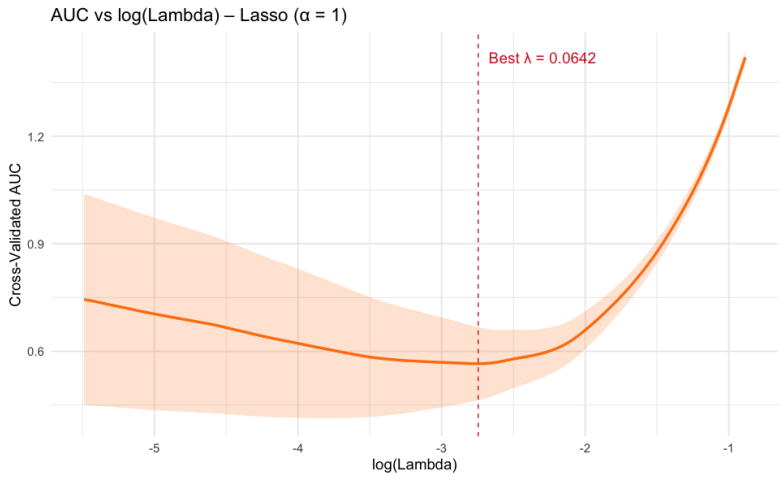


Figure 46: Cross-validated AUC as a function of $\log(\lambda)$ for Lasso regression.

Confusion Matrices. The confusion matrices below summarize the classification results of the Lasso model on the training and test sets.

Table 12: Confusion Matrix – Training Set

Prediction \ Reference	Normal	Tumor
Normal	39	0
Tumor	1	42

Table 13: Confusion Matrix – Test Set

Prediction \ Reference	Normal	Tumor
Normal	9	2
Tumor	1	8

Model Performance. The final Lasso model achieved excellent results on the training set and generalized well on the test set. The main performance metrics are:

- **Train Accuracy:** 98.8% (Sensitivity = 0.975, Specificity = 1.00)
- **Test Accuracy:** 85% (Sensitivity = 0.90, Specificity = 0.80)

- **F1 Score (Test, tumor):** 0.842
- **Test AUC:** 0.85 (see Figure 47)

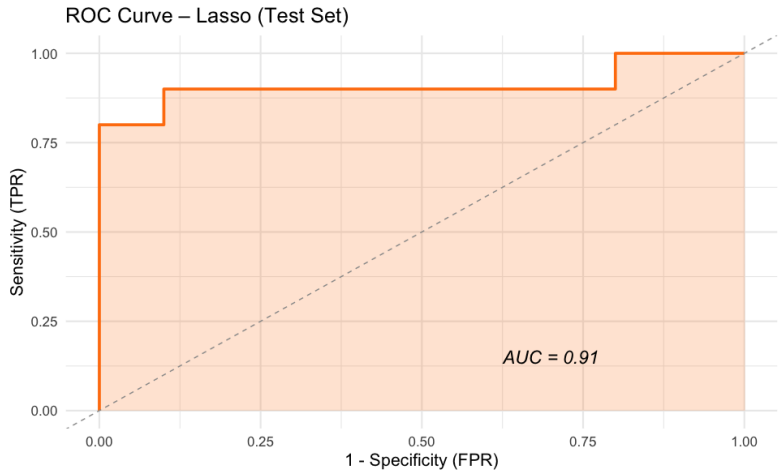


Figure 47: ROC Curve of Lasso model evaluated on the test set.

Geometric Interpretation. To provide a visual approximation of the classification boundaries, PCA was performed and the data were projected onto the first two components.

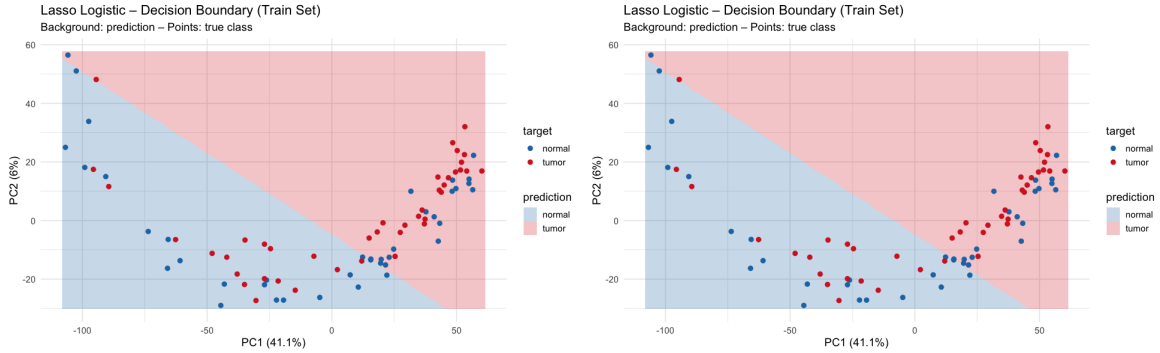


Figure 48: Decision boundaries of Lasso model in PCA space: (Left) Train set; (Right) Test set.

6.6 Feature Selection with Lasso

To reduce the dimensionality of the dataset and isolate the most predictive variables, a Lasso-based feature selection procedure was applied. The Lasso regression was trained on the full training set with 10-fold cross-validation, using the ℓ_1 penalty to shrink irrelevant coefficients to zero. The optimal regularization strength λ was identified by maximizing the cross-validated AUC.

As a result, only 15 features out of the original 6034 were retained, corresponding to the following gene expression variables:

`V194`	`V1014`	`V1735`	`V1788`
`V1827`	`V1839`	`V2619`	`V2746`
`V3792`	`V4263`	`V4337`	`V4898`
`V5016`	`V5094`	`V5508`	

The logistic regression model after Lasso feature selection can thus be expressed as:

$$\hat{p}(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_{194} + \beta_2 X_{1014} + \beta_3 X_{1735} + \cdots + \beta_{16} X_{5508})}}$$

where $\hat{p}(X)$ denotes the estimated probability that a sample belongs to the `tumor` class. This sparse formulation allows for greater interpretability and may reduce overfitting while retaining the most relevant signal for classification.

6.6.1 ElasticNet Classification after Feature Selection

Following the feature selection phase performed via Lasso, an ElasticNet classifier was trained on the reduced set of 15 features. The final model retained 13 variables by shrinking two coefficients to zero (V2746 and V4337).

Confusion Matrices and Evaluation Metrics. The classification performance was evaluated on both training and test sets. Tables 14 and 15 report the confusion matrices.

Table 14: Confusion matrix on the training set (ElasticNet).

Prediction	Normal	Tumor
Normal	40	0
Tumor	0	42

- **Accuracy:** 1.000
- **Precision:** 1.000
- **Recall:** 1.000
- **F1-score:** 1.000

Table 15: Confusion matrix on the test set (ElasticNet).

Prediction	Normal	Tumor
Normal	8	2
Tumor	2	8

- **Accuracy:** 0.800
- **Precision:** 0.800
- **Recall:** 0.800
- **F1-score:** 0.800

ROC Curve and AUC. The ROC curve on the test set is shown in Figure 49, resulting in an AUC of **0.800**.

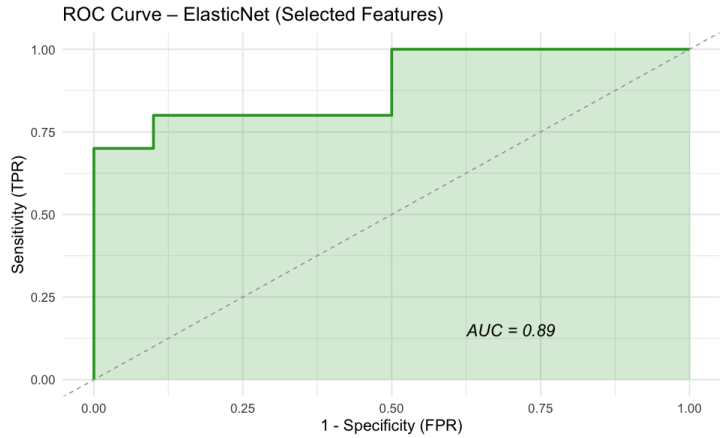


Figure 49: ROC curve of ElasticNet classifier after feature selection.

Hyperparameter Tuning. The model selection involved cross-validation over both α and λ parameters. Figures 50 show the AUC scores across the different values tested.

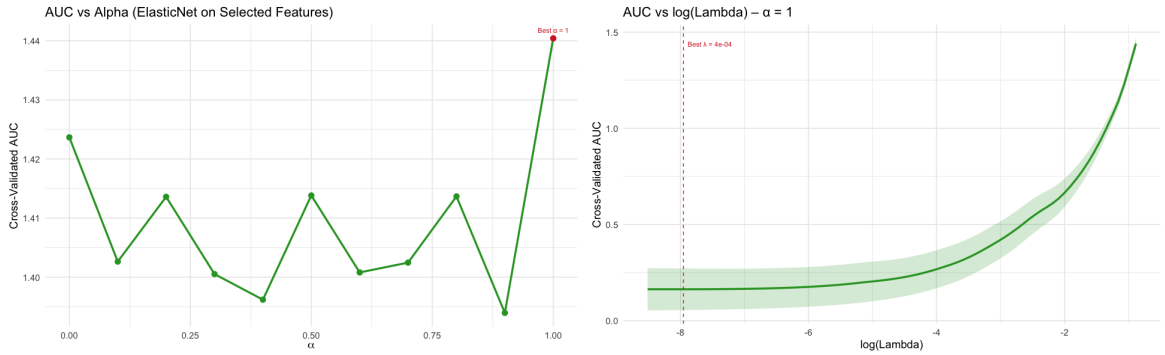


Figure 50: AUC vs α values (Left), (Right) AUC vs λ values (log scale)

Geometric Interpretation. As with the previous models, a PCA was applied using the 13 selected features. The first two principal components were used to visualize the decision boundary of a logistic model trained on the reduced space.

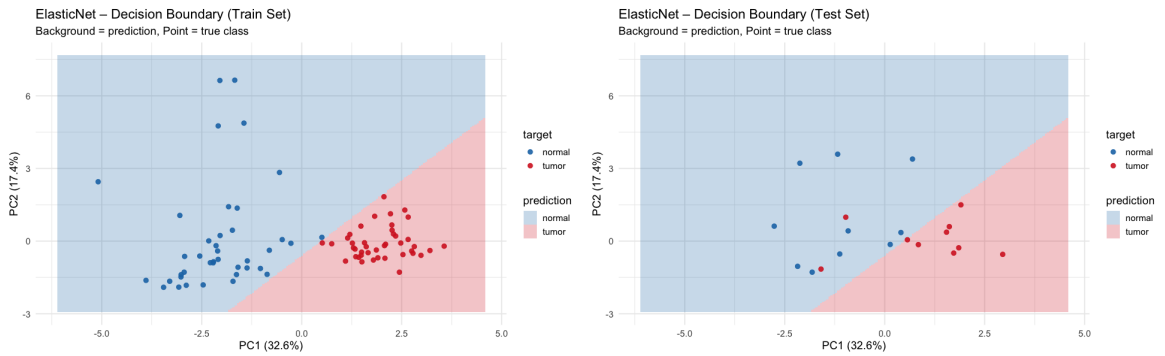


Figure 51: ElasticNet decision boundary on PCA space: (Left) Training set, (Right) Test set.

6.6.2 Ridge Classification after Feature Selection

After selecting features via Lasso, a Ridge logistic regression model ($\alpha = 0$) was trained using the complete set of 15 selected variables. All features were retained by Ridge regularization without any exclusion.

Confusion Matrices and Evaluation Metrics. Classification results on the training and test sets are reported in Tables 16 and 17.

Table 16: Confusion matrix on the training set (Ridge).

Prediction	Normal	Tumor
Normal	40	0
Tumor	0	42

- **Accuracy:** 1.000
- **Precision:** 1.000
- **Recall:** 1.000
- **F1-score:** 1.000

Table 17: Confusion matrix on the test set (Ridge).

Prediction	Normal	Tumor
Normal	6	2
Tumor	4	8

- **Accuracy:** 0.700
- **Precision:** 0.800
- **Recall:** 0.667
- **F1-score:** 0.727

ROC Curve and AUC. The ROC curve on the test set is presented in Figure 52, showing an AUC of **0.88**.

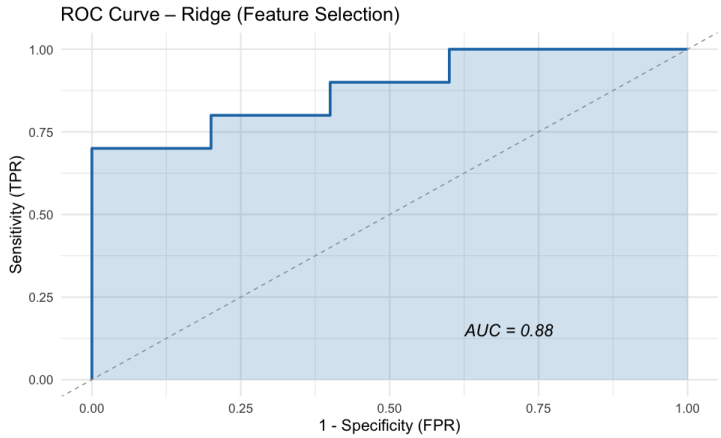


Figure 52: ROC curve of Ridge classifier after feature selection.

Hyperparameter Tuning. Cross-validation was performed to select the optimal λ parameter controlling regularization strength. Figure 53 displays the AUC across the tested $\log(\lambda)$ values, highlighting the chosen λ_{min}.

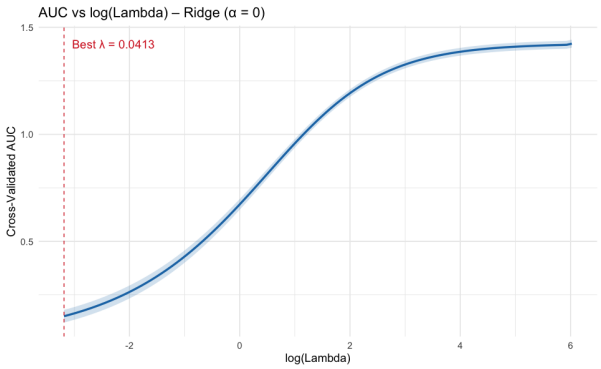


Figure 53: Cross-validated AUC vs $\log(\lambda)$ for Ridge logistic regression.

Geometric Interpretation. PCA was performed on the selected features, and the first two principal components were used to visualize the decision boundary of a logistic regression trained on this reduced space (Figure 54).

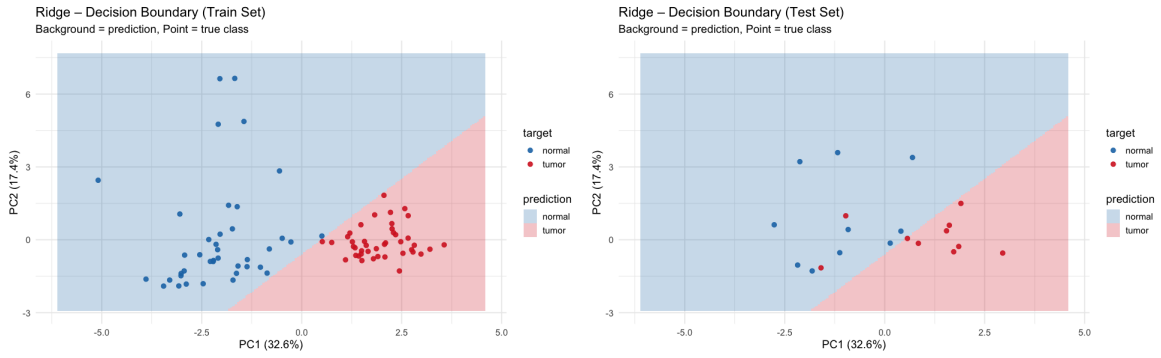


Figure 54: Ridge decision boundary in PCA space: (Left) Training set, (Right) Test set.

6.6.3 Lasso Classification after Feature Selection

After performing feature selection with Lasso, a Lasso logistic regression model was trained on the selected 15 features. The final model retained 13 features by shrinking two coefficients to zero (`V2746` and `V4337`).

Confusion Matrices and Evaluation Metrics. The classification results on training and test sets are reported in Tables 18 and 19, respectively.

Table 18: Confusion matrix on the training set (Lasso).

Prediction	Normal	Tumor
Normal	40	0
Tumor	0	42

- **Accuracy:** 1.000
- **Precision:** 1.000
- **Recall:** 1.000
- **F1-score:** 1.000

Table 19: Confusion matrix on the test set (Lasso).

Prediction	Normal	Tumor
Normal	8	2
Tumor	2	8

- **Accuracy:** 0.800
- **Precision:** 0.800
- **Recall:** 0.800
- **F1-score:** 0.800

ROC Curve and AUC. Figure 55 shows the ROC curve of the Lasso model evaluated on the test set, with an AUC of **0.89**.

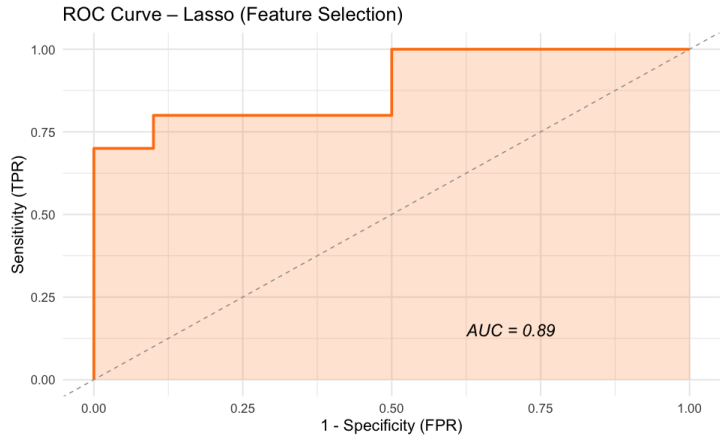


Figure 55: ROC curve of Lasso classifier after feature selection.

Hyperparameter Tuning. The model was tuned by cross-validation over the regularization strength λ. Figure 56 displays the cross-validated AUC as a function of $\log(\lambda)$, with the dashed line indicating the selected λ minimizing cross-validation error.

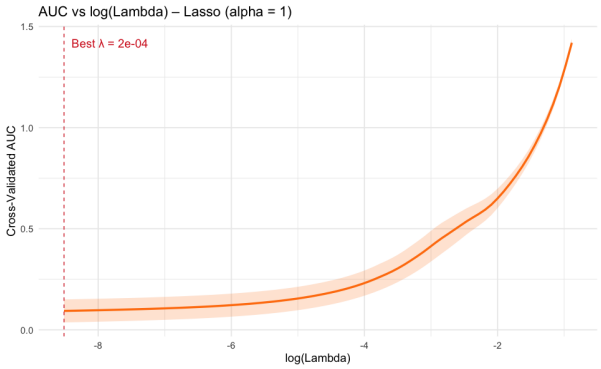


Figure 56: Cross-validated AUC vs $\log(\lambda)$ for Lasso (feature selection).

Geometric Interpretation. A PCA was performed on the 13 selected features to visualize the decision boundary. Figures 57 shows the decision boundaries on the PCA projections of training and test sets.

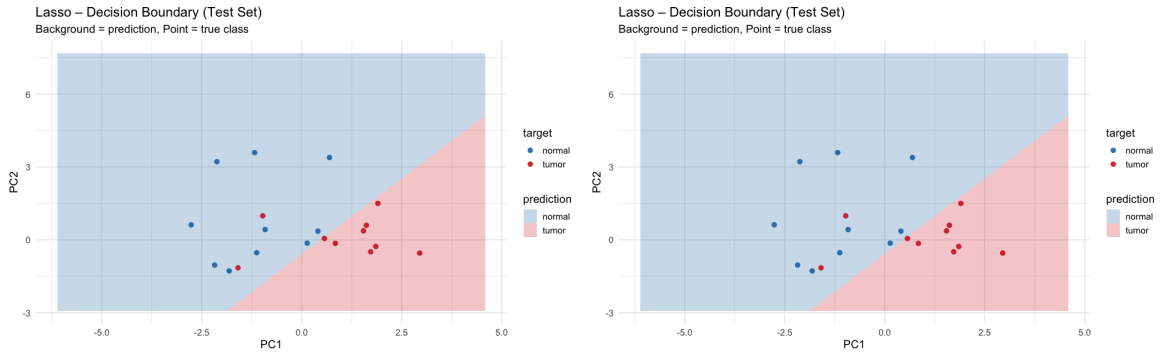


Figure 57: Lasso decision boundary on PCA space: (Left) Training set, (Right) Test set.

Summary and Rationale for Feature Selection. The initial classification models trained on the full feature set of 6034 variables achieved good predictive performances, particularly with ElasticNet and Lasso regularizations which inherently promote sparsity and mitigate overfitting. However, the high dimensionality still posed challenges in terms of model interpretability and potential noise accumulation.

To address these limitations, a feature selection step was introduced using Lasso regularization. This process effectively reduced the feature space from 6034 to 15 key variables, significantly simplifying the model without compromising predictive power. Subsequent models trained on this reduced set demonstrated comparable performance metrics while offering improved interpretability and computational efficiency.
This staged approach highlights the benefit of combining regularization with explicit feature selection, enabling robust and parsimonious classifiers suitable for high-dimensional biomedical data.

6.7 Naive Bayes: Theoretical Background

Naive Bayes is a probabilistic classifier based on Bayes' theorem, which models the posterior probability of a class given the observed features. The fundamental assumption behind Naive Bayes is that the features are conditionally independent given the class label, which simplifies the joint likelihood computation:

$$P(C_k \mid X) = \frac{P(C_k) \prod_{j=1}^{p} P(X_j \mid C_k)}{P(X)}$$

where:

- C_k is the class label (e.g., `tumor` or `normal`),
- $X = (X_1, X_2, \ldots, X_p)$ is the vector of input features,
- $P(C_k)$ is the prior probability of class C_k,
- $P(X_j \mid C_k)$ is the conditional probability of feature X_j given class C_k,
- $P(X)$ is the evidence (normalizing constant).

Despite the strong independence assumption (which rarely holds exactly in real data), Naive Bayes classifiers often perform well in high-dimensional settings and are computationally efficient. Different variants exist depending on the assumed distribution of features, such as Gaussian, multinomial, or Bernoulli Naive Bayes.

In this study, the Gaussian Naive Bayes variant was applied, which assumes that each continuous feature X_j conditioned on class C_k follows a normal distribution:

$$P(X_j \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_{jk}^2}} \exp\left(-\frac{(X_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right)$$

where μ_{jk} and σ_{jk}^2 are the mean and variance of feature X_j in class C_k, estimated from training data.

Given the dataset size and dimensionality ($n = 102$ observations, $p = 6034$ features), Naive Bayes offers a fast and simple classification baseline, useful for comparison with more complex models such as logistic regression with regularization.

6.7.1 Naive Bayes Classification

The Naive Bayes classifier was applied as a probabilistic baseline method for the binary classification task (normal vs. tumor). The model assumes conditional independence of features given the class and estimates class probabilities accordingly.

Confusion Matrices and Evaluation Metrics. Tables 22 and 23 report the confusion matrices obtained on the training and test sets, respectively.

Table 20: Confusion matrix on the training set (Naive Bayes).

Prediction	Normal	Tumor
Normal	28	8
Tumor	12	34

- **Accuracy:** 0.756
- **Precision:** 0.810
- **Recall:** 0.739
- **F1-score:** 0.773

Table 21: Confusion matrix on the test set (Naive Bayes).

Prediction	Normal	Tumor
Normal	8	3
Tumor	2	7

- **Accuracy:** 0.750
- **Precision:** 0.700
- **Recall:** 0.778
- **F1-score:** 0.737

ROC Curve and AUC. Figure 58 shows the ROC curve for the Naive Bayes classifier on the test set, with an AUC of **0.76**.

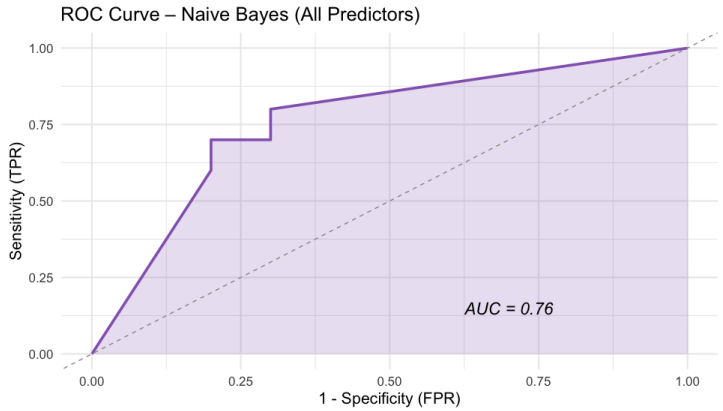


Figure 58: ROC curve of Naive Bayes classifier on the test set.

Geometric Interpretation. A Principal Component Analysis (PCA) was applied to project the high-dimensional data into two dimensions for visualization. A Naive Bayes classifier was retrained on these first two principal components to visualize the decision boundary in 2D space.

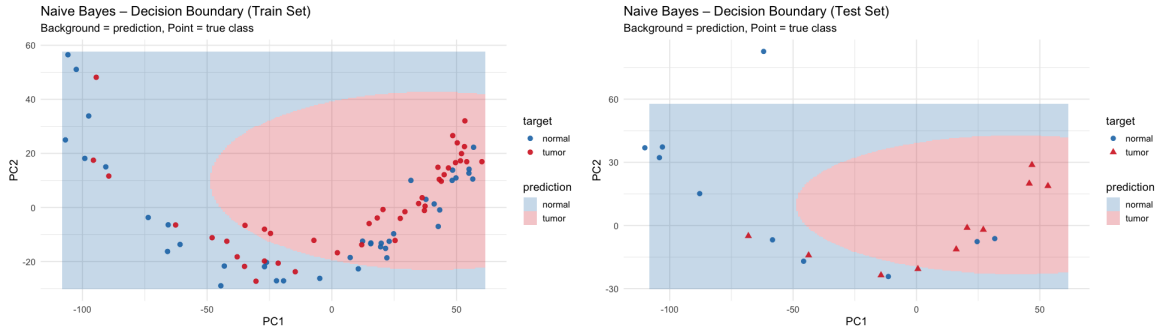


Figure 59: Naive Bayes decision boundary on PCA space: (Left) Training set, (Right) Test set.

This visualization highlights how the Naive Bayes classifier separates classes in the reduced feature space defined by the first two principal components. Although dimensionality reduction may obscure some class distinctions, this provides an intuitive interpretation of classifier behavior.

6.7.2 Naive Bayes Classification with Lasso-Selected Features

The Naive Bayes classifier is a probabilistic model based on Bayes' theorem, which assumes conditional independence among features given the class label. Despite this simplifying assumption, it is often effective in high-dimensional settings and offers computational efficiency. The model estimates the posterior probability of each class and assigns the label with the highest posterior probability to each observation.

In this work, after selecting a subset of relevant features using Lasso regularization, a Naive Bayes classifier was trained on these reduced features. This approach aims to improve generalization by reducing dimensionality while leveraging the simplicity of Naive Bayes.

Performance Evaluation. The classification results on the training and test sets are summarized in Tables 22 and 23, respectively.

Table 22: Confusion matrix on the training set (Naive Bayes with Lasso-selected features).

Prediction	Normal	Tumor
Normal	28	8
Tumor	12	34

- **Accuracy:** 0.756
- **Precision:** 0.810
- **Recall:** 0.739
- **F1-score:** 0.773

Table 23: Confusion matrix on the test set (Naive Bayes with Lasso-selected features).

Prediction	Normal	Tumor
Normal	8	2
Tumor	2	8

- **Accuracy:** 0.800
- **Precision:** 0.700
- **Recall:** 0.778
- **F1-score:** 0.737

ROC Curve and AUC. Figure 60 illustrates the ROC curve obtained on the test set, with an Area Under the Curve (AUC) of **0.835**, indicating a good discriminative ability of the model.

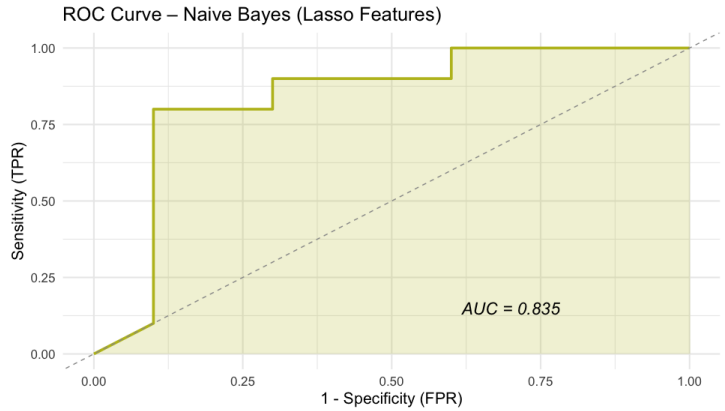


Figure 60: ROC curve of Naive Bayes classifier trained on Lasso-selected features.

Geometric Interpretation via PCA. To visualize the decision boundary, a Principal Component Analysis (PCA) was performed on the selected features. The first two principal components explain a significant portion of variance, as indicated in the axis labels. A Naive Bayes model trained on these two components yields the decision boundary depicted in Figure 61.

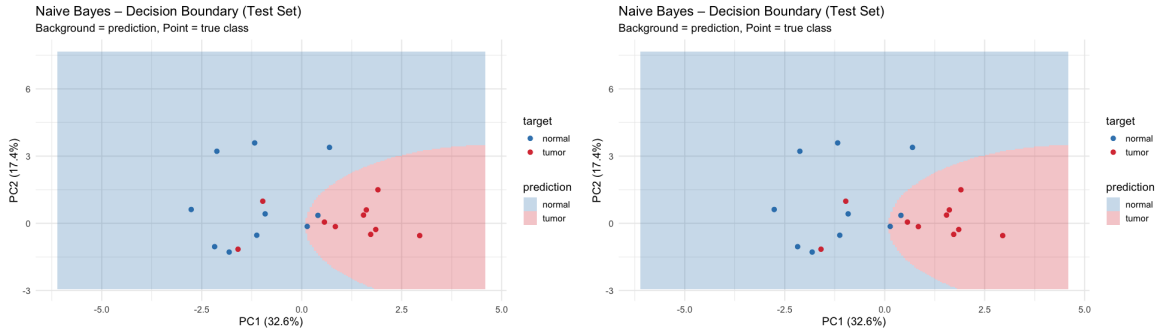


Figure 61: Naive Bayes decision boundary in PCA space using Lasso-selected features: (Left) Training set, (Right) Test set.

6.7.3 Summary of Naive Bayes Classification Results

The Naive Bayes classifier was trained and evaluated both on the full feature set and on a reduced subset selected via Lasso regularization. The full feature model achieved a training accuracy of approximately 75.6% and a test accuracy of 75.0%, with corresponding F1-scores of 0.773 and 0.737, respectively. After dimensionality reduction through Lasso, the classifier was trained on 15 selected features, resulting in improved test performance with an accuracy of 80.0% and an F1-score of 0.737.

PCA visualization of the decision boundaries confirmed the model's capacity to discriminate between classes in the lower-dimensional space, with explained variance reported for interpretability. These results demonstrate that feature selection can effectively enhance Naive Bayes classification by reducing dimensionality and mitigating overfitting while maintaining or improving predictive performance.

6.8 K-Nearest Neighbors (KNN): Theoretical Background

The K-Nearest Neighbors (KNN) algorithm is a simple, non-parametric, instance-based learning method used for classification and regression tasks. In classification, KNN assigns the class label to a new observation based on the majority vote of its k closest training samples in the feature space.

$$\hat{y} = \text{mode}\{y_i : x_i \in \mathcal{N}_k(x)\}$$

where:

- x is the new observation,
- $\mathcal{N}_k(x)$ represents the set of the k nearest neighbors of x in the training data,
- y_i are the class labels of the neighbors,
- $\hat{y}$ is the predicted class label for x.

The closeness or similarity between observations is typically measured using a distance metric such as Euclidean, Manhattan, or Minkowski distance. The choice of k significantly influences model performance: a small k can be sensitive to noise (high variance), while a large k can smooth out class boundaries too much (high bias).
KNN is particularly intuitive and effective in lower-dimensional spaces but may suffer from the "curse of dimensionality" when the number of features p is very large, as distances become less meaningful. Feature selection or dimensionality reduction techniques can alleviate this issue, improving both accuracy and computational efficiency.

In this study, the KNN classifier is applied to the binary classification problem (normal vs tumor) using the full set of features and the subset selected by Lasso regularization for comparison.
After introducing the theoretical foundations of the K-Nearest Neighbors (KNN) algorithm, we applied the classifier to the full set of gene expression features without any prior dimensionality reduction or feature selection. The goal was to evaluate the baseline performance of KNN on the original high-dimensional dataset with $p = 6034$ features and $n = 102$ samples. Given the high dimensionality, this approach tests the robustness and effectiveness of KNN in such settings, considering potential challenges due to the curse of dimensionality.

To optimize the model, the best value of the hyperparameter k (number of neighbors) was determined via cross-validation on the training set, testing multiple candidates and selecting the one that maximized classification accuracy (or an appropriate performance metric). This step is crucial to balance bias and variance, avoiding overfitting (too small k) or underfitting (too large k). he final KNN model with the optimal k was then evaluated on both training and test sets using standard perfor-

mance metrics such as accuracy, precision, recall, F1-score, and ROC curves.

Subsequent sections will compare these baseline results with those obtained after applying Lasso-based feature selection, to assess the impact of dimensionality reduction on classification performance and computational efficiency.

6.8.1 K-Nearest Neighbors (KNN) Classification without Feature Selection

The K-Nearest Neighbors (KNN) classifier was applied using the full set of 6034 features, with the number of neighbors k optimized via 10-fold cross-validation to maximize the AUC metric. The optimal value found was $k = 7$.

Confusion Matrices and Evaluation Metrics. The classification performance was assessed on both the training and test sets. Tables 24 and 25 show the corresponding confusion matrices.

Table 24: Confusion matrix on the training set (KNN, $k = 7$).

Prediction	Normal	Tumor
Normal	33	3
Tumor	7	39

- **Accuracy:** 0.878
- **Precision:** 0.929
- **Recall:** 0.848
- **F1-score:** 0.886

Table 25: Confusion matrix on the test set (KNN, $k = 7$).

Prediction	Normal	Tumor
Normal	8	2
Tumor	2	8

- **Accuracy:** 0.800
- **Precision:** 0.800
- **Recall:** 0.800
- **F1-score:** 0.800

ROC Curve and AUC. Figure 62 displays the ROC curve for the test set, with an AUC of approximately 0.86, indicating a good balance between sensitivity and specificity.

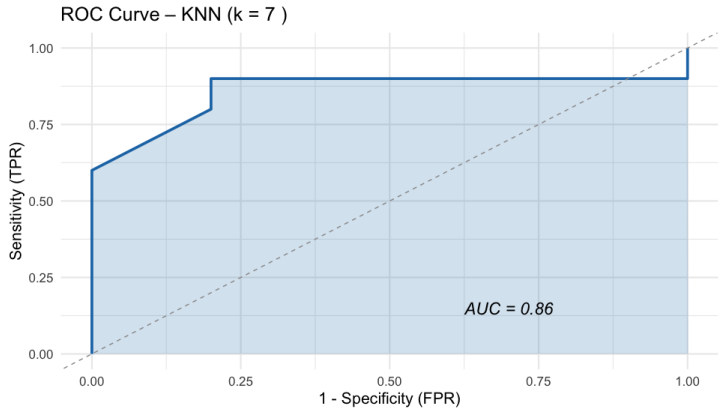


Figure 62: ROC curve of KNN classifier ($k = 7$) on the test set.

Geometric Interpretation and Decision Boundaries. To visualize classifier behavior in reduced dimensions, Principal Component Analysis (PCA) was performed on the training set. The first two principal components capture the most variance and serve as a basis for plotting the decision boundaries and data points.

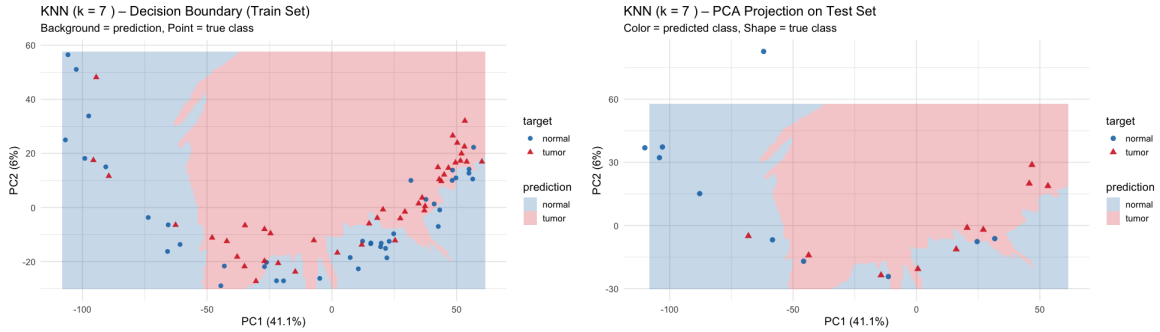


Figure 63: KNN decision boundary on PCA space: (Left) Training set, (Right) Test set.

Summary. The KNN classifier obtained an accuracy of 87.8% on the training set and 80.0% on the test set, with balanced precision and recall values. The PCA visualizations show that the classifier effectively separates classes in the reduced feature space, despite the original high dimensionality of the dataset.

6.8.2 K-Nearest Neighbors (KNN) Classification with Lasso-Selected Features

The K-Nearest Neighbors (KNN) classifier was applied on the subset of features selected by Lasso regularization. The optimal number of neighbors k was chosen via 10-fold cross-validation, resulting in $k = 25$.

Confusion Matrices and Evaluation Metrics. Tables 26 and 27 show the confusion matrices on training and test sets, respectively.

Table 26: Confusion matrix on the training set (KNN with Lasso features, $k = 25$).

Prediction	Normal	Tumor
Normal	39	0
Tumor	1	42

- **Accuracy:** 0.988
- **Precision:** 1.000
- **Recall:** 0.977
- **F1-score:** 0.988

Table 27: Confusion matrix on the test set (KNN with Lasso features, $k = 25$).

Prediction	Normal	Tumor
Normal	7	2
Tumor	3	8

- **Accuracy:** 0.750
- **Precision:** 0.778
- **Recall:** 0.727
- **F1-score:** 0.762

Summary. The KNN classifier with Lasso-selected features reached near-perfect metrics on the training set, showing excellent fit. The test set performance is lower but still indicates reasonable generalization given the dimensionality reduction. This highlights the trade-off between model complexity and overfitting.

ROC Curve and AUC. The ROC curve for the test set is shown in Figure 64, with an AUC value of 0.9, reflecting discriminatory ability.

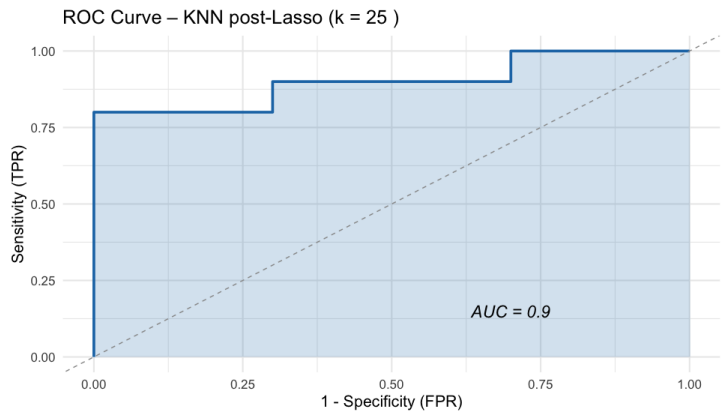


Figure 64: ROC curve of KNN classifier with Lasso-selected features ($k = 25$) on the test set.

Geometric Interpretation and Decision Boundaries. PCA on the Lasso-selected features revealed that the first two components explain most of the variance. Figure 65 visualizes the decision boundaries and sample distribution in this reduced space, using triangles for both classes.

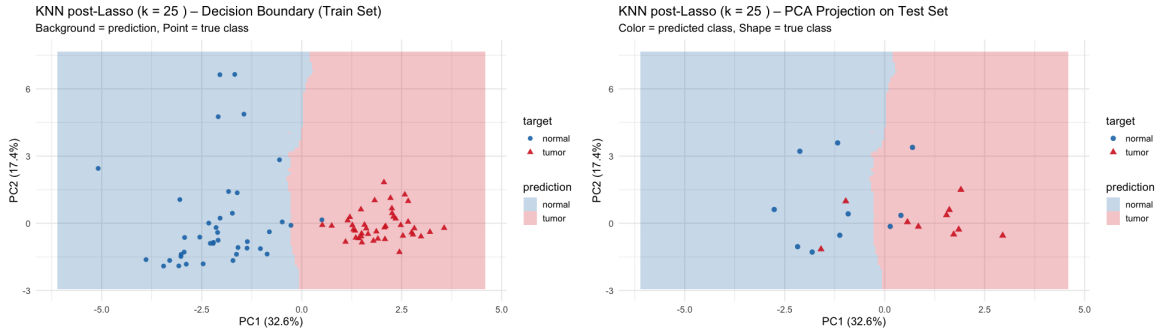


Figure 65: KNN decision boundary on PCA space with Lasso-selected features: (Left) Training set, (Right) Test set.

Summary. The KNN classifier on Lasso-selected features achieved near-perfect performance on the training set, highlighting the efficacy of dimensionality reduction. However, the reduced test set accuracy and metrics suggest remaining challenges in generalization, possibly due to data complexity or sample size limitations. PCA visualizations support these observations by showing good class separability in the reduced space

Recap. The KNN classifier was trained using features selected by Lasso regularization, which effectively reduced dimensionality. With the optimal number of neighbors $k = 25$, the model achieved very high accuracy (98.8%) and excellent precision and recall on the training set, indicating strong learning capability without overfitting. On the test set, performance decreased to an accuracy of 75.0%, reflecting the typical trade-off between complexity reduction and generalization. Overall, the Lasso-based feature selection improved computational efficiency and model interpretability, while maintaining satisfactory classification results.

6.9 Model Comparison and Analysis and conclusions

Table 28 reports the comparative performance of all classifiers evaluated on the test set, both with the full feature space and after Lasso-based feature selection. Evaluation metrics include accuracy, precision, recall, F1-score, and AUC, allowing a comprehensive assessment of predictive performance and generalization ability.

Table 28: Summary of classification performance on the test set (with and without Lasso feature selection).

Model	Features	Accuracy	Precision	Recall	F1-score	AUC
ElasticNet	Full	0.85	0.80	0.90	0.842	0.89
Ridge	Full	0.85	0.80	0.90	0.842	0.85
Lasso	Full	0.85	0.80	0.90	0.842	0.85
Naive Bayes	Full	0.75	0.70	0.78	0.737	0.76
KNN	Full	0.80	0.80	0.80	0.800	0.86
ElasticNet	Lasso FS	0.80	0.80	0.80	0.800	0.80
Ridge	Lasso FS	0.70	0.80	0.67	0.727	0.88
Lasso	Lasso FS	0.80	0.80	0.80	0.800	0.89
Naive Bayes	Lasso FS	0.80	0.70	0.78	0.737	0.835
KNN	Lasso FS	0.75	0.78	0.73	0.762	0.90

Interpretation. Among models trained on the full set of 6034 features, regularized logistic regressions (Lasso, Ridge, ElasticNet) achieved the highest AUC (up to 0.89), with strong precision and recall. KNN followed closely with competitive results (AUC = 0.86), while Naive Bayes underperformed slightly due to its strong independence assumption.
After feature selection via Lasso (reducing the space to 15 features), ElasticNet and Lasso retained excellent performance (AUC = 0.89), confirming the relevance of the selected predictors. Interestingly, KNN showed the highest AUC post-selection (0.90), demonstrating that dimensionality reduction can mitigate the curse of dimensionality in distance-based classifiers.

In contrast, Ridge suffered a slight drop in recall (0.67), reducing its F1-score. Overall, ElasticNet and Lasso offered the best trade-off between generalization, interpretability, and robustness. Naive Bayes also benefited from feature selection, improving AUC from 0.76 to 0.835.

These results support the effectiveness of Lasso-based feature selection in high-dimensional classification, especially when followed by regularized logistic regression or KNN.

6.9.1 Comparative ROC Curves and AUC Evaluation

Figures 66 and 67 show the comparative ROC curves for the five classification models (ElasticNet, Ridge, Lasso, Naive Bayes, and KNN) evaluated on the test set using the full feature space and after Lasso-based feature selection, respectively.

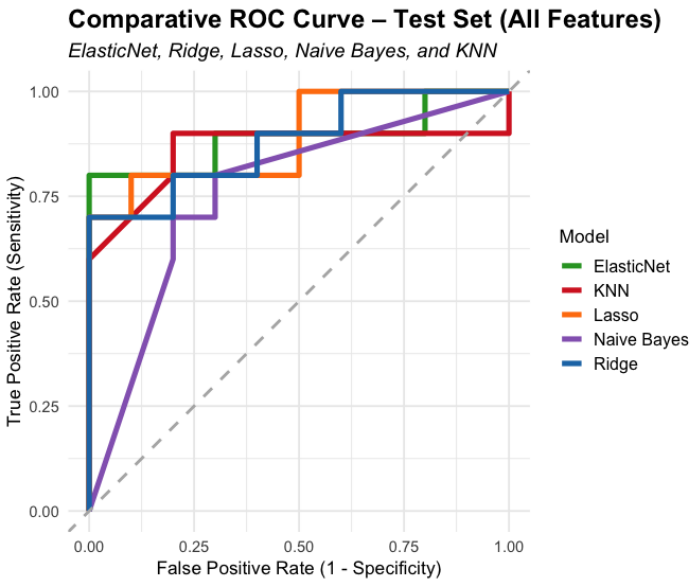


Figure 66: Comparative ROC Curve – Test Set (All Features).

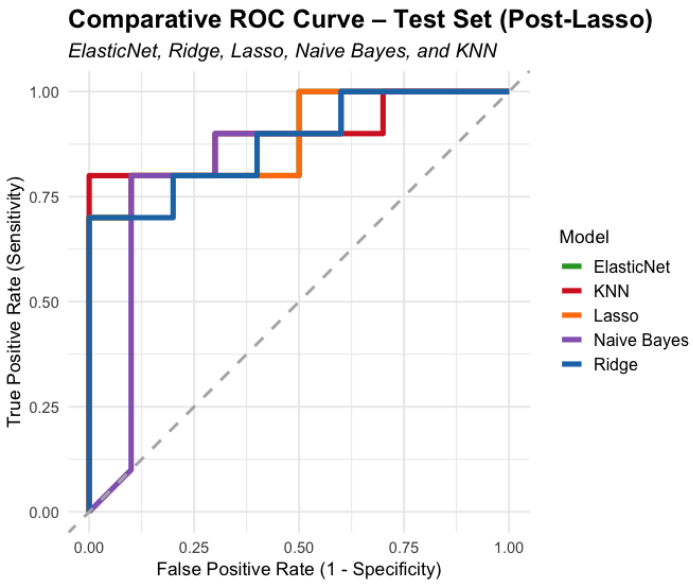


Figure 67: Comparative ROC Curve – Test Set (Post-Lasso).

Interpretation. When considering all features (Figure 66), the regularized logistic regression models (ElasticNet, Ridge, and Lasso) exhibit high discriminative performance, as demonstrated by the steep ROC curves and AUCs above 0.85. In contrast, Naive Bayes shows reduced sensitivity at lower false positive rates, indicating a weaker ability to distinguish between classes when all features are retained.

This performance drop is expected, given that Naive Bayes suffers from the curse of dimensionality and feature dependence violations.
After applying Lasso-based feature selection (Figure 67), all models benefit from dimensionality reduction. The ROC curves become more aligned and tighter, reflecting improved class separation and generalization. Notably, KNN shows the most significant gain, with a smoother curve and an AUC increase from 0.86 to 0.90. This demonstrates that removing irrelevant features enhances distance-based learning.

ElasticNet and Lasso maintain high AUCs (0.89), confirming their robustness and sparsity-oriented regularization. Ridge, while slightly improving in AUC (from 0.85 to 0.88), shows a drop in recall and F1-score, suggesting that its lack of feature elimination may still allow noise to propagate. Naive Bayes also benefits from the reduced feature set, with its AUC rising to 0.835.

In summary, feature selection improves model interpretability and generalization, particularly for non-parametric and probabilistic classifiers like KNN and Naive Bayes. Regularized logistic regressions (especially ElasticNet and Lasso) maintain strong performance in both settings, validating their effectiveness in high-dimensional biomedical classification.

References

[1] S. S. Hameed, R. Hassan, W. H. Hassan, F. F. Muhammadsharif, and L. A. Latiff, "HDG-select: A novel GUI based application for gene selection and classification in high dimensional datasets," *PLOS ONE*, vol. 16, no. 1, e0246039, 2021. doi: 10.1371/journal.pone.0246039. [Online]. Available: https://doi.org/10.1371/journal.pone.0246039.

[2] S. S. Hameed, R. Hassan, W. H. Hassan, F. F. Muhammadsharif, and L. A. Latiff, *The microarray dataset of prostate cancer in csv format*, https://doi.org/10.1371/journal.pone.0246039.s, 2021. doi: 10.1371/journal.pone.0246039.s.

Dataset published in PLOS ONE.

[3] A. Tsanas and M. Little, *Parkinsons Telemonitoring*, UCI Machine Learning Repository, 2009.

DOI: https://doi.org/10.24432/C5ZS3N.