University of Cape Town

STA5090Z

Advanced Topics in Regression

Assignment I

Author:
Roger Bukuru

Student Number:
BKRROG001

July 13, 2024

Contents

1 Question 1: Splines

1.1 Introduction and Aim

Investigating cubic regression splines is the goal of this question. We will focus on achieving the following:

- Building and visualizing a set of six basis functions
- Using a penalized regression spline with 30 knots and evaluate the overfitting and complexity of the model.
- Generating confidence bands to illustrate the uncertainty around the the fitted spline.
- Analysing the Generalized Cross-Validation (GCV) score as a function of the smoothing parameter.

1.2 Methodology

1. Constructing the Basis: We used the cubic spline basis described by Wood (2006) and Gu (2002). The basis functions are:

$$b_1(x) = 1, \quad b_2(x) = x, \quad b_{k+2}(x) = R(x, x_k^*) \quad \text{for} \quad k = 1, \ldots, q-2,$$

where x_k^* are the knot locations.

2. Penalized Regression Spline: To balance the smoothness of the model fit , a penalized regression spline includes a penalty term in the least squares model. A $\lambda = 0.000001$ was chosen for this question. The penalty matrix also defined by Wood (2006) is defined as:

$$S_{i+2,j+2} = R(x_i^*, x_j^*) \quad \text{where} \quad i, j = 1, \ldots, q-2.$$

The function $R(x, z)$ is defined as:

$$R(x,z) = \left((z-0.5)^2 - \frac{1}{12}\right)\left((x-0.5)^2 - \frac{1}{12}\right)/4 - \left((|x-z|-0.5)^4 - 0.5(|x-z|-0.5)^2 + \frac{7}{240}\right)/24.$$

3. Confidence Bands: Confidence bands visually depict the uncertainty surrounding the fitted splines. Using the fitted values's standard errors, we computed these intervals.

4. Generalized Cross-Validation (GCV): GCV helps select the smoothing paramter by balancing smoothness and fit, thereby reducing the risk of overfitting. The GCV score is given by:

$$\text{GCV}(\lambda) = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\left(1 - \frac{\text{tr}(A)}{n}\right)^2}$$

where y_i are the observed values, $\hat{y}_i$ are the fitted values, A is the smoothing matrix, and λ is the smoothing parameter.

5. Data Fitted The data to which the spline model was fitted on was created as follow:

$$Y_i \sim \mathcal{N}(\mu_i, 0.5^2),$$

$$\mu_i = 5 + \sin(3\pi(X_i - 0.6)),$$

$$X_i \sim \mathcal{N}(0, 1).$$

We simulated $n = 100$ samples from the above model.

1.3 Results and Analysis

1. Basis Functions (1a): We built and visualized the basis functions for the cubic spline on the range $x \in [0, 1]$. Figure 1 shows the set of 6 basis functions used in building the spline.

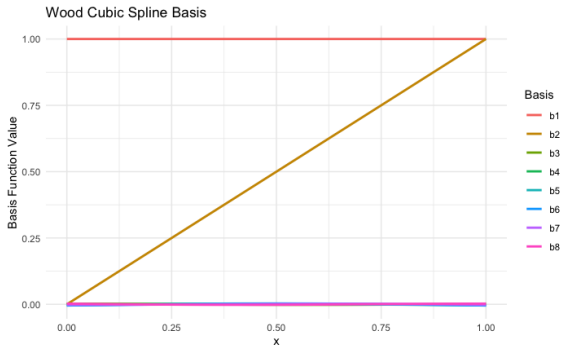

Figure 1: Basis Functions

We observe that the flexibility and smoothness of the spline are significantly influenced by the basis functions that are chosen. For smooth transitions, continuity up to the second derivative is guaranteed by the constructed basis.

2. Penalized Regression Spline Fit (1b): A penalized regression spline was fitted to the simulated data points using 30 equally spaced knots. In figure 2 we show how the spline captures the underlying pattern in the data while controlling for overfitting.

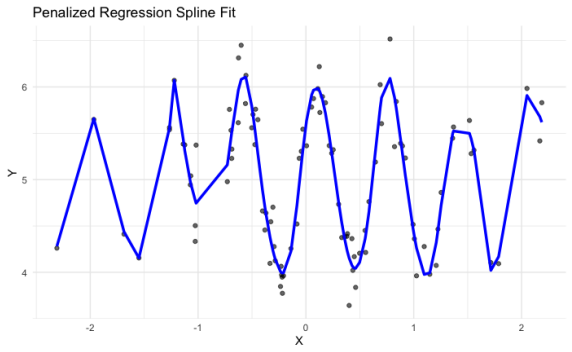

Figure 2: Penalized Regression Spline Fit

We observed that the penalized spline fit managed to effectively balance capturing the data trend whilst avoiding overfitting. The use of 30 knots provided sufficient flexibility to model the data accurately, as shown in Figure 2.

3. Confidence Bands (1c): We obtained confidence bands for the penalized regression spline. We visualized this in Figure 3 which includes the fitted spline with confidence bands, indicating the variability of the fit at different points.

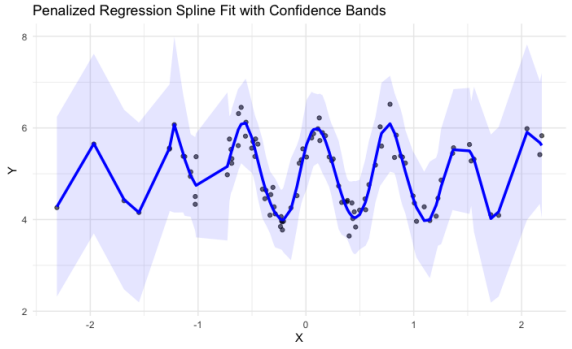

Figure 3: Confidence Bands

The confidence bands provide information about how reliable the fit is. Wider bands indicate greater uncertainty, typically at the boundaries or in regions with sparse data. Figure 3 shows wide confidence bands at the boundaries indicating greater uncertainty compared to the more narrow bands in the middle, suggesting less uncertainty and a greater fit within this region.

4. GCV as a Function of the Smoothing Parameter (1d): The GCV score was plotted as a function of the smoothing parameter. In figure 4 we observe how the GCV score varies with different values of the smoothing parameter, helping in determining the optimal level of smoothness.

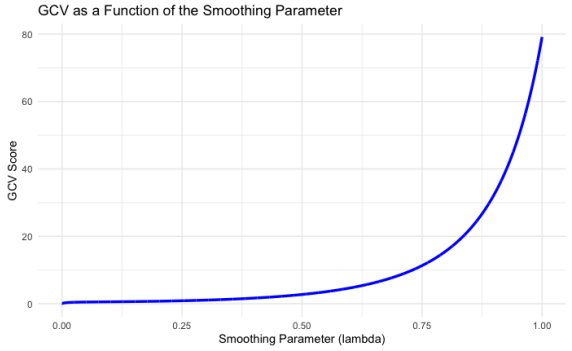

Figure 4: GCV Score

The GCV plot helps in selecting the optimal smoothing parameter. A lower GCV score indicates a better balance between fit and smoothness. In figure 4 we observe an increase in GCV with higher smoothing parameters, this indicates, that higher values of the smoothing parameter will result in over smoothing.

2 Question 2: Two-Dimensional Splines

2.1 Introduction and Aim

The goal of this question is to use two-dimensional splines to model the median Fundamental Power Quotient (medFPQ) as a function of voxel positions (X and Y). More specifically, we want to:

- Implement a tensor product spline with 10 basis functions for each dimension and apply penalization upon performing the regression.
- Fit 100 equally spaced knots to a thin-plate spline basis, penalizing for over-fitting.

The goal is to compare the performances of these spline approaches and assess how well they reflect the spatial dependency in medFPQ values. We will be implementing these spline both from a first principles approach and using the *mgcv* package in R to compare our results.

2.2 Methodology

1. Tensor Product Spline: To build tensor product splines, basis functions for both dimensions (X and Y) are created, then combined to create a grid. A tensor product spline's general form is as follows:

$$f(x, y) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \beta_{ij} B_i(x) B_j(y)$$

where $B_i(x)$ and $B_j(y)$ are basis functions for the dimensions x and y respectively, and β_{ij} are the coefficients to be estimated. Knot locations were selected evenly across the range of X and Y coordinates to ensure an adequate spread. The basis function including the penalty matrix from question one was used to enforce smoothness and avoid overfitting[1].

2. Thin-Plate Spline: When fitting spatial data, thin-plate splines are known for their adaptability and smoothness. The thin-plate spline basis function is given by:

$$f(x, y) = \sum_{i=1}^{N} \alpha_i \phi(\|\mathbf{X} - \mathbf{X}_i\|)$$

where $\phi(r) = r^2 \log(r)$ for $r > 0$ and $\phi(0) = 0$, and $\mathbf{X}_i$ are the knot locations. We used 100 knots to cover the spatial domain. Similar to the tensor product spline, a penalty was applied to control the smoothness and prevent overfitting[1].

2.3 Results and Analysis

1. Original Data: The original data plot serves as a baseline for comparison by displaying the observed medFPQ values at various voxel sites. This is seen in Figure 5.

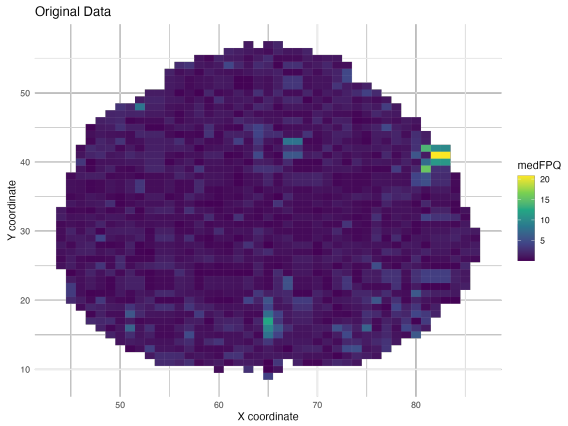

Figure 5: Original Data

2. Tensor Product Spline Fit: The tensor product spline fit from first princples managed to capture the general trend of the medFPQ values across the brain slice as shown in figure 6, however we note a poor fit in that, large parts of the fitted values do not align with the observed data. This may be a the result of a poor smoothing penalty, for this fit, we used a λ of 0.00001, without cross-validation. Another element that may have contributed to the poor fit could be our choice of basis functions.

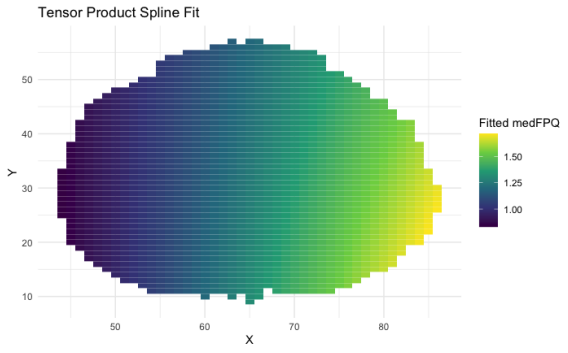

Figure 6: Tensor Product Spline Fit

3. Thin-Plate Spline Fit: The thin plate spline fit shown in figure 7, similarly managed to capture the general pattern of the medFPQ values, however we note that it shows a different pattern to the tensor product spline. Again the choice of smoothing parameter could also have played a role in a poor fit compared to the observed data, as we note that the model introduced some smoothness that might oversimplify certain areas of the brain activity.

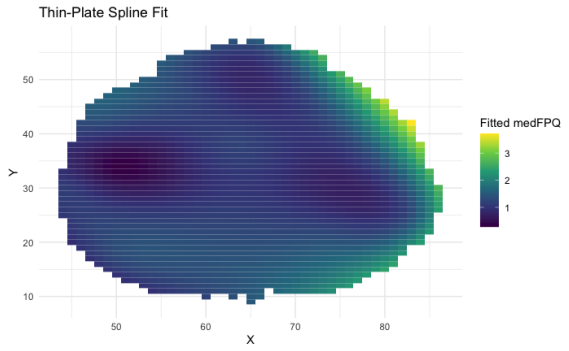

Figure 7: Thin-Plate Spline: Original vs Fitted Values

4. Comparison with mgcv package:

We compared our first principles approach with the results of the `mgcv` package.

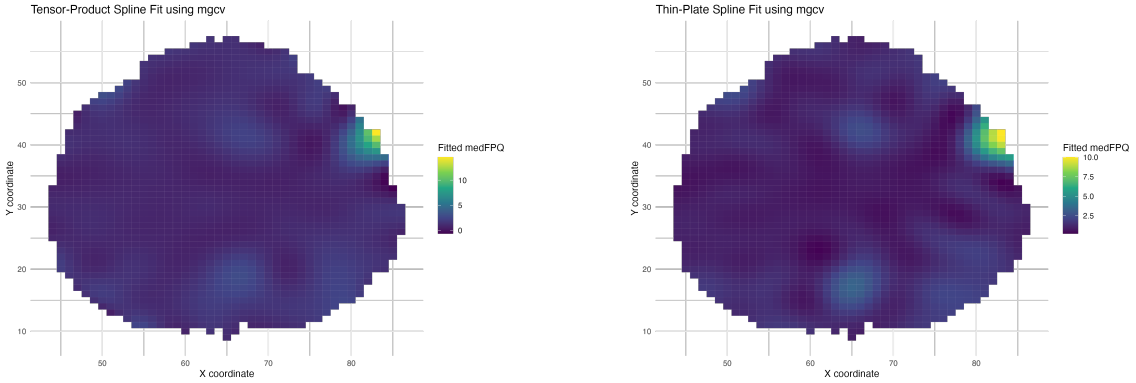

Figure 8: mgcv: Tensor Product Spline and Thin Plate Spline Fits

As can be seen in figure 8 above, the tensor product spline fit of the `mgcv` package shows trends that are comparable to our first principle approach, but it provides a more refined smoothing because of the optimized penalized that the `mgcv` implementation provides. In a similar vein, the thin plate spline fit provided by the `mgcv` package exhibits smoother transitions and more accurately represents the underlying structure of the brain activity than our first principles method. Since the results of the `mgcv` package are comparable to the observed data, they yield results that are generally more credible. Moreover the tensor product spline and thin-splate spline results when using the `mgcv` package are comparable as they produce similar results. As a final remark we do however observe over-smoothing in the fitted values when using the `mgcv` package, which is similar to our first principle approach.

2.4 Conclusion

In our first principles approach the findings demonstrate that the thin-plate spline and tensor product models offer distinct, but varying, fits to the brain data. In contrast to our first principles approach, the `mgcv` package provides optimal penalization and enhanced smoothing, leading to not only better fitting models but also comparable results between the two mult-dimensional spline approaches. Overall, the outcomes of our first principles method were subpar could be improved via cross-validation

3 Question 3: GAMs and MARS

3.1 Introduction and Aim

In order to determine whether a candidate star is a pulsar, this question compares the performance of three distinct models: generalized additive models (GAMs), multivariate adaptive regression splines (MARS), and as a baseline model logistic regression. The dataset has eight continuous variables that characterize various features of a star, while one class variable defines whether the star is a pulsar or not. For our analysis we partitioned the data set into a 20% subset for testing and an 80% subset for training.

3.2 Methodology

Generalized Additive Models (GAMs): The first method we will be implementing will be Generalized Additive Models (GAMs), GAMs serve as an extension of Generalized Linear Models as they provide the ability to incorporate non-linear functions in the predictor variable. This provides one with the ability to model more complex data patterns while yet preserving interpretability. We formally specify the model as:

$$g(E[Y]) = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p)$$

where g is the link function, $E[Y]$ is the expected value of the response variable, and f_i are smooth functions of the predictor variables[3].

Multivariate Adaptive Regression Splines (MARS): MARS is an adaptable method of modeling that automatically accounts for interactions and non-linearities among variables. A weighted sum of basis functions, each of which is a piecewise linear function (spline), is used to represent the model. The MARS model is:

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m B_m(X)$$

where $B_m(X)$ are the basis functions and β_m are the coefficients[1].

Logistic Regression: As a baseline method we made use of a logistic regeression which is a common classification method, in which the likelihood of the default class is expressed as a logistic function of the predictor variables' linear combination. The logistic function is defined as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+...+\beta_p X_p)}}$$

where Y is the dependent variable, X_i are the independent variables, and β_i are the coefficients[3].

3.3 Results and Analysis

Table 1 summarizes each model's evaluation metrics. For every model, the accuracy, precision, recall, and F1 score are listed in the table.

Model	Accuracy	Precision	Recall	F1 Score
GAM	0.981	0.98	0.92	0.98
MARS	0.981	0.98	0.96	0.99
Logistic Regression	0.979	0.98	0.91	0.98

Table 1: Model Evaluation Metrics

Generalized Additive Models (GAMs): With an accuracy of 0.981, the GAM model performed marginally better than the logistic regression model, but it exactly matched the accuracy of the MARS model. For precision, recall, and F1 score, the model produced scores of 0.98, 0.92, and 0.98, respectively.

Multivariate Adaptive Regression Splines (MARS): The MARS model also marginally outperformed the logistic regression model in accuracy, with a matching accuracy of 0.981 to the GAM model. For precision, recall, and F1 score, the model produced scores of 0.99, 0.96, and 0.98, respectively.

Logistic Regression: The aim of the logist regression model was to use it as a baseline performance measure, upon reviewing the results of the GAM and MARS model we can confidently say that these models matched the baseline and in some cases marginally outperformed it.

3.4 Conclusion

We observed really high quality results from all three models, in order to determine which model to use, we will have to assess our modelling goals given each model has

performed really well. If performance and interpretability are important and we aim to have a balance between the two then the GAM model would be recommended. If we wish to capture complex interactions then MARS would be more appropriate in capturing these interactions. Finally, our baseline Logistic Regression model could be preferable for quick and understandable results where simplicity is key. In this classification task, it was similar to the more complex GAM and MARS models.

4 Question 4: Wavelets

4.1 Introduction and Aim

In order to differentiate differentiate between the spoken sounds "aa" and "ao" also referred to as phonemes we aim to use a penalized logistic regression model following a wavelet modification in order to classify these sounds. The data set to be used will be the phonemes data set that can be found in the `fds` package. We will proceed by first using wavelets to transform these sounds and thereafter we will apply a penalized logistic regression in order to classify the sounds.

4.2 Methodology

Data Transformation: A discrete wavelet transform (DWT) was used to convert each signal in the phonemes data into a wavelet coefficient. . This involves:

$$\mathbf{X}_{ij} \rightarrow \mathbf{W}\mathbf{x}_{ij}$$

where $\mathbf{W}$ is the wavelet basis matrix[1]. Specifically, we used the Daubechies wavelet (DaubLeAsymm) with a filter number of 10. The wavelet transform was applied to the first 256 wavelengths of the signals.

Logistic Regression Model: With t denoting the temporal or spatial index, let $x_i(t)$ be the functional data that represents the i-th signal. After $x_i(t)$ is wavelet transformed, a collection of wavelet coefficients W_{ij} is produced, with the coefficients being indexed by j. The following is a formulation for the logistic regression model[1][3]:

$$\log\left(\frac{P(y_i = 1)}{P(y_i = 0)}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j W_{ij}$$

where y_i is the binary response indicating the class (0 for "aa" and 1 for "ao"), β_0 is the intercept, and β_j are the coefficients associated with the wavelet features. With the above setup we proceed by training a model by means of a penalized likelihood

in order to avoid overfitting, we control the penalty term by the parameter λ, and use cross-validation in order to determine the optimal value.

$$\text{Penalized Likelihood} = \log L(\beta) - \lambda \sum_{j=1}^{p} |\beta_j|$$

Wavelet Coefficient Transformation: The "aa" and "ao" class signals were converted into wavelet coefficients. Sample signals and their related wavelet coefficients for both classes are shown in Figure 9.

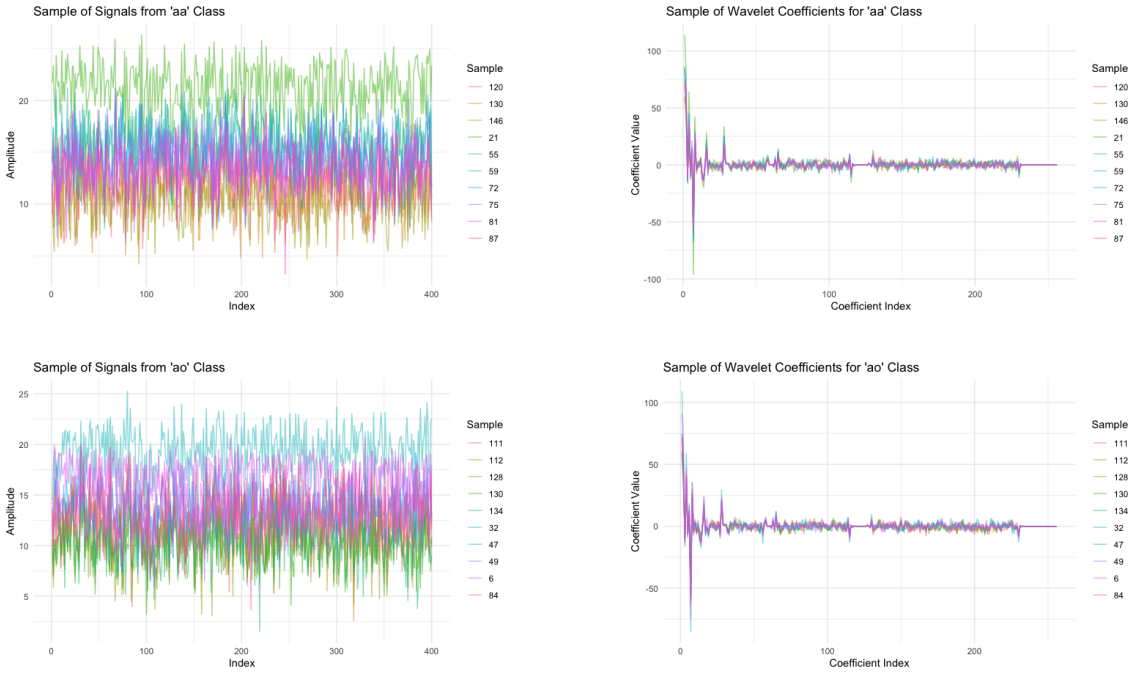

Figure 9: Sample signals and their corresponding wavelet coefficients for "aa" and "ao" classes.

Model Training and Evaluation: The wavelet coefficients were used as features in a penalized logistic regression model. 30% of the data were used for testing after the model had been trained on 70% of it. We used cross-validation to determine the optimal penalty parameter (λ).

4.3 Results and Analysis

Model Performance: The model performance was perfect, as on the test set we managed to achieve a prediction accuracy of 100% which corresponds to the perfect

model performance metrics shown in table 2. The perfect score shows that the wavelet transformation was able to detect and capture the important regions within the data such that it could effectively differentiate between the "aa" and "ao" sounds.

Metric	Value
Accuracy	1.00
Precision	1.00
Recall	1.00
F1 Score	1.00

Table 2: Model Evaluation Metrics

Feature Importance: One of the important output of this analysis was to determined which areas of the spectrum were important for predicting which sounds were spoken, upon evaluation of the wavelet coefficients, the top ten most important coefficients were captured and can be seen below in figure 10.

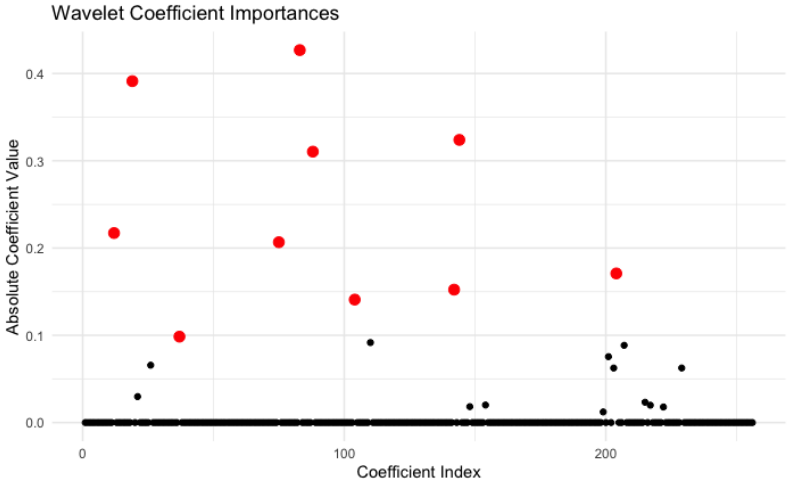

Figure 10: Wavelet Coefficient Importances

Indices 83, 19, 144, 88, 12, 75, 204, 142, 104, and 37 were the top 10 most significant wavelet coefficients. These coefficients contributed significantly to distinguishing between the "aa" and "ao" phonemes.

4.4 Conclusion

Wavelet transformation was shown to be an efficient method for extracting features in the categorization of phoneme signals. The penalized logistic regression model

effectively utilized these features, resulting in a classification accuracy of 100

5 Question 5: Functional Data Analysis

5.1 Introduction and Aim

Fuel's "knocking" resistance is based on its octane rating. An engine can run at a higher compression ratio the higher the octane rating. The conventional method of measuring octane in special variable compression ratio engines is to compare the knocking resistance of fuel samples to standard mixes. In comparison to getting a sample's near-infrared (NIR) spectrum, this is a costly procedure. Predicting the octane rating from the spectrum might be beneficial.

The purpose of this question is to construct a scalar-on-function regression model to model the octane rating as a function of the NIR spectrum from the petrol sample using the octane rating data in the R package `gamair (gas)`. The model was implemented from a first-principles approach.

5.2 Methodology

Data Preparation: The NIR spectra and octane numbers of sixty petrol samples are included in the data set. Diffuse reflectance was used to measure the NIR spectra at 401 wavelengths, ranging from 900 nm to 1700 nm in 2 nm intervals. Using B-spline basis functions, the NIR spectra data were preprocessed and transformed into a functional data object.

Functional Data Analysis Model: Define Y_i as the octane rating for the i-th sample and $X_i(t)$ as the NIR spectrum as a function of wavelength t. The equation for the model is as follows[2]:

$$Y_i = \beta_0 + \int_{900}^{1700} \beta(t) X_i(t)\, dt + \epsilon_i$$

Here, $\beta(t)$ represents the coefficient function that needs to be estimated, and ϵ_i represents the error term.

The coefficient function $\beta(t)$ is estimated using B-spline basis functions $\phi_k(t)$ and it is defined as follows:

$$\beta(t) \approx \sum_{k=1}^{K} b_k \phi_k(t)$$

where $\phi_k(t)$ are the B-spline basis functions and b_k are the

This transforms the functional regression model into a standard linear regression problem:

$$Y_i \approx \beta_0 + \sum_{k=1}^{K} b_k \int_{900}^{1700} \phi_k(t) X_i(t)\, dt + \epsilon_i$$

Basis Expansion: NIR spectra data were extended using the basis functions of the B-spline. Thereafter we used the expanded data to fit a standard linear regression model to predict the octane rating.

Model Fitting: We fitted a linear model using the coefficients of the B-spline basis expansion as predictors. The model was assessed using the R^2 value and the residual standard error.

5.3 Results and Analysis

NIR Spectra: In figure 11 we show NIR spectra of petrol samples. What stands out is that the spectra exhibits some high fluctuations throughout the range of wavelengths, using B-spline basis functions we aimed to capture these fluctuations.

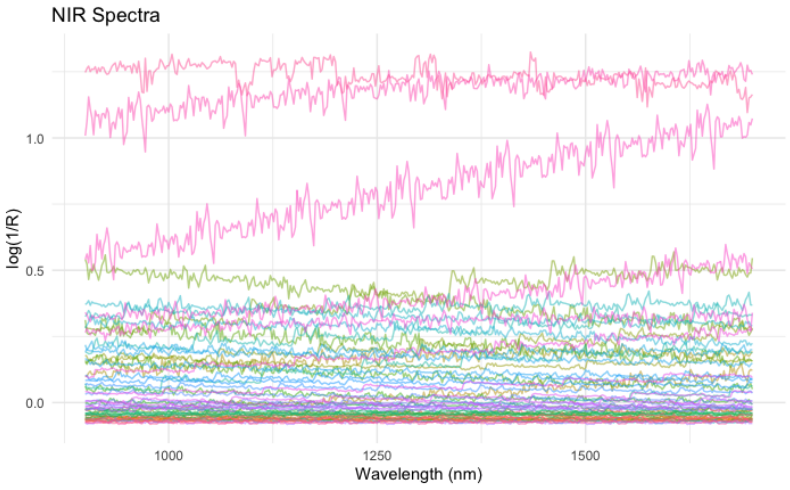

Figure 11: NIR Spectra of Petrol Samples

Model Summary: In Table 3, the fitted linear model summary is shown. Although some of the variance in the octane ratings may be explained by the model, as seen by the R^2 value, there is still room for improvement.

Metric	Value
Residual Standard Error	83.7
Multiple R^2	0.3856
Adjusted R^2	0.07842
F-statistic	1.255
P-value	0.2637

Table 3: Model Summary

Actual vs Predicted Octane Ratings: In figure 12 we show the comparison between the actual and predicted octane ratings. The figure suggests that although the model detects some patterns, it also reveals discrepancies, suggesting that additional features or model refining may be necessary to enhance the accuracy of predictions.

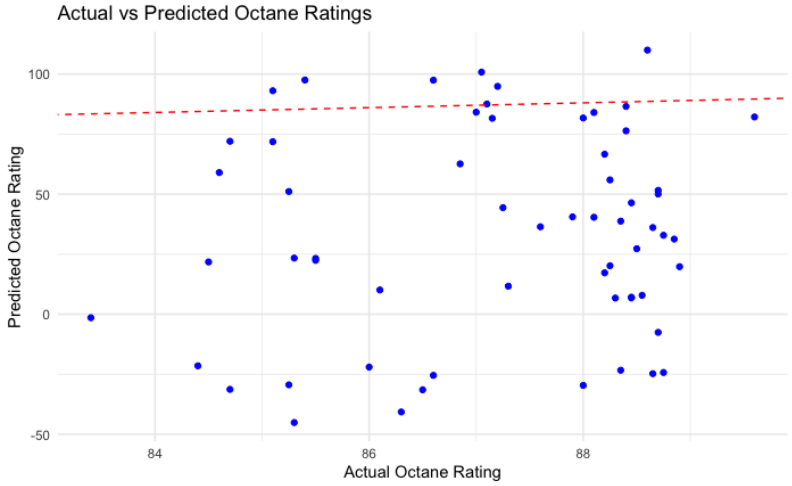

Figure 12: Actual vs Predicted Octane Ratings

5.4 Conclusion

The scalar-on-function regression model with B-spline basis functions could be used to estimate the octane rating from the NIR spectra of gasoline samples. The model was able to explain a moderate percentage of the volatility in the octane ratings, however there is still room for improvement. By including new predictors or looking at different basis functions, the model's prediction ability may be enhanced.

References

[1] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer series in statistics. Springer, 2009.

[2] J. Ramsay and B.W. Silverman. *Functional Data Analysis.* Springer Series in Statistics. Springer New York, 2006.

[3] S. Wood. *Generalized Additive Models: An Introduction with R.* Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.