Faculty of Mathematics and Informatics

VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

CLASSIFICATION OF SPEECH SIGNAL USING FUNCTIONAL DATA

ŠNEKOS SIGNALO KLASIFIKAVIMAS TAIKANT FUNKCINIUS DUOMENIS

Master's thesis

Author: Judita Vengalienė
VU email address: judita.vengaliene@mif.stud.vu.lt

Supervisor: Assoc. Prof. Dr. Jurgita Markevičiūtė

Vilnius
2023

Classification of Speech Signal using Functional Data

Abstract

The objective of this study is to classify Lithuanian words recorded in audio files by predicting the speaker's gender. Initially, the Hilbert transform was applied to the speech signals. Subsequently, after finding the optimal parameters, the smoothing of the speech signals was performed. Finally, the classification was done by using three classifiers: K-Nearest Neighbor, Support Vector Machine and Random Forest. All classifiers were applied to both functional and multivariate data after utilizing Functional Data Analysis. Evaluation of the results revealed that the Random Forest classifier for multivariate data was the most effective, achieving an accuracy of 82.60 % in predicting the speaker's gender.

Key words: speech signal, gender classification, k-nearest neighbor, support vector machine, random forest.

Šnekos signalo klasifikavimas taikant funkcinius duomenis

Santrauka

Šio tyrimo tikslas - klasifikuoti garso failuose įrašytus lietuviškus žodžius pagal kalbėtojo lytį. Iš pradžių kalbos signalams buvo pritaikyta Hilberto transformacija. Vėliau, suradus optimalius parametrus, atliktas kalbos signalų glodinimas. Galiausiai klasifikavimas atliktas naudojant tris klasifikatorius: K-Artimiausio Kaimyno, Atraminių Vektorių Mašinos ir Atsitiktinio Miško. Visi klasifikatoriai buvo taikomi tiek funkciniams, tiek daugiamačiams duomenims, panaudojus Funkcinę Duomenų Analizę. Įvertinus rezultatus paaiškėjo, kad Atsitiktinio Miško klasifikatorius, skirtas daugiamačiams duomenims, buvo veiksmingiausias - jo tikslumas prognozuojant kalbėtojo lytį siekė 82,60 %.

Raktiniai žodžiai: šnekos signalas, lyties klasifikavimas, k-artimiausias kaimynas, atraminių vektorių mašina, atsitiktinis miškas.

Contents

1 Introduction

As the demand for Human-Computer Interaction (HCI) systems rises, speech processing becomes pivotal for enhancing these systems. Speech classification plays a crucial role in the domain of speech signal processing. In essence, it involves categorizing spoken language into distinct classes or categories based on various features extracted from the audio signal. Researchers primarily focus on identifying and classifying key attributes such as the speaker's gender, age, and emotional state from speech signals. Various techniques combining machine learning and signal processing methods, are employed for effective classification. Some commonly used methods include Neural Network (NN), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), Gaussian Mixture Model (GMM), etc.

Gender classification from speech signals is a captivating field of research that plays an essential role in various applications, ranging from voice assistants and telecommunications to security systems. In most previous research, classification is performed by considering various speech signal features, such as pitch, formant or a combination of both. This study performs classification using all extracted information of sound waves in the form of continuous curves/functions. The main objective is to classify female and male speakers from Lithuanian words recorded in audio files using K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF) classifiers, also evaluate which one is the most accurate and effective in predicting gender. In addition to functional KNN, SVM and RF, the same classifiers are applied to multivariate data after taking advantage of Functional Data Analysis (FDA).

To present the information clearly and comprehensibly, the work is subdivided into distinct sections. The related researches are briefly reviewed in Section 2, while Section 3 elaborates on the employed methodology with a concise mathematical background. The details regarding the used data, methods application for data transformation and smoothing, as well as the discussion of classification and testing results, are presented in Section 4. Section 5 concludes the paper and Section 6 gives some references. Appendices, included at the end of the work, contain additional detailed information of the classification results.

2 Literature review

Gender is an essential aspect of speech, and pitch serves as a fundamental feature for gender classification due to its distinction between male and female voices. Researchers have implemented classifiers using pitch extraction algorithms based on computing the short-time auto-correlation function of the speech signal.[1] The average pitch value, derived through the auto-correlation method, reveals a notable distinction between male and female voice samples. This discrepancy in pitch values serves as the foundation for a gender classifier. The operational mechanism establishes a threshold pitch value for male and female voice samples. By setting these thresholds, the gender classifier can predict the gender of the speaker within a voice signal through analysis.

While the previous method has its merits, it may not be suitable in cases where pitch alone is insufficient for accurate gender classification. In response to these challenges, other research papers propose a solution by rectifying the limitations of pitch-based methods. One research achieved it by extracting alternative features such as Mel Frequency Cepstral Coefficient (MFCC), energy entropy and frame energy from real-time male and female voices. The gender classification is then performed using advanced techniques like Artificial Neural Network (ANN) and Support Vector Machines (SVM).[2] Another research extracted three features from the speech signal, which are Mel Frequency Cepstrum Coefficient (MFCC), Linear Prediction Coding (LPC) and Linear Prediction Coding Coefficient (LPCC). While for the classification, two classifiers are used, which are Support Vector Machine (SVM) and K-Nearest Neighbour (KNN).[3] These approaches enhance the classification accuracy by considering a broader set of features beyond pitch alone.

In the realm of emotional state classification, the integration of formant features has proven instrumental. A formant-tracking algorithm was employed to meticulously extract formant-based features, setting the stage for emotion classification.[4] The study conducted a comparative analysis between formant features and a Linear Predictive Coding (LPC) based algorithm for evaluation. Results indicated that employing formant features in isolation led to a 2.1 percentage point improvement in unweighted accuracy compared to the LPC-based algorithm. Furthermore, combining formant features with other acoustic features resulted in a more substantial enhancement, achieving a 2.7 percentage point increase in accuracy. In contrast, relying solely on LPC-based features exhibited a more modest improvement, with a mere one percentage point increase.

A novel approach was presented for age classification, combining regression and classification to achieve competitive classification accuracy.[5] Support Vector Machine (SVM) regression was used to generate finer age estimates, which were combined with the posterior probabilities of well-trained discriminative gender classifiers to predict both the age and gender of a speaker. It was proven that this combination performs better than direct 7-class classifiers. The regressors and classifiers were trained using long-term features such as pitch and formants, as well as short-term (frame-based) features derived from Maximum A Posteriori (MAP) adaptation of Gaussian Mixture Models (GMMs) that were trained on Mel Frequency Cepstral Coefficient (MFCCs).

Many different methods, techniques and combinations of them have been proposed to classify a speaker's gender, age or emotional state, but there is a limited number of scientific papers specifically addressing the classification of speech signals using Functional Data Analysis (FDA). One of the newest research papers introduces an innovative approach to enhance Speech Emotion Recognition (SER) performance.[6] It involves interpreting Mel Frequency Cepstral Coefficients (MFCC) as a multivariate functional data object. The MFCCs are treated as functional data by preprocessing them as images and applying resizing techniques. This representation allows for a better understanding of the temporal dynamics of speech, capturing emotional cues more effectively. The improvement contributes significantly to the learning process of SER methods without compromising performance. The paper further applies a functional Support Vector Machine (fSVM) directly on the MFCC represented as functional data, enabling the utilization of the full functional information for more accurate emotion recognition.

3 Methodology

3.1 Hilbert transform

The Hilbert transform is a mathematical operation that widely used in signal processing to extract the envelope from modulated signals. When applied to a function, it produces a new function representing the analytic signal associated with the original one. The analytic signal has a real part, which is the original data, and an imaginary part, which contains the Hilbert transform. The imaginary part is a version of the original real sequence with a 90 degrees ($\pi/2$ radians) phase shift.

The Hilbert transform of a function $f(x)$ is defined by[7]

$$Hf(x) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(y)}{x-y} dy.$$

3.2 B-spline smoothing

A B-spline is a piecewise-defined polynomial function that is represented as a linear combination of basis functions. These basis functions are defined over local intervals, and they are connected end-to-end at specific values known as knots, breaks, or join points in a way that ensures the overall function is smooth and continuous. The B-spline basis functions are defined recursively using the Cox-de Boor recursion formula[8]

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & u < u_i, u_{i+1} \leq u \end{cases}$$
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u).$$

Here, $N_{i,p}(u)$ represent the i-th B-spline basis function of degree p and nondecreasing knot vector $U = (u_0, u_1, ..., u_{m-1}, u_m)$ defined over the parameter u.

In the context of B-spline smoothing, the goal is often to find a smooth curve that fits the given data points. This involves adjusting the positions of the control points. The influence on the B-spline curve by each control point, based on the B-spline basis functions, can be expressed as[8]

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)P_i, \quad a \leq u \leq b,$$

where $P_0, P_1, ..., P_n$ are the control points and the $N_{i,p}$ are the degree p B-spline basis functions defined on the nondecreasing $m+1$ knot vector $U = u_0, u_1, ..., u_{m-1}, u_m$ where $u_0 = u_1 = ... = u_p = a$ and $u_{m-p} = u_{m-p+1} = ... = u_m = b$.

3.3 K-nearest neighbor classification

The K-Nearest Neighbor algorithm, also known as KNN or k-NN, is a non-parametric, supervised machine learning classifier that uses proximity to make classifications or predictions about the grouping of an individual data point. KNN is a distance-based classifier, meaning that it implicitly assumes that the smaller the distance between two points, the more similar they are.

For the algorithm to perform best on a particular data set, the most appropriate distance metric must be selected accordingly. There are a lot of different distance metrics available, such as Minkowski, Manhattan, Euclidean, Cosine, Jaccard or Hamming. The most popular of these is the Euclidean distance function, which is the one used in this work. For two points $p = (p_1, p_2, ..., p_n)$ and $q = (q_1, q_2, ..., q_n)$, the Euclidean distance is calculated as[9]

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

KNN to generate a prediction for a given data point, finds the k-nearest data points and then predicts the majority class of these k points. This is often done by a simple majority voting scheme. If $C_1, C_2, ..., C_k$ are the classes of the k nearest neighbors, the predicted label y is:

$$y_{KNN} = \text{argmax}_c \left(\sum_{i=1}^{k} I(C_i = c) \right),$$

where I is the indicator function (1 if true, 0 otherwise).

3.4 Support vector machine classification

A Support Vector Machine (SVM) is a powerful machine learning algorithm that uses supervised learning models to solve complex classification problems by performing optimal data transformations that determine boundaries between data points based on predefined classes. The primary objective of SVM is to establish a hyperplane with a maximal margin, where the margin represents the distance between the hyperplane and the nearest data points of each class. This maximal margin approach not only aids in robust classification but also enhances the algorithm's generalization to new, unseen data.

Support Vector Machine is broadly classified into two types: simple or linear SVM and kernel or non-linear SVM. This research used a kernel or non-linear SVM as non-linear data can not be segregated into distinct categories with the help of a straight line. SVM with a Gaussian radial basis function (RBF) kernel, often referred to as the radial kernel SVM or RBF SVM, was selected as the best one. The RBF kernel is defined as [10]

$$K(x_i, x_j) = \exp\left(-\gamma ||x_i - x_j||^2\right).$$

Here, x_i and x_j are input data points, $||x_i - x_j||^2$ is the squared Euclidean distance, and $\gamma > 0$ is a parameter controlling the kernel width.

The decision function for SVM is expressed as[11]

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b\right).$$

In this formula, $f(x)$ is the decision function for a given input x, α_i are Lagrange multipliers, y_i is the class label, x_i is a training example and b is the bias term. The decision is made based on the sign of $f(x)$: if $f(x) > 0$, the input is classified as one class, if $f(x) < 0$, the input is classified as the other class.

3.5 Random forest classification

Random Forest is a supervised machine learning algorithm which is built upon the foundation of decision trees. The decision tree algorithm recursively splits the data based on feature thresholds to create a tree-like structure. The key components of a decision tree are the splitting criteria and the leaf node predictions.

Random Forest builds each tree on a different subset of the training data through bootstrapping. At each node of a decision tree, only a random subset of features is considered for splitting. This helps in decorrelating the trees and improving generalization. The number of features to consider at each split is often denoted as parameter m and is typically the square root of the total number of features.

The final prediction of the Random Forest is obtained through a majority vote. Each tree in the forest "votes" for a class, and the class with the most votes is the predicted class for a given input

$$y_{RF} = \mathrm{argmax_c} \left(\sum_{i=1}^{N_{trees}} I(y_{tree_i} = c) \right).$$

Here, y_{RF} is the predicted class by the Random Forest, N_{trees} is the number of trees in the forest, y_{tree_i} is the predicted class by the i-th tree, and I is the indicator function (1 if true, 0 otherwise).

3.6 Friedman test

The Friedman test is the non-parametric alternative to the one-way ANOVA with repeated measures. It tests whether the k paired samples ($k > 2$) of n size, are from the same population or the samples from populations having similar properties, considering the position parameter. In simple terms, this test helps determine whether there are any differences in the central tendencies (typically medians) of related groups.

When conducting a Friedman test, the null hypothesis (H_0) involves comparing the differences between the medians and predicts that there is no difference in the distribution of the dependent variable among the groups. In other words, the medians of the groups are equal. The alternative hypothesis (H_1) then states that at least two groups have different distributions, indicating a statistically significant difference among the medians of related groups.

$$\begin{aligned} &H_0 : \eta_1 = \eta_2 = ... = \eta_k \\ &H_1 : \exists\ i, j : \eta_i \neq \eta_j, \qquad \text{where } i \neq j \text{ and } i, j = 1, 2, ..., k. \end{aligned}$$

The Friedman test statistics is used to determine whether to support or reject the null hypothesis and is computed, comparing the mean ranks across groups[12]

$$\chi_r^2 = \frac{12n}{k(k+1)} \sum_{j=1}^{k} \left(\bar{R}_j - \frac{1}{2}(k+1) \right)^2,$$

where $\bar{R}_j$ is the sum of the ranks for sample j, n is the number of independent blocks and k is the number of groups or treatment levels.

3.7 T-test

A t-test, also known as Student's t-test, is used to evaluate whether a single group differs from a known value (a one-sample t-test), whether two groups differ from each other (an independent two-sample t-test) or whether there is a significant difference in paired measurements (a paired, dependent samples, or correlated t-test). In this study, a paired t-test was used to determine whether there was a statistically significant difference between the mean values of two dependent groups.

When conducting a paired t-test, the null hypothesis (H_0) involves comparing the mean difference μ_d to a hypothesized constant μ_0. In many cases, this constant is set to zero, especially when the objective is to test whether the mean difference is significantly different from zero. The alternative hypothesis (H_1) then states that there is a significant difference between the means of related groups and the hypothesized constant.

$$H_0 : \mu_d = \mu_0$$
$$H_1 : \mu_d \neq \mu_0.$$

The t-statistic, also known as t-value or t-score, is used in a t-test to determine whether to support or reject the null hypothesis. The formula for calculating the t-statistic in a paired t-test is as follows[13]:

$$t = \frac{\bar{X}_d - \mu_0}{s_d/\sqrt{n}}$$

Here, $\bar{X}_d$ and s_d are the average and standard deviation of the differences between all pairs, n represents the number of pairs, the constant μ_0 is typically set to zero when testing whether the average of the differences is significantly different.

4 Analysis

4.1 Data set

A data set used for the research consists of 111 different Lithuanian words, which were collected by the Image and Signal Analysis group of the Data Science and Digital Technologies Institute. After data cleaning, a set of 70 different Lithuanian words remained (bolded). A list of words is given in the table below.

1.	**Būti**	29.	Dalis	**57.**	**Dažnai**	85.	Viskas
2.	**Kuris**	**30.**	**Įstatymas**	**58.**	**Skirti**	86.	Tyrimas
3.	**Galėti**	**31.**	**Straipsnis**	**59.**	**Veikla**	87.	Vanduo
4.	**Visas**	**32.**	**Įmonė**	60.	Eiti	88.	Matyti
5.	Kaip	33.	Žodis	61.	Atlikti	89.	Grupė
6.	**Lietuva**	**34.**	**Norėti**	62.	Pasakyti	90.	Priemonė
7.	Kitas	**35.**	**Kalba**	63.	Gyventi	91.	Vyriausybė
8.	**Turėti**	**36.**	**Šalis**	64.	Priimti	92.	Būdas
9.	**Savas**	**37.**	**Sudaryti**	**65.**	**Valstybinis**	93.	Naudoti
10.	**Darbas**	38.	Asmuo	**66.**	**Mokslas**	94.	Medžiaga
11.	**Žmogus**	**39.**	**Naujas**	**67.**	**Akis**	**95.**	**Nors**
12.	**Metai**	**40.**	**Sistema**	**68.**	**Geras**	**96.**	**Procesas**
13.	**Labai**	41.	Sakyti	69.	Atvejis	**97.**	**Pasaulis**
14.	**Vienas**	**42.**	**Todėl**	**70.**	**Dirbti**	98.	Ūkis
15.	**Nebūti**	43.	Kartas	**71.**	**Antras**	99.	Kiek
16.	**Reikėti**	**44.**	**Gauti**	**72.**	**Mažas**	**100.**	**Rašyti**
17.	**Žinoti**	**45.**	**Áukštas**	**73.**	**Miestas**	**101.**	**Nulis**
18.	**Didelis**	**46.**	**Žemė**	74.	Ranka	**102.**	**Du**
19.	**Tačiau**	47.	Metas	**75.**	**Bendras**	**103.**	**Trys**
20.	**Teisė**	48.	Vieta	76.	Įstaiga	**104.**	**Keturi**
21.	**Laikas**	49.	Niekas	**77.**	**Mokykla**	**105.**	**Penki**
22.	**Diena**	**50.**	**Įvairus**	**78.**	**Teismas**	**106.**	**Šeši**
23.	Dabar	51.	Lietuviai	79.	Kalbėti	**107.**	**Septyni**
24.	Pagal	**52.**	**Svarbus**	**80.**	**Forma**	108.	Aštuoni
25.	Valstybė	**53.**	**Vaikas**	**81.**	**Bankas**	109.	Devyni
26.	**Jeigu**	**54.**	**Gerai**	**82.**	**Tada**	110.	Pradžia
27.	**Respublika**	55.	Prieš	**83.**	**Kultūra**	111.	Pabaiga
28.	**Nustatyti**	56.	Tarp	**84.**	**Sąlyga**		

Table 4.1.1: Data set of Lithuanian words

Each word was recorded in audio files (WAV or Waveform Audio File Format) by 36 women and 26 men repeating it 10 times. In addition to 10 original files (without noise or 0 dB), there are audio files with added background/noise in different loudness (15 dB, 20 dB, 25 dB and 30 dB). In total, there are 50 audio files per word, per speaker, 3100 per word, per all speakers and 217000 per all words, per all speakers.

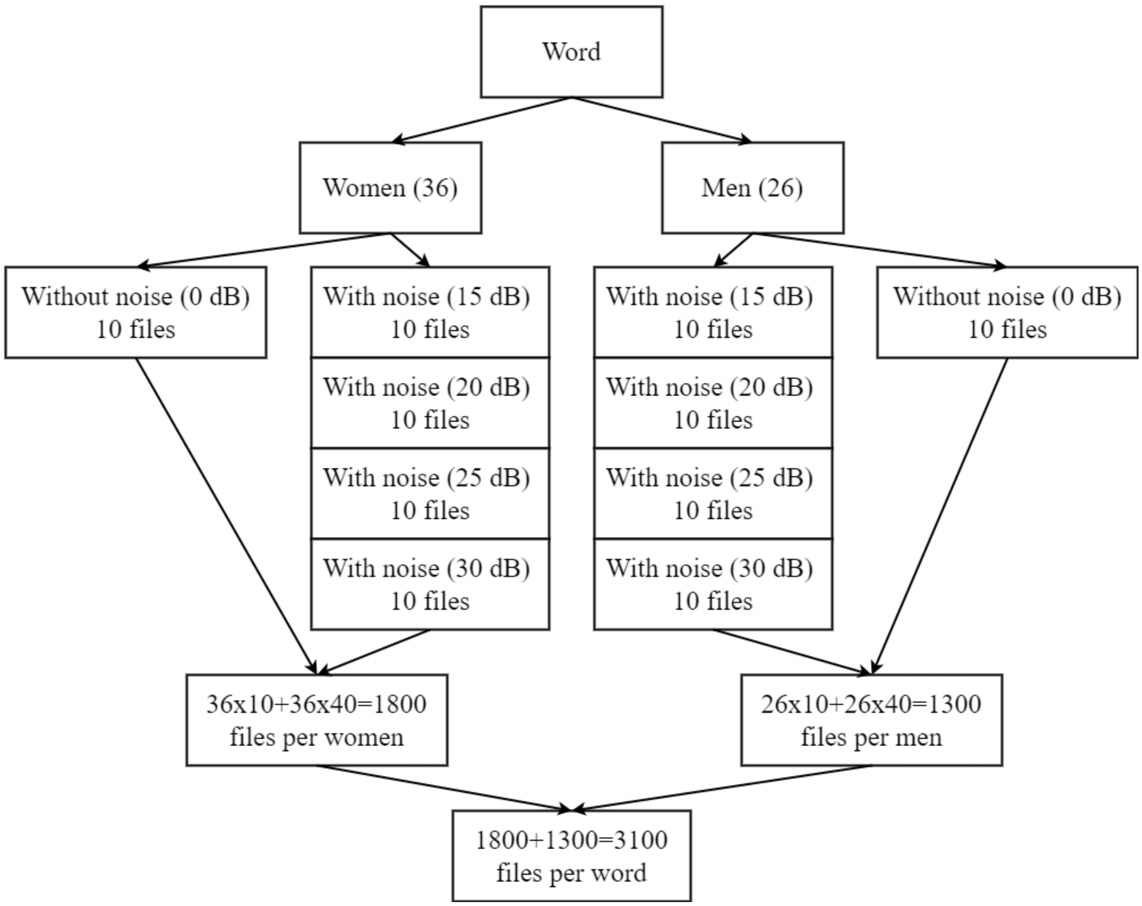

Figure 4.1.1: Scheme of word

It was decided to randomly select 20 different words for further work. All audio files for each word were taken, i.e., all speakers' sessions without and with added noise. The final data set consists of 62000 files. A list of selected words is given in the table below.

1.	Kuris	11.	Valstybinis
2.	Lietuva	12.	Mokslas
3.	Darbas	13.	Dirbti
4.	Metai	14.	Forma
5.	Vienas	15.	Kultūra
6.	Diena	16.	Sąlyga
7.	Šalis	17.	Procesas
8.	Žemė	18.	Du
9.	Įvairus	19.	Šeši
10.	Gerai	20.	Septyni

Table 4.1.2: Final data set

4.2 Data transformation

Each speech signal contains a wealth of information. To extract it, the Hilbert transform was applied. Using the function `env()` (package "`seewave`"), the amplitude envelope was returned as the modulus of the analytical signal of a wave obtained through the Hilbert transform. This amplitude envelope provides a valuable representation of the signal's variations over time, capturing the underlying modulations in amplitude. It also enhances the ability to discern key features and patterns within the speech signal, contributing to a more nuanced understanding of its characteristics, as well as offering a structured and accessible format for subsequent analysis.

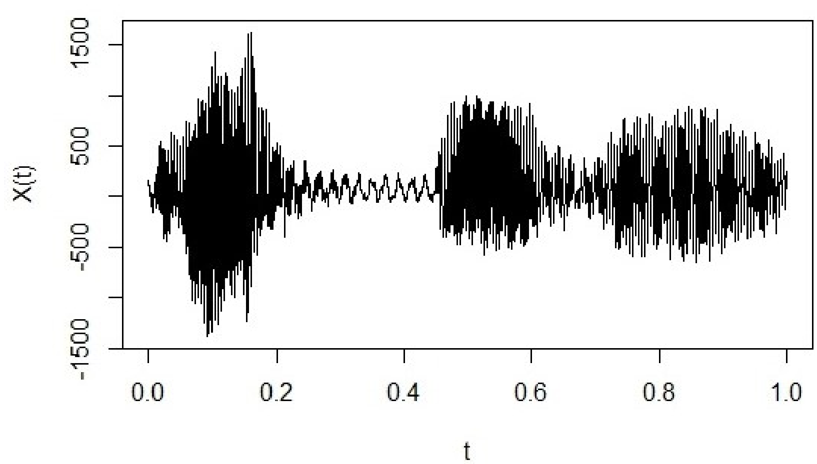

Figure 4.2.1: Original speech signal

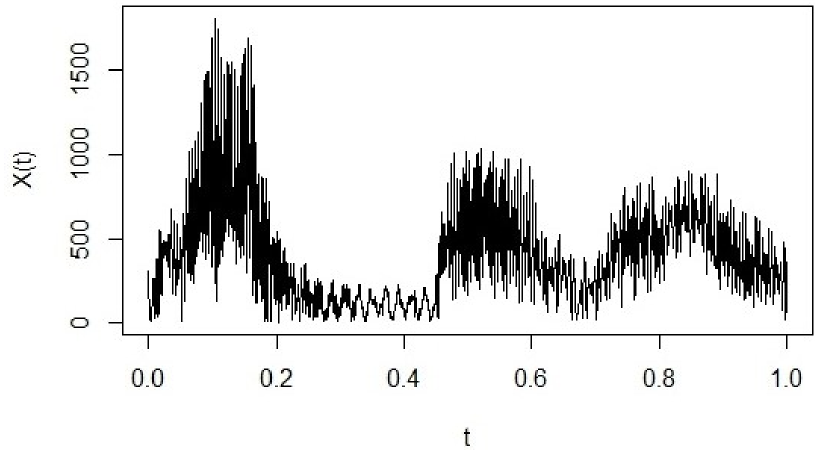

Figure 4.2.2: Speech signal with the Hilbert transform

4.3 Data smoothing

Data smoothing was performed using the B-spline method. A total of 20 women and 20 men speech signals were randomly selected from any session to determine the smoothing parameters for each word. The optimal parameters were selected based on the minimum generalized cross-validation or GCV criterion. In each case, the final parameter was obtained by calculating the median of the 20 optimal parameters. The number of basis functions ranges from 30 to 49, as indicated in the table below, while $\lambda = 0$ (roughness penalty) was the optimal value for all words.

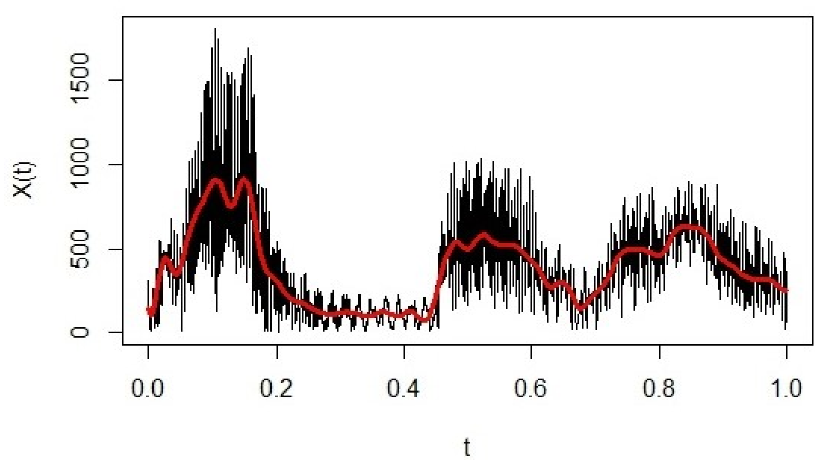

Figure 4.3.1: Speech signal after smoothing (red line)

Word	Nbasis (women)	Nbasis (men)
Lietuva	49	48
Kultūra	49	47
Vienas	48	47
Žemė	48	38
Šeši	48	48
Darbas	48	48
Diena	47	34
Metai	46	40
Forma	48	46
Gerai	48	42
Dirbti	47	48
Procesas	48	48
Du	44	30
Įvairus	49	48
Kuris	48	48
Šalis	48	48
Valstybinis	49	49
Mokslas	48	46
Septyni	48	48
Sąlyga	47	48

Table 4.3.1: Smoothing parameters

4.4 Data classification

Since the final data set is very large (62000 speech signals) and therefore, not all classifiers can handle such an amount of data, classification was done in segments. The data set was divided into five parts based on noise (0 dB, 15 dB, 20 dB, 25 dB and 30 dB), where each part contains 12400 speech signals. Those were split into training and testing sets using a 3:1 ratio. In all cases, the classification was carried out into two classes, predicting between female and male speakers.

The following tables present the performance indicators of K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF) classifiers applied to all 20 words. Each classifier was applied to multivariate and functional data respectively. The indicators include accuracy, sensitivity, specificity and the Youden index. For every case, accuracy, sensitivity, and specificity were calculated by averaging the corresponding indicators derived from the noise-based segments of the classification for each word.

Accuracy measures the overall correctness of a classifier by considering both true positives (correctly identified positive instances) and true negatives (correctly identified negative instances). It gives a general sense of how well the classifier is performing across all classes. Accuracy is calculated using the formula

$$Accuracy = \frac{True\,Positives + True\,Negatives}{Total\,Population}.$$

Sensitivity measures the ability of a classifier to correctly identify positive instances among all actual positive instances. It is important when the cost of missing positive instances (false negatives) is high. Sensitivity is calculated using the formula

$$Sensitivity = \frac{True\,Positives}{True\,Positives + False\,Negatives}.$$

Specificity measures the ability of a classifier to correctly identify negative instances among all actual negative instances. It is important when the cost of missing negative instances (false positives) is high. Specificity is calculated using the formula

$$Specificity = \frac{True\,Negatives}{True\,Negatives + False\,Positives}.$$

The Youden Index, also known as the Youden's J statistic, is a metric used to assess the overall performance of a classifier. It is calculated using sensitivity and specificity. Mathematically, the Youden Index (J) is expressed as

$$J = Sensitivity + Specificity - 1.$$

The Youden Index ranges from 0 to 1. A value of 0 indicates that the test is no better than random, while a value of 1 indicates perfect performance.

A more detailed breakdown of classification results from the noise-based segments can be found in the Appendices (Classification tables).

4.4.1 K-Nearest Neighbor

The K-Nearest Neighbor classification of multivariate data was carried out using the `knn()` function from the `"class"` package. All segments were classified in about 15 minutes. The table reveals a diversity of performance across different words. The indicator values exhibit a range, with accuracy spanning from 0.743 to 0.836, sensitivity - from 0.811 to 0.891, specificity - from 0.655 to 0.789 and the Youden Index fluctuating between 0.498 and 0.664. While, the word *Vienas* stands out with the highest overall performance, achieving an accuracy of 0.836, sensitivity of 0.887, specificity of 0.777 and Youden Index of 0.664, other words such as *Dirbti*, *Šeši* and *Mokslas* also exemplify good classification. Conversely, the word *Du* appears to have relatively lower performance across all indicators, with an accuracy of 0.743, sensitivity of 0.843, specificity of 0.655 and a Youden Index of 0.498.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	0.787	0.844	0.725	0.569
Kultūra	0.792	0.857	0.722	0.579
Vienas	**0.836**	**0.887**	**0.777**	**0.664**
Žemė	0.766	0.811	0.711	0.522
Šeši	0.818	0.878	0.752	0.630
Darbas	0.801	0.857	0.738	0.595
Diena	0.769	0.864	0.682	0.546
Metai	0.804	0.853	0.746	0.599
Forma	0.808	0.821	0.789	0.610
Gerai	0.783	0.858	0.706	0.564
Dirbti	0.822	0.873	0.764	0.637
Procesas	0.786	0.852	0.715	0.567
Du	**0.743**	**0.843**	**0.655**	**0.498**
Įvairus	0.795	0.852	0.732	0.584
Kuris	0.752	0.818	0.681	0.499
Šalis	0.795	0.833	0.746	0.579
Valstybinis	0.795	0.831	0.748	0.579
Mokslas	0.814	0.891	0.737	0.628
Septyni	0.774	0.837	0.706	0.543
Sąlyga	0.785	0.842	0.720	0.562

Table 4.4.1: K-Nearest Neighbor classification of multivariate data (mKNN)

The K-Nearest Neighbor classification of functional data, performed using the function `classif.knn()` (`"fda.usc"` package), yielded results similar to those obtained through multivariate data. However, the classification time was much longer - all segments were classified within 800 minutes. The indicator values span a range as follows: accuracy from 0.734 to 0.823, sensitivity from 0.732 to 0.845, specificity from 0.672 to 0.849 and Youden index from 0.473 to 0.653. The word *Vienas* was the second best performance and the word *Šeši* demonstrates the highest overall performance across all indicators. It achieves the highest accuracy of 0.823, sensitivity of 0.823, specificity of 0.830 and Youden Index of 0.653. The word *Du* once again appears to have relatively lower performance across all indicators, with accuracy of 0.734, sensitivity of 0.801, specificity of 0.672 and a Youden Index of 0.473.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	0.794	0.790	0.806	0.596
Kultūra	0.782	0.835	0.721	0.556
Vienas	0.819	0.832	0.807	0.639
Žemė	0.750	0.732	0.798	0.530
Šeši	**0.823**	**0.823**	**0.830**	**0.653**
Darbas	0.799	0.841	0.750	0.591
Diena	0.801	0.807	0.793	0.600
Metai	0.796	0.811	0.782	0.593
Forma	0.810	0.799	0.836	0.635
Gerai	0.799	0.798	0.801	0.599
Dirbti	0.813	0.841	0.780	0.621
Procesas	0.783	0.790	0.784	0.574
Du	**0.734**	**0.801**	**0.672**	**0.473**
Įvairus	0.812	0.794	0.849	0.643
Kuris	0.761	0.764	0.767	0.531
Šalis	0.785	0.793	0.778	0.571
Valstybinis	0.778	0.804	0.745	0.549
Mokslas	0.809	0.840	0.781	0.621
Septyni	0.779	0.792	0.763	0.555
Sąlyga	0.782	0.845	0.716	0.561

Table 4.4.2: K-Nearest Neighbor classification of functional data (fKNN)

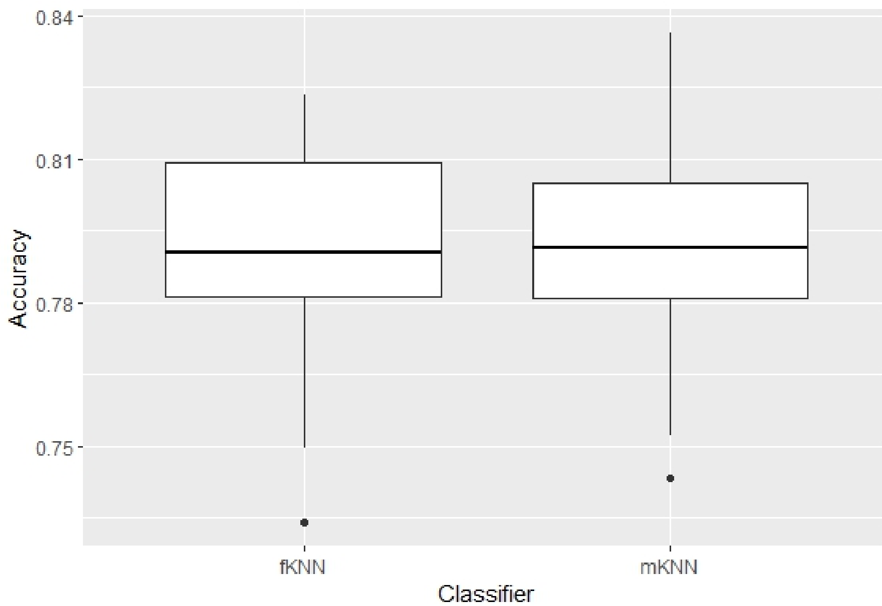

Figure 4.4.1: K-Nearest Neighbor classification

4.4.2 Support Vector Machine

The Support Vector Machine classification of multivariate data was performed using the `svm()` function from the `"e1071"` package. The classification time for this classifier was the longest at around 815 minutes for all segments. The results of SVM classifier differ slightly from those of KNN classifier. In this context, accuracy ranges from 0.739 to 0.854, sensitivity - from 0.763 to 0.902, specificity - from 0.696 to 0.852 and Youden index - from 0.468 to 0.702. The word *Mokslas* got the best classification outcome with an accuracy of 0.854, sensitivity of 0.902, specificity of 0.800 and Youden index of 0.702. Words *Dirbti*, *Šeši* and *Vienas* can also be examples of good classification. In contrast, the word *Žemė* was classified as the worst and the values of the indicators were distributed as follows: accuracy 0.739, sensitivity 0.763, specificity 0.705 and Youden index 0.468.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	0.806	0.838	0.765	0.603
Kultūra	0.814	0.847	0.772	0.619
Vienas	0.837	0.867	0.799	0.666
Žemė	**0.739**	**0.763**	**0.705**	**0.468**
Šeši	0.846	0.877	0.809	0.686
Darbas	0.818	0.863	0.763	0.626
Diena	0.787	0.832	0.739	0.571
Metai	0.797	0.814	0.772	0.586
Forma	0.764	0.788	0.730	0.518
Gerai	0.827	0.816	0.851	0.667
Dirbti	0.850	0.849	0.852	0.701
Procesas	0.775	0.791	0.751	0.542
Du	0.775	0.835	0.711	0.546
Įvairus	0.830	0.833	0.825	0.658
Kuris	0.792	0.849	0.730	0.579
Šalis	0.836	0.842	0.829	0.671
Valstybinis	0.832	0.831	0.838	0.669
Mokslas	**0.854**	**0.902**	**0.800**	**0.702**
Septyni	0.765	0.827	0.696	0.523
Sąlyga	0.775	0.792	0.761	0.553

Table 4.4.3: Support Vector Machine classification of multivariate data (mSVM)

The Support Vector Machine classifier for functional data, executed using the function `classif.svm()` (`"fda.usc"` package), demonstrates slightly lower indicator values compared with those obtained through multivariate data. Nevertheless, classification took much less time - about 155 minutes for all segments. The indicators display a spectrum of values: accuracy varies from 0.676 to 0.787, sensitivity - from 0.689 to 0.805, specificity - from 0.643 to 0.825 and Youden index - from 0.339 to 0.576. The words *Mokslas* and *Darbas* both achieved the best accuracy value of 0.787. For *Mokslas*, the other indicators are as follows: sensitivity of 0.805, specificity of 0.771, and a Youden index of 0.576. Similarly, *Darbas* demonstrates the following indicators: sensitivity of 0.795, specificity of 0.775, and a Youden index of 0.570. In this case, the word *Du* again became the worst classified with an accuracy of 0.676, sensitivity of 0.689, specificity of 0.650 and Youden index of 0.339.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	0.761	0.754	0.780	0.534
Kultūra	0.750	0.746	0.759	0.505
Vienas	0.759	0.764	0.754	0.518
Žemė	0.732	0.731	0.747	0.478
Šeši	0.785	0.804	0.756	0.560
Darbas	**0.787**	**0.795**	**0.775**	**0.570**
Diena	0.733	0.756	0.700	0.456
Metai	0.747	0.768	0.714	0.482
Forma	0.721	0.730	0.706	0.436
Gerai	0.772	0.760	0.797	0.557
Dirbti	0.735	0.735	0.740	0.475
Procesas	0.741	0.732	0.764	0.496
Du	**0.676**	**0.689**	**0.650**	**0.339**
Įvairus	0.764	0.758	0.776	0.534
Kuris	0.690	0.719	0.643	0.362
Šalis	0.768	0.764	0.778	0.542
Valstybinis	0.745	0.748	0.738	0.486
Mokslas	**0.787**	**0.805**	**0.771**	**0.576**
Septyni	0.738	0.747	0.726	0.473
Sąlyga	0.748	0.723	0.825	0.548

Table 4.4.4: Support Vector Machine classification of functional data (fSVM)

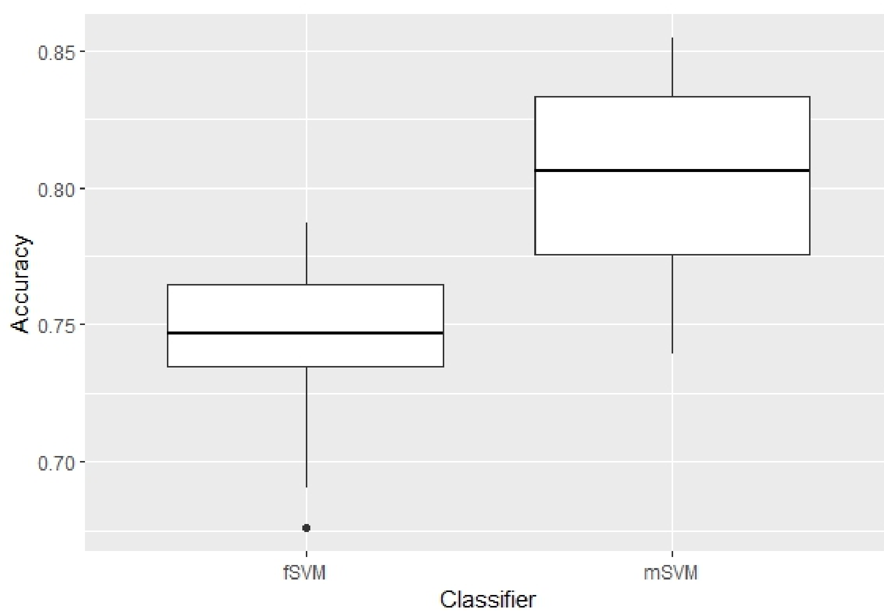

Figure 4.4.2: Support Vector Machine classification

4.4.3 Random Forest

The Random Forest classifier for multivariate data, carried out using the `randomForest()` function from the `"randomForest"` package, achieved the highest classification results among all classifications. Despite that, the classification time was long enough - about 625 minutes for all segments. The indicator values exhibit a range, with accuracy spanning from 0.766 to 0.861, sensitivity - from 0.770 to 0.875, specificity - from 0.712 to 0.908 and the Youden Index fluctuating between 0.523 and 0.721. While, the word *Lietuva* stands out with the highest overall performance, achieving an accuracy of 0.861, sensitivity of 0.875, specificity of 0.841 and Youden Index of 0.716, other words such as *Mokslas*, *Darbas* and *Įvairus* also exemplify good classification. Contrariwise, the word *Du* appears to have comparatively lower performance across all indicators, with an accuracy of 0.766, sensitivity of 0.811, specificity of 0.712 and a Youden Index of 0.523.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	**0.861**	**0.875**	**0.841**	**0.716**
Kultūra	0.834	0.806	0.893	0.699
Vienas	0.834	0.818	0.863	0.681
Žemė	0.774	0.776	0.771	0.547
Šeši	0.840	0.825	0.869	0.694
Darbas	0.858	0.862	0.856	0.718
Diena	0.800	0.782	0.839	0.621
Metai	0.844	0.854	0.830	0.684
Forma	0.781	0.770	0.806	0.576
Gerai	0.827	0.816	0.852	0.668
Dirbti	0.846	0.826	0.885	0.711
Procesas	0.832	0.819	0.857	0.676
Du	**0.766**	**0.811**	**0.712**	**0.523**
Įvairus	0.850	0.830	0.888	0.718
Kuris	0.788	0.807	0.766	0.573
Šalis	0.848	0.835	0.871	0.706
Valstybinis	0.831	0.798	0.908	0.706
Mokslas	0.859	0.856	0.865	0.721
Septyni	0.803	0.788	0.833	0.621
Sąlyga	0.836	0.818	0.874	0.692

Table 4.4.5: Random Forest classification of multivariate data (mRF)

The Random Forest classifier for functional data, employed with the function `classif.randomForest()` ("fda.usc" package), demonstrates lower indicator values compared with those obtained through multivariate data. However, the classifier performed the task more efficiently, completing the classification of all segments in a shorter time frame of around 20 minutes. The indicators display a spectrum of values: accuracy varies from 0.694 to 0.787, sensitivity - from 0.717 to 0.817, specificity - from 0.641 to 0.796 and Youden index - from 0.374 to 0.575. Among the words, *Šalis* attained the highest accuracy value at 0.787, closely followed by *Darbas* and *Gerai*, both achieving an accuracy of 0.785. For *Šalis*, the other indicators are as follows: sensitivity of 0.816, specificity of 0.748, and a Youden index of 0.564. Meanwhile, the word *Du* again became the worst classified with an accuracy of 0.694, sensitivity of 0.733, specificity of 0.641 and Youden index of 0.374.

Word	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	0.742	0.746	0.735	0.481
Kultūra	0.759	0.760	0.763	0.523
Vienas	0.763	0.784	0.733	0.517
Žemė	0.725	0.730	0.715	0.445
Šeši	0.781	0.817	0.739	0.556
Darbas	0.785	0.806	0.754	0.560
Diena	0.766	0.757	0.786	0.543
Metai	0.741	0.757	0.714	0.471
Forma	0.712	0.717	0.706	0.423
Gerai	0.785	0.779	0.796	0.575
Dirbti	0.733	0.730	0.739	0.469
Procesas	0.719	0.720	0.720	0.440
Du	**0.694**	**0.733**	**0.641**	**0.374**
Įvairus	0.760	0.766	0.751	0.517
Kuris	0.716	0.756	0.662	0.418
Šalis	**0.787**	**0.816**	**0.748**	**0.564**
Valstybinis	0.738	0.734	0.747	0.481
Mokslas	0.777	0.773	0.784	0.557
Septyni	0.765	0.761	0.774	0.535
Sąlyga	0.748	0.748	0.751	0.499

Table 4.4.6: Random Forest classification of functional data (fRF)

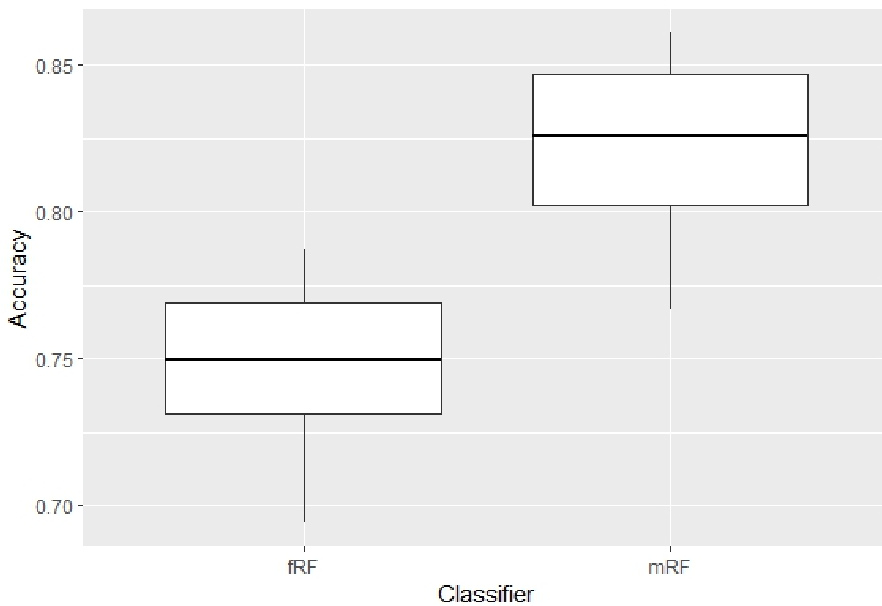

Figure 4.4.3: Random Forest classification

4.4.4 Total classification

The final classification results were determined by averaging the corresponding indicators across all words and all noise-based segments of the classification. The best indicator values for women and men classification were achieved by Random Forest classifier for multivariate data. It is noteworthy that the K-Nearest Neighbor classifier yielded nearly identical overall results for both the multivariate data and the functional data. Meanwhile, Support Vector Machine and Random Forest classifiers present strong overall performance on multivariate data, but their effectiveness seems to diminish when applied to functional data.

Classifier	Accuracy	Sensitivity	Specificity	Youden Index
mKNN	0.791	0.850	0.728	0.578
fKNN	0.790	0.806	0.778	0.584
mSVM	0.806	0.833	0.775	0.608
fSVM	0.747	0.751	0.745	0.496
mRF	**0.826**	**0.819**	**0.844**	**0.663**
fRF	0.750	0.760	0.738	0.498

Table 4.4.7: Total classification

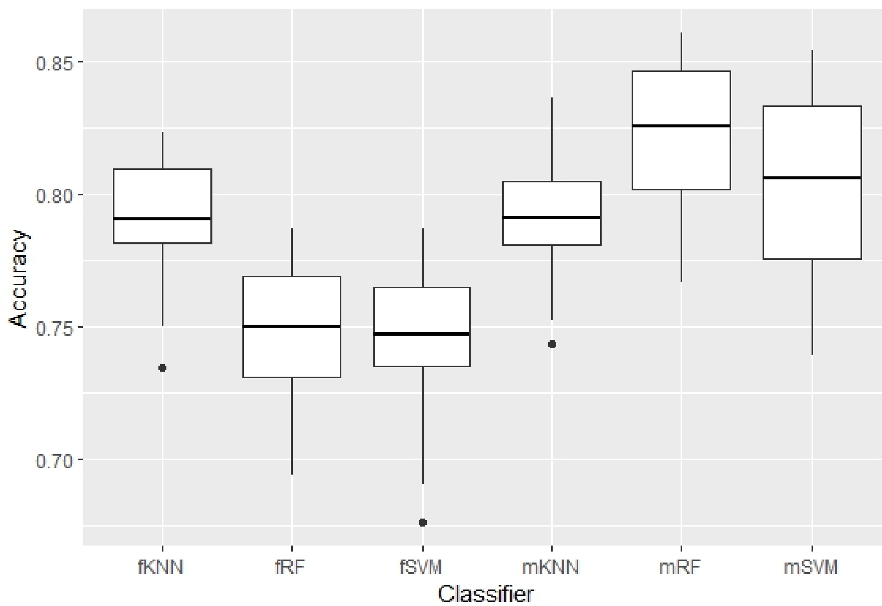

Figure 4.4.4: Total classification

4.5 Tests

In order to conclude classification results, Friedman and t-tests were applied to assess the statistical significance of the performance differences among the employed classifiers.

4.5.1 Friedman test

The Friedman test was applied to ascertain whether a statistically significant difference exists among the accuracy medians of noise-based (0 dB, 15 dB, 20 dB, 25 dB, and 30 dB) segments. In other words, the question is whether or not added noise to speech signals affects the classification accuracy of women and men. The null hypothesis states that the accuracy medians of classification are the same regardless of the added noise to speech signals, while the alternative hypothesis assumes, that there is a statistically significant difference between accuracy medians of classification when the noise to speech signals is added.

$$H_0 : \eta_{0dB} = \eta_{15dB} = \eta_{20dB} = \eta_{25dB} = \eta_{30dB}$$
$$H_1 : \exists\ i, j : \eta_i \neq \eta_j, \qquad \text{where } I \neq j \text{ and } I, j = 0dB, 15dB, 20dB, 25dB, 30dB.$$

Classifier	Chi-square	df	p-value
mKNN	3.008	4	0.557
fKNN	8.989	4	0.061
mSVM	22.994	4	0.000
fSVM	6.016	4	0.198
mRF	26.670	4	0.000
fRF	24.332	4	0.000

Table 4.5.1: Friedman rank sum test

The test outcomes indicate that the *p-value* is less than 0.05 for the three classifiers, signifying statistical significance, while for the remaining classifiers, the *p-value* exceeds 0.05, indicating a lack of statistical significance. Because the *p-value* for mSVM, mRF and fRF is less than the significance level of 0.05, the null hypothesis is rejected and it concludes that at least 1 of 5 parts based on noise has a different classification. On the other hand, the *p-value* for mKNN, fKNN and fSVM is greater than the significance level of 0.05, therefore the null hypothesis is accepted and it can be said that added noise to speech signals did not affect the classification accuracy of women and men.

4.5.2 T-test

A paired t-test was used to determine whether there is a statistically significant difference between the accuracy means of two classifiers, i.e., whether two classifiers classify women and men equally well. The null hypothesis is that both classifiers' accuracy means are equal, while the alternative hypothesis assumes, that there is a statistically significant difference between the accuracy means of the two classifiers.

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2$$

Firstly, a paired t-test was performed among all the same classifiers for multivariate and for functional data. The test results indicate that the *p-value* exceeds 0.05 only for classifiers mKNN versus fKNN. Consequently, the null hypothesis is accepted and it can be said that there is no statistically significant difference between these classifiers and they classify women and men equally well. The *p-value* is found to be less than 0.05 in the comparisons of mSVM versus fSVM and mRF versus fRF. This indicates a statistically significant difference between the accuracy means of these classifiers, leading to the conclusion that their classification performances are indeed unequal.

Classifier	t	df	p-value	95% CI	Mean difference
mKNN vs. fKNN	-0.296	19	0.771	-0.0068 to 0.0051	-0.0008
mSVM vs. fSVM	-9.550	19	0.000	-0.0723 to -0.0463	-0.0593
mRF vs. fRF	-13.602	19	0.000	-0.0876 to -0.0642	-0.0759

Table 4.5.2: Paired t-test

Secondly, a paired t-test was executed between all different classifiers for multivariate data. Across all cases, the *p-value* was found to be less than 0.05, leading to the rejection of the null hypothesis and indicating a statistically significant difference. Hence, it can be asserted that all classifiers for multivariate data, when compared with each other, yield distinct performance outcomes.

Classifier	t	df	p-value	95% CI	Mean difference
mKNN vs. mSVM	-2.642	19	0.016	-0.0267 to -0.0031	-0.0149
mKNN vs. mRF	-6.746	19	0.000	-0.0450 to -0.0237	-0.0343
mSVM vs. mRF	-3.740	19	0.001	-0.0303 to -0.0086	-0.0194

Table 4.5.3: Paired t-test

Thirdly, a paired t-test was carried out across all different classifiers for functional data. The test outcomes reveal that the *p-value* exceeds 0.05 only for classifiers fSVM versus fRF. Hence, the null hypothesis is accepted, suggesting no statistically significant difference between these classifiers, indicating equally well classification of women and men. On the other hand, the *p-value* is found to be less than 0.05 in the comparisons of fKNN versus fSVM and fKNN versus fRF. This indicates a statistically significant difference between the accuracy means of these classifiers, leading to the conclusion that their classification performances are indeed unequal.

Classifier	t	df	p-value	95% CI	Mean difference
fKNN vs. fSVM	9.079	19	0.000	0.0335 to 0.0536	0.0435
fKNN vs. fRF	7.620	19	0.000	0.0295 to 0.0519	0.0407
fSVM vs. fRF	-0.821	19	0.422	-0.0101 to 0.0044	-0.0028

Table 4.5.4: Paired t-test

5 Conclusions

It is noticeable that all classifiers found it more difficult to predict the gender of the speaker from the speech signal when the word is very short. The main reason for this may be that a short sound wave contains less information about the speaker, making it harder for classifiers to distinguish the difference between men and women. The word *Du* has the lowest accuracy rate across all classifications except for Support Vector Machine (SVM) in multivariate data. In contrast, longer words such as *Darbas*, *Dirbti*, *Mokslas*, *Šeši* and *Vienas* consistently achieved high accuracy rates across all classifiers, making them top-classified words in predicting male and female speakers from the speech signals.

To summarise the accuracy results, the Random Forest (RF) classifier for multivariate data is the most efficient in this case. It achieved 82.60 % accuracy in predicting the gender of the speaker. Overall, higher accuracy rates were achieved with classifiers for multivariate data than with the same classifiers for functional data. However, even though these classifiers are more efficient due to their higher accuracy, the main disadvantage of them often lies in their prolonged running time.

Functional SVM and RF classifiers outperformed multivariate ones, operating more than 5 and 30 times faster, respectively. Conversely, in the K-Nearest Neighbor (KNN) classification, the scenario is reversed - functional KNN demonstrated a runtime more than 50 times slower than its multivariate equivalent. Thus, when summarising the classifiers in terms of running time, it is important to stress that in this case, the classifier for multivariate data was both the fastest (KNN with around 15 minutes) and the slowest (SVM with around 815 minutes).

In terms of potential avenues for further research, other data transformations of speech signals can be applied to improve the extraction of relevant audio features. Also, possible consideration of other classifiers beyond KNN, SVM and RF that might provide improved results in classifying female and male speakers from speech signals.

6 References

[1] B. L. Jena, B. P. Panigrahi. Gender Classification by Pitch Analysis. (2012)

[2] G. S. Archana, M. Malleswari. Gender Identification and Performance Analysis of Speech Signals. (2015)

[3] N. A. Nazifa, C. Y. Fook, L. C. Chin, V. Vijean, E. S. Kheng. Gender Prediction by Speech Analysis. (2019)

[4] J. C. Kim, M. A. Clements. Formant-based Feature Extraction for Emotion Classification from Speech. (2015)

[5] C. van Heerden, E. Barnard, M. Davel, C. van der Walt, E. van Dyk, M. Feld, C. Müller. Combining Regression and Classification Methods for Improving Automatic Speaker Age Recognition. (2010)

[6] M. Saumard. Enhancing Speech Emotions Recognition using Multivariate Functional Data Analysis. (2023)

[7] F. R. Kschischang. The Hilbert Transform. (2006)

[8] L. Piegl, W. Tiller. The NURBS Book. 2nd ed. (1997)

[9] J. Tabak. Geometry: The Language of Space and Form. (2014)

[10] J. P. Vert, K. Tsuda, B. Schölkopf. A Primer on Kernel Methods. (2004)

[11] S. Busuttil. Support Vector Machines. (2003)

[12] W. J. Conover. Practical Nonparametric Statistics. 3rd ed. (1999)

[13] T. K. Kim. T test as a Parametric Statistic. (2015)

[14] F. Rossi, N. Villa. Support Vector Machine for Functional Data Classification. (2007)

[15] K. Fuchs, J. Gertheiss, G. Tutz. Nearest Neighbor Ensembles for Functional Data with Interpretable Feature Selection. (2015)

[16] B. Gregorutti, B. Michel, P. Saint-Pierre. Grouped Variable Importance with Random Forests and Application to Multiple Functional Data Analysis. (2015)

[17] F. Maturo, R. Verde. Combining Unsupervised and Supervised Learning Techniques for Enhancing the Performance of Functional Data Classifiers. (2022)

A Appendices

A.1 Classification tables

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	3	0.826	0.839	0.806	0.645
Kultūra	2	0.800	0.864	0.730	0.594
Vienas	2	0.865	0.926	0.797	0.723
Žemė	2	0.800	0.873	0.724	0.597
Šeši	6	0.794	0.863	0.720	0.583
Darbas	3	0.716	0.788	0.640	0.428
Diena	6	0.723	0.783	0.653	0.436
Metai	2	0.832	0.856	0.800	0.656
Forma	4	0.806	0.819	0.787	0.606
Gerai	4	0.742	0.821	0.662	0.483
Dirbti	4	0.800	0.824	0.766	0.590
Procesas	2	0.826	0.871	0.771	0.642
Du	3	0.735	0.802	0.662	0.465
Įvairus	2	0.865	0.888	0.833	0.721
Kuris	2	0.826	0.862	0.779	0.641
Šalis	4	0.794	0.837	0.739	0.576
Valstybinis	3	0.858	0.878	0.831	0.709
Mokslas	4	0.832	0.881	0.775	0.656
Septyni	3	0.781	0.818	0.731	0.550
Sąlyga	5	0.781	0.841	0.712	0.554

Table A.1.1: K-Nearest Neighbor classification (without noise) of multivariate data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.768	0.821	0.704	0.526
Kultūra	4	0.813	0.886	0.737	0.623
Vienas	6	0.806	0.866	0.740	0.606
Žemė	2	0.774	0.809	0.727	0.536
Šeši	15	0.826	0.862	0.779	0.641
Darbas	5	0.832	0.881	0.775	0.656
Diena	6	0.768	0.865	0.679	0.544
Metai	5	0.794	0.854	0.726	0.580
Forma	10	0.800	0.824	0.766	0.590
Gerai	2	0.794	0.872	0.714	0.586
Dirbti	5	0.826	0.889	0.757	0.646
Procesas	4	0.761	0.812	0.700	0.512
Du	15	0.742	0.868	0.644	0.511
Įvairus	2	0.774	0.824	0.714	0.538
Kuris	6	0.735	0.810	0.658	0.468
Šalis	3	0.794	0.815	0.762	0.577
Valstybinis	4	0.781	0.818	0.731	0.550
Mokslas	4	0.794	0.872	0.714	0.586
Septyni	10	0.768	0.846	0.688	0.534
Sąlyga	18	0.787	0.828	0.735	0.563

Table A.1.2: K-Nearest Neighbor classification (with 15 dB noise) of multivariate data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.800	0.864	0.730	0.594
Kultūra	3	0.794	0.854	0.726	0.580
Vienas	4	0.852	0.894	0.800	0.694
Žemė	3	0.742	0.778	0.692	0.470
Šeši	6	0.813	0.867	0.750	0.617
Darbas	8	0.819	0.869	0.761	0.630
Diena	3	0.787	0.901	0.690	0.592
Metai	4	0.794	0.854	0.726	0.580
Forma	6	0.794	0.822	0.754	0.576
Gerai	3	0.787	0.861	0.711	0.571
Dirbti	6	0.832	0.872	0.783	0.655
Procesas	3	0.781	0.850	0.707	0.557
Du	6	0.729	0.824	0.642	0.466
Įvairus	4	0.781	0.841	0.712	0.554
Kuris	2	0.729	0.808	0.649	0.457
Šalis	4	0.794	0.822	0.754	0.576
Valstybinis	6	0.774	0.824	0.714	0.538
Mokslas	4	0.813	0.896	0.731	0.627
Septyni	8	0.781	0.850	0.707	0.557
Sąlyga	17	0.787	0.835	0.729	0.564

Table A.1.3: K-Nearest Neighbor classification (with 20 dB noise) of multivariate data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.768	0.838	0.693	0.531
Kultūra	3	0.774	0.831	0.708	0.540
Vienas	6	0.826	0.871	0.771	0.642
Žemė	8	0.755	0.795	0.701	0.497
Šeši	4	0.826	0.899	0.750	0.649
Darbas	3	0.826	0.889	0.757	0.646
Diena	3	0.787	0.890	0.695	0.586
Metai	4	0.800	0.864	0.730	0.594
Forma	7	0.819	0.830	0.803	0.633
Gerai	3	0.800	0.864	0.730	0.594
Dirbti	4	0.826	0.871	0.771	0.642
Procesas	3	0.774	0.867	0.688	0.554
Du	6	0.768	0.875	0.675	0.550
Įvairus	2	0.781	0.878	0.691	0.570
Kuris	6	0.723	0.797	0.645	0.442
Šalis	4	0.800	0.855	0.736	0.592
Valstybinis	3	0.787	0.828	0.735	0.563
Mokslas	4	0.813	0.896	0.731	0.627
Septyni	4	0.768	0.821	0.704	0.526
Sąlyga	5	0.781	0.868	0.696	0.565

Table A.1.4: K-Nearest Neighbor classification (with 25 dB noise) of multivariate data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.774	0.857	0.692	0.549
Kultūra	3	0.781	0.850	0.707	0.557
Vienas	6	0.832	0.881	0.775	0.656
Žemė	8	0.761	0.798	0.712	0.510
Šeši	4	0.832	0.900	0.760	0.660
Darbas	8	0.813	0.859	0.757	0.616
Diena	4	0.781	0.878	0.691	0.570
Metai	6	0.800	0.839	0.750	0.589
Forma	9	0.819	0.810	0.836	0.646
Gerai	3	0.794	0.872	0.714	0.586
Dirbti	2	0.826	0.909	0.744	0.653
Procesas	3	0.787	0.861	0.711	0.571
Du	10	0.742	0.847	0.651	0.498
Įvairus	3	0.774	0.831	0.708	0.540
Kuris	4	0.748	0.815	0.676	0.490
Šalis	3	0.794	0.837	0.739	0.576
Valstybinis	3	0.774	0.809	0.727	0.536
Mokslas	4	0.819	0.908	0.734	0.642
Septyni	8	0.774	0.848	0.697	0.545
Sąlyga	8	0.787	0.835	0.729	0.564

Table A.1.5: K-Nearest Neighbor classification (with 30 dB noise) of multivariate data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	7	0.794	0.822	0.754	0.576
Kultūra	6	0.787	0.806	0.758	0.565
Vienas	5	0.819	0.888	0.747	0.634
Žemė	2	0.761	0.757	0.769	0.527
Šeši	6	0.774	0.816	0.721	0.537
Darbas	3	0.703	0.782	0.623	0.405
Diena	2	0.697	0.694	0.705	0.398
Metai	2	0.781	0.759	0.830	0.589
Forma	3	0.781	0.792	0.763	0.554
Gerai	2	0.748	0.758	0.732	0.490
Dirbti	2	0.787	0.771	0.820	0.591
Procesas	2	0.794	0.709	0.800	0.590
Du	3	0.729	0.800	0.653	0.453
Įvairus	2	0.826	0.806	0.865	0.671
Kuris	3	0.813	0.867	0.750	0.617
Šalis	2	0.774	0.762	0.800	0.562
Valstybinis	3	0.832	0.856	0.800	0.656
Mokslas	2	0.819	0.792	0.878	0.670
Septyni	8	0.768	0.800	0.723	0.523
Sąlyga	5	0.774	0.840	0.703	0.542

Table A.1.6: K-Nearest Neighbor classification (without noise) of functional data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.787	0.766	0.833	0.600
Kultūra	3	0.774	0.831	0.708	0.540
Vienas	4	0.800	0.811	0.783	0.594
Žemė	2	0.716	0.705	0.744	0.450
Šeši	2	0.852	0.825	0.904	0.729
Darbas	5	0.819	0.869	0.761	0.630
Diena	2	0.794	0.802	0.780	0.582
Metai	5	0.794	0.854	0.726	0.580
Forma	8	0.806	0.800	0.818	0.618
Gerai	2	0.813	0.802	0.833	0.635
Dirbti	5	0.819	0.878	0.753	0.631
Procesas	2	0.774	0.762	0.800	0.562
Du	15	0.742	0.868	0.644	0.511
Įvairus	2	0.794	0.769	0.851	0.620
Kuris	2	0.729	0.722	0.745	0.467
Šalis	3	0.794	0.815	0.762	0.577
Valstybinis	4	0.761	0.752	0.780	0.532
Mokslas	8	0.800	0.855	0.736	0.592
Septyni	4	0.768	0.787	0.738	0.525
Sąlyga	18	0.800	0.817	0.774	0.591

Table A.1.7: K-Nearest Neighbor classification (with 15 dB noise) of functional data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.800	0.804	0.793	0.597
Kultūra	3	0.794	0.854	0.726	0.580
Vienas	4	0.832	0.827	0.842	0.669
Žemė	2	0.742	0.719	0.805	0.524
Šeši	4	0.826	0.825	0.828	0.652
Darbas	8	0.826	0.839	0.806	0.645
Diena	2	0.839	0.849	0.823	0.672
Metai	6	0.806	0.819	0.787	0.606
Forma	4	0.813	0.808	0.821	0.630
Gerai	2	0.806	0.800	0.818	0.618
Dirbti	3	0.813	0.843	0.773	0.615
Procesas	3	0.781	0.850	0.707	0.557
Du	2	0.729	0.722	0.745	0.467
Įvairus	2	0.813	0.790	0.860	0.650
Kuris	2	0.761	0.743	0.804	0.547
Šalis	8	0.794	0.796	0.789	0.585
Valstybinis	3	0.755	0.802	0.696	0.498
Mokslas	2	0.806	0.826	0.778	0.604
Septyni	10	0.787	0.813	0.750	0.563
Sąlyga	5	0.781	0.868	0.696	0.565

Table A.1.8: K-Nearest Neighbor classification (with 20 dB noise) of functional data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.787	0.777	0.808	0.584
Kultūra	3	0.774	0.831	0.708	0.540
Vienas	4	0.819	0.816	0.825	0.641
Žemė	2	0.768	0.741	0.837	0.578
Šeši	4	0.832	0.833	0.831	0.664
Darbas	3	0.819	0.878	0.753	0.631
Diena	2	0.832	0.833	0.831	0.664
Metai	4	0.806	0.813	0.797	0.609
Forma	8	0.826	0.800	0.880	0.680
Gerai	4	0.813	0.828	0.790	0.618
Dirbti	3	0.819	0.852	0.776	0.628
Procesas	2	0.781	0.769	0.804	0.573
Du	12	0.742	0.821	0.662	0.483
Įvairus	2	0.813	0.802	0.833	0.635
Kuris	2	0.755	0.745	0.776	0.521
Šalis	2	0.774	0.757	0.813	0.570
Valstybinis	3	0.774	0.809	0.727	0.536
Mokslas	5	0.806	0.905	0.716	0.621
Septyni	4	0.787	0.777	0.808	0.584
Sąlyga	5	0.774	0.857	0.692	0.549

Table A.1.9: K-Nearest Neighbor classification (with 25 dB noise) of functional data

Word	K	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.800	0.781	0.840	0.621
Kultūra	3	0.781	0.850	0.707	0.557
Vienas	4	0.826	0.818	0.839	0.657
Žemė	2	0.761	0.735	0.833	0.568
Šeši	2	0.832	0.814	0.868	0.682
Darbas	6	0.826	0.839	0.806	0.645
Diena	2	0.845	0.859	0.825	0.684
Metai	4	0.794	0.809	0.770	0.579
Forma	8	0.826	0.794	0.896	0.690
Gerai	2	0.813	0.802	0.833	0.635
Dirbti	3	0.826	0.862	0.779	0.641
Procesas	2	0.787	0.777	0.808	0.584
Du	10	0.729	0.793	0.658	0.450
Įvairus	2	0.813	0.802	0.833	0.635
Kuris	2	0.748	0.743	0.760	0.503
Šalis	3	0.787	0.835	0.729	0.564
Valstybinis	3	0.768	0.800	0.723	0.523
Mokslas	2	0.813	0.821	0.800	0.621
Septyni	4	0.787	0.782	0.796	0.578
Sąlyga	7	0.781	0.841	0.712	0.554

Table A.1.10: K-Nearest Neighbor classification (with 30 dB noise) of functional data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	123	0.806	0.813	0.797	0.609
Kultūra	55	0.684	0.741	0.614	0.355
Vienas	32	0.800	0.831	0.758	0.589
Žemė	20	0.761	0.791	0.719	0.510
Šeši	15	0.774	0.831	0.708	0.540
Darbas	3	0.710	0.753	0.652	0.404
Diena	60	0.735	0.845	0.643	0.488
Metai	117	0.768	0.776	0.754	0.530
Forma	11	0.761	0.812	0.700	0.512
Gerai	112	0.768	0.787	0.738	0.525
Dirbti	171	0.806	0.813	0.797	0.609
Procesas	17	0.768	0.787	0.738	0.525
Du	12	0.723	0.737	0.696	0.434
Įvairus	55	0.826	0.832	0.817	0.648
Kuris	31	0.768	0.875	0.675	0.550
Šalis	82	0.819	0.852	0.776	0.628
Valstybinis	25	0.858	0.840	0.891	0.731
Mokslas	5	0.781	0.859	0.701	0.560
Septyni	425	0.735	0.795	0.667	0.462
Sąlyga	20	0.768	0.829	0.699	0.528

Table A.1.11: Support Vector Machine classification (without noise) of multivariate data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	5	0.813	0.843	0.773	0.615
Kultūra	6	0.852	0.894	0.800	0.694
Vienas	14	0.845	0.867	0.815	0.682
Žemė	1	0.723	0.733	0.704	0.436
Šeši	1	0.858	0.862	0.852	0.714
Darbas	7	0.839	0.892	0.778	0.669
Diena	1	0.787	0.788	0.786	0.574
Metai	5	0.806	0.826	0.778	0.604
Forma	2	0.761	0.768	0.750	0.518
Gerai	3	0.845	0.837	0.860	0.696
Dirbti	12	0.852	0.853	0.850	0.703
Procesas	3	0.761	0.768	0.750	0.518
Du	3	0.781	0.850	0.707	0.557
Įvairus	4	0.826	0.825	0.828	0.652
Kuris	7	0.800	0.839	0.750	0.589
Šalis	24	0.832	0.827	0.842	0.669
Valstybinis	5	0.839	0.835	0.845	0.680
Mokslas	14	0.858	0.895	0.812	0.707
Septyni	19	0.781	0.841	0.712	0.554
Sąlyga	1	0.774	0.748	0.841	0.589

Table A.1.12: Support Vector Machine classification (with 15 dB noise) of multivariate data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	3	0.806	0.849	0.754	0.602
Kultūra	3	0.858	0.878	0.831	0.709
Vienas	8	0.845	0.875	0.806	0.681
Žemė	11	0.742	0.784	0.687	0.471
Šeši	11	0.865	0.897	0.824	0.720
Darbas	4	0.839	0.882	0.786	0.668
Diena	6	0.813	0.851	0.765	0.615
Metai	4	0.794	0.802	0.780	0.582
Forma	4	0.768	0.793	0.730	0.524
Gerai	1	0.839	0.816	0.885	0.700
Dirbti	10	0.871	0.872	0.869	0.741
Procesas	5	0.781	0.798	0.754	0.552
Du	3	0.794	0.863	0.720	0.583
Įvairus	9	0.826	0.832	0.817	0.648
Kuris	8	0.800	0.847	0.743	0.590
Šalis	20	0.839	0.842	0.833	0.675
Valstybinis	2	0.819	0.810	0.836	0.646
Mokslas	8	0.865	0.906	0.814	0.720
Septyni	30	0.781	0.841	0.712	0.554
Sąlyga	28	0.787	0.813	0.750	0.563

Table A.1.13: Support Vector Machine classification (with 20 dB noise) of multivariate data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.800	0.839	0.750	0.589
Kultūra	2	0.832	0.856	0.800	0.656
Vienas	9	0.852	0.885	0.809	0.694
Žemė	2	0.735	0.753	0.707	0.459
Šeši	13	0.865	0.888	0.833	0.721
Darbas	3	0.845	0.884	0.797	0.681
Diena	5	0.806	0.833	0.769	0.603
Metai	6	0.813	0.835	0.781	0.616
Forma	2	0.761	0.779	0.733	0.512
Gerai	1	0.839	0.816	0.885	0.700
Dirbti	7	0.858	0.854	0.864	0.719
Procesas	5	0.794	0.815	0.762	0.577
Du	3	0.787	0.852	0.716	0.568
Įvairus	2	0.826	0.825	0.828	0.652
Kuris	7	0.800	0.847	0.743	0.590
Šalis	3	0.845	0.844	0.847	0.691
Valstybinis	5	0.819	0.810	0.836	0.646
Mokslas	11	0.890	0.940	0.833	0.773
Septyni	15	0.761	0.827	0.689	0.516
Sąlyga	3	0.768	0.770	0.764	0.534

Table A.1.14: Support Vector Machine classification (with 25 dB noise) of multivariate data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	5	0.806	0.849	0.754	0.602
Kultūra	3	0.845	0.867	0.815	0.682
Vienas	11	0.845	0.875	0.806	0.681
Žemė	2	0.735	0.753	0.707	0.459
Šeši	11	0.871	0.907	0.826	0.733
Darbas	4	0.858	0.905	0.803	0.708
Diena	5	0.794	0.845	0.732	0.578
Metai	5	0.806	0.833	0.769	0.603
Forma	2	0.768	0.787	0.738	0.525
Gerai	1	0.845	0.824	0.887	0.710
Dirbti	7	0.865	0.856	0.879	0.735
Procesas	4	0.774	0.789	0.750	0.539
Du	10	0.794	0.872	0.714	0.586
Įvairus	7	0.845	0.851	0.836	0.687
Kuris	14	0.794	0.837	0.739	0.576
Šalis	2	0.845	0.844	0.847	0.691
Valstybinis	48	0.826	0.862	0.779	0.641
Mokslas	9	0.877	0.908	0.838	0.746
Septyni	20	0.768	0.829	0.699	0.528
Sąlyga	15	0.781	0.798	0.754	0.552

Table A.1.15: Support Vector Machine classification (with 30 dB noise) of multivariate data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	292	0.761	0.743	0.804	0.547
Kultūra	474	0.723	0.716	0.739	0.455
Vienas	249	0.723	0.764	0.667	0.431
Žemė	460	0.768	0.814	0.710	0.524
Šeši	61	0.748	0.763	0.724	0.487
Darbas	448	0.755	0.777	0.721	0.498
Diena	493	0.690	0.744	0.623	0.367
Metai	365	0.697	0.722	0.655	0.377
Forma	16	0.716	0.735	0.684	0.419
Gerai	24	0.716	0.709	0.733	0.442
Dirbti	240	0.755	0.777	0.721	0.498
Procesas	257	0.742	0.750	0.727	0.477
Du	54	0.677	0.692	0.647	0.339
Įvairus	72	0.800	0.804	0.793	0.597
Kuris	59	0.723	0.753	0.677	0.430
Šalis	455	0.819	0.830	0.803	0.633
Valstybinis	368	0.800	0.811	0.783	0.594
Mokslas	348	0.781	0.868	0.696	0.565
Septyni	285	0.774	0.809	0.727	0.536
Sąlyga	50	0.748	0.738	0.771	0.509

Table A.1.16: Support Vector Machine classification (without noise) of functional data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	31	0.768	0.755	0.796	0.551
Kultūra	217	0.755	0.755	0.755	0.510
Vienas	459	0.742	0.727	0.778	0.505
Žemė	148	0.729	0.711	0.780	0.491
Šeši	3	0.787	0.794	0.776	0.570
Darbas	362	0.781	0.786	0.772	0.558
Diena	17	0.761	0.785	0.726	0.511
Metai	152	0.755	0.777	0.721	0.498
Forma	40	0.723	0.728	0.712	0.440
Gerai	127	0.806	0.794	0.830	0.624
Dirbti	185	0.729	0.731	0.725	0.456
Procesas	92	0.729	0.711	0.780	0.491
Du	6	0.677	0.692	0.647	0.339
Įvairus	34	0.729	0.722	0.745	0.467
Kuris	98	0.658	0.691	0.603	0.294
Šalis	58	0.716	0.713	0.723	0.436
Valstybinis	27	0.742	0.745	0.736	0.481
Mokslas	25	0.781	0.792	0.763	0.554
Septyni	141	0.723	0.716	0.739	0.455
Sąlyga	380	0.748	0.726	0.810	0.535

Table A.1.17: Support Vector Machine classification (with 15 dB noise) of functional data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	362	0.768	0.750	0.809	0.559
Kultūra	54	0.761	0.752	0.780	0.532
Vienas	480	0.768	0.765	0.774	0.538
Žemė	5	0.716	0.709	0.733	0.442
Šeši	22	0.800	0.817	0.774	0.591
Darbas	16	0.787	0.794	0.776	0.570
Diena	27	0.748	0.752	0.741	0.493
Metai	148	0.768	0.781	0.746	0.527
Forma	3	0.710	0.710	0.708	0.419
Gerai	5	0.781	0.769	0.804	0.573
Dirbti	500	0.735	0.721	0.773	0.493
Procesas	211	0.742	0.731	0.766	0.497
Du	51	0.671	0.689	0.635	0.324
Įvairus	32	0.748	0.738	0.771	0.509
Kuris	488	0.665	0.694	0.614	0.308
Šalis	12	0.748	0.738	0.771	0.509
Valstybinis	3	0.716	0.721	0.706	0.427
Mokslas	1	0.781	0.775	0.792	0.567
Septyni	268	0.735	0.733	0.740	0.473
Sąlyga	2	0.755	0.724	0.846	0.570

Table A.1.18: Support Vector Machine classification (with 20 dB noise) of functional data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	58	0.761	0.768	0.750	0.518
Kultūra	23	0.748	0.748	0.750	0.498
Vienas	476	0.781	0.786	0.772	0.558
Žemė	17	0.723	0.708	0.762	0.470
Šeši	19	0.787	0.813	0.750	0.563
Darbas	44	0.800	0.804	0.793	0.597
Diena	39	0.729	0.750	0.695	0.445
Metai	40	0.755	0.783	0.714	0.497
Forma	38	0.729	0.735	0.717	0.452
Gerai	1	0.774	0.757	0.813	0.570
Dirbti	466	0.723	0.716	0.739	0.455
Procesas	87	0.735	0.738	0.731	0.469
Du	390	0.671	0.673	0.667	0.339
Įvairus	97	0.768	0.760	0.784	0.544
Kuris	311	0.697	0.726	0.650	0.376
Šalis	55	0.768	0.755	0.796	0.551
Valstybinis	5	0.729	0.735	0.717	0.452
Mokslas	11	0.794	0.796	0.789	0.585
Septyni	121	0.729	0.735	0.717	0.452
Sąlyga	1	0.748	0.714	0.861	0.575

Table A.1.19: Support Vector Machine classification (with 25 dB noise) of functional data

Word	C	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	28	0.748	0.752	0.741	0.493
Kultūra	24	0.761	0.757	0.769	0.527
Vienas	131	0.781	0.780	0.782	0.562
Žemė	34	0.723	0.712	0.750	0.462
Šeši	15	0.800	0.831	0.758	0.589
Darbas	28	0.813	0.814	0.810	0.625
Diena	46	0.735	0.747	0.714	0.462
Metai	73	0.761	0.779	0.733	0.512
Forma	38	0.729	0.740	0.709	0.449
Gerai	25	0.781	0.769	0.804	0.573
Dirbti	459	0.735	0.733	0.740	0.473
Procesas	495	0.755	0.732	0.814	0.546
Du	107	0.684	0.699	0.654	0.353
Įvairus	61	0.774	0.767	0.788	0.555
Kuris	199	0.710	0.732	0.672	0.404
Šalis	21	0.787	0.782	0.796	0.578
Valstybinis	30	0.735	0.729	0.750	0.479
Mokslas	1	0.800	0.792	0.815	0.607
Septyni	254	0.729	0.740	0.709	0.449
Sąlyga	1	0.742	0.712	0.838	0.550

Table A.1.20: Support Vector Machine classification (with 30 dB noise) of functional data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	7	0.884	0.883	0.885	0.768
Kultūra	49	0.852	0.819	0.920	0.739
Vienas	34	0.832	0.820	0.855	0.675
Žemė	42	0.832	0.814	0.868	0.682
Šeši	48	0.832	0.802	0.898	0.700
Darbas	9	0.787	0.766	0.833	0.600
Diena	6	0.800	0.766	0.886	0.652
Metai	10	0.832	0.840	0.820	0.660
Forma	3	0.800	0.761	0.905	0.666
Gerai	5	0.800	0.766	0.886	0.652
Dirbti	35	0.819	0.810	0.836	0.646
Procesas	7	0.858	0.827	0.922	0.748
Du	8	0.742	0.760	0.712	0.472
Įvairus	18	0.877	0.851	0.926	0.777
Kuris	4	0.845	0.817	0.902	0.719
Šalis	5	0.897	0.885	0.915	0.801
Valstybinis	6	0.865	0.842	0.907	0.749
Mokslas	41	0.884	0.875	0.898	0.773
Septyni	15	0.852	0.813	0.938	0.751
Sąlyga	21	0.845	0.811	0.918	0.730

Table A.1.21: Random Forest classification (without noise) of multivariate data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.845	0.867	0.815	0.682
Kultūra	1	0.800	0.766	0.886	0.652
Vienas	1	0.813	0.785	0.875	0.660
Žemė	39	0.748	0.768	0.717	0.485
Šeši	7	0.826	0.818	0.839	0.657
Darbas	25	0.852	0.860	0.839	0.699
Diena	2	0.794	0.784	0.811	0.596
Metai	29	0.819	0.844	0.785	0.629
Forma	10	0.774	0.772	0.778	0.550
Gerai	19	0.845	0.837	0.860	0.696
Dirbti	18	0.839	0.822	0.870	0.692
Procesas	21	0.774	0.767	0.788	0.555
Du	41	0.761	0.812	0.700	0.512
Įvairus	17	0.813	0.790	0.860	0.650
Kuris	10	0.755	0.789	0.708	0.497
Šalis	8	0.832	0.833	0.831	0.664
Valstybinis	3	0.806	0.773	0.889	0.662
Mokslas	2	0.845	0.837	0.860	0.696
Septyni	30	0.787	0.777	0.808	0.584
Sąlyga	31	0.819	0.816	0.825	0.641

Table A.1.22: Random Forest classification (with 15 dB noise) of multivariate data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	1	0.845	0.859	0.825	0.684
Kultūra	1	0.832	0.808	0.882	0.690
Vienas	5	0.852	0.838	0.875	0.713
Žemė	39	0.781	0.780	0.782	0.562
Šeši	19	0.839	0.828	0.857	0.685
Darbas	25	0.858	0.862	0.852	0.714
Diena	8	0.787	0.771	0.820	0.591
Metai	40	0.858	0.862	0.852	0.714
Forma	4	0.768	0.770	0.764	0.534
Gerai	16	0.832	0.820	0.855	0.675
Dirbti	4	0.852	0.825	0.904	0.729
Procesas	33	0.826	0.812	0.852	0.664
Du	2	0.768	0.800	0.723	0.523
Įvairus	33	0.845	0.824	0.887	0.710
Kuris	2	0.768	0.793	0.730	0.524
Šalis	7	0.839	0.828	0.857	0.685
Valstybinis	1	0.826	0.784	0.932	0.716
Mokslas	2	0.845	0.844	0.847	0.691
Septyni	13	0.781	0.775	0.792	0.567
Sąlyga	28	0.839	0.822	0.870	0.692

Table A.1.23: Random Forest classification (with 20 dB noise) of multivariate data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.858	0.886	0.821	0.707
Kultūra	2	0.845	0.811	0.918	0.730
Vienas	7	0.832	0.820	0.855	0.675
Žemė	11	0.755	0.760	0.745	0.505
Šeši	17	0.852	0.838	0.875	0.713
Darbas	11	0.890	0.901	0.875	0.776
Diena	8	0.806	0.788	0.843	0.632
Metai	10	0.845	0.851	0.836	0.687
Forma	24	0.787	0.777	0.808	0.584
Gerai	46	0.826	0.825	0.828	0.652
Dirbti	5	0.852	0.832	0.889	0.721
Procesas	7	0.839	0.828	0.857	0.685
Du	32	0.774	0.840	0.703	0.542
Įvairus	13	0.858	0.847	0.877	0.724
Kuris	4	0.787	0.813	0.750	0.563
Šalis	2	0.819	0.798	0.863	0.661
Valstybinis	9	0.826	0.789	0.913	0.702
Mokslas	5	0.865	0.871	0.855	0.726
Septyni	17	0.787	0.782	0.796	0.578
Sąlyga	30	0.852	0.838	0.875	0.713

Table A.1.24: Random Forest classification (with 25 dB noise) of multivariate data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	1	0.871	0.880	0.857	0.738
Kultūra	20	0.839	0.828	0.857	0.685
Vienas	2	0.839	0.828	0.857	0.685
Žemė	33	0.755	0.760	0.745	0.505
Šeši	9	0.852	0.838	0.875	0.713
Darbas	31	0.903	0.921	0.879	0.800
Diena	3	0.813	0.802	0.833	0.635
Metai	7	0.865	0.871	0.855	0.726
Forma	11	0.774	0.772	0.778	0.550
Gerai	20	0.832	0.833	0.831	0.664
Dirbti	4	0.871	0.843	0.925	0.768
Procesas	1	0.865	0.863	0.867	0.730
Du	45	0.787	0.843	0.722	0.566
Įvairus	1	0.858	0.840	0.891	0.731
Kuris	1	0.787	0.820	0.742	0.563
Šalis	15	0.852	0.832	0.889	0.721
Valstybinis	2	0.832	0.802	0.898	0.700
Mokslas	6	0.858	0.854	0.864	0.719
Septyni	15	0.806	0.794	0.830	0.624
Sąlyga	1	0.826	0.800	0.880	0.680

Table A.1.25: Random Forest classification (with 30 dB noise) of multivariate data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	6	0.787	0.794	0.776	0.570
Kultūra	7	0.781	0.750	0.860	0.610
Vienas	4	0.800	0.792	0.815	0.607
Žemė	1	0.781	0.780	0.782	0.562
Šeši	5	0.819	0.804	0.849	0.653
Darbas	4	0.735	0.743	0.722	0.465
Diena	7	0.774	0.757	0.813	0.570
Metai	4	0.774	0.796	0.742	0.538
Forma	2	0.748	0.730	0.795	0.525
Gerai	1	0.742	0.755	0.719	0.474
Dirbti	3	0.774	0.752	0.826	0.578
Procesas	7	0.794	0.759	0.884	0.643
Du	7	0.761	0.768	0.750	0.518
Įvairus	4	0.787	0.777	0.808	0.584
Kuris	6	0.819	0.830	0.803	0.633
Šalis	3	0.826	0.832	0.817	0.648
Valstybinis	2	0.813	0.796	0.846	0.642
Mokslas	6	0.852	0.825	0.904	0.729
Septyni	5	0.852	0.825	0.904	0.729
Sąlyga	4	0.735	0.738	0.731	0.469

Table A.1.26: Random Forest classification (without noise) of functional data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	1	0.748	0.743	0.760	0.503
Kultūra	5	0.748	0.768	0.717	0.485
Vienas	4	0.735	0.775	0.682	0.457
Žemė	2	0.716	0.713	0.723	0.436
Šeši	1	0.768	0.807	0.716	0.523
Darbas	2	0.787	0.806	0.758	0.565
Diena	2	0.761	0.757	0.769	0.527
Metai	1	0.742	0.750	0.727	0.477
Forma	7	0.677	0.696	0.642	0.338
Gerai	5	0.774	0.772	0.778	0.550
Dirbti	1	0.723	0.724	0.720	0.444
Procesas	4	0.690	0.702	0.667	0.369
Du	3	0.645	0.692	0.578	0.270
Įvairus	2	0.723	0.728	0.712	0.440
Kuris	7	0.652	0.700	0.585	0.285
Šalis	3	0.774	0.809	0.727	0.536
Valstybinis	6	0.735	0.729	0.750	0.479
Mokslas	1	0.742	0.745	0.736	0.481
Septyni	3	0.768	0.776	0.754	0.530
Sąlyga	1	0.735	0.721	0.773	0.493

Table A.1.27: Random Forest classification (with 15 dB noise) of functional data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	4	0.729	0.745	0.702	0.447
Kultūra	1	0.748	0.752	0.741	0.493
Vienas	6	0.761	0.779	0.733	0.512
Žemė	4	0.703	0.716	0.679	0.395
Šeši	1	0.774	0.816	0.721	0.537
Darbas	1	0.774	0.802	0.734	0.537
Diena	1	0.774	0.772	0.778	0.550
Metai	2	0.735	0.753	0.707	0.459
Forma	2	0.703	0.712	0.686	0.398
Gerai	7	0.787	0.782	0.796	0.578
Dirbti	3	0.716	0.717	0.714	0.431
Procesas	4	0.710	0.723	0.685	0.408
Du	2	0.677	0.733	0.609	0.341
Įvairus	1	0.723	0.733	0.704	0.436
Kuris	2	0.671	0.719	0.606	0.325
Šalis	2	0.794	0.822	0.754	0.576
Valstybinis	1	0.716	0.713	0.723	0.436
Mokslas	1	0.755	0.760	0.745	0.505
Septyni	1	0.729	0.722	0.745	0.467
Sąlyga	3	0.748	0.758	0.732	0.490

Table A.1.28: Random Forest classification (with 20 dB noise) of functional data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	2	0.716	0.717	0.714	0.431
Kultūra	2	0.761	0.768	0.750	0.518
Vienas	7	0.748	0.780	0.703	0.483
Žemė	3	0.723	0.728	0.712	0.440
Šeši	5	0.774	0.831	0.708	0.540
Darbas	2	0.794	0.815	0.762	0.577
Diena	1	0.755	0.745	0.776	0.521
Metai	1	0.735	0.747	0.714	0.462
Forma	6	0.690	0.706	0.660	0.366
Gerai	1	0.800	0.781	0.840	0.621
Dirbti	6	0.729	0.731	0.725	0.456
Procesas	1	0.697	0.705	0.680	0.385
Du	2	0.684	0.730	0.621	0.352
Įvairus	6	0.787	0.806	0.758	0.565
Kuris	2	0.710	0.753	0.652	0.404
Šalis	1	0.794	0.830	0.746	0.576
Valstybinis	3	0.716	0.721	0.706	0.427
Mokslas	1	0.768	0.765	0.774	0.538
Septyni	5	0.742	0.745	0.736	0.481
Sąlyga	6	0.755	0.760	0.745	0.505

Table A.1.29: Random Forest classification (with 25 dB noise) of functional data

Word	Mtry	Accuracy	Sensitivity	Specificity	Youden Index
Lietuva	1	0.729	0.731	0.725	0.456
Kultūra	1	0.755	0.760	0.745	0.505
Vienas	7	0.768	0.793	0.730	0.524
Žemė	2	0.703	0.716	0.679	0.395
Šeši	6	0.768	0.829	0.699	0.528
Darbas	1	0.832	0.864	0.791	0.655
Diena	1	0.768	0.755	0.796	0.551
Metai	1	0.716	0.740	0.678	0.418
Forma	5	0.742	0.740	0.745	0.485
Gerai	2	0.819	0.804	0.849	0.653
Dirbti	3	0.723	0.728	0.712	0.440
Procesas	5	0.703	0.712	0.686	0.398
Du	3	0.703	0.744	0.646	0.391
Įvairus	3	0.781	0.786	0.772	0.558
Kuris	3	0.729	0.779	0.667	0.446
Šalis	1	0.748	0.787	0.697	0.483
Valstybinis	3	0.710	0.710	0.708	0.419
Mokslas	4	0.768	0.770	0.764	0.534
Septyni	2	0.735	0.738	0.731	0.469
Sąlyga	5	0.768	0.765	0.774	0.538

Table A.1.30: Random Forest classification (with 30 dB noise) of functional data

A.2 R codes

```
library(tuneR)
library(seewave)
library(fda.usc)
library(class)
library(randomForest)
library(e1071)
library(caret)

#Smoothing parameters
param = matrix(nrow = length(file_name), ncol = 3)

for (p in 1:length(file_name)){
x = readWave(paste0("C:/Users/Desktop/Magistras/Duomenys_wav/",
file_name[p]))
y = attributes(x)$left
```

```
t = 1
n = length(y) - 1
delta = t / n
s = seq(0, t, delta)
hilbe = env(x)
fhy = fdata(t(hilbe), argvals = s)
lambdas = c(0, 0.0001, 0.001, 0.01, 0.1)
basis = seq(4, 50, by = 1)
long = length(lambdas) * length(basis)
mean.gcv = rep(0, long)
backtrack = list()
k = 1
for(i in lambdas){
for(j in basis){
dataf = fhy
bbasis = create.bspline.basis(rangeval = dataf$rangeval, nbasis = j)
curv.Lfd = int2Lfd(2)
curv.fdPar = fdPar(bbasis, curv.Lfd, lambda = i)
tempSmooth = smooth.basis(argvals = dataf$argvals, y = hilbe,
fdParobj = curv.fdPar)
mean.gcv[k] = mean(tempSmooth$gcv)
backtrack[[k]] = c(i,j)
k = k + 1
}
}
best = which.min(mean.gcv)
lambdabest = backtrack[[best]][1]
basisbest = backtrack[[best]][2]
param[p,] = c(file_name[p], lambdabest, basisbest)
}

med = c()
fin = lapply(1:40, function(i) (1:20) + (i - 1) * 20)

k = 1

for(i in 1:length(fin)){
med[k] = round(median(as.numeric(param[as.numeric(fin[[i]]), 3])),
```

```
digits = 0)
k = k + 1
}

#mKNN
results.mknn = list()

acc.mknn = c()
sen.mknn = c()
spe.mknn = c()

k = 2:50

for(j in k){
pred = knn(train = train_scaledf, test = test_scaledf, cl = fcr, k = j)
actual = fct
conf = confusionMatrix(actual, pred)
sen.mknn[j] = conf$byClass[["Sensitivity"]]
spe.mknn[j] = conf$byClass[["Specificity"]]
acc.mknn[j] = conf$overall[["Accuracy"]]
}

best.acc.mknn = max(na.omit(acc.mknn))
nr = which(acc.mknn == max(na.omit(acc.mknn)))[1]
results.mknn[[i]] = c(nr, best.acc.mknn, sen.mknn[nr], spe.mknn[nr])

#fKNN
results.fknn = list()

acc.fknn = c()
sen.fknn = c()
spe.fknn = c()

k = 2:50

for(j in k){
fknn = classif.knn(fcr, ftrain, knn = j)
pred = predict(fknn, ftest)
```

```
actual = fct
conf = confusionMatrix(actual, pred)
sen.fknn[j] = conf$byClass[["Sensitivity"]]
spe.fknn[j] = conf$byClass[["Specificity"]]
acc.fknn[j] = conf$overall[["Accuracy"]]
}

best.acc.fknn = max(na.omit(acc.fknn))
nr = which(acc.fknn == max(na.omit(acc.fknn)))[1]
results.fknn[[i]] = c(nr, best.acc.fknn, sen.fknn[nr], spe.fknn[nr])

#mSVM
results.msvm = list()

acc.msvm = c()
sen.msvm = c()
spe.msvm = c()

c = 1:500

for(j in c){
svmfit = svm(fcr ~ ., data = train_scaled, kernel = 'radial',
type = 'C-classification', cost = j)
pred = predict(svmfit, test_scaled)
actual = fct
conf = confusionMatrix(actual, pred)
sen.msvm[j] = conf$byClass[["Sensitivity"]]
spe.msvm[j] = conf$byClass[["Specificity"]]
acc.msvm[j] = conf$overall[["Accuracy"]]
}

best.acc.msvm = max(na.omit(acc.msvm))
nr = which(acc.msvm == max(na.omit(acc.msvm)))[1]
results.msvm[[i]] = c(nr, best.acc.msvm, sen.msvm[nr], spe.msvm[nr])

#fSVM
results.fsvm = list()
```

```
acc.fsvm = c()
sen.fsvm = c()
spe.fsvm = c()

c = 1:500

for(j in c){
svmfit = classif.svm(fcr ~ x, data = dat, kernel = "radial",
type = "C-classification", cost = j)
newdat = list("x" = ftest)
pred = predict(svmfit, newdat)
actual = fct
conf = confusionMatrix(actual, pred)
sen.fsvm[j] = conf$byClass[["Sensitivity"]]
spe.fsvm[j] = conf$byClass[["Specificity"]]
acc.fsvm[j] = conf$overall[["Accuracy"]]
}

best.acc.fsvm = max(na.omit(acc.fsvm))
nr = which(acc.fsvm == max(acc.fsvm))[1]
results.fsvm[[i]] = c(nr, best.acc.fsvm, sen.fsvm[nr], spe.fsvm[nr])

#mRF
results.mrf = list()

acc.mrf = c()
sen.mrf = c()
spe.mrf = c()

m = 1:50

for(j in m){
mrf = randomForest(fcr ~ ., data = train, mtry = j)
pred = predict(mrf, test)
actual = fct
conf = confusionMatrix(actual, pred)
sen.mrf[j] = conf$byClass[["Sensitivity"]]
spe.mrf[j] = conf$byClass[["Specificity"]]
```

```
acc.mrf[j] = conf$overall[["Accuracy"]]
}

best.acc.mrf = max(na.omit(acc.mrf))
nr = which(acc.mrf == max(na.omit(acc.mrf)))[1]
results.mrf[[i]] = c(nr, best.acc.mrf, sen.mrf[nr], spe.mrf[nr])

#fRF
results.frf = list()

acc.frf = c()
sen.frf = c()
spe.frf = c()

m = 1:50

for(j in m){
frf = classif.randomForest(fcr ~ x, data = dat, mtry = j)
newdat = list("x" = ftest)
pred = predict(frf, newdat)
actual = fct
conf = confusionMatrix(actual, pred)
sen.frf[j] = conf$byClass[["Sensitivity"]]
spe.frf[j] = conf$byClass[["Specificity"]]
acc.frf[j] = conf$overall[["Accuracy"]]
}

best.acc.frf = max(na.omit(acc.frf))
nr = which(acc.frf == max(acc.frf))[1]
results.frf[[i]] = c(nr, best.acc.frf, sen.frf[nr], spe.frf[nr])

#Friedman test
mknn_friedman = friedman.test(mknn_acc)
fknn_friedman = friedman.test(fknn_acc)
msvm_friedman = friedman.test(msvm_acc)
fsvm_friedman = friedman.test(fsvm_acc)
mrf_friedman = friedman.test(mrf_acc)
frf_friedman = friedman.test(frf_acc)
```

```r
#T-test
knn_ttest = t.test(acc ~ type, data = knn, paired = TRUE)
svm_ttest = t.test(acc ~ type, data = svm, paired = TRUE)
rf_ttest = t.test(acc ~ type, data = rf, paired = TRUE)
```