LDATS2470 - Project 2023

Classification of people based on their health status by the use of SVM

Rousseau Mathieu, 67001800

,

UCLouvain
Belgium
23/12/2022

1 Introduction

The aim of this project is to analyse a range of biomedical voice measurements from 31 people where 23 of parkinson disease. There are around 6 voice measurements per patient so that in total we have a collection of 195 observations. Each one contains severall voice measures that are detailled below. The ‘status‘ column indicate is the patient has the parkinson disease or not.

2 Research question

Using support vector machine algorithms we want to discriminate the patients based on their health status.

Firstly, we will begin with a basic descriptive analysis of the different variables composing this dataset. Then we will first try to perform an Hard Margin SVM. If needed, in case of the presence of outliers or a non linear dataset, we could investigate respectively the Soft Margin SVM or the use of the kernel trick.

3 Exploratory data analysis

4 SVM

4.1 Preparing the datas

One of the first thing we noticed in the exploratory data analysis was that the target classes are unbalanced (**??**). Indeed, among the 195 observations, we had 48 healthy people and 147 people having the parkinson disease. This is roughly a $3:1$ ratio. Having unbalanced classes can be misleading as the algorithm could have "good score" even if it only predicts the majority class.

In order to fix the imbalance, we upsampled the datas. The idea is to sample the datas with replacement by making multiple copy of observations belonging to the minority target class. The consequence is having a perfectly balanced dataset with $1:1$ target class ratio.

Next, we split the dataset into a training and testing test. We kept 30% of the datas for testing purpose.

Plotting the datas for the training and testing sets on the two first principal components, we notice there is no trivial separation of the datas. The two first principal components accounts for roughly 64% of the variation in the datas (respectively 46.8% and 17.3% for the first and second principal component) which is correct but not completely representative of the reality. However, we can already assume that a linear kernel will not be effective. Let’s try it first though.

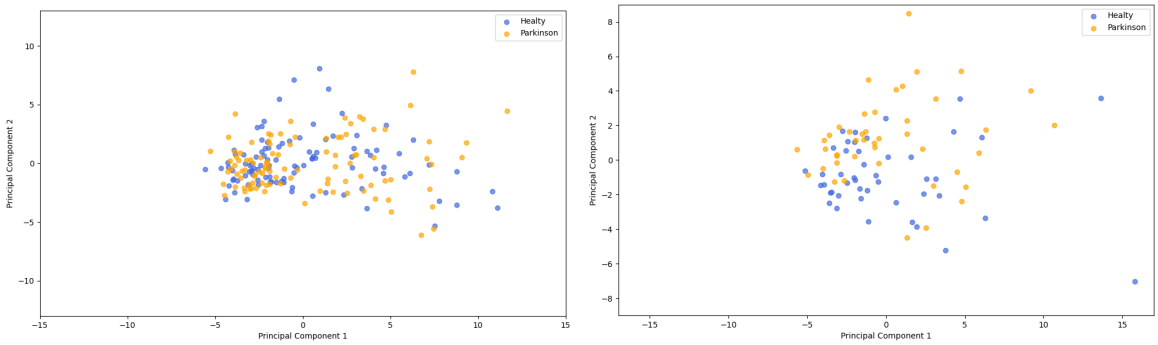

Figure 1 – *Plot of the training set (at left) and testing set (at right) on the two first principal components.*

4.2 Hard margin SVM

We first try to find a separating hyperplane using a linear kernel.

Let $D = \{ (\vec{x}_i, \vec{y}_i) \}_{i=1}^{n}$ be our set of data points with $x_i \in \mathbb{R}^d$ (where d is the dimension of the feature space) and $y_i \in \{ +1, -1 \}$.

Let $h : \mathbb{R}^d \to \mathbb{R}$, the equation of a plane defined $\forall x \in \mathbb{R}^d$ by,

$$h(\vec{x}) = \vec{w}^T \cdot \vec{x} + b \tag{1}$$

where $\vec{w} \in \mathbb{R}^d$ is a vector of weights and b is the bias.

A separating hyperplane is the set of points $\vec{x} \in \mathbb{R}^d$ that satisfy,

$$h(\vec{x}) = 0 \tag{2}$$

such that we have $\forall \vec{x}_i \in D$ the following inequality,

$$y_i \cdot h(\vec{x}_i) \geq 1 \tag{3}$$

The hyperplane should yields the maximum margin among all possible separating hyperplanes, that is the parameters $\vec{w}$ and b are the one that maximize $1/||\vec{w}||$.

An equivalent formulation is to say we want to minimize the following objective function,

$$\min_{\vec{w},b} \frac{||\vec{w}||^2}{2} \tag{4}$$

subject to the linear constraint **??**.

The found hyperplane gets us an accuracy of 0.752 on the testing set. However, a better way to assess the quality of a classifier is to look at the confusion matrix.

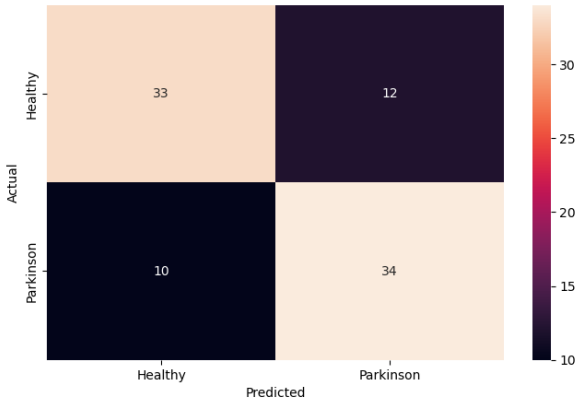

Figure 2 – *Confusion matrix on the testing set for a linear SVM*

We notice that we have effectifely $\sim 25\%$ of missclassification. Therefore, we can conclude that a linear kernel is not ideal. On the following plot, we project the datas points of the training set on the two first principal components along with the support vectors. We also show the separating hyperplane and the margins. We have a confirmation of our first intuition, a linear kernel is not effective in separating these datas.

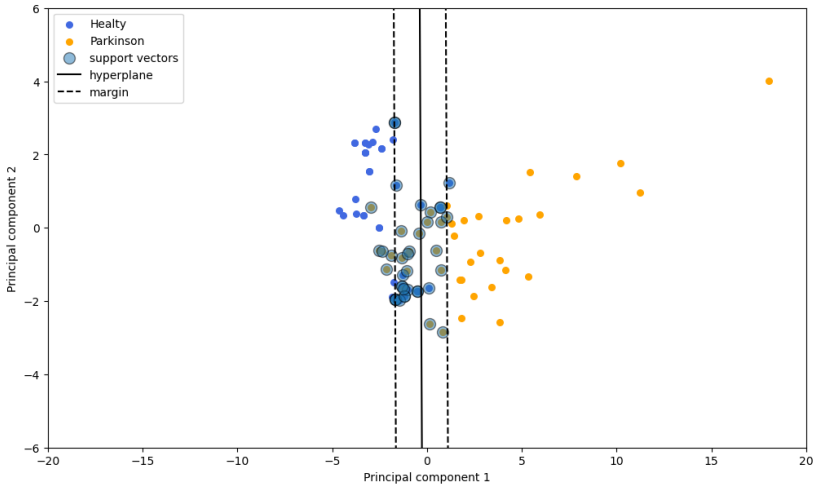

Figure 3 – *Margins and separating hyperplane for a linear kernel SVM*

4.3 Hard margin SVM

Despite the poor performance of the linear kernel. We can still try to improve it by introducing a *slack variable* ξ_i, $i = 1, \ldots, n$ that for each data point x_i indicates how much it violates the separability condition [1]

$\forall \vec{x}_i \in D$ the inequality becomes,

$$y_i \cdot h(\vec{x}_i) \geq 1 - \xi_i \tag{5}$$

1. The separability condition ensures that the point is at least $\frac{1}{||\vec{w}||}$ from the hyperplane.

The goal is then to minimize the same objective function as before plus a penalty term,

$$\min_{\vec{w},b,\xi_i} \frac{||\vec{w}||^2}{2} - C\sum_{i=1}^{n}(x_i)^k \tag{6}$$

where $C \in \mathbb{R}$ is a regularization constant and $k \in \mathbb{R}$.

This objective function is subject to the constraint above (**??**) as well as $x_i \geq 0 \ \forall \vec{x}_i \in D$.

Performing a grid search with 5-fold cross-validation, we found an optimal regularization constant $C = 0.9$ giving a mean accuracy of 0.829 on the validations sets and an accuracy of 0.753 on the testing set. That is a slighlty better result than the hard margin case. Let's look at the confusion matrix,

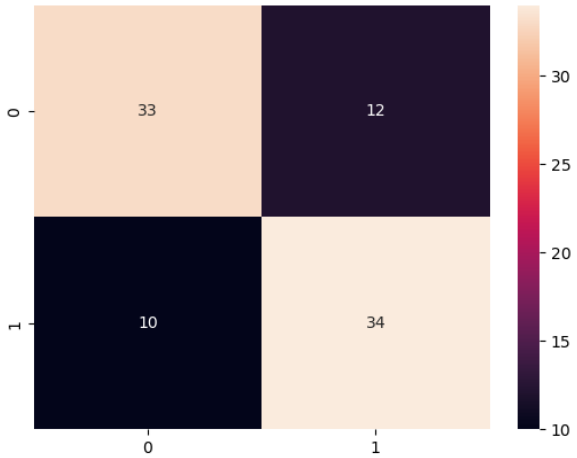

Figure 4 – *Confusion matrix on the testing set for the soft margin case*

We notice that our classificer did not improve with the regularization parameters. So the linear classifier is definitely not helpful in separating these datas.

4.4 Kernel trick

A to fix this problem is to map the data points in a high-dimension space by performing a non-linear transformation,

$$\phi : \mathbb{R}^d \to \mathbb{R}^l, \quad \vec{x}_i \to \phi(\vec{x}_i) \tag{7}$$

By this way, there is far more probability for the data to be linearly separable and because we perform a non-linear transformation, a linear separation in the feature space correspond to non linear decision region in the original data space.

We used a gaussian radial basis function for the kernel,

$$\phi : \mathbb{R}^d \to \mathbb{R}^l, \quad \vec{x}_i \to \exp\left(-(\gamma\vec{x}_i)^2\right) \tag{8}$$

The kernel is then,

$$K(x_i, x_j) = \phi(x_i)^T\phi(x_j) \tag{9}$$

$$= \exp\left(-\gamma(\vec{x}_i - \vec{x}_j)^2\right) \tag{10}$$

The γ parameter has to be found by cross-validation. Performing a grid search on the C and γ parameters with a 5-fold cross-validation, we found the optimal parameters to be $C = 4.51$ and $\gamma = 0.005$. With these, we get a mean accuracy score of 0.961 on the validation sets and an accuracy score of 0.888 on the testing set. That's way better than using a linear kernel. We proove it by showing the confusion matrix,

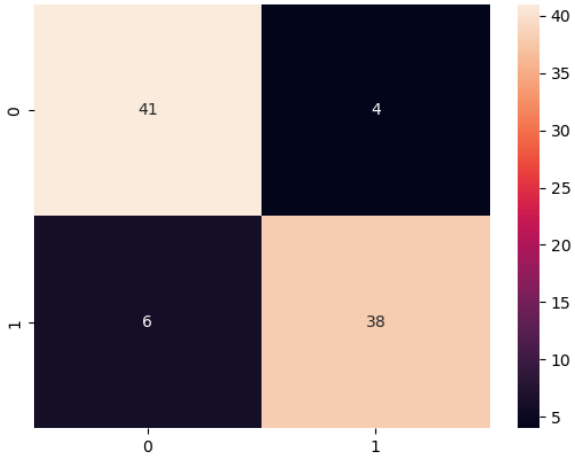

Figure 5 – *Confusion matrix on the testing set for a Gaussian kernel SVM*

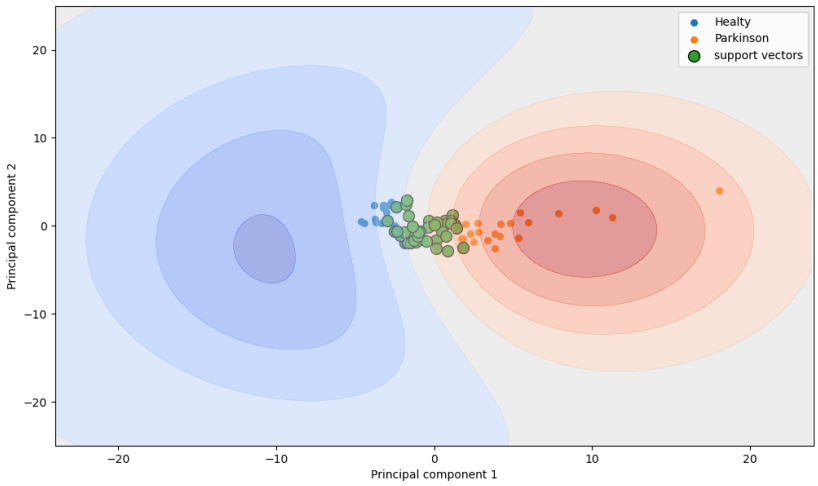

Figure 6 – *Decision region for a Gaussian kernel SVM*

Appendix

Description of the different variables

The **response variable** is *status* : 1 if the subject has the Parkinson disease and 0 if not.

The **explanatory variables** are the following :

- *name* : the suject name along the recording number.
- *mdvp.fo* : the **average** local fundamental frequency (Hz).
- *mdvp.fhi* : the **maximum** local fundamental frequency (Hz).
- *mdvp.flo* : the **minimum** local fundamental frequency (Hz).
- *mdvp.jitter_ perc (%)*, *mdvp.jitter_ abs (Abs)*, *mdvp.rap*, *mdvp.ppq*, *jitter.ddp* : these are several measures of variation in fundamental frequency.
- *mdvp.apq*, *mdvp.shimmer*, *mdvp.shimmer_ db*, *shimmer.apq3*, *shimmer.apq5*, *shimmer.dda* : these are several measures of variation in amplitude.
- *nhr*, *hnr* : 2 measures of noise to tonal components in the voice.
- *rpde*, *d2* : 2 nonlinear dynamical complexity measures.
- *dfa* : signal fractal scaling exponent.
- *spread1, spread2*, *ppe* : 3 nonlinear measures of fundamental frequency variation.