Universitat Politècnica de Catalunya

Grau en Ciència i Enginyeria de Dades

Prediction of UPDRS scores in Parkinson's patients from voice recordings

AA1-Machine learning (GCED)

Sonia Castro & Alba Perna

10 maig 2022

Table of contents

1 Introduction

1.1 Goals of this project

For our project we have chosen a dataset that contains a series of biomedical voice measurements of 42 people with early-stage Parkinson's disease. These people were part of a study that monitored the progression of symptoms remotely for six months with a remote monitoring device.

The main goal is to predict motor and total UPDRS scores from 16 voice measurements. The unified Parkinson's disease rating scale (UPDRS) is the most widely used scale in the clinical study of Parkinson's disease, which is a progressive nervous system disorder that affects movement with symptoms that begin gradually. Although there is no cure for Parkinson's disease, medication may greatly improve symptoms so an early detection may suppose such an important improvement.

The early detection of a pathology is one of the pillars that usually accompany a good functioning of the treatment. An early diagnosis of Parkinson's disease is closely linked to the proper functioning of the treatment to be followed which will greatly facilitate the action of medications and the different existing treatments to alleviate and control the symptoms of Parkinson's.

1.2 Dataset description

The dataset was created by Athanasios Tsanas and Max Little of the University of Oxford, in collaboration with 10 medical centers in the US and Intel Corporation who developed the telemonitoring device to record the speech signals. The dataset can be found in this Repository.

This dataset has a total of 5875 observations and 22 variables which are explained in the following table:

Category	Explanation	Type
subject	Integer that uniquely identifies each subject	Integer
age	Subject age	Integer
sex	Subject gender	Nominal: (0-male, 1-female)
test$_time$	Time since recruitment into the trial. The integer part is the number of days since recruitment.	Integer
motor$_UPDRS$	Clinician's motor UPDRS score, linearly interpolated	Decimal
total$_UPDRS$	Clinician's total UPDRS score, linearly interpolated	Decimal
Jitter(%), Jitter(Abs), Jitter(RAP), Jitter:PPQ5, Jitter:DDP	Several measures of variation in fundamental frequency	Decimal
Shimmer, Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA	Several measures of variation in amplitude	Decimal
NHR, HNR	Measures of ratio of noise to tonal components in the voice	Decimal
RPDE	A nonlinear dynamical complexity measure	Decimal
DFA	Signal fractal scaling exponent	Decimal
PPE	A nonlinear measure of fundamental frequency variation	Decimal

Table 1: Description of the variables in the dataset.

2 Data exploration process

2.1 Pre-processing

In this part the data will be pre-processed and visualized so that it will be easier to handle later.

In the dataset description it is specified that the missing values are encoded as “N/A”, so we have checked it and discovered that there are no missing values. After a further inspection of the dataset we have also confirmed that there are not any outliers either, since any of the variables takes an excessively large or small value.

However, we found some incoherent values as some of the test_time values (that measures the elapsed time since the experiment started) were negative. Those samples were the recollected from subjects 34 and 42. At first we thought about erasing those incoherent rows as they were only 12 out of 5875. In the case of subject 34 that is what we did indeed, as the subject already had samples of the 3rd day so we don’t know where those measures on day “-3” belong. Nevertheless, we arrived at the conclusion that the error in subject 42 had been made when introducing the sign, as they were lacking samples from those days. Therefore, in this case we only changed the sign of those values. So the dataset used has 5869 samples instead of 5875 as declared before.

We have also eliminated two redundant variables as their correlation was exactly one, indicating that both provide the same information. The variables were Jitter.RAP and Jitter.DDP, and Shimmer.APQ3 and Shimmer.DDA. We have eliminated Jitter.DDP and Shimmer.DDA.

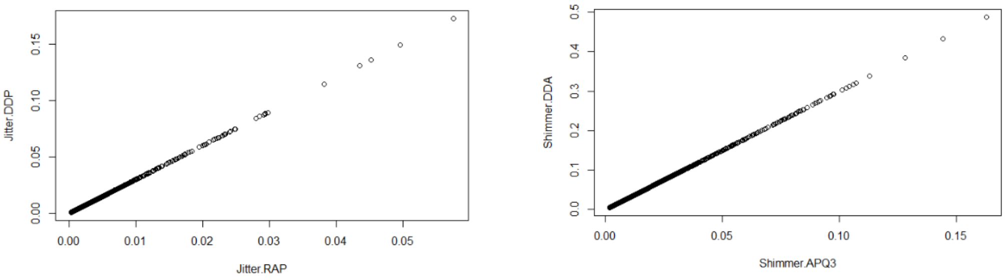


Figure 1: Correlation between Jitter DDP and Jitter.RAP and between Jitter DDA and Jitter.APQ3

We also found out that the numeric variables were not gaussian distributed, so we normalized them using the Box-Cox power transformation.

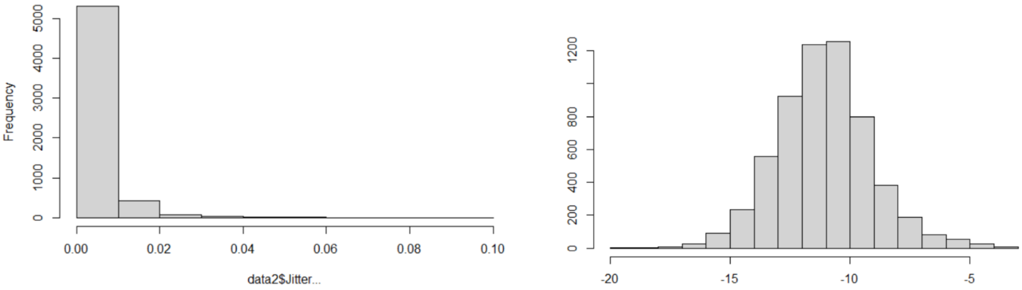


Figure 2: Jitter... variable before and after a Box-Cox power transformation.

We also shuffled the data to avoid possible biases.

2.2 Visualization

The main goal of this section is to make it easier to understand data and visualize possible patterns or trends graphically. Therefore, some graphics will be plotted in order to analyze them.

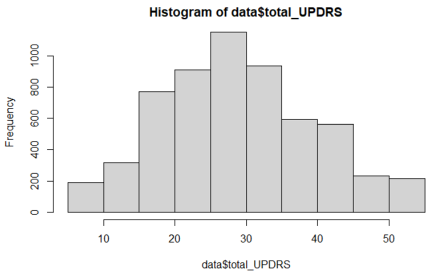


Figure 3: Histogram of total_UPDRS.

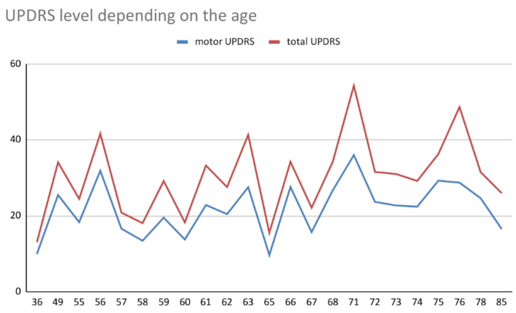


Figure 4: Plot of the total_UPDRS and motor_UPDRS in different ages

On Figure 3, we can observe the distribution of our response variable, later on in this project we will divide it in two classes (0-30) early stage parkinson and (30-60) a more advanced stage.

On figure 4, when plotting the motor and the total UPDRS levels regarding the age, we can see a clear dependence between these two measures since they follow more or less the same pattern for the different ages. Besides, it will be interesting to study how the motor level affects the total one. Furthermore, we can observe that the mean of these two measures increases when the age takes higher values. In spite of that, we can observe a possible outlier for the only subject who is 85 years old, whose level of parkinson is much smaller than expected.

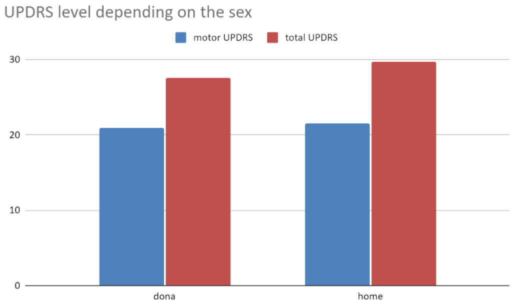


Figure 5: Plot of the total_UPDRS and motor_UPDRS depending on the sex.

We can see how both levels of UPDRS are a little bit lower when the subjects are women, and how the relationship between motor UPDRS and total UPDRS is maintained on both genders.

3 Linear regression

We will first treat the response variable as a numerical variable to estimate the patient's specific value on the UPDRS scale. To achieve this we will use linear regression, specifically the standard, the ridge and the LASSO. We will compare the algorithm's performances by conducting a 10x400 cross-validation using the best hyperparameters (lambda) found previously.

3.1 Standard linear regression

The best model found based on the AIC takes into account the following variables: age, sex, test_time, motor_UPDRS, Jitter..., Jitter.Abs., Jitter.RAP, Jitter.PPQ5, Shimmer, HNR, RPDE Shimmer.APQ5, Shimmer.APQ11, DFA and PPE.

If we look at the p-values we can see that the most significant ones are age, sex and motor_UPDRS (Parkinson's disease scale based only on the examination of motor aspects).This model explains 87.3% of the total_UPDRS value as the Adjusted R-squared shows.

3.2 Ridge regression

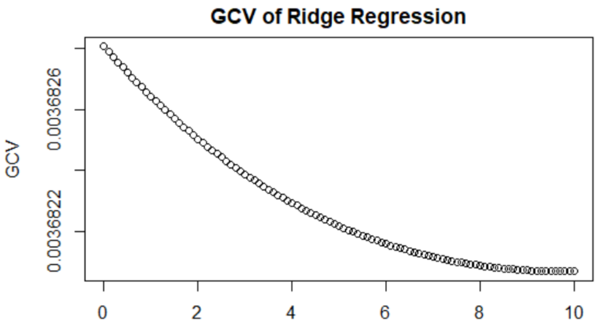


Figure 6: GVC of Ridge Regression.

We choose the best lambda to use before training the model, in this case the best lambda value is 9.8.

3.3 LASSO regression

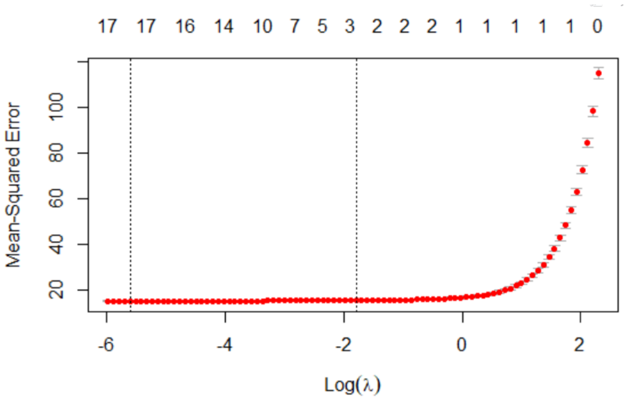


Figure 7: Meaned-Squared Error for different λ values in LASSO regression.

We choose the best lambda to use before training the model, in this case the best lambda value is 0.003676411.

3.4 Best linear regression model

To select the best model we now use a 10X400- CV with our training data. Both the standard and the ridge regression take into account the sex variable (categorical), while the LASSO does not, since it only works with numerical variables.

The mean error for standard linear regression is 0.12763, for ridge linear regression is 0.12761 and for lasso linear regression is 0.12953. All values are pretty close, but we choose as the best model the ridge linear regression.

We train the final model (ridge regression with lambda = 10) on the whole training set. Finally we estimate the generalization error of the final model (using the test set), and the result is 0.122526.

4 Logistic regression

First we will convert the response variable (total UPDRS score) into a categorical of two classes, one for an earlier stage of Parkinson and another for a more advanced stage. The minimum value of the variable total_UPDRS is 7, the maximum 54.99 and the mean 29.01.

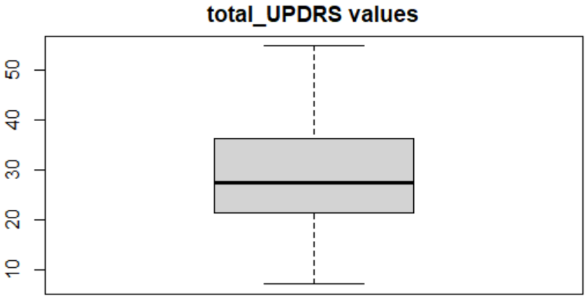


Figure 8: Boxplot of total_UPDRS

So we have finally decided to divide the patients in two groups depending on their total_UPDRS value early_stage (0-30) and late_stage (30-60).

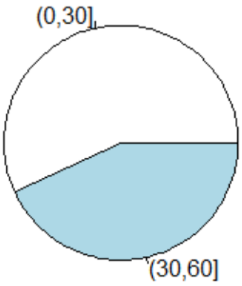


Figure 9: Piechart of total_UPDRS

The first group we have created has 3341 samples and the second group 2528 which won't be a problem since there is not a big imbalance between levels.

We will analyze by means of classification algorithms in which of the two stages of Parkinson a patient is. Specifically we will use: generalized linear model and support vector machine.

4.1 Generalized linear model

We have trained the glm with a train set of 4000 samples and we have chosen the best glm model based on the value of its corresponding AIC. Again the most meaningful variables according to the glm are sex, age and motor_UPDRS.

The logistic regression model has as default P= 0.5 as the probability threshold. If the filter assigns early_stage with probability at least 0.5 we predict early_stage. We have created a function so that we can try different values of P.

If we run the function with P=0.5 we obtain a 4.3 % training error and a 4.22 % test error.

		Prediction	
		late_stage	early_stage
Truth	late_stage	1637	70
	early_stage	102	2191

Table 2: Table of the truth values and predictions with P=0.5 in the trainning set.

		Prediction	
		late_stage	early_stage
Truth	late_stage	786	35
	early_stage	44	1004

Table 3: Table of the truth values and predictions with P=0.5 in the test set.

´

Although the errors are quite low, we should try to lower the probability of predicting earlystage when the patient is in a more advanced stage. Since what we are trying to do is detect the disease in order to act on it as soon as possible, we cannot properly treat a more sicker patient if we believe that they are still in an initial phase of the disease.

We can do this by increasing the value of P. We have decided the best value for P in this case would be 0.7 since the increase in the errors is very low but the probability of misclassifying a late stage of Parkinson's disease is lower.

		Prediction	
		late_stage	early_stage
Truth	late_stage	1673	34
	early_stage	141	2152

Table 4: Table of the truth values and predictions with P=0.7 in the trainning set.

		Prediction	
		late_stage	early_stage
Truth	late_stage	800	21
	early_stage	57	991

Table 5: Table of the truth values and predictions with P=0.7 in the test set.

If we run the function with P=0.7 we obtain a 4.375 % training error and a 4.173 % test error. As we can see, the training error is practically the same and the test error has even decreased. And we get a much better Parkinson's disease stage filter.

4.2 KNN

For the following project methods we will use the attached jupyter notebook (pyhton) instead of R studio. We divide our previously preprocessed data into three groups: train (2000 samples), validation (2000) and test (1869).

We will take as our baseline the KNN classifier model with the default parameters. This model can be a good baseline for comparing with SVM because it is simple, fast, interpretable and distance based. We try to divide the data in two clusters with it, to see if it can predict the stage of the parkinson disease.

The result is quite good since for the validation data it has an accuracy of 0.752922, but it should be better as we are treating a serious disease.

4.3 Support vector machine

We will use a SVM classifier, the powerful part of this model is using the kernel trick. This model has the next hyperparameters:

- C: Regularization parameter. It will avoid the model to use too many suport vectors.
- Kernel: The function that will modify the original space.

In our case, we will use linear SVM, Radial Basis Function (RBF), Sigmoid, and Polynomial. For each method, we will first run it with the default parameters. Then we will try to optimize the hyperparameters with cv to improve the result.

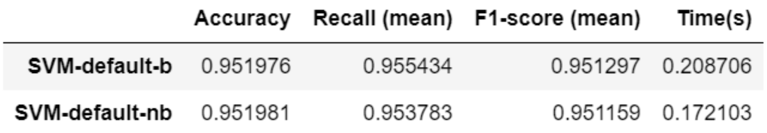
SVM has the advantage that we can use it with unbalanced data. They can weight the C hyperparameter based on the number of samples of each class, penalyzing this way the majoritary classes. So we will have this into account when choosing our models.

To compare between the different models, we will use the value of the accuracy, recall (as having false negative predictions is very harmful in our context) and F1-score.

Recall measures how much the model is predicting correctly a class with respect all the real values of this class : $recall_c == \frac{tp}{tp+fn}$ Where tp are the true positives and fn are the false negatives. This metric is used when having false negative predictions is very harmful, exactly what happens in our case since having a false low prediction may prevent early detection of the disease.

4.3.1 Lineal SVM

For lineal SVM with the default parametres balancig (b) /not balancing (nb) the data we get:



	Accuracy	Recall (mean)	F1-score (mean)	Time(s)
SVM-default-b	0.951976	0.955434	0.951297	0.208706
SVM-default-nb	0.951981	0.953783	0.951159	0.172103

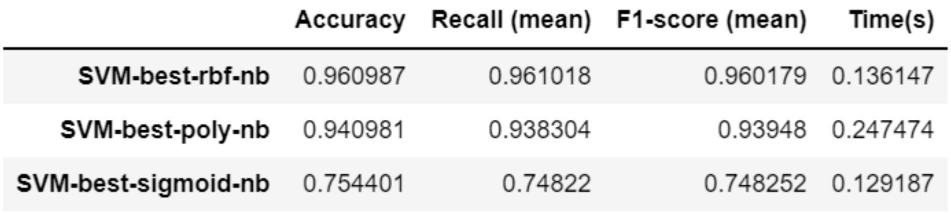
Figure 10: Table with the results metrics for SVM balanced(b) and not (nb).

As we can see the model without balancing the data gives us better validation results (higher Accuracy and Recall), however we will compare both situations also in the non-linear SVM. In both cases we exceed the results obtained with KNN.

If we optimize the parameters we get slightly better results, the best one from the SVM not balanced with an Accuracy of 0.954984 and a Recall of 0.955178.

4.3.2 Non-lineal SVM

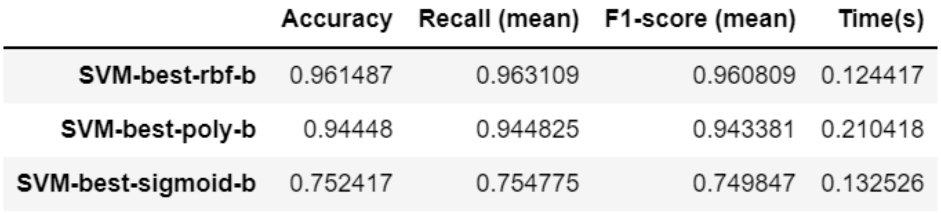
We will now test three nonlinear SVM methods, Radial Basis Function (RBF), Sigmoid, and Polynomial. We optimize its parameters and compare the resulting metrics. For each method we will take into account the balanced version and the unbalanced version.



	Accuracy	Recall (mean)	F1-score (mean)	Time(s)
SVM-best-rbf-nb	0.960987	0.961018	0.960179	0.136147
SVM-best-poly-nb	0.940981	0.938304	0.93948	0.247474
SVM-best-sigmoid-nb	0.754401	0.74822	0.748252	0.129187

Figure 11: Table with the results metrics for RBF, Sigmoid and Polynomial SVM not balanced.

In this case we obtain the best rbf with C = 6, the best polynomial with C = 6 and the best sigmoid with C = 0.1.



	Accuracy	Recall (mean)	F1-score (mean)	Time(s)
SVM-best-rbf-b	0.961487	0.963109	0.960809	0.124417
SVM-best-poly-b	0.94448	0.944825	0.943381	0.210418
SVM-best-sigmoid-b	0.752417	0.754775	0.749847	0.132526

Figure 12: Table with the results metrics for RBF, Sigmoid and Polynomial SVM balanced.

In this case we obtain the best rbf with C = 6, the best polynomial with C = 5 and the best sigmoid with C = 0.1.

As we can see in both cases the best result has been obtained with the optimizied RBF, being the best case when we balance the data. With non-lineal SVM we obtain better results when balancing the data in any of the applied methods.

4.4 Best logistic regression method

As we can see in the table above, the best results are provided by RBF SVM when balancing the data. We run it with the test group to see how well they generalize. We obtain an accuracy of 0.965761 and a recall of 0.967271, so we conclude that the model correctly generalizes.

The last step is to compare it with the GLM previously discuted. As the glm had an error aproximately of the 4% even though we changed P, and the RBF SVM has an error of the 0.0342%, we conclude that the RBF SVM with balanced data is the best logistic regression in our case.

5 Scientific and personal conclusions

In the first case, when treating the response variable as a numerical variable to estimate the patient's specific value on the UPDRS scale, we have arrived at the conclusion that the linear regression model that provides the best results for this data sample is the ridge regression. We obtained a competent value for the generalization error: 0.122626, which will be analogous to an accuracy of approximately 0.877.

On the other hand, we wanted to estimate the degree of parkinson of a patient by dividing the total score into two categories: an early stage (for values between 0 and 30 on the total UPDRS scale) and a late stage (between 30 and 60). We arrived at the conclusion that the best logistic method is the non-lineal SVM, specially for RBF for balanced data, which gave us an accuracy of 0.95761. Besides, this model lowers the probability of predicting an early stage when actually the patient is in a more advanced stage so that we could apply the specific treatment as soon as possible.

When comparing linear and logistic regressions, we can see that we have obtained a better accuracy and a smaller error for the logistic one. However, we have to take into consideration the aim of our research before selecting only one of these models.

For example, if we want to know the specific value of a patient in order to better adapt their personal treatment, we would use regression (as it is not the same having a score of 31 than of 55, although they are both classified as late stages). But to divide the patients of a medical center or a medical study in a simple way into more advanced or more premature stages, we would use classification.

As personal conclusions, we have realized the importance of good data preprocessing since without it, the resulting models and results can be wrong or difficult to achieve. Moreover, the importance of knowing and adapting to the context of the dataset, for example in our case being aware that it is more important to reduce the error of predicting a high level of Parkinson's as low, than the opposite.

Finally, the objective of this project is providing a good evaluation of the Parkinson's level of new patients from the voice data collected from previous patients, all this through the use of statistical algorithms and machine learning techniques. We have seen how the development of these models capable of predicting with a certain precision can be crucial for medical, scientific and economic reasons, among others.

6 References

Information about ridge regression: [link$_1$, link$_2$, link$_3$]

Information about ridge and LASSO regression: [link$_4$]

Information about LASSO regression: [link$_5$, link$_6$]

Information about CV: [link$_7$]

Information about glm: [link$_8$, link$_9$]

Information about converting a numerical variable into a category: [link$_{10}$]

Infomration about how to shuffle rows of a dataset in R: [link$_{11}$]

Information about reading and writting data with R: [link$_{12}$, link$_{13}$]

Documentation about SVM: [link$_{12}$]

Documentation about RBF SVM in Python: [link$_{13}$]

Information about the most used non-linear SVM: [link$_{14}$, link$_{15}$]

Information about encoding categorical variables in Python: [link$_{16}$, link$_{17}$]

Information about F1-score [link$_{18}$]

Uundoubtedly one of the main sources of information for acomplishing this project have been the presentations and notes given in class of Machine Learning 1 (AA1) and others subjects in the course such as Mathematical optimization (OM), wich gave a good insight of SVM methods.