Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

Parkinson Prediction with logistic regression

Introduction

Machine learning is one of the most important tendences of future technologies, it is used to make machines learn with different techniques, one of them is Logistic Regression which is a statistical method for predicting binary classes, it is also one of the simplest and most widely used Machine Learning algorithms for classifying two classes. For this project I created a model to predict if a person has Parkinson or not by using different medical exam results. As it is said, this model will require a boolean result, that is why i decided to use logistic regression for my prediction.

To understand how this prediction works you need to understand why this was an interesting topic from the medical point of view. Parkinson disease is a central nervous system disorder that affects movement and often causes tremors. Damage to nerve cells in the brain causes a drop in dopamine levels, which causes the symptoms of Parkinson's disease. Parkinson's disease usually begins with a tremor in one hand. Other symptoms are slow movement, stiffness, and loss of balance. Medicines can only control Parkinson's symptoms.

Justification

The main reason why i decided to make a prediction about this topic is because Parkinson's disease is difficult to diagnose because there is no specific test for the condition. The symptoms of Parkinson's disease vary from person to person and a number of other diseases have similar symptoms. For these reasons, incorrect diagnoses are sometimes made and that ends up with incorrect medication that can cause several problems in future. This model will help doctors to make the right diagnosis and reduce false positive results.

Objective

Predict if a person has Parkinson and analyse the difference between framework results and by hand results.

Procedure

For this process i downloaded a Parkinson database from oxford university, this dataset used the next attributes:

- name - ASCII subject name and recording number
- MDVP:Fo(Hz) - Average vocal fundamental frequency
- MDVP:Fhi(Hz) - Maximum vocal fundamental frequency
- MDVP:Flo(Hz) - Minimum vocal fundamental frequency
- MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP - Several measures of variation in fundamental frequency
- MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude
- NHR,HNR - Two measures of ratio of noise to tonal components in the voice
- status - Health status of the subject (one) - Parkinson's, (zero) - healthy

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

- RPDE,D2 - Two nonlinear dynamical complexity measures
- DFA - Signal fractal scaling exponent
- spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation

As we can see i have a couple of features that are similar, this means these features can be correlated so i need to check what are the columns that i can drop. Thanks to a heatmap I can know which are the ones that i don't need.

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

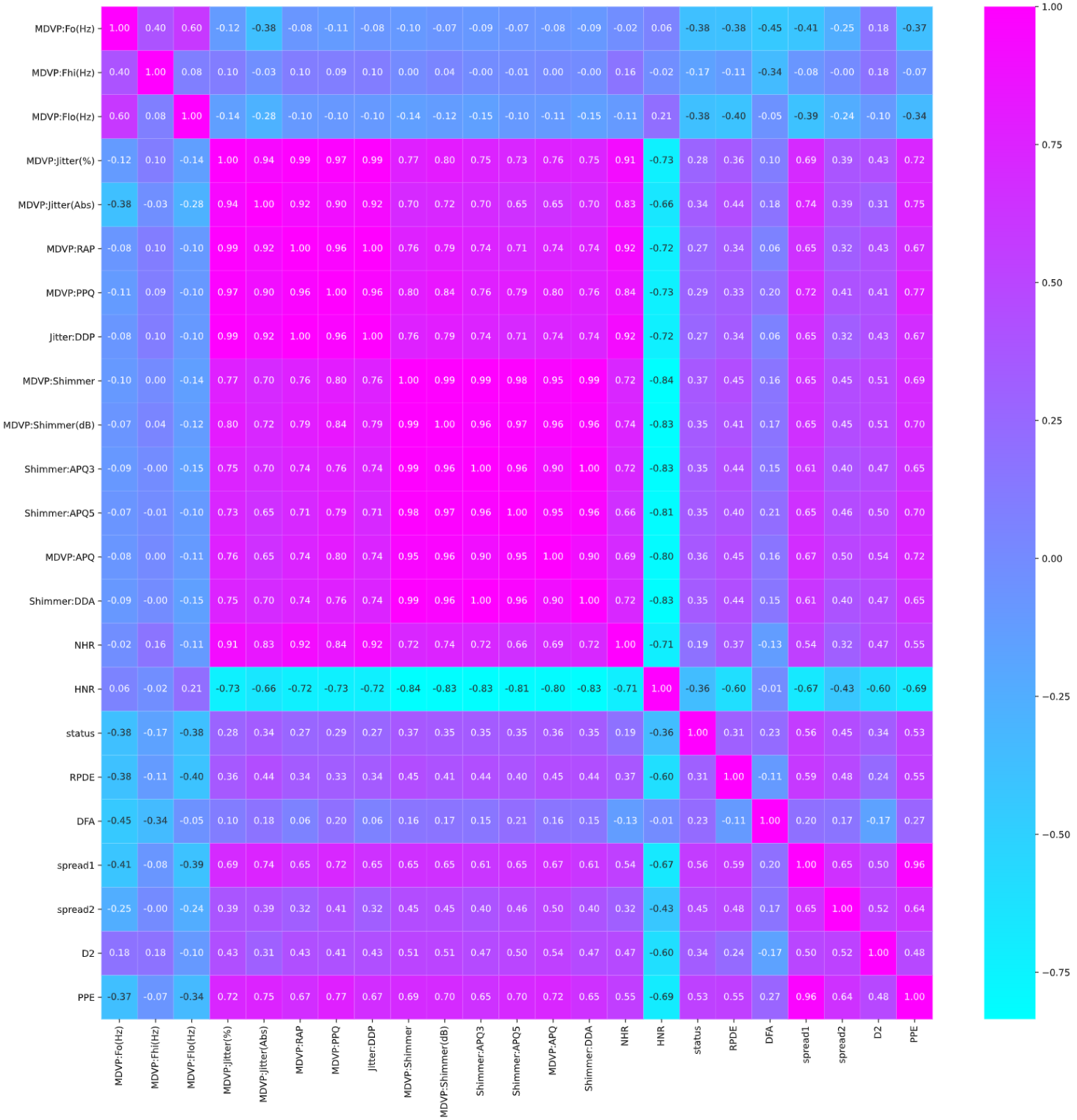

Figure 1. Heat map for parkinson database.

After this process I proceed to drop the columns that have a value of 1.0 inside the heat map. Then I analysed what are the possible cases for all patients. I ended up with 147 Parkinson cases and 48 healthy patients.

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

```
Counting positive and negative values:
1    147
0     48
```

Figure 2. Positive and negative values for diagnosis.

Now we checked that this is a small data set so the solver liblinear should be enough to get a good prediction. Also I decided to split my model into 20% test and 80% train. For my first run I used 30% and 70% but i realized that 20% and 80% were more accurate.

```
Calculated coeficients are:
[[-3.12945840e-03 -1.38219743e-02  8.99852607e-02  6.32404830e-04
   5.49755320e-02  5.49284397e-02  3.90196712e-01  5.63585898e-01
   1.71030282e+00  1.99198138e+00]]

Calculated intercept:
[1.60328106]
```

Figure 3. Calculated coefficients and intercept for framework logistic regression.

Once I had both variables calculated I wanted to know how accurate my model was so I printed the confusion matrix of my test. The results were satisfactory because at the end i got 3 true positives, 30 true negatives, 4 false positives and 2 false negatives. This means my model accuracy is .8461

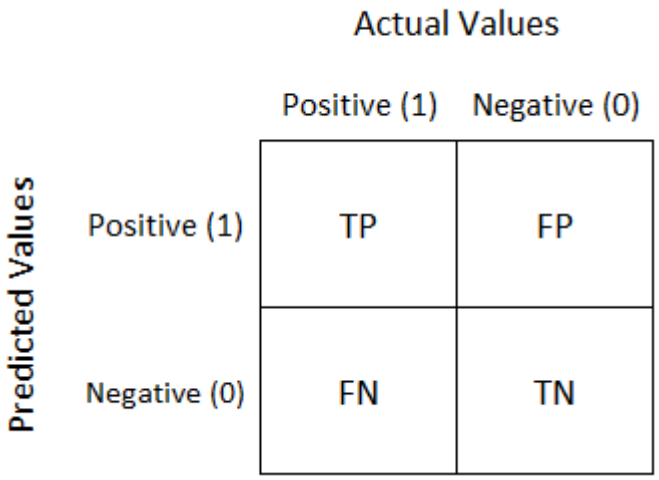

Figure 4. Confusion matrix model.

```
Confusion matrix results for test model 20%
[[ 3  4]
 [ 2 30]]
Accuracy score for Logistic Regresion with framework is 0.846154
```

Figure 5. Confusion matrix for the test model.

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project
Now I want to compare my framework model with another made by hand because i want to see which one is more accurate.

To make a logistic regression model by hand I only used a numpy library to solve all my equations. In this case I will use the sigmoid function to predict the boolean value which represents the status of my patient disease. I decided to use a learning rate of .0001 as default and 5000 iterations.

$$\hat{y} = h_\theta(x) = \frac{1}{1 + e^{-wx+b}}$$

Figure 6. Logistic regression model function.

Sigmoid Function

$$s(x) = \frac{1}{1 + e^{-x}}$$

Figure 7. Sigmoid function.

$$J'(\theta) = \begin{bmatrix} \frac{dJ}{dw} \\ \frac{dJ}{db} \end{bmatrix} = [\dots] = \begin{bmatrix} \frac{1}{N}\sum 2x_i(\hat{y} - y_i) \\ \frac{1}{N}\sum 2(\hat{y} - y_i) \end{bmatrix}$$

Figure 8. Gradient descent function.

I initialize my variables and then I apply the gradient descent, at the end I make the prediction and print the accuracy with the confusion matrix.

```
Confusion matrix results for hand test model 20%
[[ 3  4]
 [ 2 30]]
LR by hand accuracy: 0.8205128205128205
```

Figure 9. LR

```
LR by hand accuracy: 0.8205128205128205
```

Figure 10. LR by hand accuracy.

Results
Comparing both models with the same random input data:

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

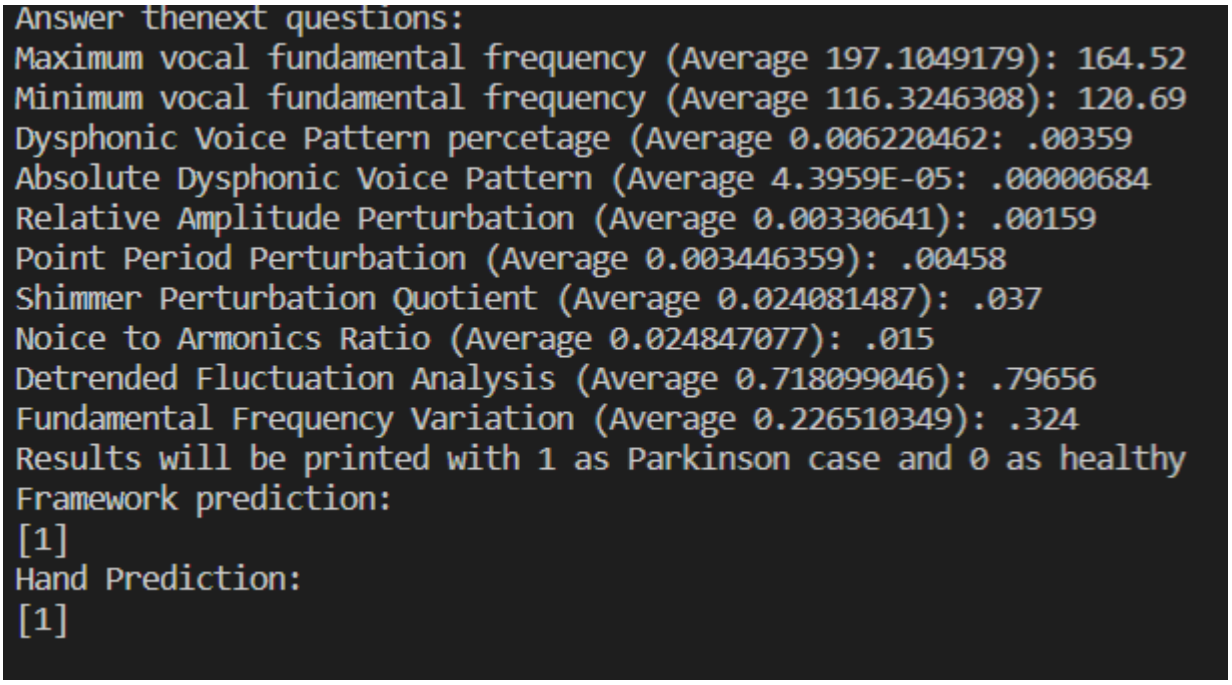

Figure 11. Random input predictions.

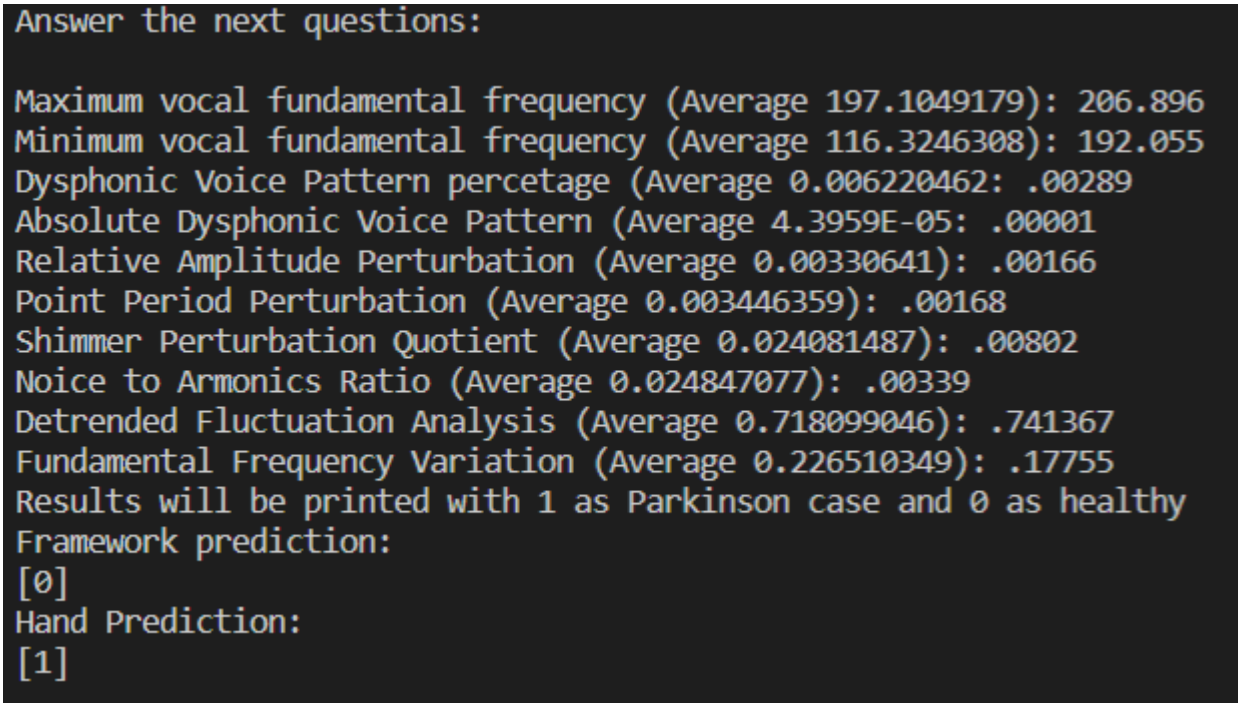

Figure 12. Healthy input data predictions.

After the comparison I can conclude that the framework process is a little bit more accurate than the one programmed by hand.

Bibliography:

- Oxford University, O. (2021). UCI Machine Learning Repository: Parkinsons Data Set. Retrieved 21 April 2021, from https://archive.ics.uci.edu/ml/datasets/parkinsons
- ¿Cómo se Diagnostica la enfermedad de Parkinson? | Parkinson y yo. (2021). Retrieved 21 April 2021, from http://terapiaparkinson.com/como-se-diagnostica-la-enfermedad-de-parkinson/