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

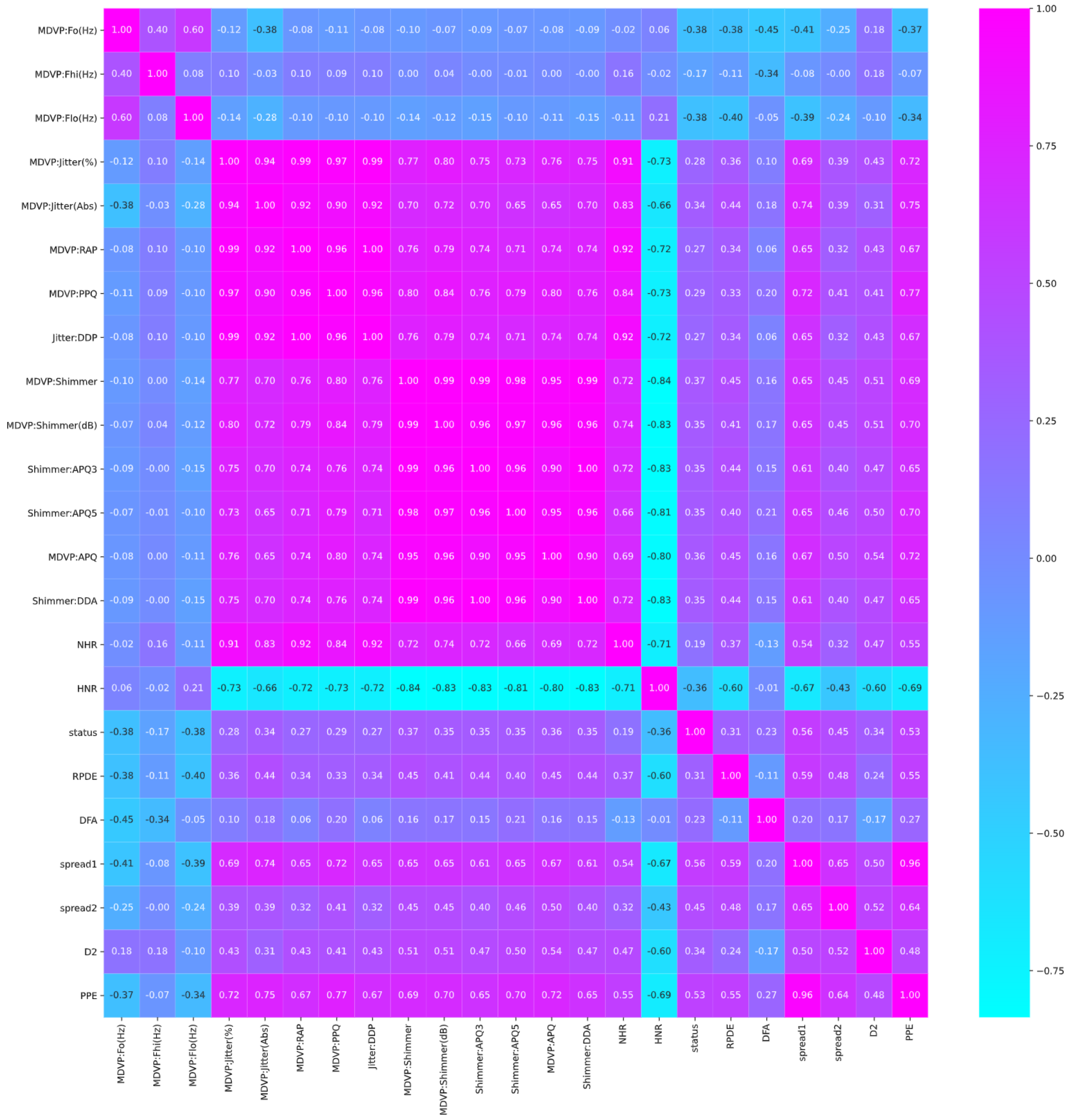
For this process i downloaded a Parkinson database from oxford university, this dataset used the next attributes:

- 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.

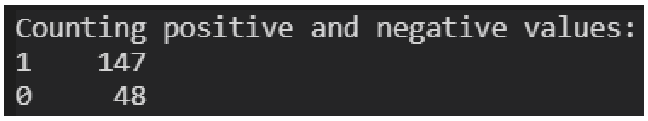


	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(dB)	Shimmer:APQ3	Shimmer:APQ5	MDVP:APQ	Shimmer:DDA	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	PPE
MDVP:Fo(Hz)	1.00	0.40	0.60	-0.12	-0.38	-0.08	-0.11	-0.08	-0.10	-0.07	-0.09	-0.07	-0.08	-0.09	-0.02	0.06	-0.38	-0.38	-0.45	-0.41	-0.25	0.18	-0.37
MDVP:Fhi(Hz)	0.40	1.00	0.08	0.10	-0.03	0.10	0.09	0.10	0.00	0.04	-0.00	-0.01	0.00	-0.00	0.16	-0.02	-0.17	-0.11	-0.34	-0.08	-0.00	0.18	-0.07
MDVP:Flo(Hz)	0.60	0.08	1.00	-0.14	-0.28	-0.10	-0.10	-0.10	-0.14	-0.12	-0.15	-0.10	-0.11	-0.15	-0.11	0.21	-0.38	-0.40	-0.05	-0.39	-0.24	-0.10	-0.34
MDVP:Jitter(%)	-0.12	0.10	-0.14	1.00	0.94	0.99	0.97	0.99	0.77	0.80	0.75	0.73	0.76	0.75	0.91	-0.73	0.28	0.36	0.10	0.69	0.39	0.43	0.72
MDVP:Jitter(Abs)	-0.38	-0.03	-0.28	0.94	1.00	0.92	0.90	0.92	0.70	0.72	0.70	0.65	0.65	0.70	0.83	-0.66	0.34	0.44	0.18	0.74	0.39	0.31	0.75
MDVP:RAP	-0.08	0.10	-0.10	0.99	0.92	1.00	0.96	1.00	0.76	0.79	0.74	0.71	0.74	0.74	0.92	-0.72	0.27	0.34	0.06	0.65	0.32	0.43	0.67
MDVP:PPQ	-0.11	0.09	-0.10	0.97	0.90	0.96	1.00	0.96	0.80	0.84	0.76	0.79	0.80	0.76	0.84	-0.73	0.29	0.33	0.20	0.72	0.41	0.41	0.77
Jitter:DDP	-0.08	0.10	-0.10	0.99	0.92	1.00	0.96	1.00	0.76	0.79	0.74	0.71	0.74	0.74	0.92	-0.72	0.27	0.34	0.06	0.65	0.32	0.43	0.67
MDVP:Shimmer	-0.10	0.00	-0.14	0.77	0.70	0.76	0.80	0.76	1.00	0.99	0.99	0.98	0.95	0.99	0.72	-0.84	0.37	0.45	0.16	0.65	0.45	0.51	0.69
MDVP:Shimmer(dB)	-0.07	0.04	-0.12	0.80	0.72	0.79	0.84	0.79	0.99	1.00	0.96	0.97	0.96	0.96	0.74	-0.83	0.35	0.41	0.17	0.65	0.45	0.51	0.70
Shimmer:APQ3	-0.09	-0.00	-0.15	0.75	0.70	0.74	0.76	0.74	0.99	0.96	1.00	0.96	0.90	1.00	0.72	-0.83	0.35	0.44	0.15	0.61	0.40	0.47	0.65
Shimmer:APQ5	-0.07	-0.01	-0.10	0.73	0.65	0.71	0.79	0.71	0.98	0.97	0.96	1.00	0.95	0.96	0.66	-0.81	0.35	0.40	0.21	0.65	0.46	0.50	0.70
MDVP:APQ	-0.08	0.00	-0.11	0.76	0.65	0.74	0.80	0.74	0.95	0.96	0.90	0.95	1.00	0.90	0.69	-0.80	0.36	0.45	0.16	0.67	0.50	0.54	0.72
Shimmer:DDA	-0.09	-0.00	-0.15	0.75	0.70	0.74	0.76	0.74	0.99	0.96	1.00	0.96	0.90	1.00	0.72	-0.83	0.35	0.44	0.15	0.61	0.40	0.47	0.65
NHR	-0.02	0.16	-0.11	0.91	0.83	0.92	0.84	0.92	0.72	0.74	0.72	0.66	0.69	0.72	1.00	-0.71	0.19	0.37	-0.13	0.54	0.32	0.47	0.55
HNR	0.06	-0.02	0.21	-0.73	-0.66	-0.72	-0.73	-0.72	-0.84	-0.83	-0.83	-0.81	-0.80	-0.83	-0.71	1.00	-0.36	-0.60	-0.01	-0.67	-0.43	-0.60	-0.69
status	-0.38	-0.17	-0.38	0.28	0.34	0.27	0.29	0.27	0.37	0.35	0.35	0.35	0.36	0.35	0.19	-0.36	1.00	0.31	0.23	0.56	0.45	0.34	0.53
RPDE	-0.38	-0.11	-0.40	0.36	0.44	0.34	0.33	0.34	0.45	0.41	0.44	0.40	0.45	0.44	0.37	-0.60	0.31	1.00	-0.11	0.59	0.48	0.24	0.55
DFA	-0.45	-0.34	-0.05	0.10	0.18	0.06	0.20	0.06	0.16	0.17	0.15	0.21	0.16	0.15	-0.13	-0.01	0.23	-0.11	1.00	0.20	0.17	-0.17	0.27
spread1	-0.41	-0.08	-0.39	0.69	0.74	0.65	0.72	0.65	0.65	0.65	0.61	0.65	0.67	0.61	0.54	-0.67	0.56	0.59	0.20	1.00	0.65	0.50	0.96
spread2	-0.25	-0.00	-0.24	0.39	0.39	0.32	0.41	0.32	0.45	0.45	0.40	0.46	0.50	0.40	0.32	-0.43	0.45	0.48	0.17	0.65	1.00	0.52	0.64
D2	0.18	0.18	-0.10	0.43	0.31	0.43	0.41	0.43	0.51	0.51	0.47	0.50	0.54	0.47	0.47	-0.60	0.34	0.24	-0.17	0.50	0.52	1.00	0.48
PPE	-0.37	-0.07	-0.34	0.72	0.75	0.67	0.77	0.67	0.69	0.70	0.65	0.70	0.72	0.65	0.55	-0.69	0.53	0.55	0.27	0.96	0.64	0.48	1.00

Figure 1. Heat map for parkinson database.

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

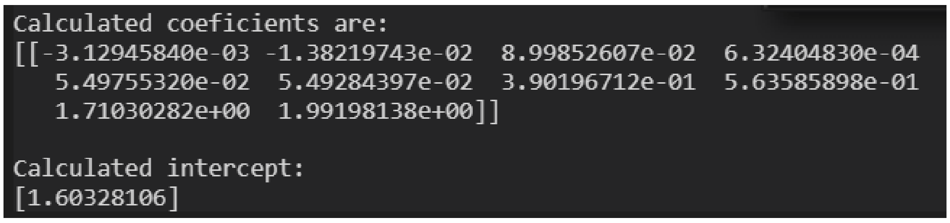
After this process I proceed to drop the columns that have more than one value of 1.0 or close inside the heat map (light pink). Then I analysed what are the possible cases for all patients. I ended up with 147 Parkinson cases and 48 healthy patients.



```
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 because the number of used attributes for this prediction are 10 which is a low value. 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

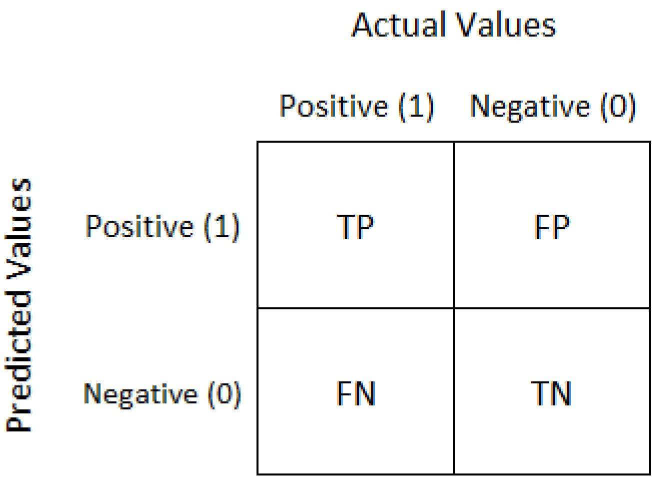
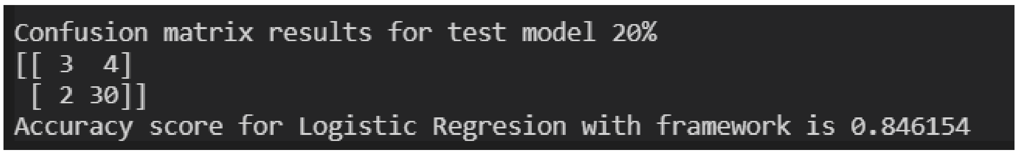


Figure 4. Confusion matrix model.



```
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 with framework, random state=42 and test model size =.20.

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's disease. I decided to use a learning rate of .01 as default and 1000 iterations.

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

Figure 6. Logistic regression model function.

Sigmoid Function

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

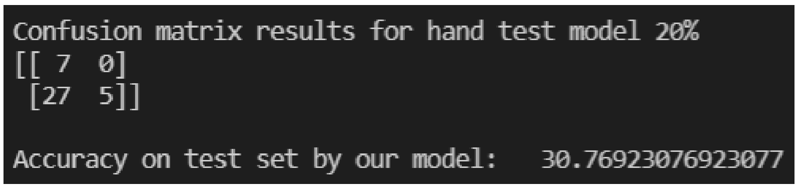
Figure 7. Sigmoid function.

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

Figure 8. Gradient descent function.

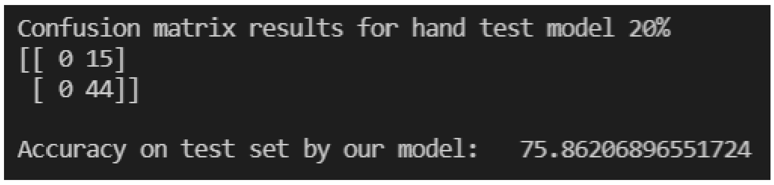
I initialize my variables and then I apply the gradient descent, at the end I make the prediction and print the accuracy with the confusion matrix.

At my first attempt I realized my model was really inaccurate if I used the same conditions for the framework example. After that I decided to make different examples, making a variation of random state and the size of the test model.



```
Confusion matrix results for hand test model 20%
[[ 7  0]
 [27  5]]

Accuracy on test set by our model:   30.76923076923077
```

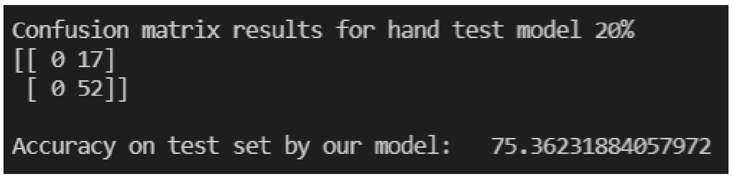
Figure 9. Confusion matrix for hand model with random state = 42 and test model size of 20%.



```
Confusion matrix results for hand test model 20%
[[ 0 15]
 [ 0 44]]

Accuracy on test set by our model:   75.86206896551724
```

Figure 10. Confusion matrix for hand model with random state = 42 and test model size of 30%.

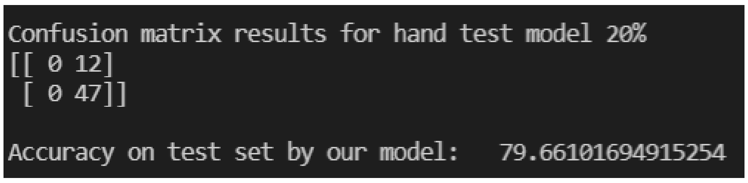
Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project



```
Confusion matrix results for hand test model 20%
[[ 0 17]
 [ 0 52]]

Accuracy on test set by our model:   75.36231884057972
```

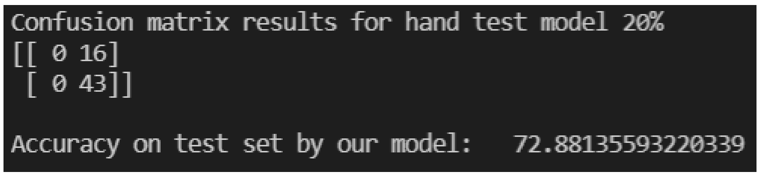
Figure 11. Confusion matrix for hand model with random state = 42 and test model size of 35%.

Accuracy of 75.86% was the highest value I could get changing only the test model size but i wanted to have at least an accuracy of 80% so I decided to change the random state value and keep the test size value equal to 30%.



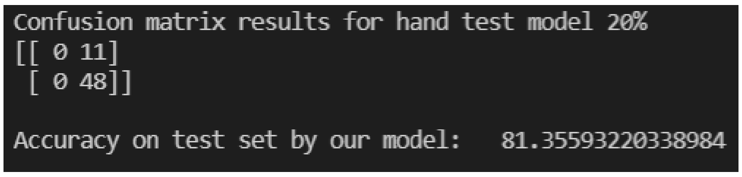
```
Confusion matrix results for hand test model 20%
[[ 0 12]
 [ 0 47]]

Accuracy on test set by our model:   79.66101694915254
```

Figure 12. Confusion matrix for hand model with random state = 100 or 60 and test model size of 30%.



```
Confusion matrix results for hand test model 20%
[[ 0 16]
 [ 0 43]]

Accuracy on test set by our model:   72.88135593220339
```

Figure 13. Confusion matrix for hand model with random state = 101 and test model size of 30%.



```
Confusion matrix results for hand test model 20%
[[ 0 11]
 [ 0 48]]

Accuracy on test set by our model:   81.35593220338984
```

Figure 14. Confusion matrix for hand model with random state = 30 and test model size of 30%.

A value of 81.35% of accuracy was the highest value I could get so I decided to leave values of random state = 30 and test model size of 30% only for hand prediction, then I modified print information.

But I saw a big error, even though my model gives me a high accuracy that is not the best way to calculate because if my model have a really good behaviour by predicting true negatives it had a really bad behaviour by predicting true negatives, in fact my model wasn't able to predict true positives like my confusion matrix says in value [1,1].

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project

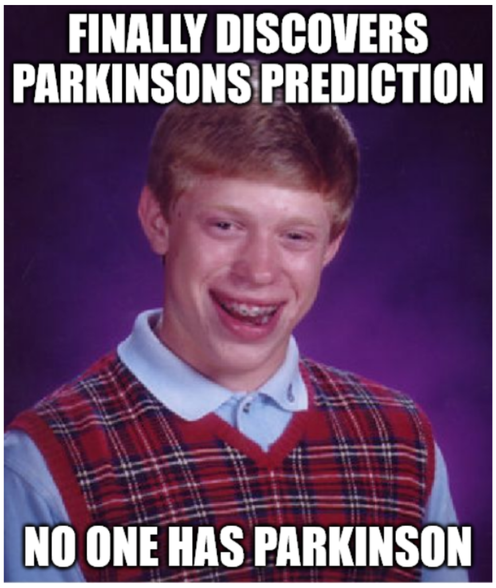
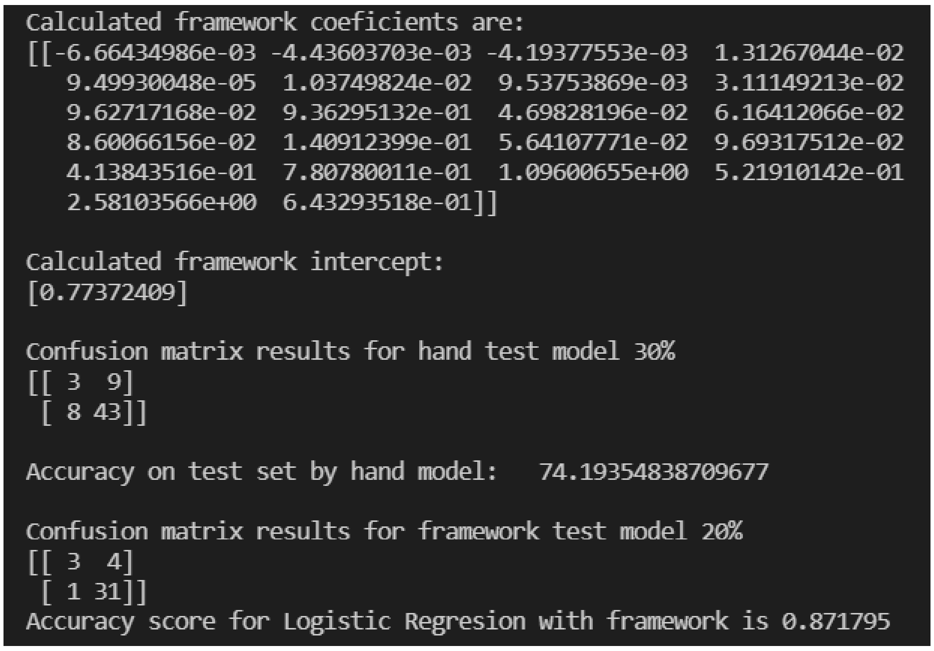


Figure 15. Accurate meme for project situation.

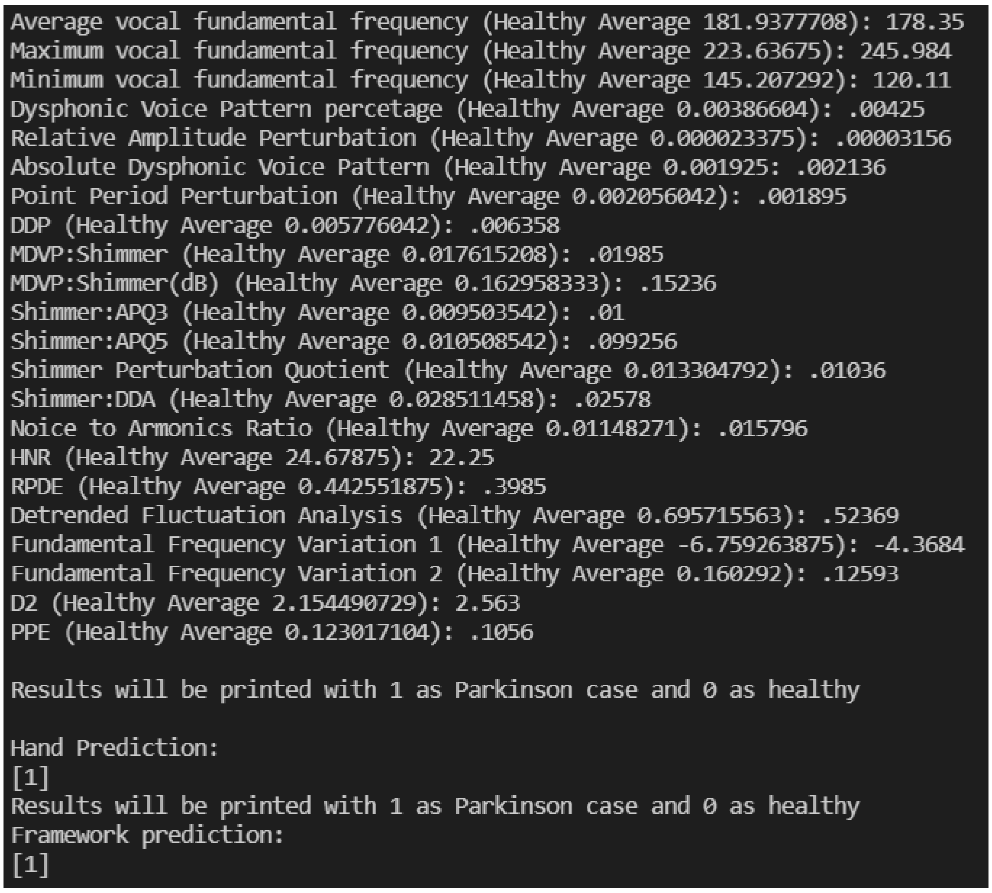
I started to look for reasons why my model was acting so unpredictable and I realized my hand model was highly biased because I had less positive values than negative values and by deleting correlated attributes I was making this problem worse because if my model had little information from the beginning I was taking out most of that information. I decided to apply an Oversampling technique with SMOTE(Synthetic Minority Oversampling Technique). With SMOTE, the minority class is over-sampled by creating “synthetic” examples rather than by over-sampling with replacement. These introduced synthetic examples are based along the line segments joining a defined number of k minority class nearest neighbors, which in the imblearn package is set at five by default. This method let me increase my real hand model accuracy.



```
Calculated framework coeficients are:
[[-6.66434986e-03 -4.43603703e-03 -4.19377553e-03  1.31267044e-02
   9.49930048e-05  1.03749824e-02  9.53753869e-03  3.11149213e-02
   9.62717168e-02  9.36295132e-01  4.69828196e-02  6.16412066e-02
   8.60066156e-02  1.40912399e-01  5.64107771e-02  9.69317512e-02
   4.13843516e-01  7.80780011e-01  1.09600655e+00  5.21910142e-01
   2.58103566e+00  6.43293518e-01]]

Calculated framework intercept:
[0.77372409]

Confusion matrix results for hand test model 30%
[[ 3  9]
 [ 8 43]]

Accuracy on test set by hand model:   74.19354838709677

Confusion matrix results for framework test model 20%
[[ 3  4]
 [ 1 31]]
Accuracy score for Logistic Regresion with framework is 0.871795
```

Figure 16. Final results for hand and framework models. Hand model accuracy = 74.19% vs Framework model accuracy = 87.17%.

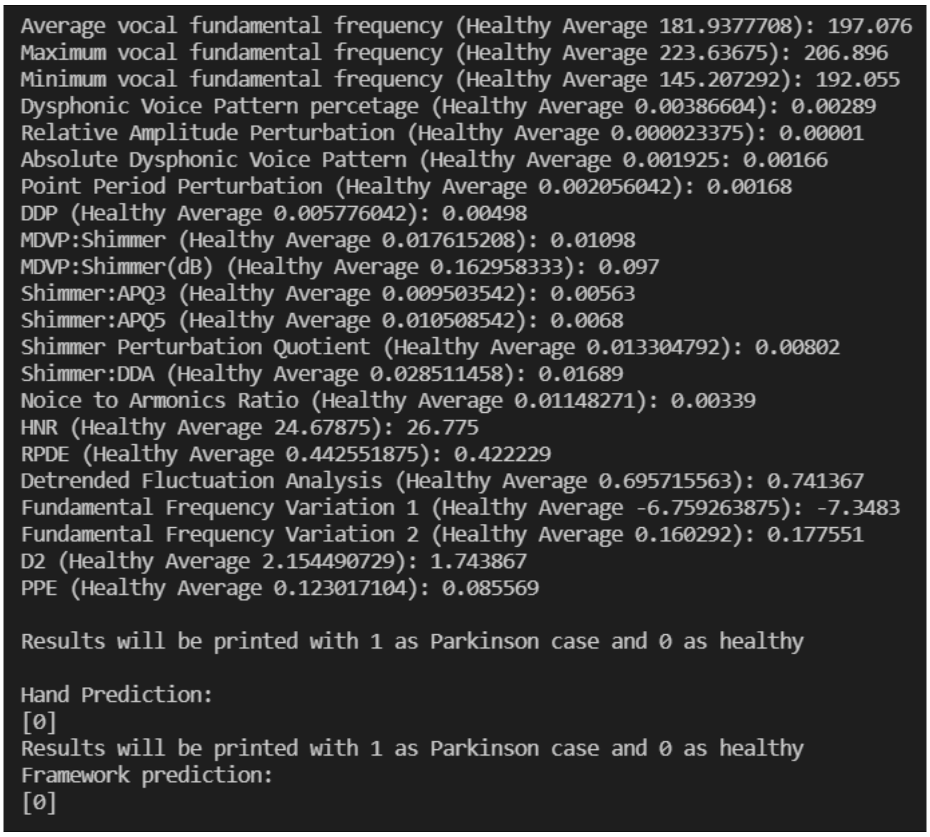
Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project
Results
Comparing both models with the same input data:



```
Average vocal fundamental frequency (Healthy Average 181.9377708): 178.35
Maximum vocal fundamental frequency (Healthy Average 223.63675): 245.984
Minimum vocal fundamental frequency (Healthy Average 145.207292): 120.11
Dysphonic Voice Pattern percetage (Healthy Average 0.00386604): .00425
Relative Amplitude Perturbation (Healthy Average 0.000023375): .00003156
Absolute Dysphonic Voice Pattern (Healthy Average 0.001925: .002136
Point Period Perturbation (Healthy Average 0.002056042): .001895
DDP (Healthy Average 0.005776042): .006358
MDVP:Shimmer (Healthy Average 0.017615208): .01985
MDVP:Shimmer(dB) (Healthy Average 0.162958333): .15236
Shimmer:APQ3 (Healthy Average 0.009503542): .01
Shimmer:APQ5 (Healthy Average 0.010508542): .099256
Shimmer Perturbation Quotient (Healthy Average 0.013304792): .01036
Shimmer:DDA (Healthy Average 0.028511458): .02578
Noice to Armonics Ratio (Healthy Average 0.01148271): .015796
HNR (Healthy Average 24.67875): 22.25
RPDE (Healthy Average 0.442551875): .3985
Detrended Fluctuation Analysis (Healthy Average 0.695715563): .52369
Fundamental Frequency Variation 1 (Healthy Average -6.759263875): -4.3684
Fundamental Frequency Variation 2 (Healthy Average 0.160292): .12593
D2 (Healthy Average 2.154490729): 2.563
PPE (Healthy Average 0.123017104): .1056

Results will be printed with 1 as Parkinson case and 0 as healthy

Hand Prediction:
[1]
Results will be printed with 1 as Parkinson case and 0 as healthy
Framework prediction:
[1]
```

Figure 15. Random input predictions.

Andrea Carolina Flores Ramirez
A01350993
Intelligent Systems Project



```
Average vocal fundamental frequency (Healthy Average 181.9377708): 197.076
Maximum vocal fundamental frequency (Healthy Average 223.63675): 206.896
Minimum vocal fundamental frequency (Healthy Average 145.207292): 192.055
Dysphonic Voice Pattern percetage (Healthy Average 0.00386604): 0.00289
Relative Amplitude Perturbation (Healthy Average 0.000023375): 0.00001
Absolute Dysphonic Voice Pattern (Healthy Average 0.001925: 0.00166
Point Period Perturbation (Healthy Average 0.002056042): 0.00168
DDP (Healthy Average 0.005776042): 0.00498
MDVP:Shimmer (Healthy Average 0.017615208): 0.01098
MDVP:Shimmer(dB) (Healthy Average 0.162958333): 0.097
Shimmer:APQ3 (Healthy Average 0.009503542): 0.00563
Shimmer:APQ5 (Healthy Average 0.010508542): 0.0068
Shimmer Perturbation Quotient (Healthy Average 0.013304792): 0.00802
Shimmer:DDA (Healthy Average 0.028511458): 0.01689
Noice to Armonics Ratio (Healthy Average 0.01148271): 0.00339
HNR (Healthy Average 24.67875): 26.775
RPDE (Healthy Average 0.442551875): 0.422229
Detrended Fluctuation Analysis (Healthy Average 0.695715563): 0.741367
Fundamental Frequency Variation 1 (Healthy Average -6.759263875): -7.3483
Fundamental Frequency Variation 2 (Healthy Average 0.160292): 0.177551
D2 (Healthy Average 2.154490729): 1.743867
PPE (Healthy Average 0.123017104): 0.085569

Results will be printed with 1 as Parkinson case and 0 as healthy

Hand Prediction:
[0]
Results will be printed with 1 as Parkinson case and 0 as healthy
Framework prediction:
[0]
```

Figure 16. Healthy input data predictions.

After the comparison I can conclude that the framework process is more accurate than the one programmed by hand, my hypothesis is that the framework process has filters that make my model accurate even though I have few positive values.

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/