UDEM

UNIVERSIDAD DE MONTERREY

Artificial Intelligence

Project 01:

Logistic Classification

'Classifying Patients as suffering from Heart Disease or not, using a Public Health Dataset'

Technical Report

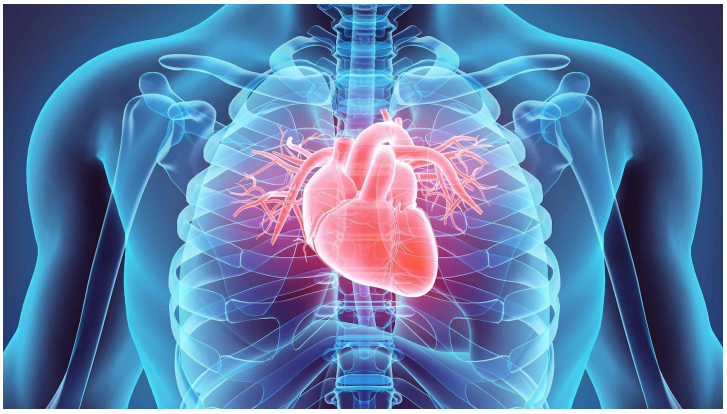

Figure 1. Heart Disease Visual Dramatization.

PROFESSOR: DR. ANDRÉS HERNÁNDEZ GUTIÉRREZ

STUDENTS: Elizondo Benet, Diego
567003

Flores Ramones, Alejandro
537489

Lira Rangel, Karla Alejandra
526389

Martínez Sánchez, Daniel Alberto
534032

Monday November 16th, 2020

We hereby declare that we have worked on this project with academic integrity.

0. Index

1. Introduction **3**

2. Methodology **3**

3.1. General Exploratory Data Analysis 3

3.2. Particular Exploratory Data Analysis 9

3.3. Training 11

4. Experimental Results: Testing **13**

5. Conclusions **15**

6. References **16**

1. Introduction

For this project, the task at hand consists in being able to classify a person or patient into one of two categories: suffering from heart disease, or not suffering from heart disease. To tackle this problem, as it will be explained afterwards in this report with further detail, logistic regression was the method used.

Health Research Australia (2018) succinctly states on their website: 'heart disease is an umbrella term for a range of conditions that affect your heart'. That is, many conditions, such as those related to the blood vessels or the heart's rhythm, among many others, fall under the term 'heart disease'.

The relevance of working with this problem is evident. According to the World Health Organization (2018), heart disease has been the leading cause of death over the past 15 years. Heart disease claimed over 9 million lives during the year of 2016. For this reason, working with problems related to being able to identify people at risk of suffering from heart disease is of great importance.

This project consists in writing a logistic classification algorithm / classifier, which will be trained with data from a real-world heart disease public health dataset obtained from Kaggle. It's worth mentioning that first, said dataset was analyzed in the stage of exploratory data analysis. Following that, the classifier was trained, and in the end it was put to the test with a portion of the dataset not seen during training, that is, the testing data. From this test, the confusion matrix was obtained, as well as several performance metrics derived from those values.

2. Methodology

3.1. General Exploratory Data Analysis

The heart disease dataset used for this project was found on Kaggle (https://www.kaggle.com/ronitf/heart-disease-uci). It contains samples from the following places, recorded by the following doctors:

- Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
- University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
- University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
- V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

The dataset consists of 303 samples obtained from real cases observed by said doctors at the previously mentioned health institutions. It contains 13 different attributes, which are: age, sex, and 11 heart-related health metrics. All of the values in this dataset are numeric, with only one of the features consisting of floating point values, while the remaining 12 consist of whole numbers. Additionally, there are 0 NaN entries throughout the entire dataset.

Now, each feature will be shown, along with its distribution, its minimum and maximum value, its mean and median, as well as the number of NaN values.

1. **Age (x_1)**

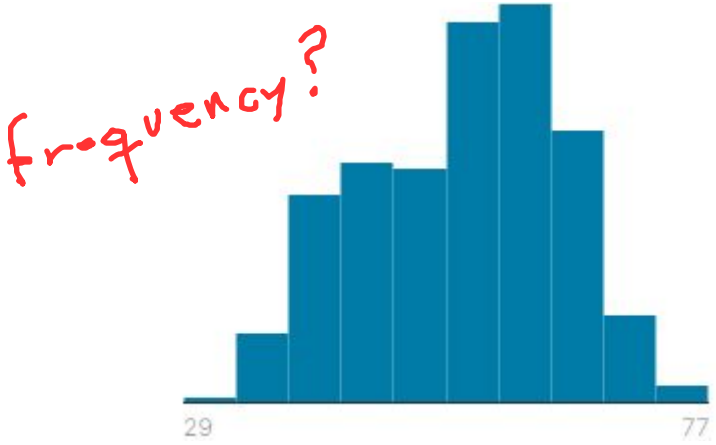

Figure 2. Age distribution.

Aspect	Value
Min	29
Max	77
Mean	55.4
Median	55
# of NaN	0

Table 1. Age statistics.

2. **Sex (x_2)**

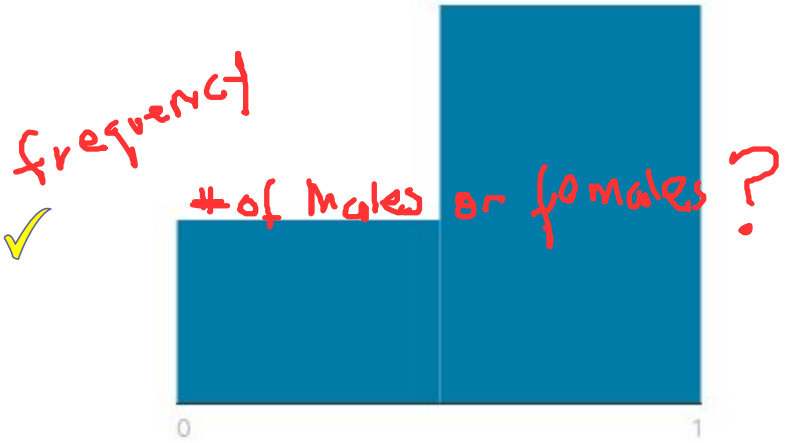

Figure 3. Sex distribution.

Aspect	Value
Min	0
Max	1
Mean	0.68
Median	1
# of NaN	0

Table 2. Sex statistics.

3. **Chest pain type (4 values) (x_3)**

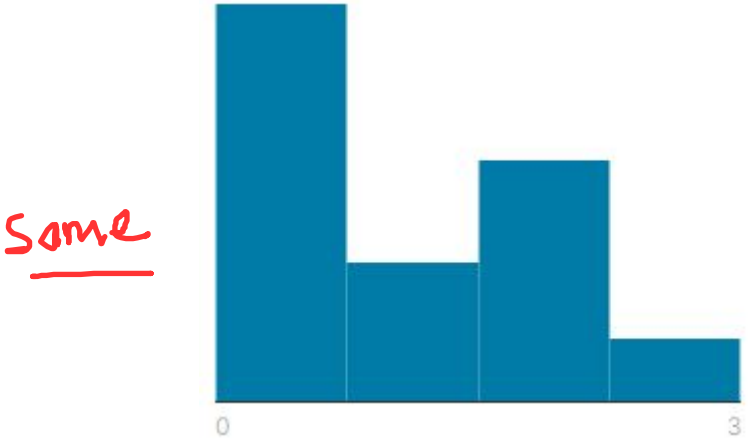

Figure 4. Cp distribution.

Aspect	Value
Min	0
Max	3
Mean	0.97
Median	1
# of NaN	0

Table 3. Cp statistics.

4. **Resting blood pressure (x_4)**

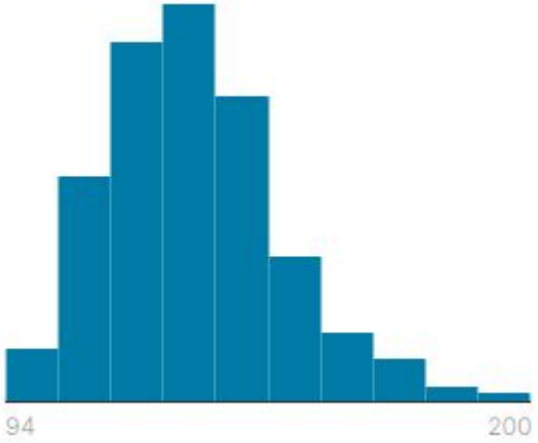

Figure 5. Trestbps distribution.

Aspect	Value
Min	94
Max	200
Mean	132
Median	130
# of NaN	0

Table 4. Trestbps statistics.

5. **Serum cholesterol in mg/dl (x_5)**

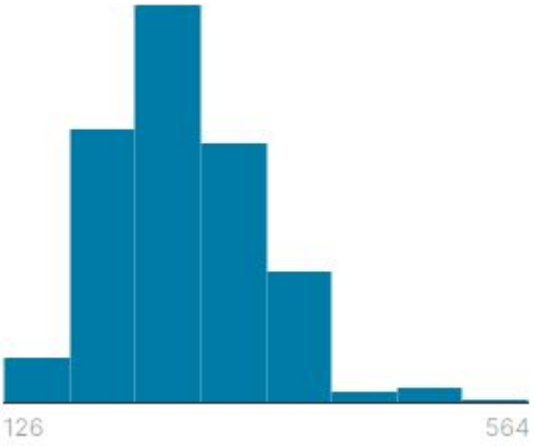

Figure 6. Chol distribution.

Aspect	Value
Min	126
Max	564
Mean	246
Median	240
# of NaN	0

Table 5. Chol statistics.

6. **Fasting blood sugar > 120 mg/dl (x_6)**

Figure 7. Fbs distribution.

Aspect	Value
Min	0
Max	1
Mean	0.15
Median	0
# of NaN	0

Table 6. Fbs statistics.

7. **Resting electrocardiographic results (values 0, 1, 2) (x_7)**

Figure 8. Restecg distribution.

Aspect	Value
Min	0
Max	2
Mean	0.53
Median	1
# of NaN	0

Table 7. Restecg statistics.

8. **Maximum heart rate achieved (x_8)**

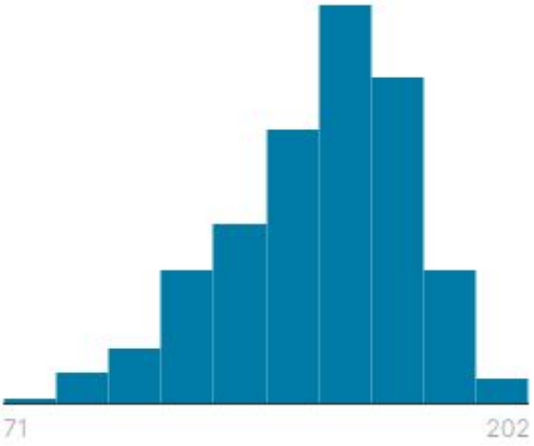

Figure 9. Thalach distribution.

Aspect	Value
Min	71
Max	202
Mean	150
Median	153
# of NaN	0

Table 8. Thalach statistics.

9. **Exercise induced angina (x_9)**

Figure 10. Exang distribution.

Aspect	Value
Min	0
Max	1
Mean	0.33
Median	0
# of NaN	0

Table 9. Exang statistics.

10. Oldpeak = ST depression (electrocardiograph) induced by exercise relative to rest (x_{10})

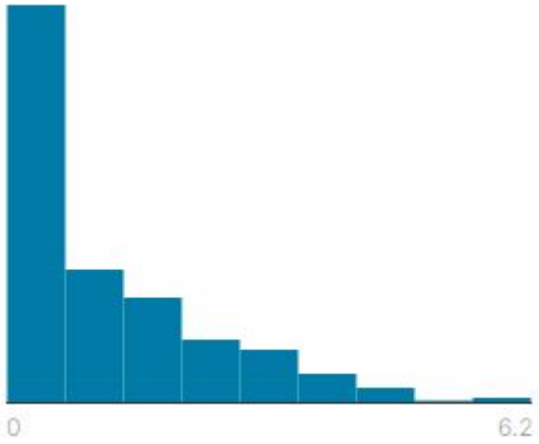

Figure 11. Oldpeak distribution.

Aspect	Value
Min	0
Max	6.2
Mean	1.04
Median	0.8
# of NaN	0

Table 10. Oldpeak statistics.

11. The slope of the peak exercise ST segment (x_{11})

Figure 12. Slope distribution.

Aspect	Value
Min	0
Max	2
Mean	1.4
Median	1
# of NaN	0

Table 11. Slope statistics.

12. Number of major vessels (0-3) colored by fluoroscopy (x_{12})

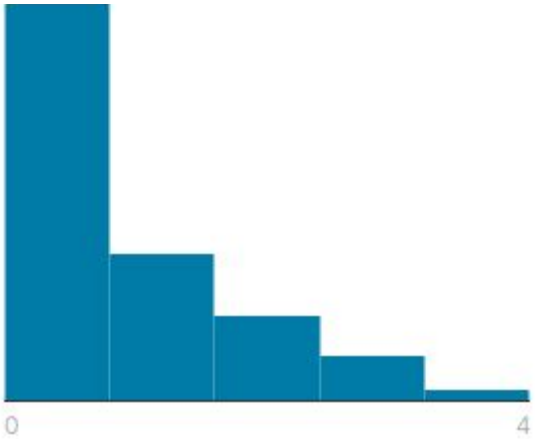

Figure 13. Ca distribution.

Aspect	Value
Min	0
Max	4
Mean	0.73
Median	0
# of NaN	0

Table 12. Ca statistics.

13. **Thallium test: 3 = normal; 6 = fixed defect; 7 = reversible defect (x_{13})**

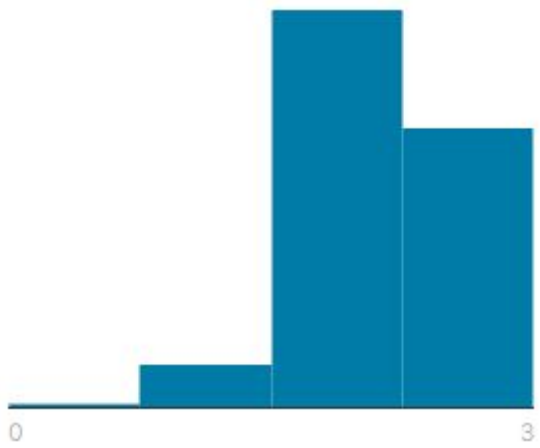

Figure 14. Thal distribution.

Aspect	Value
Min	0
Max	3
Mean	2.31
Median	2
# of NaN	0

Table 13. Thal statistics.

In regards to the labels, a sample is either labelled with a 0, which represents an absence of heart disease; or with a 1, which represents that the patient suffers from heart disease. Just like the 13 features shown previously, here's the numeric information about the labels:

14. **Target (y)**

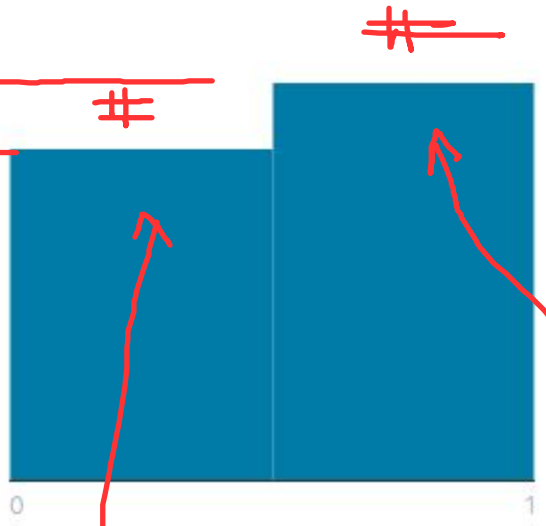

Figure 15. Class distribution.

Aspect	Value
Min	0
Max	1
Mean	0.54
Median	2
# of NaN	0

Table 14. Class statistics.

As it can be observed by the previous distribution of the labels, this dataset is fairly balanced, consisting of 138 (about 45.54% of the dataset) negative classes; and 165 (about 54.46% of the dataset) positive clases.

3.2. Particular Exploratory Data Analysis

First, the dataset is read from a .csv file, that is, a comma-separated values file. The first line of the file contains the titles of each of the attributes, each separated by a comma, while the remaining lines are the dataset samples. Again, each of the lines corresponding to the samples contains the value for each feature separated by a comma. This .csv file is read, and afterwards it is randomly shuffled. Then, it is split: leaving the first 80% of the dataset to be used as the training dataset, and the other 20% for the testing dataset.

To get an idea about the samples included in the dataset, 10 samples from the training dataset are randomly selected and visualised. For example, the following 10 samples:

Training Dataset														
Sample	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	y
1	52	1	2	138	223	0	1	169	0	0.0	2	4	2	1
2	59	1	0	138	271	0	0	182	0	0.0	2	0	2	1
3	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
4	54	1	2	120	258	0	0	147	0	0.4	1	0	3	1
5	51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
6	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
7	49	0	0	130	269	0	1	163	0	0.0	2	0	2	1
8	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
9	44	1	2	120	226	0	1	169	0	0.0	2	0	2	1
10	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1

Table 15. Training dataset randomly selected samples.

Just like with the training dataset, another 10 samples were visualised, but this time they were randomly selected from the testing dataset:

Testing Dataset														
Sample	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	y
1	54	1	0	0	124	266	0	1	109	1	2.2	1	1	0
2	64	0	2	2	140	313	0	0	133	0	0.2	2	0	1
3	43	1	2	2	130	315	0	0	162	0	1.9	2	1	1
4	69	1	3	3	160	234	1	0	131	0	0.1	1	1	1
5	57	0	0	0	140	241	0	1	123	1	0.2	1	0	0
6	55	0	1	1	132	342	0	1	166	0	1.2	2	0	1
7	44	1	2	2	150	232	0	0	165	0	1.6	2	0	1
8	53	1	2	2	130	197	1	0	152	0	1.2	0	0	1
9	63	1	0	0	140	187	0	0	144	1	4.0	2	2	0
10	39	0	2	2	138	220	0	1	152	0	0.0	1	0	1

Table 16. Testing dataset randomly selected samples.

Additionally, some statistics are computed about each of the 13 features, both for the training dataset and for the testing dataset. The results are included in the following tables:

Training Dataset													
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
Min	29	0	0	94	126	0	0	71	0	0	0	0	0
Max	77	1	3	200	564	1	2	202	1	6.2	2	4	3
Mean	54.4	0.70	0.96	131.8	245.5	0.16	0.52	150.2	0.32	0.98	1.42	0.71	2.28
Median	55	1	1	130	241.5	0	1	153	0	0.6	1	0	2

Table 17. Training dataset statistics before normalisation.

Testing Dataset													
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
Min	34	0	0	100	160	0	0	95	0	0	0	0	1
Max	69	1	3	192	407	1	2	195	1	5.6	2	4	3
Mean	54.4	0.62	1	131.1	249.1	0.12	0.54	147.5	0.36	1.26	1.31	0.8	2.46
Median	56	1	1	130	240	0	1	152	0	1.2	1	0	2

Table 18. Testing dataset statistics before normalisation.

3.3. Training

The first step for the training is to normalise the training dataset, as well as the testing dataset. To achieve this, the method using the mean and the variance was used, as requested by the client. After normalisation, the data values for each feature fall between -1 and 0, which is great for numeric stability and for the algorithm to be able to converge in less iterations with less overshoots.

For the logistic classifier, as its name implies, the logistic function was used. This function is used as the hypothesis function, which results in the predictions falling between 0 and 1. Additionally, the cost function, also known as the loss function or cross-entropy, comes from the use of logarithms, and it 'punishes' wrong predictions, but doesn't punish correct predictions. For this reason, the gradient operator, which helps to minimise the cost function, when combined with the loss function, leads to improved weights for the model over several iterations.

However, the hyperparameter α, which is known as the learning rate, also needs to be chosen properly for the algorithm to converge in a timely manner and to a correct model. The learning rate multiplies the result from the gradient operator, which means that a bigger α changes the weights faster, which may lead to a faster convergence, however, a balance needs to be found, as with a too high of a learning rate, the algorithm may overshoot or even diverge.

On the other hand, the L2 norm stop is another parameter that is used for the gradient descent algorithm. When the L2 norm goes below the threshold dictated by the L2 norm stop, the algorithm is stopped.

Included below there's a table that shows several combinations of different learning rates with different L2 norm stops. For each combination, several performance metrics are included. Those metrics include the number of iterations and computing time, to take time constraints into consideration; as well as the final cost, accuracy, and recall, to also take into account how well samples are classified, which is a crucial point of the project. In the 'Testing' section below it will be explained why recall is considered an important metric for this problem. The table values were obtained by running the main program inside a nested for loop, the outer loop being iterating through the different learning rates and the inner loop through the various L2 norm stops.

quite well written and explained ;)

Legend:

ni = number of iterations ct = computing time (seconds) fc = final cost ac = accuracy re = recall

L2 Norm Stopping Criteria	Learning Rate α						
	[0.0005]	[0.001]	[0.005]	[0.01]	[0.05]	[0.1]	[0.5]
[0.0005]	ni: 193,470 ct: 32.98 fc: 93.48 ac: 0.89 re: 0.97	ni: 96,734 ct: 16.97 fc: 93.48 ac: 0.89 re: 0.97	ni: 19,346 ct: 3.34 fc: 93.48 ac: 0.89 re: 0.97	ni: 9,673 ct: 1.69 fc: 93.48 ac: 0.89 re: 0.97	ni: 1,934 ct: 0.36 fc: 93.48 ac: 0.89 re: 0.97	ni: 967 ct: 0.17 fc: 93.48 ac: 0.89 re: 0.97	ni: 193 ct: 0.05 fc: 93.48 ac: 0.89 re: 0.97
[0.001]	ni: 156,673 ct: 28.14 fc: 93.48 ac: 0.9 re: 0.97	ni: 78,336 ct: 14.14 fc: 93.48 ac: 0.9 re: 0.97	ni: 15,667 ct: 2.94 fc: 93.48 ac: 0.9 re: 0.97	ni: 7,833 ct: 1.36 fc: 93.48 ac: 0.9 re: 0.97	ni: 1,566 ct: 0.30 fc: 93.48 ac: 0.9 re: 0.97	ni: 783 ct: 0.17 fc: 93.48 ac: 0.9 re: 0.97	ni: 156 ct: 0.08 fc: 93.48 ac: 0.9 re: 0.97
[0.005]	ni: 79,128 ct: 13.92 fc: 93.54 ac: 0.9 re: 0.97	ni: 39,564 ct: 7.10 fc: 93.54 ac: 0.9 re: 0.97	ni: 7,913 ct: 1.39 fc: 93.54 ac: 0.9 re: 0.97	ni: 3,956 ct: 0.7 fc: 93.54 ac: 0.9 re: 0.97	ni: 791 ct: 0.16 fc: 93.54 ac: 0.9 re: 0.97	ni: 395 ct: 0.09 fc: 93.54 ac: 0.9 re: 0.97	ni: 79 ct: 0.02 fc: 93.54 ac: 0.9 re: 0.97
[0.01]	ni: 53,538 ct: 9.48 fc: 93.71 ac: 0.92 re: 0.97	ni: 26,769 ct: 4.42 fc: 93.71 ac: 0.92 re: 0.97	ni: 5,354 ct: 0.64 fc: 93.71 ac: 0.92 re: 0.97	ni: 2,677 ct: 0.48 fc: 93.71 ac: 0.92 re: 0.97	ni: 535 ct: 0.08 fc: 93.71 ac: 0.92 re: 0.97	ni: 268 ct: 0.05 fc: 93.71 ac: 0.92 re: 0.97	**ni: 53** **ct: 0.02** **fc: 93.71** **ac: 0.92** **re: 0.97**
[0.05]	ni: 16,522 ct: 2.48 fc: 96.44 ac: 0.92 re: 0.97	ni: 8,261 ct: 1.62 fc: 96.44 ac: 0.92 re: 0.97	ni: 1,653 ct: 0.28 fc: 96.44 ac: 0.92 re: 0.97	ni: 827 ct: 0.17 fc: 96.44 ac: 0.92 re: 0.97	ni: 166 ct: 0.05 fc: 96.42 ac: 0.92 re: 0.97	ni: 83 ct: 0.03 fc: 96.42 ac: 0.92 re: 0.97	ni: 17 ct: 0.03 fc: 96.29 ac: 0.92 re: 0.94
[0.1]	ni: 8,626 ct: 1.64 fc: 101.33 ac: 0.92 re: 0.97	ni: 4,313 ct: 0.91 fc: 101.33 ac: 0.92 re: 0.97	ni: 863 ct: 0.17 fc: 101.33 ac: 0.92 re: 0.97	ni: 432 ct: 0.11 fc: 101.32 ac: 0.92 re: 0.97	ni: 87 ct: 0.05 fc: 101.27 ac: 0.92 re: 0.97	ni: 44 ct: 0.03 fc: 101.18 ac: 0.92 re: 0.97	ni: 9 ct: 0.02 fc: 101.11 ac: 0.92 re: 0.97
[0.5]	ni: 264 ct: 0.08 fc: 159.06 ac: 0.92 re: 0.97	ni: 133 ct: 0.05 fc: 159.03 ac: 0.92 re: 0.97	ni: 28 ct: 0.02 fc: 158.84 ac: 0.92 re: 0.97	ni: 15 ct: 0.05 fc: 158.52 ac: 0.92 re: 0.97	ni: 4 ct: 0.03 fc: 157.81 ac: 0.92 re: 0.97	ni: 3 ct: 0.03 fc: 154.72 ac: 0.92 re: 0.97	ni: 2 ct: 0.02 fc: 137.62 ac: 0.92 re: 0.97

This is a fantastic analysis ;)

Table 19. Learning rate and L2 norm stop combination's metrics.

In this case the chosen learning rate was 0.5, while the chosen L2 norm stop was 0.01 (cell marked with a green background). The reasoning behind this is that with said combination there's the least amount of iterations and computing time when considering those combinations where the cost is relatively low (at 93.71) and the accuracy and recall are the highest amongst all (0.92 and 0.97, respectively).

Besides training the algorithm, the cost function was plotted, to be able to see how the cost decreases over the iterations. Said graph can be seen below:

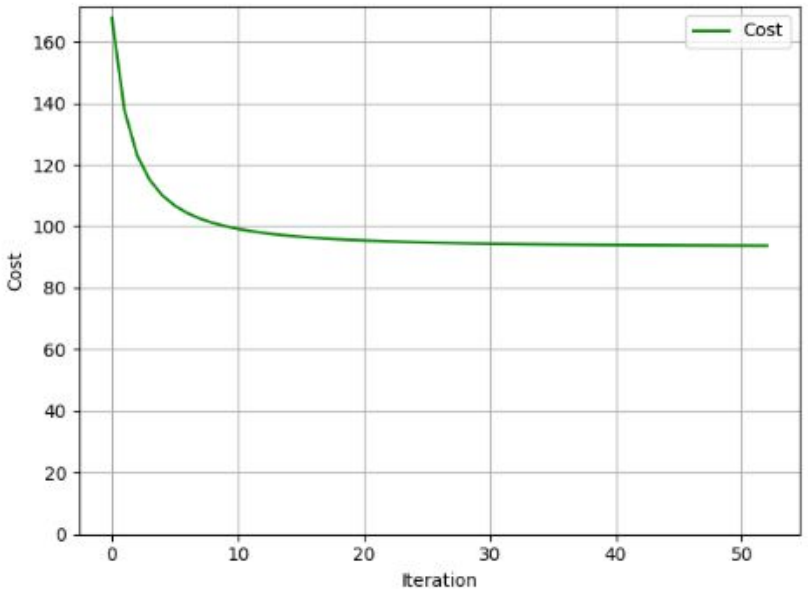

Figure 16. Graph of the cost function.

As it can be seen, the cost decreases monotonically, and it seems to have asymptotic behavior with the asymptote at a cost of slightly over 90. Additionally, the cost decreases faster in the initial iterations, where there are bigger improvements to be made, and then the decrements become much lower as the model approaches convergence. This behaviour resembles the law of diminishing returns, however, with the amount of samples in this dataset, the time it takes for convergence is not much. Therefore, it's better to use a small value for the L2 norm stop, as described above.

It's worth mentioning that the evaluation of the cost function, the hypothesis function, and the update of the gradient descent are vectorised. This means that their code is not defined 'manually', but rather it works with vector operations, which take into account the dimensions of said vectors. For this reason the code can work for other datasets, for example, with a different number of features and samples, without needing to rewrite it. Additionally, vectorising those evaluations avoids loops, which results in them being performed computationally faster.

4. Experimental Results: Testing

After the algorithm has been trained, it was tested using the testing dataset, which as mentioned earlier, was the 20% of the original dataset, obtained after it was randomly shuffled. The testing dataset consists of 61 samples, and as mentioned on the previous section, it was normalised just like the training dataset was.

After the algorithm was trained and the final weights for the model were obtained, the hypothesis function was used to calculate the prediction for each training dataset sample. If a prediction was greater than or equal to 0.5, the final prediction was set to 1, and it was set to 0 otherwise, that is, for values less than 0.5.

Afterwards, these predictions were compared to the actual classes of the samples to obtain the confusion matrix, which resulted as shown below:

		Actual Class	
		Has heart disease (1)	No heart disease (0)
Predicted Class	Has heart disease (1)	True Positive 30	False Positive 4
	No heart disease (0)	False Negative 1	True Negative 26

Table 20. Confusion matrix obtained from the testing dataset.

From the previous confusion matrix, several metrics were obtained, as shown below. Said metrics will be discussed next.

Metric	Value
Accuracy	0.92
Precision	0.88
Recall / Sensitivity	0.97
Specificity	0.87
F1-score	0.92

Table 21. Testing metrics calculated from the confusion matrix.

First, looking at the accuracy, the classifier seems to perform well, at least when working with the samples gathered for the testing dataset. Accuracy is a valid metric for this dataset, because as it was stated in an earlier section of the report, the dataset is fairly balanced, consisting of 138 (about 45.54% of the dataset) negative classes; and 165 (about 54.46% of the dataset) positive clases. In this case, where the dataset is balanced and precision and recall are similar, it's not necessary to resort to the F1-score, but it still shows a good performance. As an important note, recall was the highest metric, which is a good sign for this classifier, whose objective is to point if someone suffers from heart disease or not.

As it can be seen from the confusion matrix and the obtained metrics, the classifier performs better at classifying positive-class samples compared to negative-class samples. This can be seen with the higher recall (0.97) compared to the specificity (0.87), which is a desired behaviour, as it will be explained next.

For the problem that this project is dealing with, that is, identifying someone as suffering from heart disease or not, recall is a very important metric. Recall or sensitivity measures the fraction of samples correctly predicted as positive out of all of the samples that were actually positive. That is, out of all the patients who suffered from heart disease, how many of them were actually predicted to suffer from heart disease.

If someone actually has heart disease but they're told that they're actually fine (false negative), without any heart disease, there can be grave consequences. For example, maybe the individual could have received medical treatment or made some lifestyle adjustments to help with the heart disease that they actually suffer from, but because they were told that they don't suffer from heart disease, they continued going through life without any positive adjustments, and in the future their health could complicate even more. On the other hand, if they were correctly diagnosed with a heart disease, they could be treated for it to the extent possible and improve their health prospects or life expectancy.

While telling someone that they suffer from heart disease when in fact they actually don't (false positive, related metric: precision) is also bad, it doesn't have as bad of an outcome as the one related to false negatives. False positives can be devastating for certain conditions, such as those where the individual is told that they have months remaining to live, so their life could spiral away if they don't see the point, for example, of continue working with the little time left they think they posses. However, with heart disease that's not the case, and telling someone that they suffer from it while they don't could even lead to benefits as the individual makes lifestyle adjustments such as changing their diet or exercising more often.

Nonetheless, going by only the first opinion is not a good option, and even less with a classifier trained with as little as 13 features (referring to a problem as broad as health disease) and less than 500 samples. What's predicted by this classifier could be used to study trends, for experiments or simulations, but perhaps not as a final indicator to tell if someone suffers from heart disease or not. While this classifier performed well with the training dataset, when dealing with problems as important such as the health of a patient, it's better to deeply analyze their situation in order to make the best recommendations. Possibly, this classifier could be used as a first indicator before performing further analysis on the patient. To make better and more secure predictions, a classifier could be trained with more samples and with some additional relevant features.

Grade = 100.0

5. Conclusions

In conclusion, using cross-entropy as a cost function, along with the logistic function as the hypothesis function, through the gradient descent, a binary classifier was able to be trained with information from a public health dataset with samples from individuals who either suffer from heart disease or not.

After both general and particular exploratory data analysis were performed and the data was normalised, the classifier was trained with 80% of the dataset, with a learning rate of 0.5 and using the L2 norm as the stopping criteria, with a threshold of 0.01. Then, it was tested with the remaining 20% of the dataset, the confusion matrix was built, and several metrics were obtained.

The classifier proved to have great performance with the samples it was tested, achieving 0.92 accuracy, and perhaps most importantly, a recall of 0.97. As it was discussed previously, being able to correctly point out those individuals who actually suffer from heart disease is of great importance. This way, the patients can receive adequate medical help and adjust their lifestyle in order to improve their health to the extent possible. This is the reason that a recall is a good metric for this problem, and it's a good sign that it was the highest metric.

Still, when talking about a problem such as the health of a person, having close to perfect accuracy is crucial, especially with conditions such as heart disease, which is the leading cause of death in the world. In this case, there was 1 false negative out of the 31 positives, which is an error of around 0.03 or 3% (or the complement of recall, 1 - 0.97). This is good, but can be improved. For example, out of 1000 individuals, telling 30 of them that they don't suffer from heart disease while they actually do, could be a problem. That's why as mentioned earlier, second opinions and detailed analysis on each individual is crucial.

This binary classifier could be a way to study patterns and perform simulations, optimize hospital processes by quickly indicating those individuals who are more likely to require further heart studies, or even work as a good first indicator before continuing with further specialized analysis by more accurate methods.

In future works, it may be looked into training the binary with much bigger datasets, for example, with over 10,000 samples, to see how the performance metrics such as accuracy and recall turn out. Additionally, along with the help of heart specialists, the features included in the dataset could be improved, removing those who might not be of actual relevance if there are any, and adding some additional features which might improve the performance of the heart disease classifier when taken into consideration.

6. References

[Figure 1] Retrieved from:
https://www.nhlbi.nih.gov/sites/default/files/styles/16x9_crop/public/2017-12/internal_structures_1_0.jpg?h=6c8b8c3d&itok=BNFPxPUV

[Figure 2] through [Figure 15] Retrieved from: https://www.kaggle.com/ronitf/heart-disease-uci

Heart Research Australia. (2018). *What is Heart Disease*. Retrieved from:
https://www.heartresearch.com.au/heart-disease/what-is-heart-disease/

Hernández, A. (2020). *Logistic regression for classification* [Slides].

Ronit (Kaggle user). (2018). *Heart Disease UCI*. Retrieved from:
https://www.kaggle.com/ronitf/heart-disease-uci

World Health Organization. (2018). *The top 10 causes of death*. Retrieved from: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death