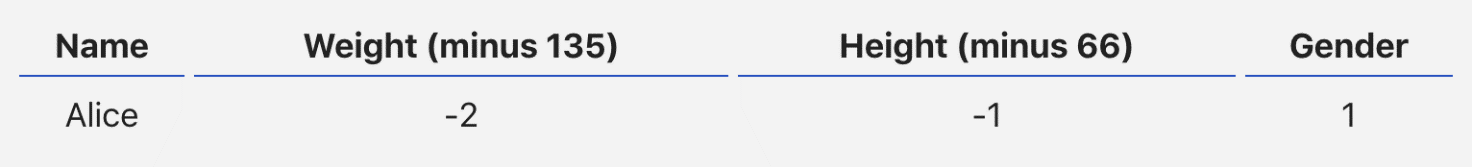
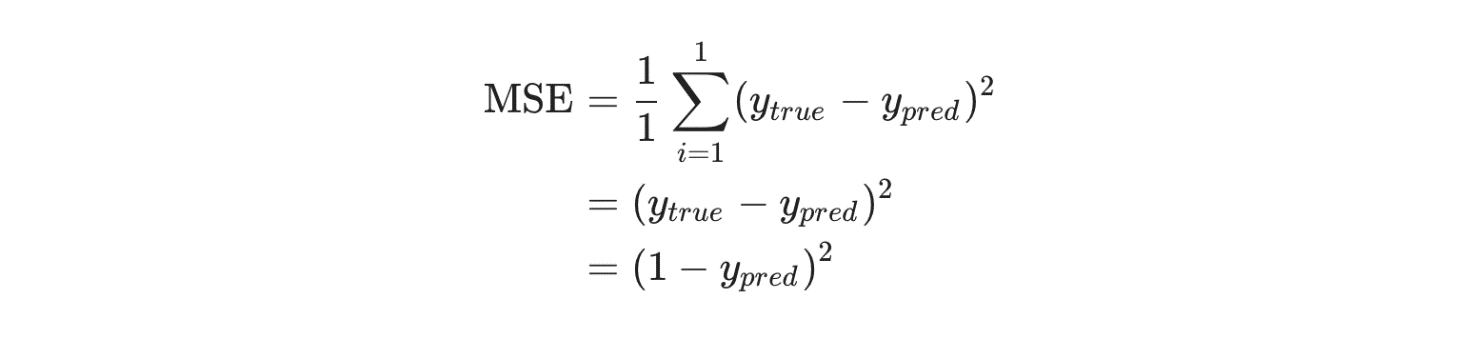
# ANNEXOS

## Exemple derivada parcial Backpropagation

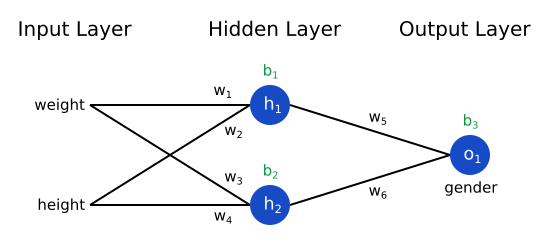
For simplicity, let’s pretend we only have Alice in our dataset:



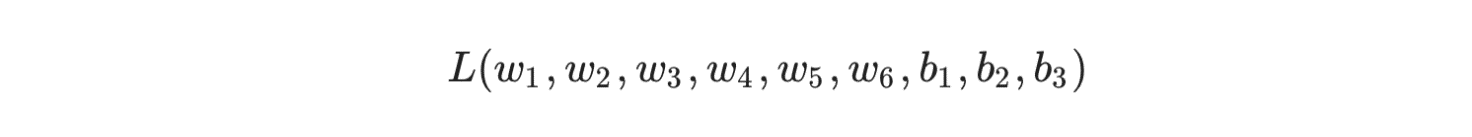
Then the mean squared error loss is just Alice’s squared error:



Another way to think about loss is as a function of weights and biases. Let’s label each weight and bias in our network:

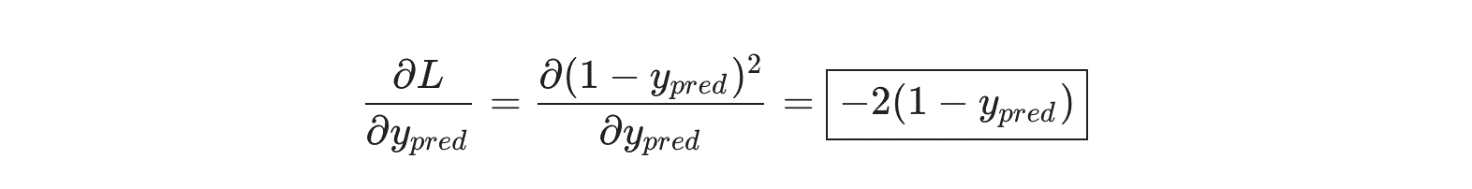


Then, we can write loss as a multivariable function:



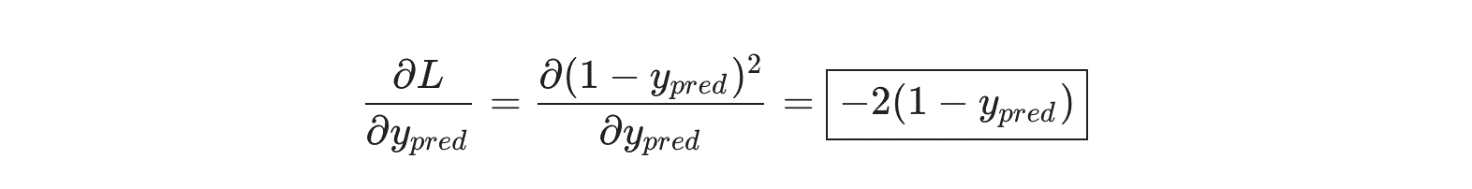
Imagine we wanted to tweak w1​. How would loss L change if we changed w1​? That’s a question the partial derivative can answer. How do we calculate it?

To start, let’s rewrite the partial derivative in terms of ∂y\_pred/∂w1​​ instead:

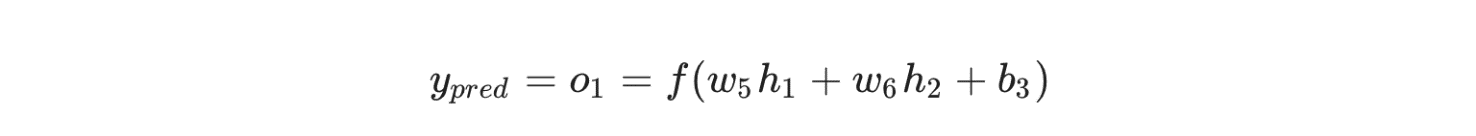


This works because of the Chain Rule.

We can calculate ∂L/∂y\_pred​ because we computed L= (1−y\_pred​)² above:

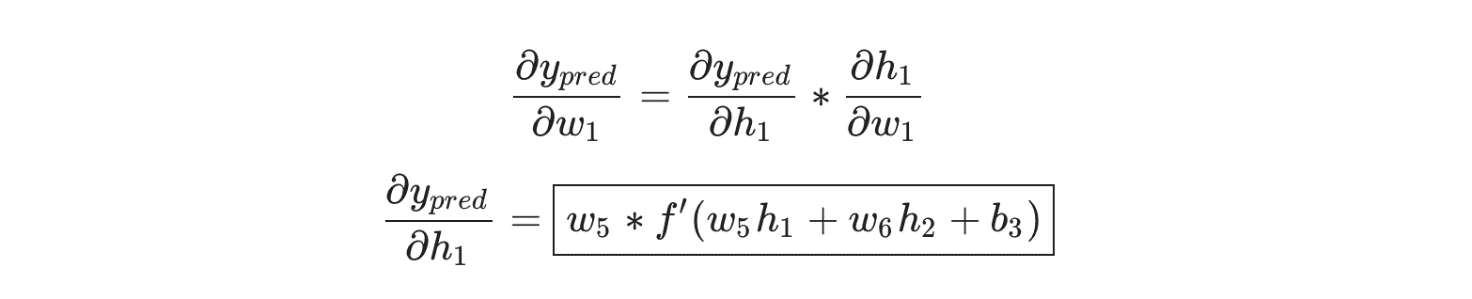


Now, let’s figure out what to do with ∂y\_pred/∂w1. Just like before, let h1​, h2​, o1​ be the outputs of the neurons they represent. Then

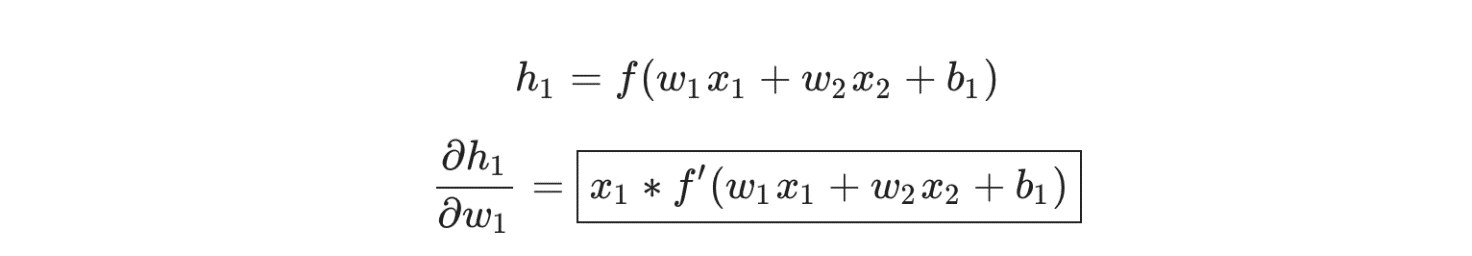


f is the sigmoid activation function, remember?

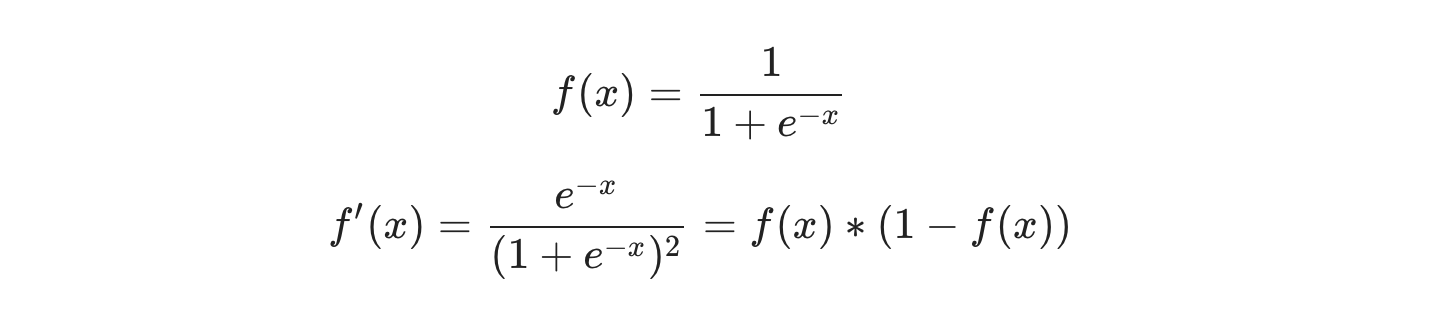
Since w1​ only affects h1​ (not h2​), we can write



We do the same thing for ∂h1​​/∂w1:

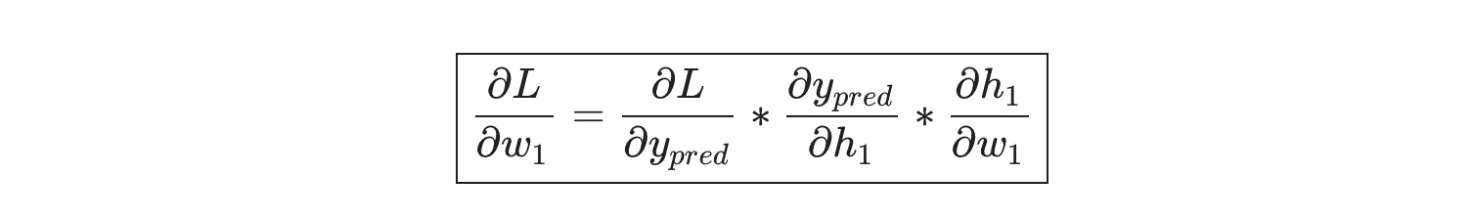


x1​ here is weight, and x2​ is height. This is the second time we’ve seen f′(x) (the derivate of the sigmoid function) now! Let’s derive it:



We’ll use this nice form for f′(x) later.

We’re done! We’ve managed to break down ∂L/∂w1​ into several parts we can calculate:



This system of calculating partial derivatives by working backwards is known as backpropagation, or “backprop”.

## ROC i AUC

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

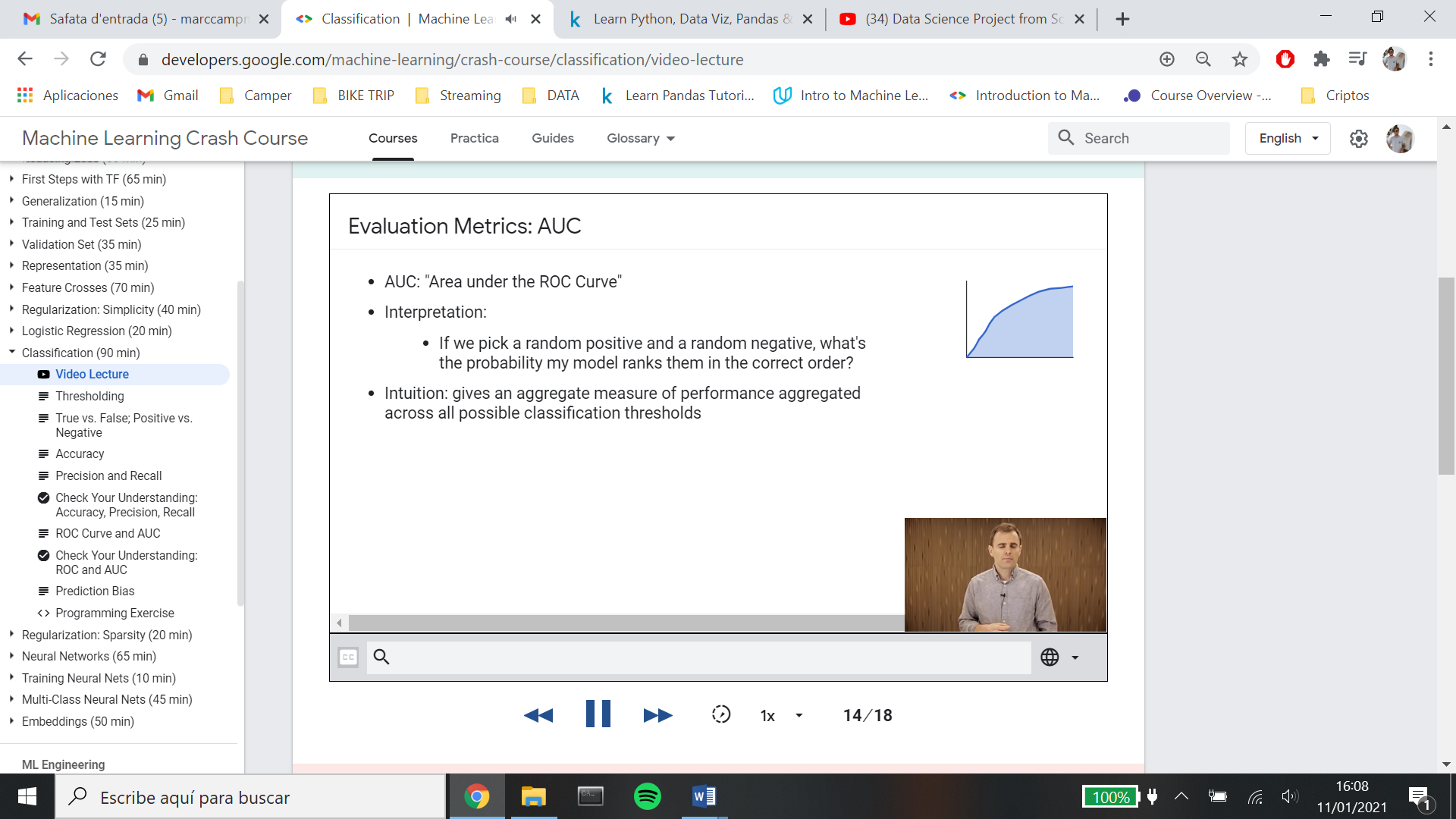
**True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:

TPR=TP/ (TP+FN)

**False Positive Rate** (**FPR**) is defined as follows:

FPR=FP/ (FP+TN)

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



**AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

## Data augmentation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.8381 | 0.8489 | 0.7291 | 0.6295 | 0.8780 |
| Val. | 0.8869 | 0.8445 | 0.7005 | 0.6603 | 0.8660 |
| Test | 0.8266 | 0.8486 | 0.7125 | 0.6610 | 0.8825 |



Confusion Matrix

[[109 0 0 2]

[ 25 0 0 0]

[ 12 0 0 1]

[ 28 0 0 32]]

Classification Report

precision recall f1-score support

0.0 0.63 0.98 0.76 111

0.33 0.00 0.00 0.00 25

0.66 0.00 0.00 0.00 13

1.0 0.91 0.53 0.67 60

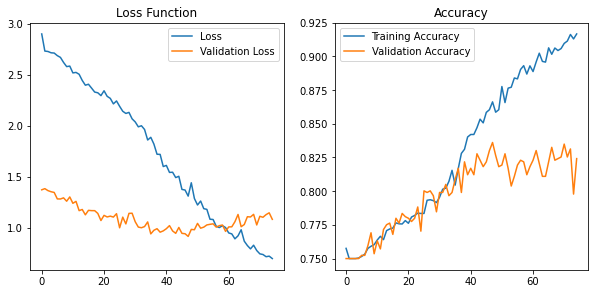
accuracy 0.67 209

macro avg 0.39 0.38 0.36 209

weighted avg 0.60 0.67 0.60 209

## Equilibri de pesos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.7328 | 0.9147 | 0.8712 | 0.7729 | 0.9654 |
| Val. | 1.0824 | 0.8242 | 0.6632 | 0.6029 | 0.8660 |
| Test | 1.0310 | 0.8476 | 0.7167 | 0.6457 | 0.8752 |



Confusion Matrix

[[76 18 2 15]

[ 8 13 2 2]

[ 6 1 4 2]

[15 3 3 39]]

Classification Report

precision recall f1-score support

0.0 0.72 0.68 0.70 111

0.33 0.37 0.52 0.43 25

0.66 0.36 0.31 0.33 13

1.0 0.67 0.65 0.66 60

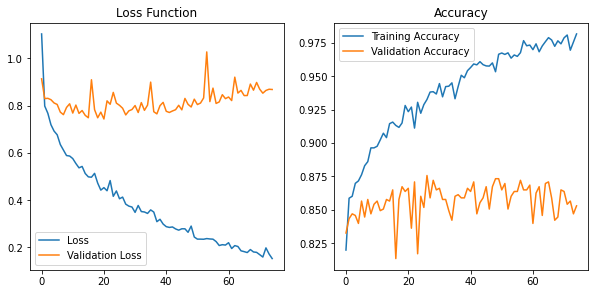
accuracy 0.63 209

macro avg 0.53 0.54 0.53 209

weighted avg 0.64 0.63 0.64 209

## Transfer Learning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.1543 | 0.9795 | 0.9695 | 0.9479 | 0.9974 |
| Val. | 0.8685 | 0.8529 | 0.7150 | 0.6842 | 0.9039 |
| Test | 0.7546 | 0.8650 | 0.7413 | 0.7065 | 0.9111 |



Confusion Matrix

[[99 3 0 9]

[17 7 0 1]

[ 7 1 1 4]

[16 2 3 39]]

Classification Report

precision recall f1-score support

0.0 0.71 0.89 0.79 111

0.33 0.54 0.28 0.37 25

0.66 0.25 0.08 0.12 13

1.0 0.74 0.65 0.69 60

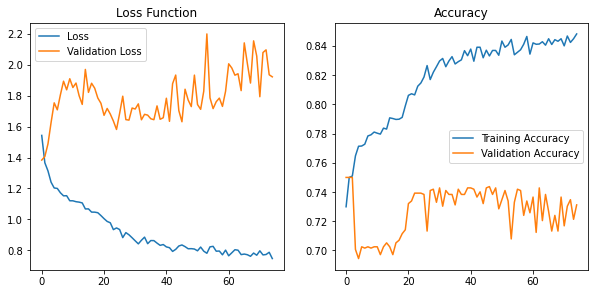
accuracy 0.70 209

macro avg 0.56 0.47 0.49 209

weighted avg 0.67 0.70 0.67 209

## Oversampling

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.7336 | 0.8517 | 0.7947 | 0.5498 | 0.9066 |
| Val. | 1.9222 | 0.7312 | 0.4444 | 0.3011 | 0.6628 |
| Test | 0.9942 | 0.8495 | 0.7590 | 0.5833 | 0.8531 |



Confusion Matrix

[[49 6 0 33]

[39 7 0 22]

[27 4 0 18]

[39 3 0 32]]

Classification Report

precision recall f1-score support

0.0 0.32 0.56 0.40 88

0.33 0.35 0.10 0.16 68

0.66 0.00 0.00 0.00 49

1.0 0.30 0.43 0.36 74

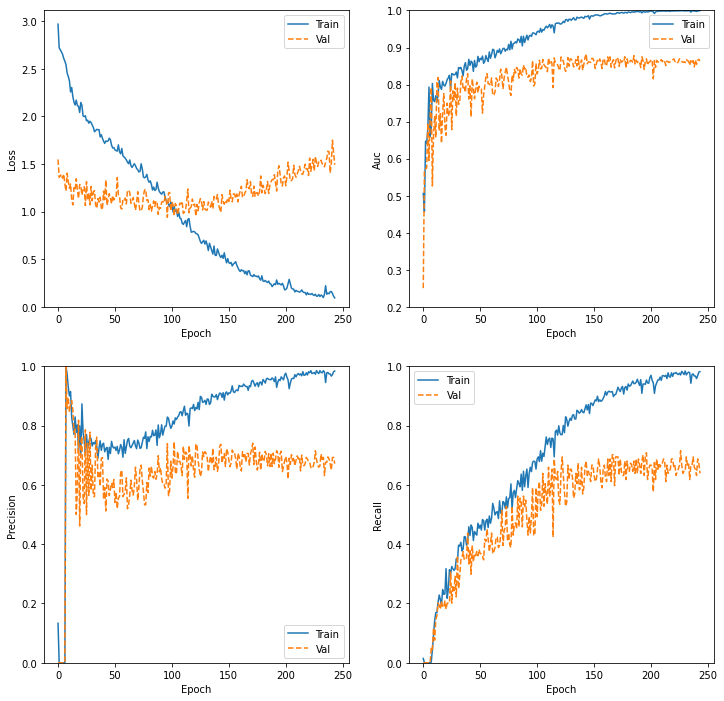
accuracy 0.32 279

macro avg 0.24 0.27 0.23 279

weighted avg 0.27 0.32 0.26 279

## Tipus mono, pesos, L2 i early stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.0897 | 0.0101 | 0.9863 | 0.9836 | 0.9996 |
| Val. | 1.4930 | 0.0087 | 0.6715 | 0.6389 | 0.8638 |
| Test | 0.7690 | 0.0 | 0.8485 | 0.7778 | 0.9175 |



Confusion Matrix

[[58 2 3 4]

[ 2 6 3 1]

[ 2 0 2 2]

[ 3 0 0 20]]

Classification Report

precision recall f1-score support

0.0 0.89 0.87 0.88 67

0.33 0.75 0.50 0.60 12

0.66 0.25 0.33 0.29 6

1.0 0.74 0.87 0.80 23

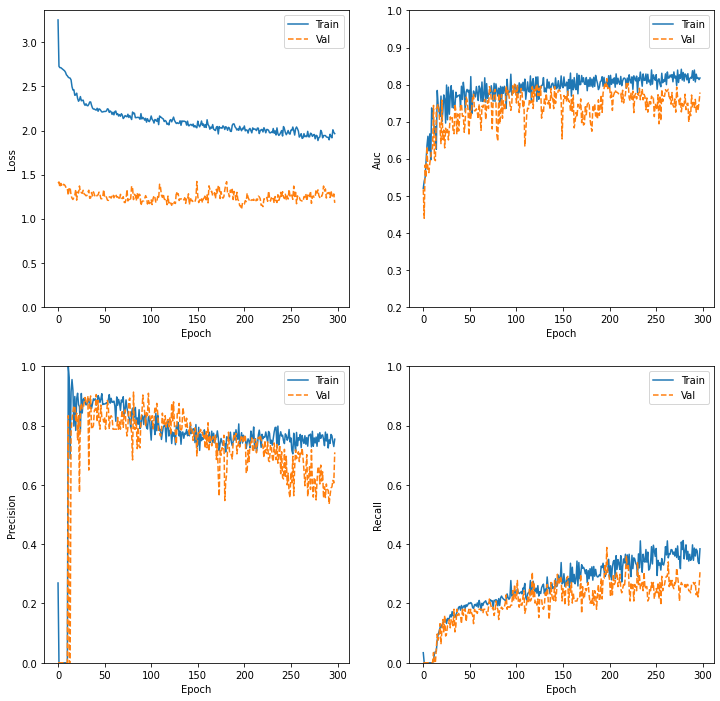
accuracy 0.80 108

macro avg 0.66 0.64 0.64 108

weighted avg 0.81 0.80 0.80 108

## Tipus mono, pesos, data augmentation, L2, early stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 2.0121 | 0.0 | 0.7647 | 0.3845 | 0.8128 |
| Val. | 1.1800 | 0.0 | 0.7097 | 0.3056 | 0.7786 |
| Test | 0.9754 | 0.0 | 0.7931 | 0.4259 | 0.8375 |



Confusion Matrix

[[46 9 8 4]

[ 5 7 0 0]

[ 2 1 2 1]

[ 5 0 2 16]]

Classification Report

precision recall f1-score support

0.0 0.79 0.69 0.74 67

0.33 0.41 0.58 0.48 12

0.66 0.17 0.33 0.22 6

1.0 0.76 0.70 0.73 23

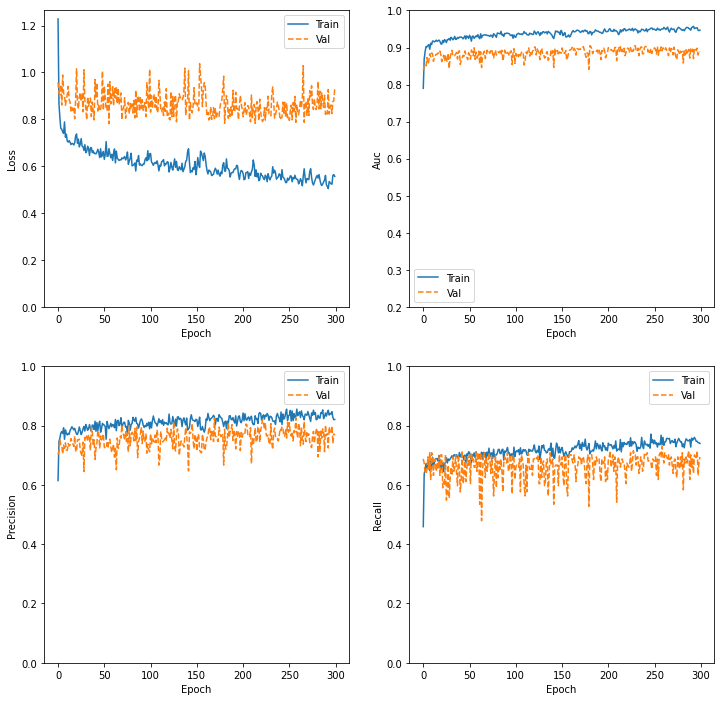
accuracy 0.66 108

macro avg 0.53 0.57 0.54 108

weighted avg 0.71 0.66 0.68 108

## Tipus mono, transfer learning, data aug, L2, early stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.5654 | 0.0120 | 0.8201 | 0.7549 | 0.9467 |
| Val. | 0.9328 | 0.0122 | 0.7692 | 0.6944 | 0.8879 |
| Test | 0.6434 | 0.0093 | 0.8000 | 0.7778 | 0.9361 |



Confusion Matrix

[[64 1 0 2]

[ 8 4 0 0]

[ 5 1 0 0]

[ 5 0 0 18]]

Classification Report

precision recall f1-score support

0.0 0.78 0.96 0.86 67

0.33 0.67 0.33 0.44 12

0.66 0.00 0.00 0.00 6

1.0 0.90 0.78 0.84 23

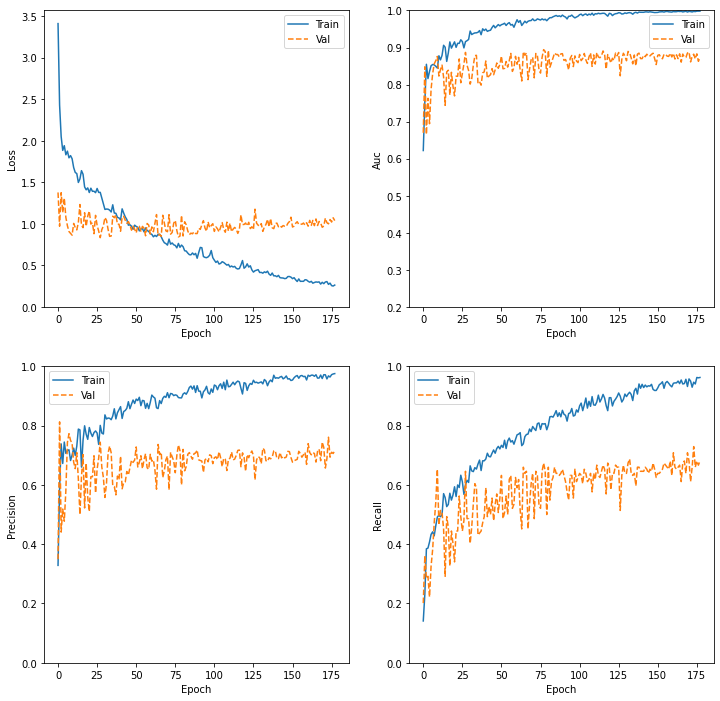
accuracy 0.80 108

macro avg 0.59 0.52 0.54 108

weighted avg 0.75 0.80 0.76 108

## Tipus mono, pesos, transfer learning, L2, early stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 0.2762 | 0.0152 | 0.9711 | 0.9543 | 0.9981 |
| Val. | 1.0397 | 0.0122 | 0.7153 | 0.6806 | 0.8730 |
| Test | 0.5998 | 0.000 | 0.8229 | 0.7315 | 0.9401 |



Confusion Matrix

[[60 3 2 2]

[ 5 6 1 0]

[ 3 1 2 0]

[ 7 0 0 16]]

Classification Report

precision recall f1-score support

0.0 0.80 0.90 0.85 67

0.33 0.60 0.50 0.55 12

0.66 0.40 0.33 0.36 6

1.0 0.89 0.70 0.78 23

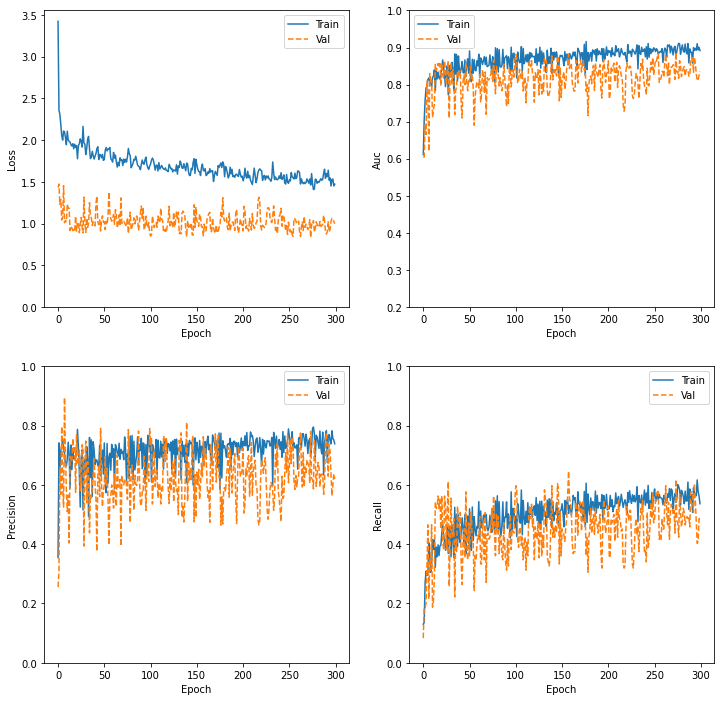
accuracy 0.78 108

macro avg 0.67 0.61 0.63 108

weighted avg 0.77 0.78 0.77 108

## Tipus mono, pesos, transfer learning, L2, early stop, data aug

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Loss | Accuracy | Precision | Recall | AUC |
| Train | 1.4723 |  | 0.7482 | 0.5397 | 0.8975 |
| Val. | 0.9978 |  | 0.6195 | 0.4861 | 0.8350 |
| Test | 0.7945 |  | 0.7294 | 0.5741 | 0.8900 |



Confusion Matrix

[[43 15 4 5]

[ 1 8 2 1]

[ 2 3 1 0]

[ 2 1 1 19]]

Classification Report

precision recall f1-score support

0.0 0.90 0.64 0.75 67

0.33 0.30 0.67 0.41 12

0.66 0.12 0.17 0.14 6

1.0 0.76 0.83 0.79 23

accuracy 0.66 108

macro avg 0.52 0.58 0.52 108

weighted avg 0.76 0.66 0.69 108