Advanced Topics in Climate Dynamics
Project 1

L. Vulto
4685784
l.vulto@students.uu.nl

June 1, 2023

(a) **Load in the data from the netcdf file. This is ERSSTv5 SST (sea surface temperature) data set which has been de-seasoned, linearly detrended, and has a 5 month look back running mean applied. See notebook.**

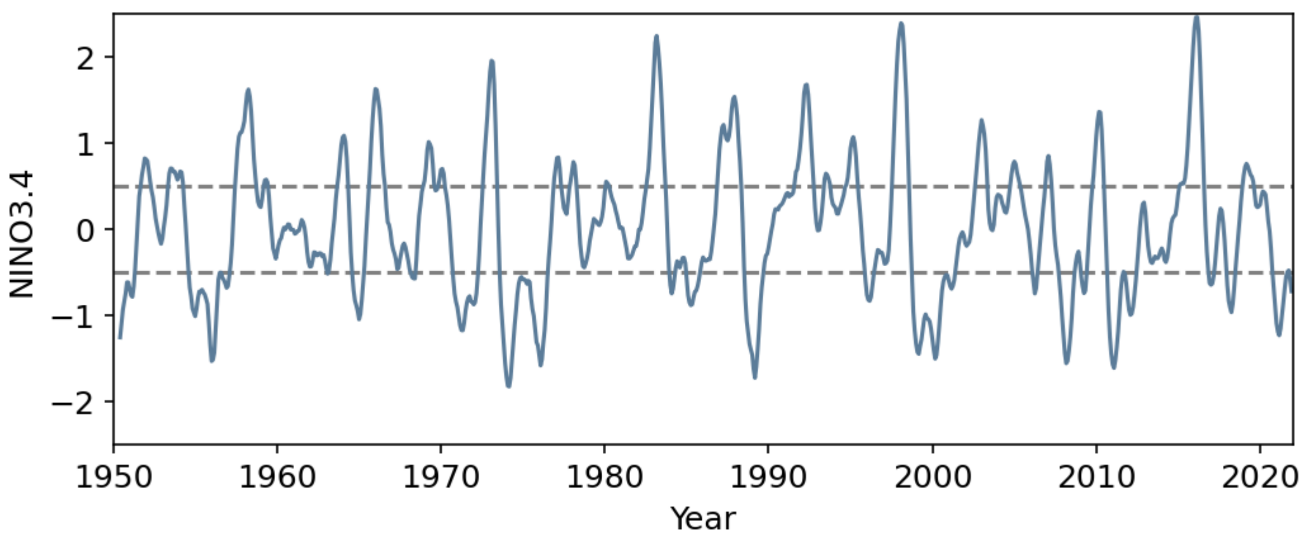


Figure 1: SST in region 3.4

(b) **Make a plot of the NINO3.4 data of the El Niño and La Niña events and show the different date sets (training, validation and testing) in different colours.**

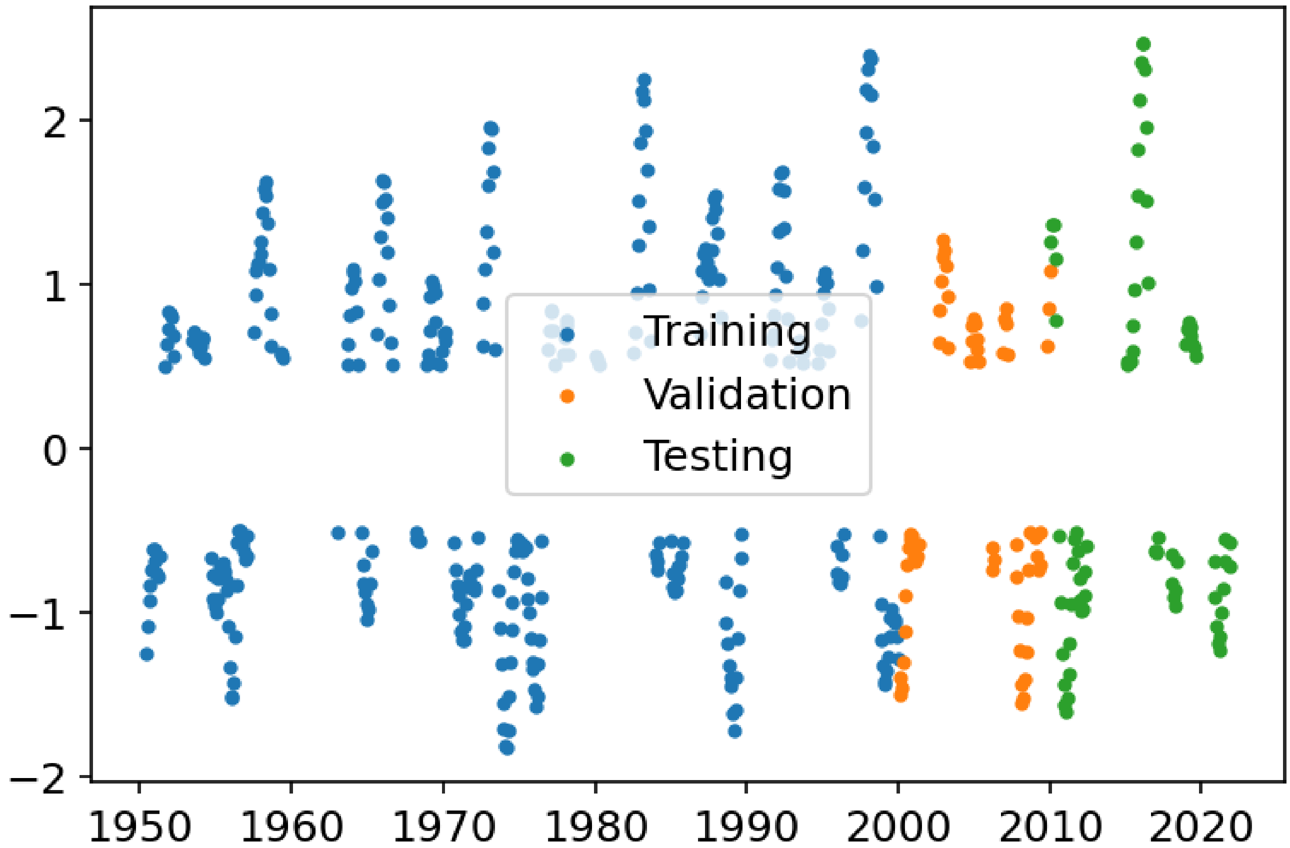


Figure 2: Data from figure 1 splitted into training, testing, and validation

(c) **Define a feedforward neural network (FNN) architecture with h hidden layers with n_h neurons, with a learning rate η, with an activation function A, a loss function J, ridge regularisation with parameter β, the number of epochs n and a batch size b. The number of output nodes is n_O = 2 (two classes). $h = 12, n_h = 12, \eta = 10^{-3}, A =$ 'Relu', $J =$ 'Cross Entropy', $\beta = 10^{-5}, n = 20$ and $b = 32$.**

See attached notebook for the Feedforward Neural Network (FNN).

(d) **Train the FNN on the training data set (using e.g. the Adam optimizer) and study the behaviour of the loss function for both training and validation data. For the standard values of the hyper-parameters, how many epochs are needed to obtain a training error of 5%? What is validation accuracy in this case?** In figure 3 the trainig and validation loss function as a function of number of Epochs. We need a total of 2 epochs in order for the training error to be lower than 0.05%. The accuracy of the validation is calculated in the code, and gives a validation accuracy of 100%. For 20 epochs, we have a training accuracy of 100% and on the test data the accuracy is 94.59%. The code used to calculate this is in the supplemental notebook.

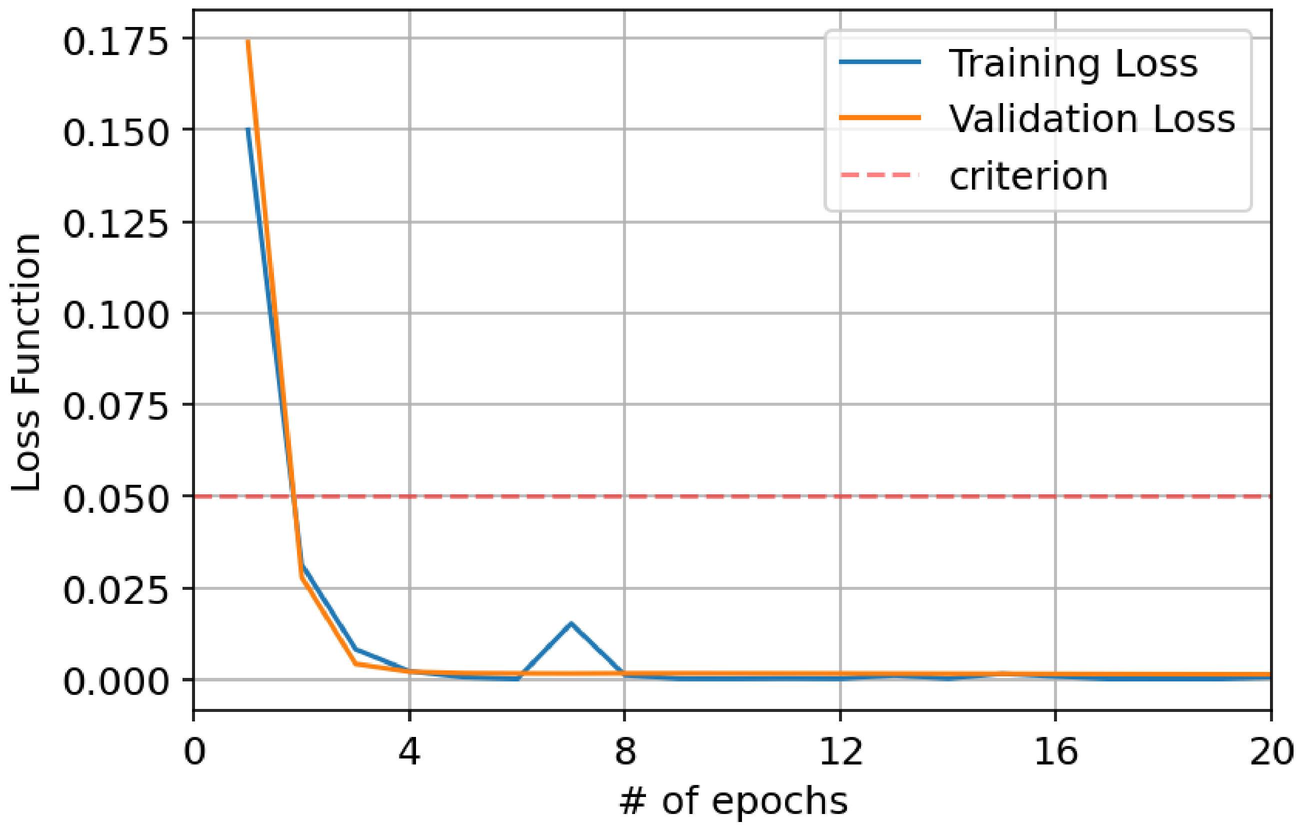


Figure 3: The Training loss and validation loss for the given hyperparameters.

(e) **What is the effect of the learning rate on the training error?**

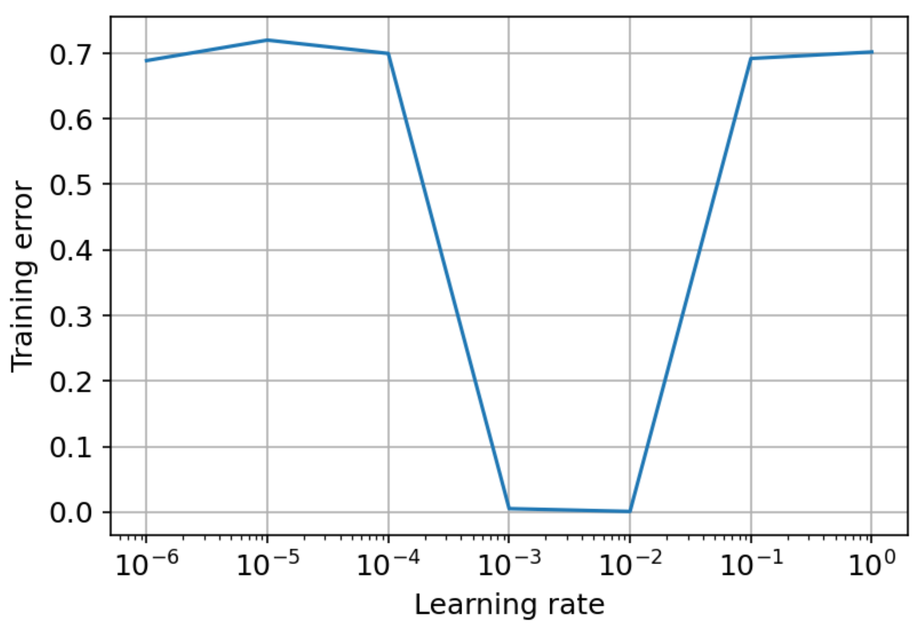


Figure 4: Effect of the learning rate on the training error.

Figure 4 was obtained by checking a learning rate in each order of magnitude between 10^{-6} and

10^0. It can be seen that the best learning rate is between $\eta = 0.0001$ and $\eta = 0.001$.

(f) **Study the effect of the activation function (apart from 'Relu', choose 'Sigmoid' or 'Softmax') on the training and validation results. Which one gives more accurate results for the same number of epochs?** I chose to compare ReLU activation function to Sigmoid, because this is a binary classification problem. The model has to either predict a 0 or a 1. This can be seen in equation (1). The Sigmoid function maps its input to a value between 0 and 1 and can be interpreted as the probability of the positive class.

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

The SoftMax activation function would make more sense in a multi-class classification problem, where there are more than 2 classes as output nodes. The Softmax function maps its input to a probability distribution over the classes. The output of the model is a vector of probabilities, one for each class, and the class with the highest probability is chosen as the prediction.

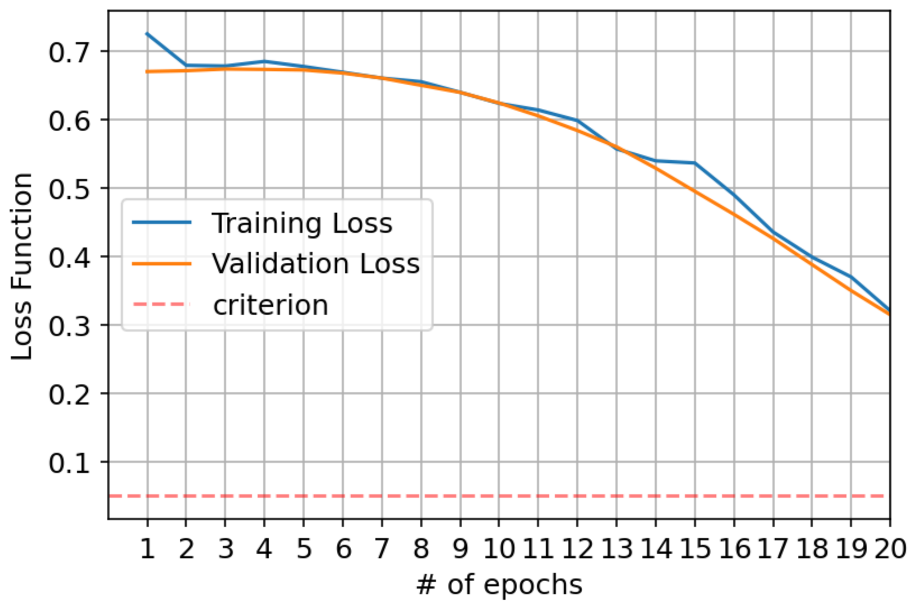
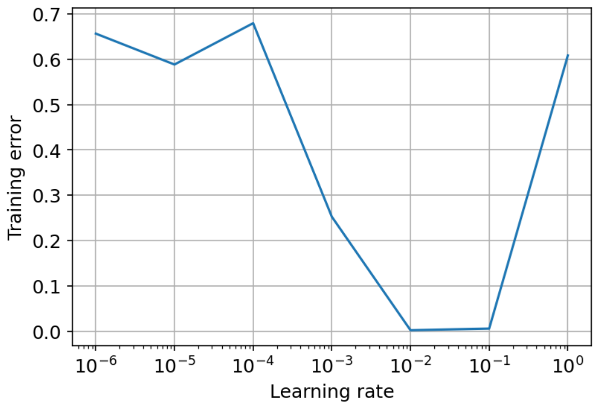
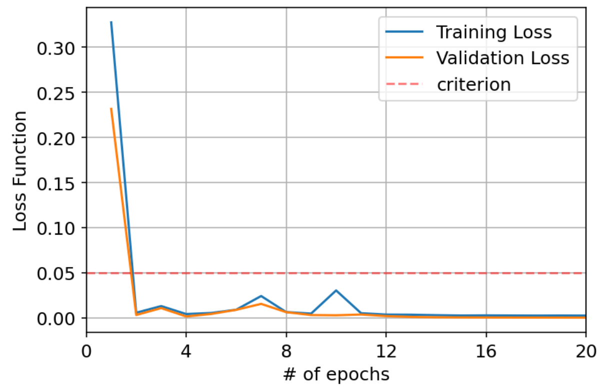


Figure 5: Caption

Comparing this figure to figure 3 ReLU seems to be better, but figure 5 has not has its hyper parameters optimised yet, so find in figure 6a the learning rate optimisation. In figure 6b the Loss function of Sigmoid for this optimised value is displayed.



(a) Learning rate graph for Sigmoid activation function



(b) Training and validation loss function for $\eta = 0.1$ for sigmoid function

Figure 6: Optimising the Sigmoid activation function

Unsurprisingly, this yields much better results. The Sigmoid activation function performs just as well as the ReLU. The training loss is also below 5% after 2 epochs, and after 20 epochs they are completely similar. In the end, the Sigmoid and ReLU activation functions both are viable options for our classification problem and both will yield similar results.

(g) **Now evaluate the model for the SST test data set and determine model accuracy in correctly classifying El Niño and La Niña. Plot also the so-called confusion matrix, showing the number of the predicted events (false/true) versus the true events.**

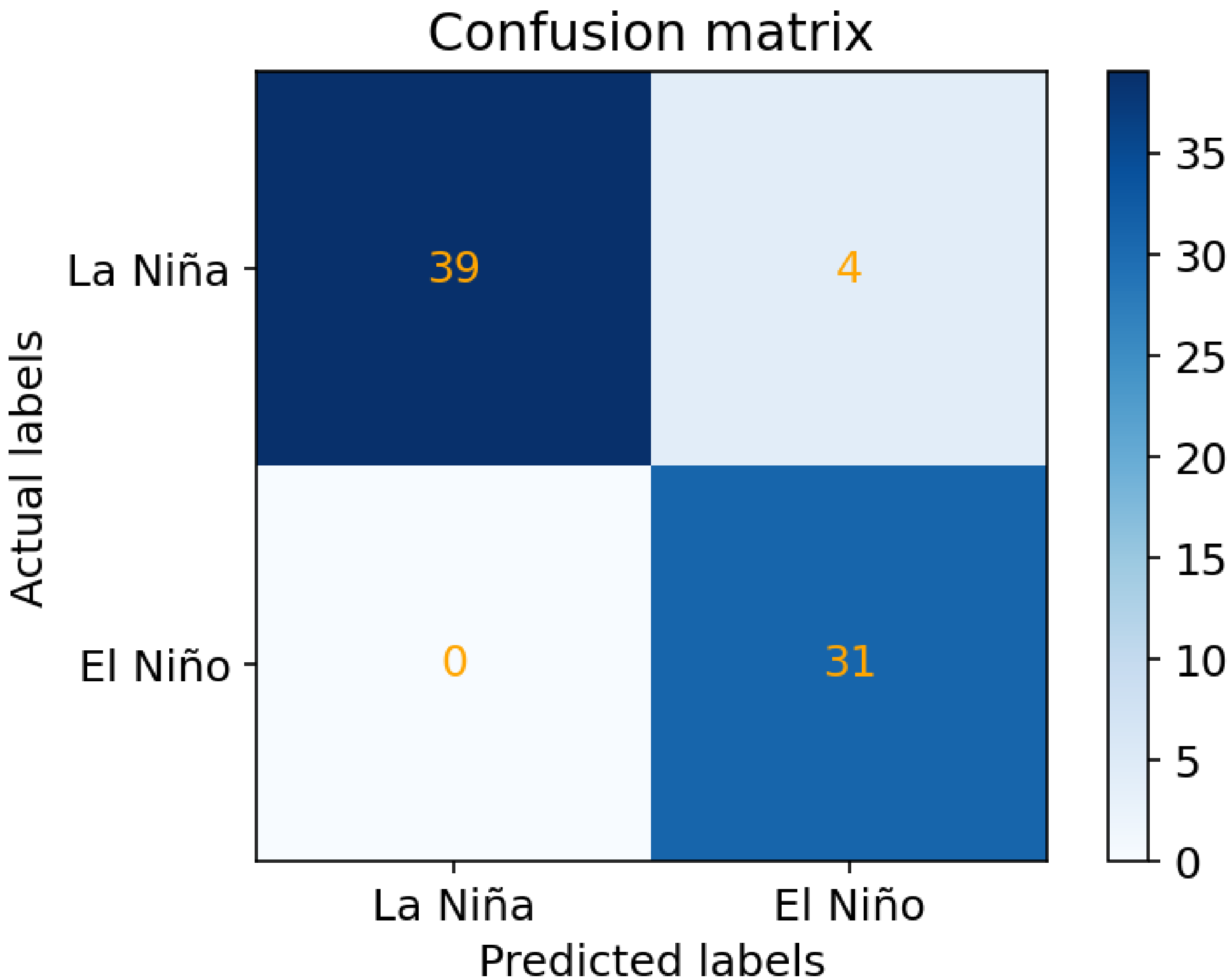


Figure 7: ReLU activation function, η=0.001

The confusion matrix tells us that the Machine Learning algoritm predicts 39/39 La Niña correctly, and 31/35 of the El Niño events correctly. For the Sigmoid activation function with $\eta = 0.1$ the same confusion matrix occurs. Taking a closer look at the mismatches between the actual class and the FNN class, see table 1.

Index	Time	Nino34	Timexr	Class	ClassML
798	2016.916667	-0.624338	2016-12-15 00:00:00	0	1
799	2017.000000	-0.644087	2017-01-15 00:00:00	0	1
800	2017.083333	-0.630447	2017-02-15 00:00:00	0	1
801	2017.166667	-0.543174	2017-03-15 00:00:00	0	1

Table 1: Mismatches between actual class and FNN assigned class

If we look at figure 1 we can see that the mismatched values are ever so slightly just a la Niña according to our predefined definitions, but for a very short time and with minimum values.

(h) **Finally, make composite plots of the global SST for El Niño events and global SST for La Niña events.**

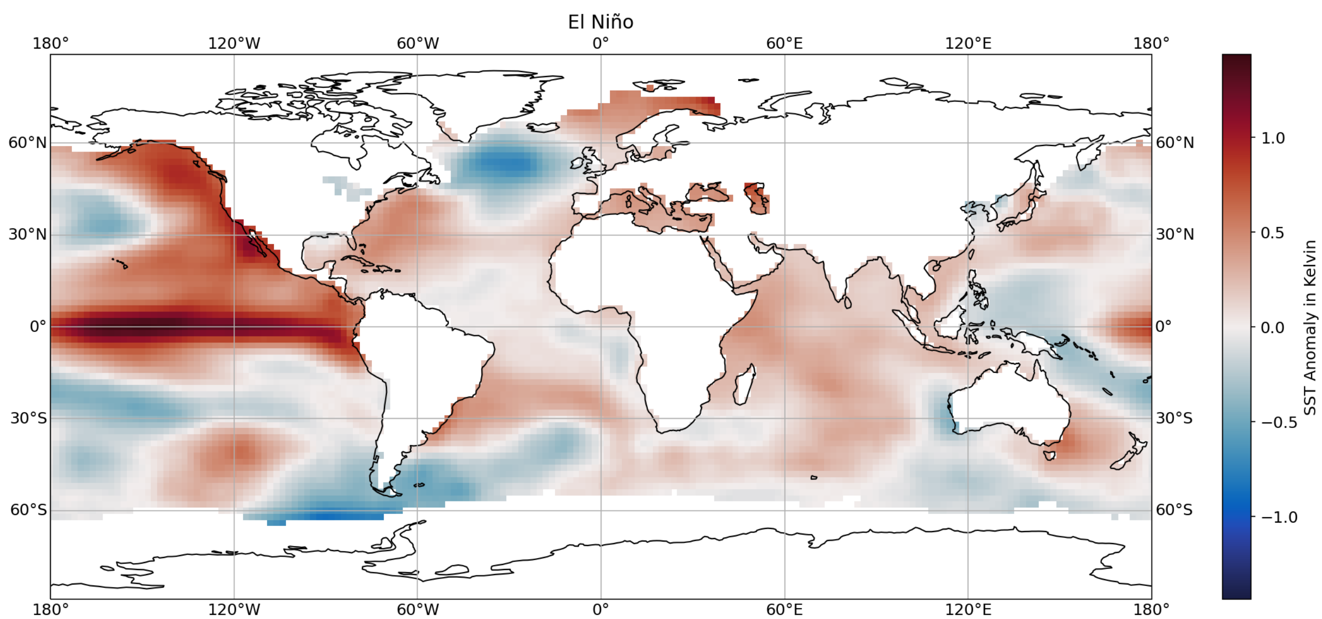


Figure 8: Global SST anomalies for correctly predicted El Niño

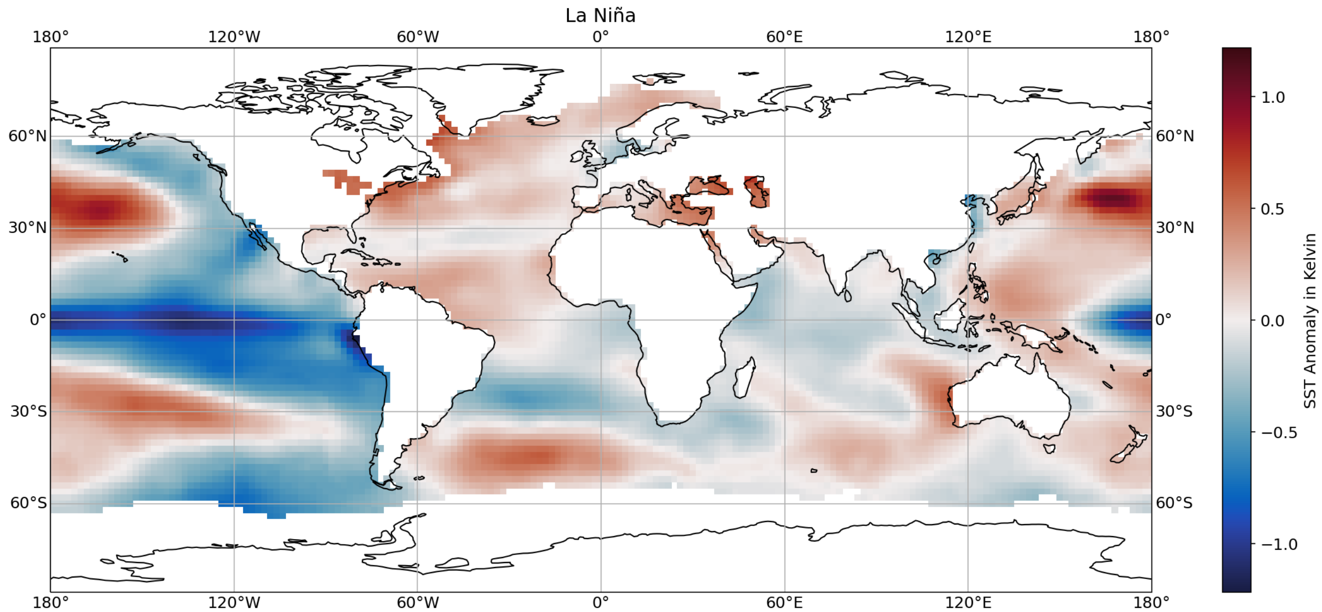


Figure 9: Global SST anomalies for correctly predicted La Niña

The classification is successful. Where the machine learning predicts La Niña we can see clear La Niña's, and where the program predicts El Niño's, we see clear El Niño's. Furthermore, it recognises most features besides from the obvious heating and cooling in the pacific ocean, it also recognises the Indian ocean dipole, and the NOA in the Atlantic Ocean. The wrongly categorised predictions from table 1 are plotted in figure 10.

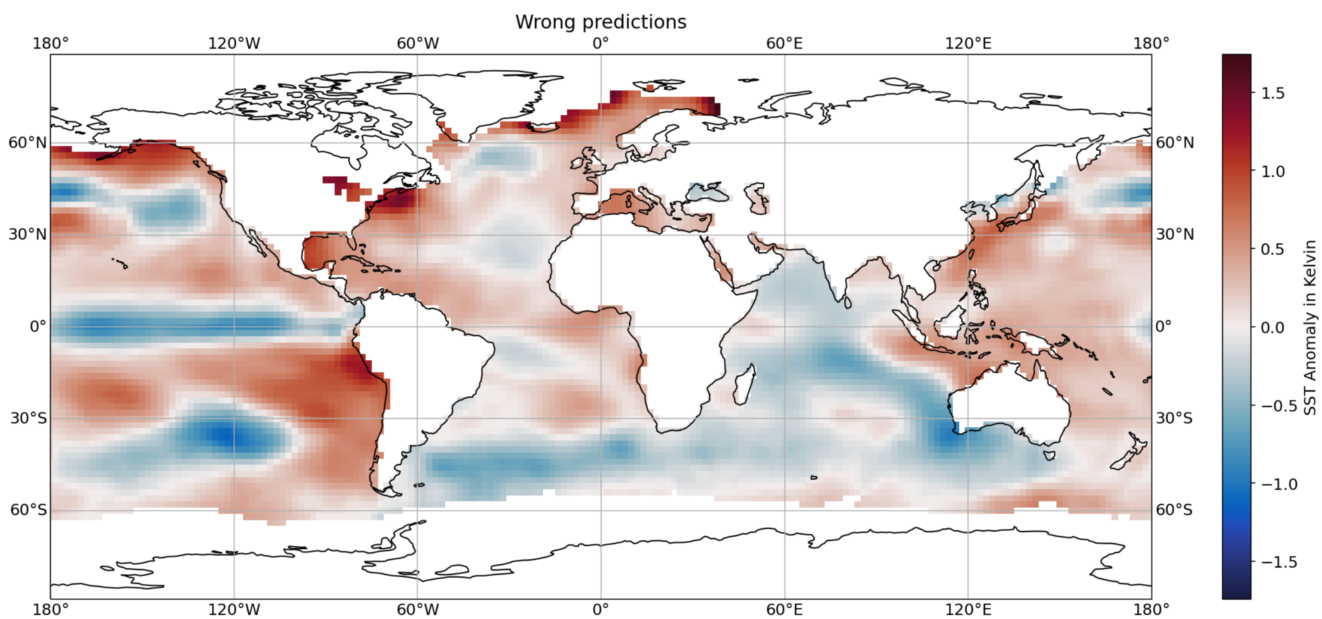


Figure 10: Misclassified predictions from table 1

It can be seen that in these SST there are no clear El Niño nor La Niña features, and since the machine cannot predict Neither an El Niño or La Niña, it is explainable that the program mispredicted these datapoints. This, since we trained the program on the global SST points and not the Niño3.4 region.

Appendix

See notebook attached