Companies bankruptcy forecast with autoencoders

Iván Cortes Garrido
Alejandro Clavera Poza

Introducction

In this presentation will explain the machine learning project that we have carried out.

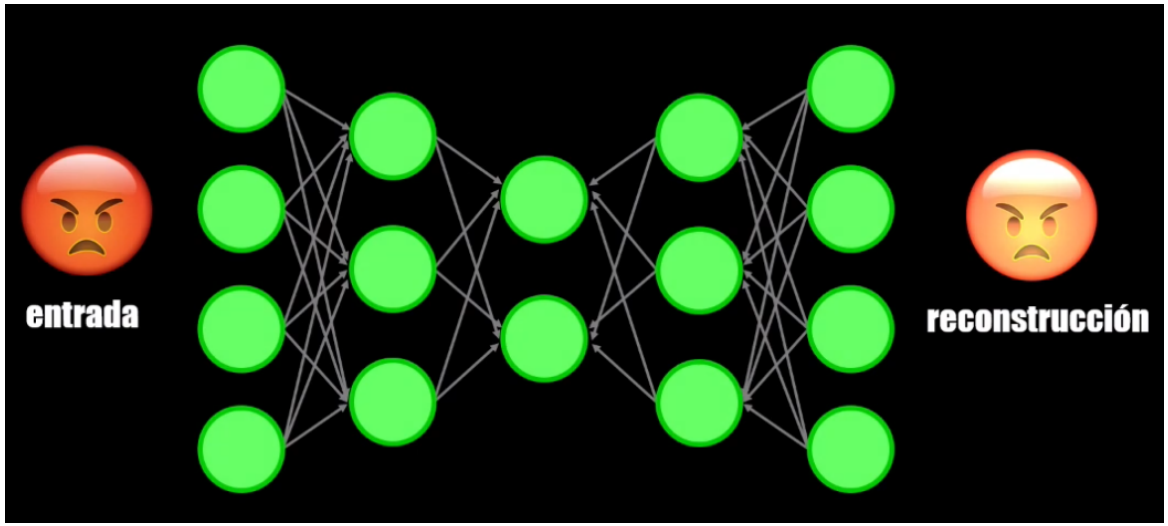
This Project consists of the prediction of the bankruptcy of a company in view of the financial situation of this, by means of autoencoders.

The initial idea was to use LSTM networks, a type of recurrent neural network, that that implements a long-term memory system.

But there was a problem, after analyzing the data we realized that they had no relationship between them.
Solution: Use of autoencoders

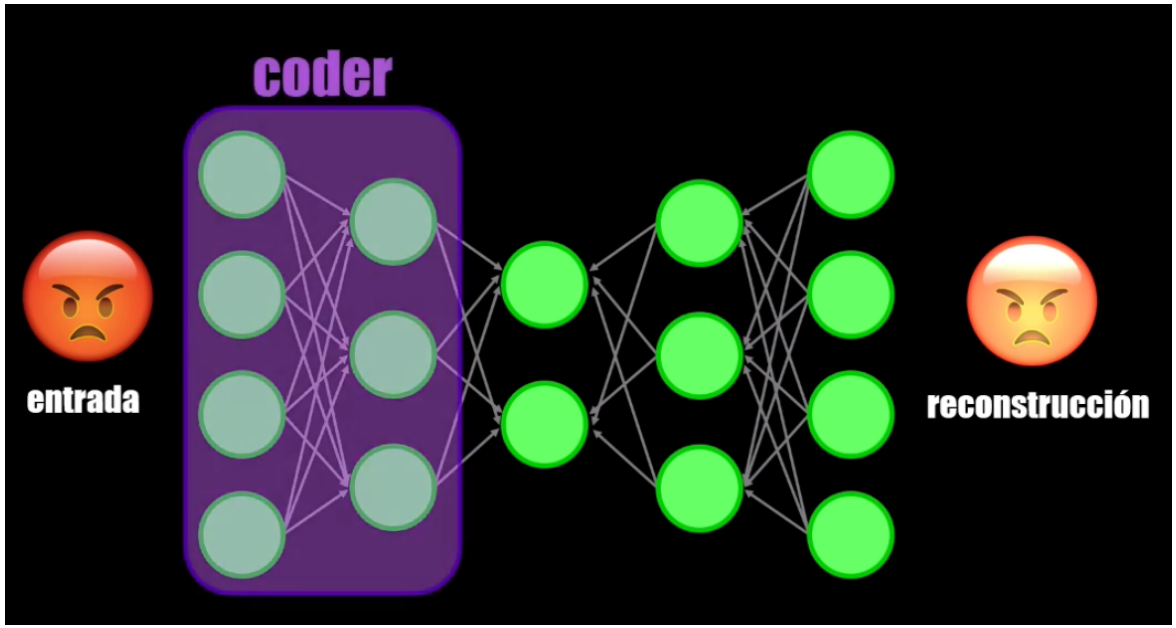
What is an autoencoder?

- Autoencoders are an unsupervised learning method
- Architecture that allows obtaining more compact versions of the input data to later generate new data



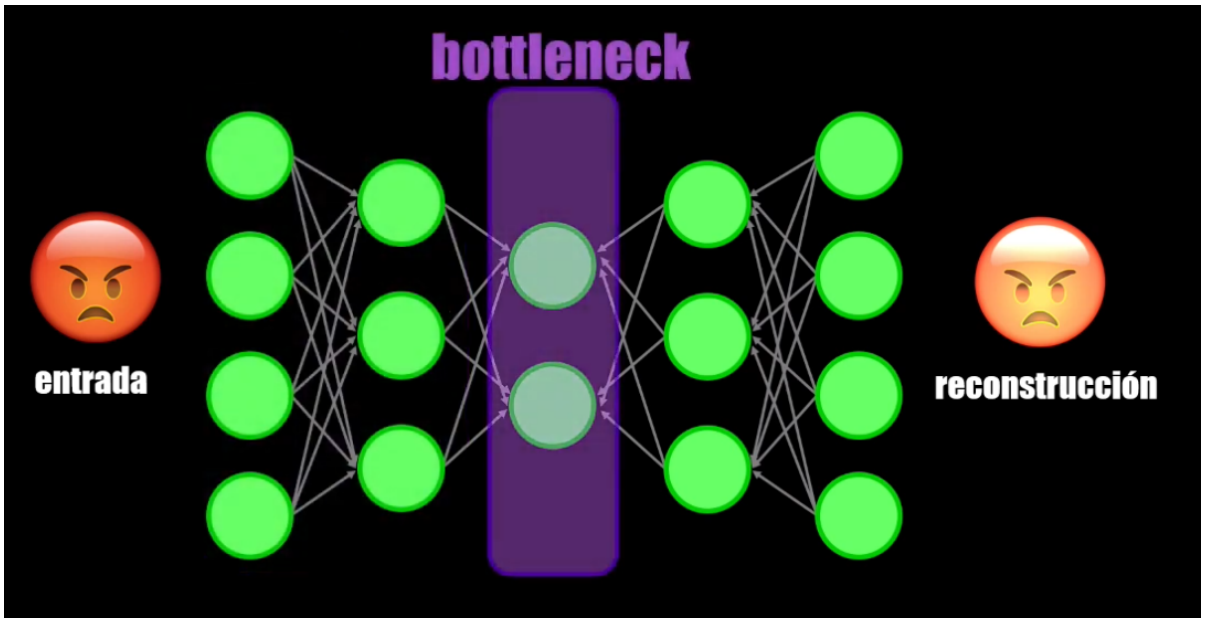
Parts of autoencoder

Encoder: The part of the network that compresses the input into a space of latent variables and can be represented.



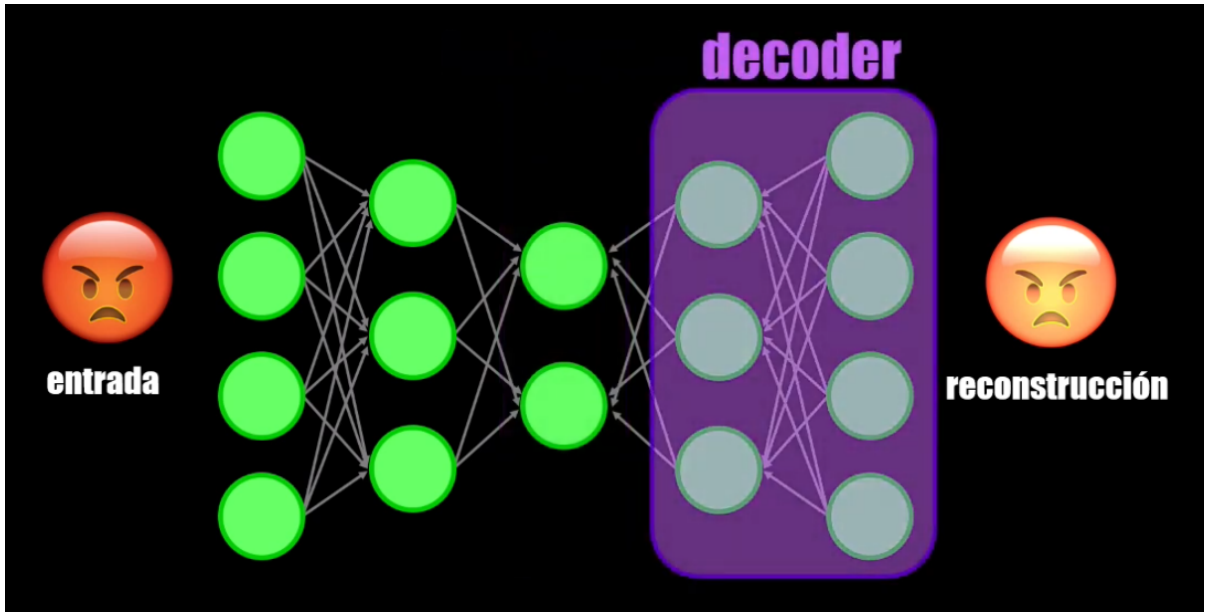
Parts of autoencoder

Bottleneck: This point of the autoencoder has a compact representation of the input.



Parts of autoencoder

Decoder: The part that tries to reconstruct the entry based on previously collected information.



Training of autoencoder

The Autoencoder is trained in a similar way to a Neural Network, but in this case the error function used to update the autoencoder coefficients is simply the result of comparing, point by point, the reconstructed data with the original data.

Aplication of autoencoder

Some of the examples of autocoders are:

- Soften the images
- Image noise removal
- Anomaly detection

Aplication of autoencoder

Some of the examples of autocoders are:

- Soften the images
- Image noise removal
- **Anomaly detection**

Anomaly detection with autoencoders

The procedure for detecting anomalies with autoencoders consists of training the neural network so that it learns to reconstruct the input data without anomalies.

This action causes that at the time of making predictions, if a data is passed with anomalies, the error is higher than if it were to introduce an no anomalous data. Therefore, by assigning a threshold value, we can determine that an anomaly has occurred when the error exceeds this threshold.

Data description

To begin with, it should be noted that we have worked with a reduced data set compared to the data sets that are normally used for the training of these architectures.

This data set contains information on different companies that are or are not bankrupt. For companies there are data on 64 characteristics of the financial status of the company

Data analysis

In addition to the characteristics discussed above, there is another attribute called class which indicates whether the company is bankrupt (1) or not (0).

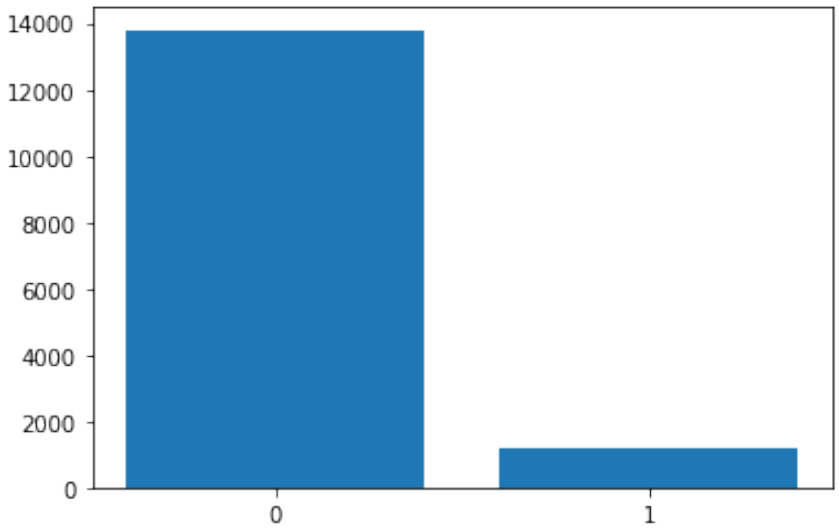


Figure: Number of samples for each class

Data analysis

Once the examples have been classified according to the class, we are going to analyze the number of different examples that we have for companies that are not bankrupt.

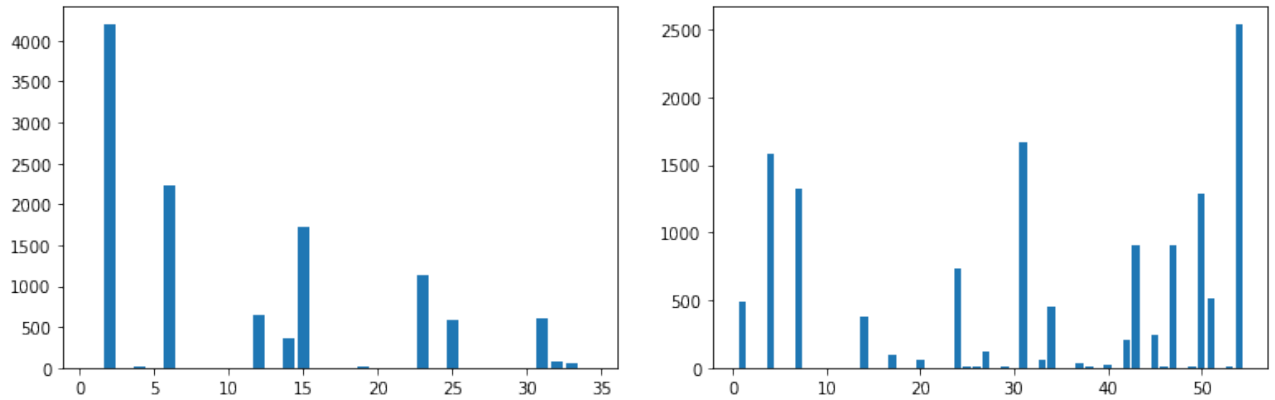


Figure: Numbers of different examples

Data Standarization

To facilitate the convergence of the algorithm the data is standardized to a mean 0 and std 1

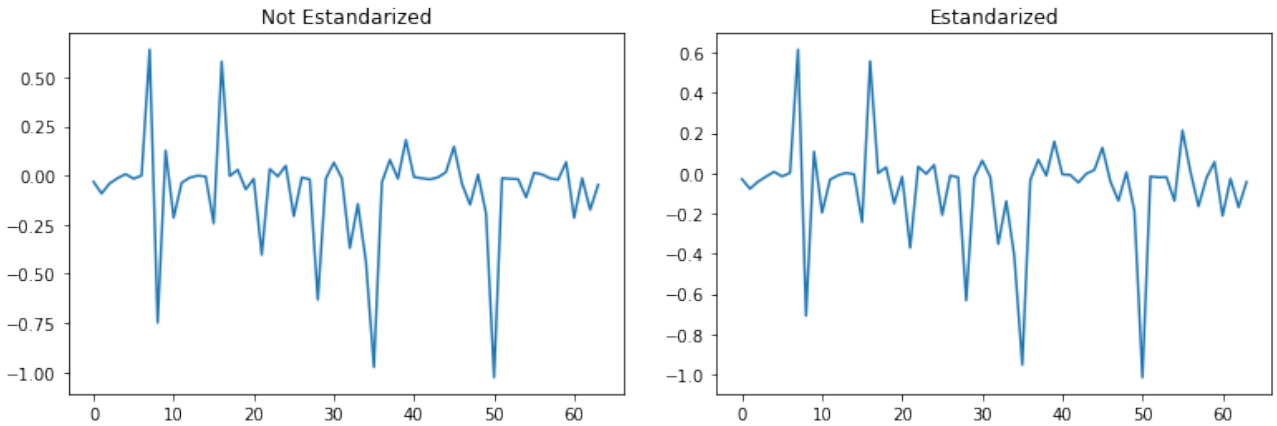


Figure: Standarizartion example

Model

For the development of the autoencoder, the Tensorflow library was used and specifically the Keras api that it offers.

This model will contain two fundamental elements:

- **Encoder**
- **Decoder**

Both parts have been implemented independently, using the Keras sequential model

Encoder

The layers used for the encoder are:

- **InputLayer:** Fixed layer that must have the same size as the input data, since in this case we work with 64 attributes, the size must be 64.
- **Hiden Layer:** Dense layer with 50 neurons and a hyperbolic tangent function
- **Output Layer:** The output layer is a Bottleneck that contain 40 neuron and Relu funtion.

Decoder

The layers used for the encoder are:

- **Hiden Layers:** Two dense layers with 50 and 54 neurons with a function of the hyperbolic tangent activation functions and Relu respectively. The first layer processes the encoder output.
- **Output Layer:** Fixed layer that must have the same size as the input data, since in this case we are working with 64 attributes, the size must be 64. The hyperbolic tangent is used as the activation function.

Complete model

A class has been used for the complete implementation of the autoencoder, this class has the encoder and the decoder as attributes.

This class when invoked for training or prediction joins both parts, passing the result of the encoder output to the decoder

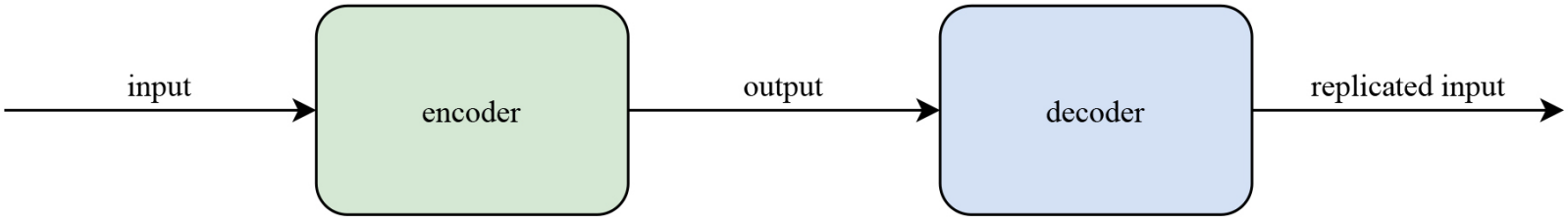


Figure: Model Arquitecture

Training

For the training of the network, how it has to replicate the input data, the values of x,y will be the same.

The number of epochs are 100 with Adam optimizer.

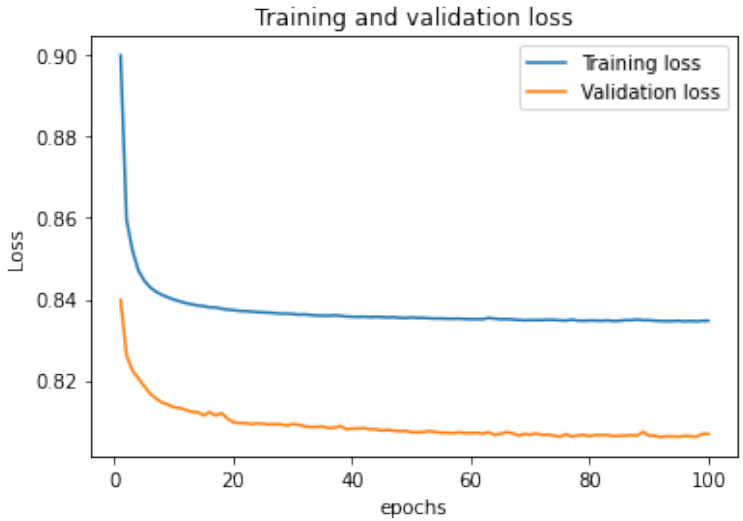


Figure: Training loss

Predictions

Steps to make a prediction:

1. Replicate the input data
2. Calculate the error of the replication with the same error function of training
3. Classify the data according to the threshold

Replication Examples

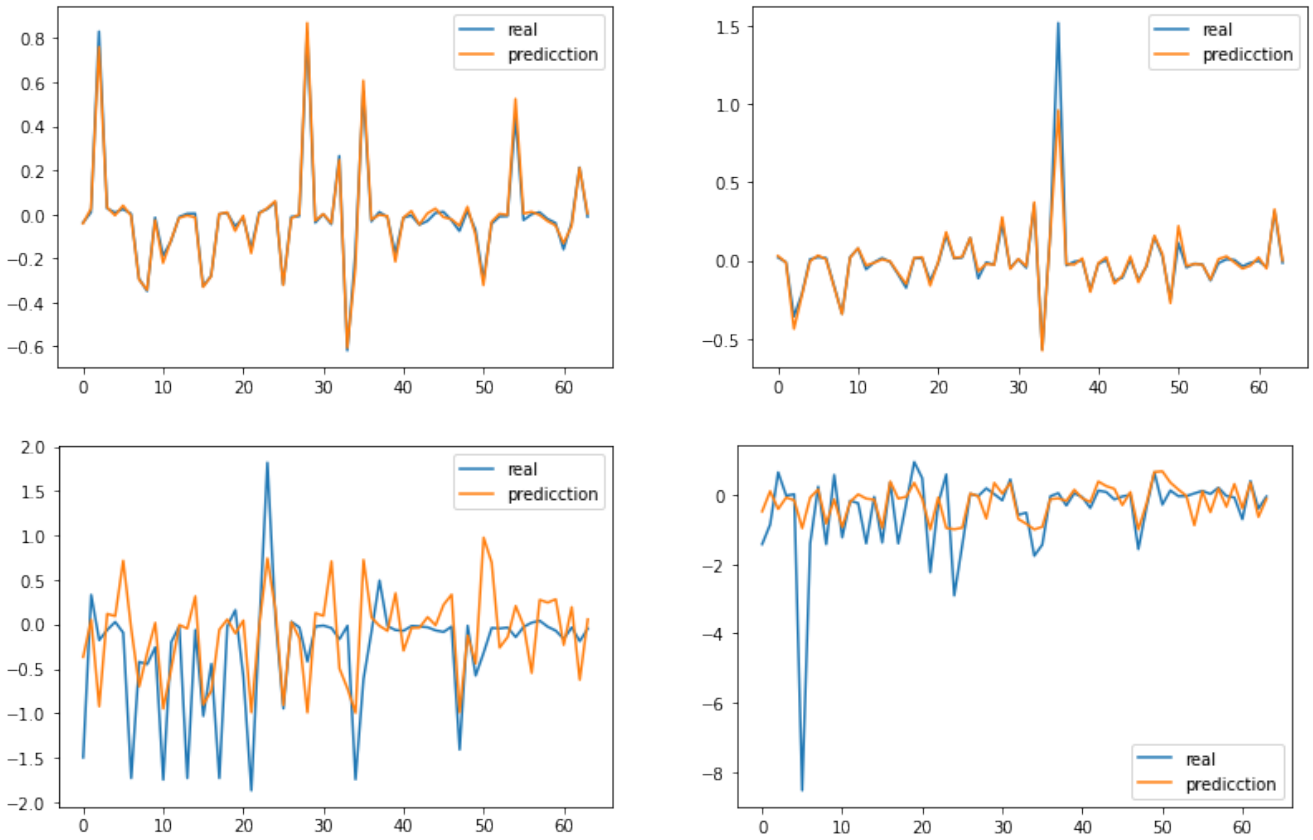


Figure: Good and bad reconstructions by the autoencoder

How to choose the threshold value

The choice of a threshold value is very important because how good the prediction can be depends on it.

An ideal would always make correct predictions, but this cannot be, therefore we have to follow some strategy to adjust this value.

Some metrics that can help us

- **Precision:** metric that measures the number of false positives generated by the model.
- **Recall:** metric that measures the number of false negatives generated by the model.

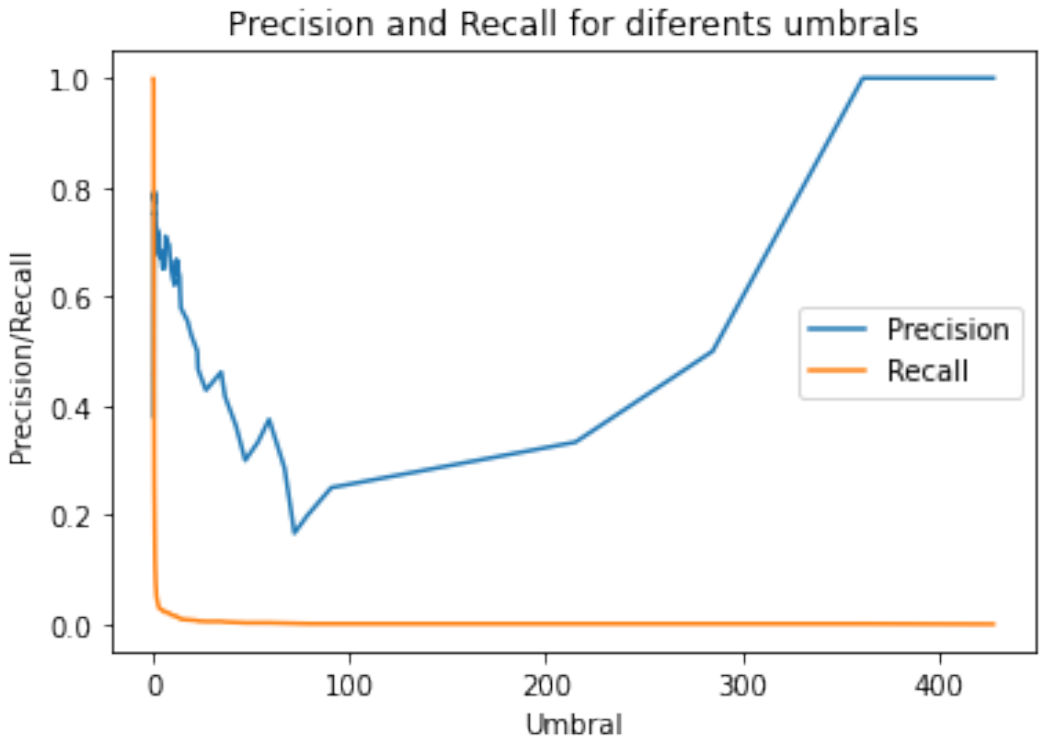


Figure: Precision and Recall example

Confuxion Matrix

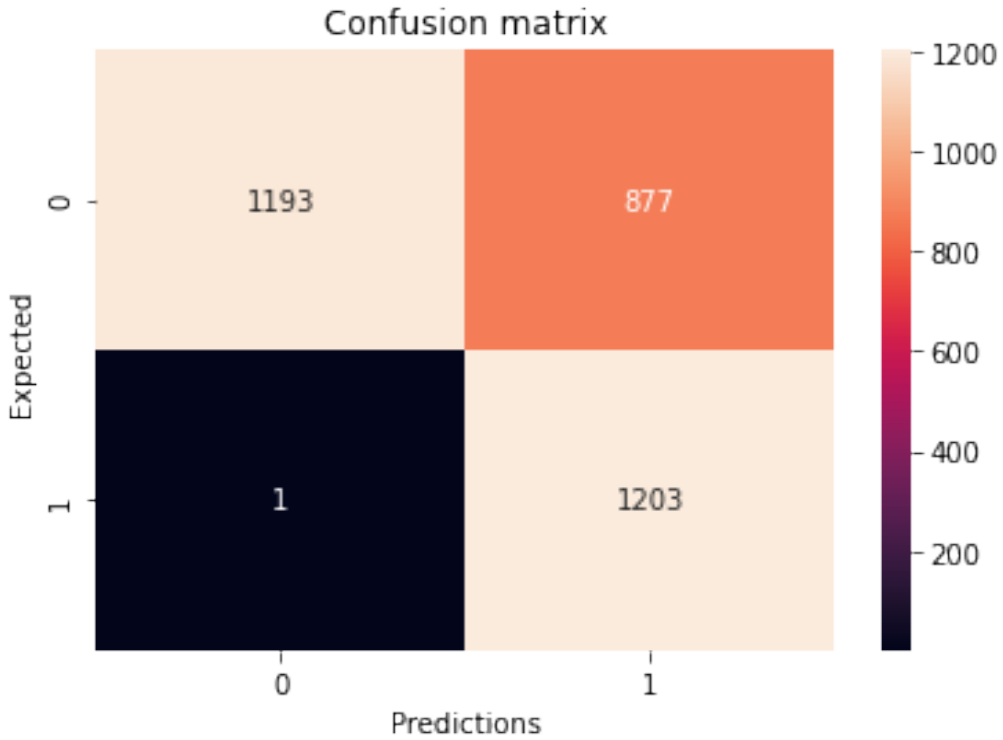


Figure: Precision and Recall example

Problems

At the time of training, we run into an underfiting problem,

Underfiting occurs when the model is not able to reduce the error of both the training data and the validation data.

Underfiting

Figure: Underfitting example

Solutions

The first thing we did was try to enlarge the number of neurons and the layer, but we didn't get many results.

Best solution: Expand the data set.

Biography


Companies bankruptcy forecast
https://www.kaggle.com/c/companies-bankruptcy-forecast/


Tensorflow Tutorial
https://www.tensorflow.org/tutorials/generative/autoencoder


Applied Deep Learning - Part 3: Autoencoders
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798


Autoencoder Example(Spanish)
https://www.codificandobits.com/blog/autoencoders-expli