Contents

[1. TASK DEFINITION 1](#_Toc42647920)

[2. EVALUATION MEASURE 2](#_Toc42647921)

[3. APPROACH 2](#_Toc42647922)

[3.1. Data preparation 2](#_Toc42647926)

[3.2. Modeling 3](#_Toc42647927)

[3.3. Accuracy and Loss 5](#_Toc42647928)

[4. ERROR ANALYSIS 6](#_Toc42647929)

[5. CONCLUSION 6](#_Toc42647930)

[6. APPENDIX 7](#_Toc42647931)

# 

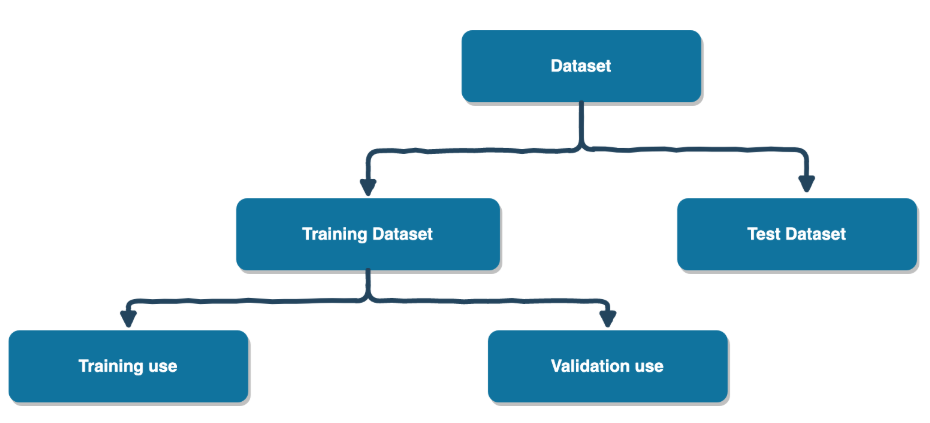
# TASK DEFINITION

The purpose of creating a model to predict in a binary way if it is a good moment to buy or not a particular stock. For this work, I decided to analyze companies in the Brazilian index Bovespa. I chose not to focus on day to day information like stock prices and volatility but to concentrate on the company's fundamentals. In our view, a buy and hold strategy, rather than day trading that uses graphic patterns, based on the concepts of investors like Benjamin Graham and Warren Buffett, proves to deliver a higher return in the long run. I also understood that it was essential to remove all financial companies from our dataset because of their particularity compared with the remaining sectors.

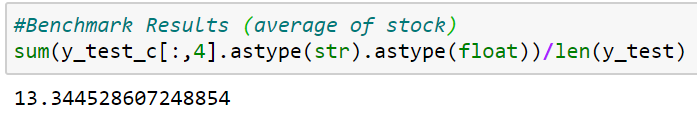
Since today many Ibsites provide a considerable amount of information on listed companies for free, I decided to collect a massive dataset on the fundamentals of dozens of companies. This information is available on the balance sheet reported quarterly or yearly by these companies. Indicators like Price to Earnings Ratio (PER), Earnings per Share (EPS), dividend yield, debt ratio provides a reliable portrait of the expected impact on the price of a stock. This type of investment is also known has value investing and involves picking stocks that are trading for less that their book value. Knowing the true value of stocks allows finding out undervalued companies and can save us a lot of money when I decide to "buy" (value = 1) or "do nothing" (value = 0).

I analyzed more than 50 predictor variables such as debt, dividends, return on assets and equity, among others. On the other hand, our response variable would be profitability in the year, that is (future price / current price) - 1. Since I wanted profitability in one year, our database was built to obtain only one observation per company per year, thus guaranteeing data independence in the different stages of the deep learning process.

After collecting the data, I organized, cleaned, and standardized the data to create a uniform dataset adjusted to the deep learning algorithms. I split the dataset into two, training and testing. With the training dataset I fit the deep learning algorithms and validate our model. Finally, with the testing dataset I verified the final results. The image shows what I did in our project.



Our goal was to try to make profit greater than a naïve approach. For the naïve approach, I simply took the average of the dataset stock's profitability which is 13,34%, which is already a very good 1-year profitability.



# EVALUATION MEASURE

I found that for our type of analysis it would be necessary to run the algorithm at most once a month. Therefore, for the evaluation measure, our concern was only related to the assertiveness of the algorithm and not to the time spent by it.

For this reason, to predict the model's capacity, I choose the standard metrics of accuracy and loss and, for the loss function I use "binary crossentropy".

# APPROACH

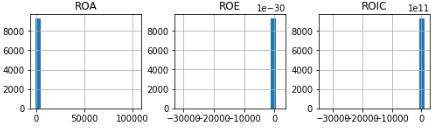


## Data preparation

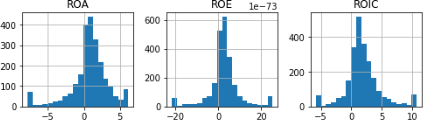
Similar to the work I did in the machine learning course, I had many columns in our dataset, so I decided need to reduce dimensionality by eliminating some variables highly correlated to others. Also, with our business understanding, I eliminated rows that didn't provide important information for us. The indexes Ire 'IPL', 'PVPA, and 'AT'. The first and the second Ire pick for its importance in the fundamental analysis domain. The third ('AT') because I decided to use it to normalize some columns of the dataset.

Other important challenge I had to overcome was the presence of some outliers. As I observed the data with histograms, I saw very high values on the extremities, so I decided to use the *Winsorization* approach to handle extreme values in the limit. This method allows the limiting of extreme values in the dataset to reduce the effect of outliers, as I can see in histograms before and after the *Winsorization.*

*Before:*

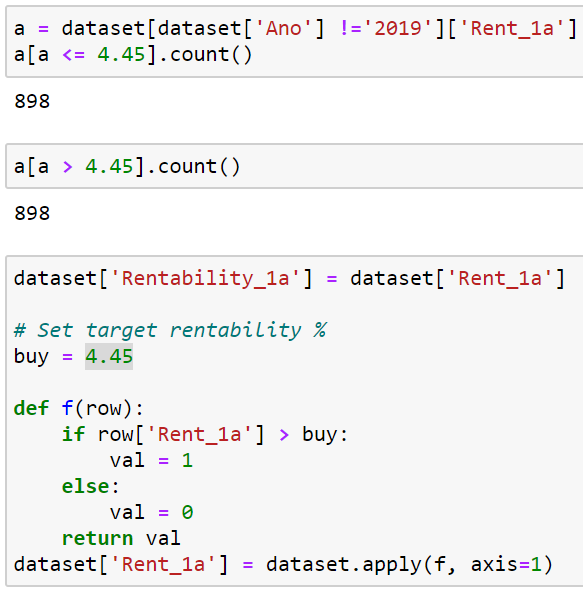


*After:*



After that, I had to choose betIen normalize and standardize the data. After some tests I observed that the standardize the data brought better results.

I also divided the dataset into two parts, the one with the highest and loIst profitability on a 1-year period. The purpose of this manipulation was to simplify and mimic the process of a "buy and hold" investor with a binary decision algorithm, whether to buy or not to buy. Therefore, to achieve a better profitability than the naïve approach, I developed a deep learning algorithm that aims to find top 50% stocks from our dataset. For this, I verify the profitability that divides the data in two equal parts.



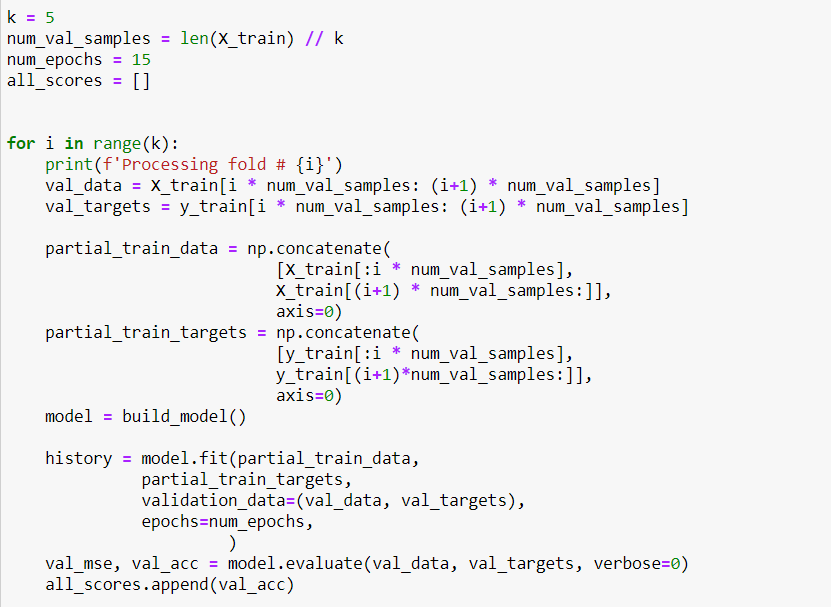
As the picture shows, the cutting profitability on a 1-year period was 4.45%. Thus, the 898 data with a profitability greater than 4.45% received category 1, while the remaining data, with a loIr profitability, received 0.

## Modeling

For modeling, I try to use different variations of deep learning algorithms. I modify different parameters; optimizers like "SGD", "rmsprop" and "Adam"; learning rates; amounts of layers and units; activations functions "relu", "sigmoid", "tanh"; epochs and batch sizes. For the loss function I tested only the binary entropy, since "Hinge Loss" is interesting only if more speed is needed.

I did more than 30 tests before finishing on a model that I found satisfactory. When performing the modeling I try to keep it simple. The final model was as shown in the image below.

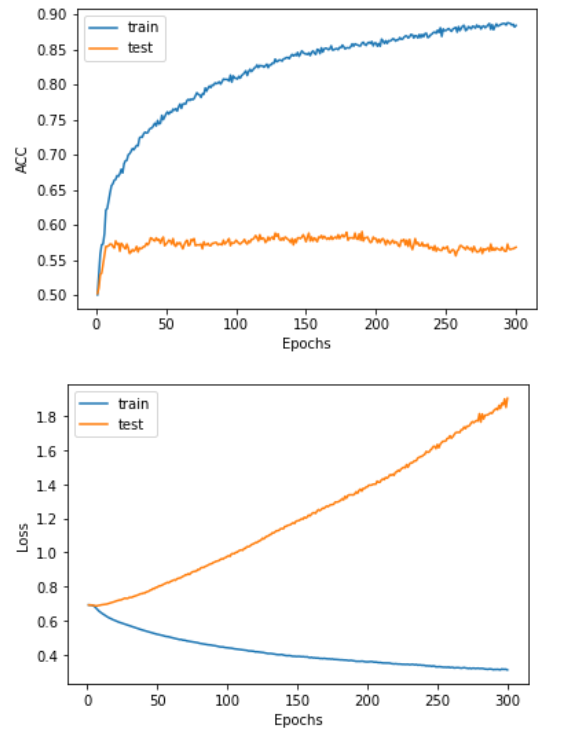




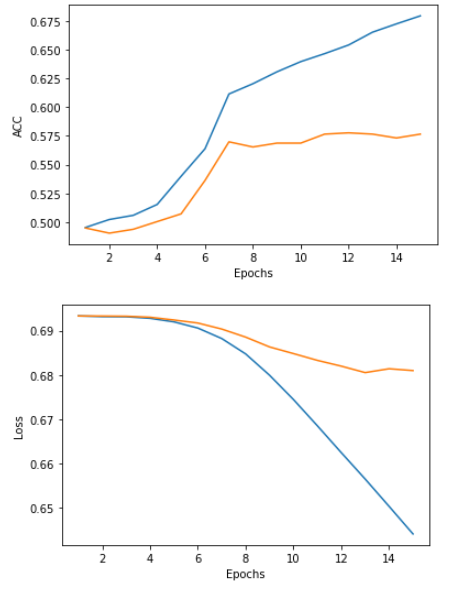
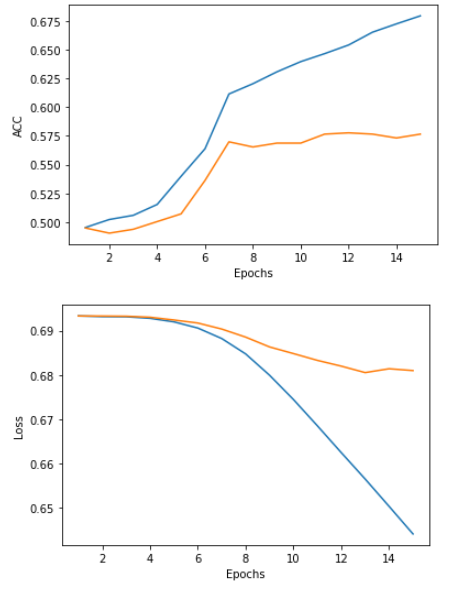
I break the dataset in 5 for cross-validation.

## Accuracy and Loss

After modeling, I checked the accuracy of the algorithm in the testing and training phase. It became evident for us after some tests that I couldn't insert a high number of epochs due to overfitting. As it is possible to observe on the figure bellow:



After few epochs the accuracy stops rising, while lost function rises continually. Therefore, I tried few numbers of epochs in order to catch the minimum loss. Evidently, I did not expect results in which the test curve was very similar to the training curve, as I are trying to predict future stock prices and dealing with real stock exchange data. But, although the accuracy value in training was much higher than that of the tests, I Ire satisfied with the results of the tests that shoId an increase, indicating that, despite a certain overfitting, there was real learning of the algorithm.

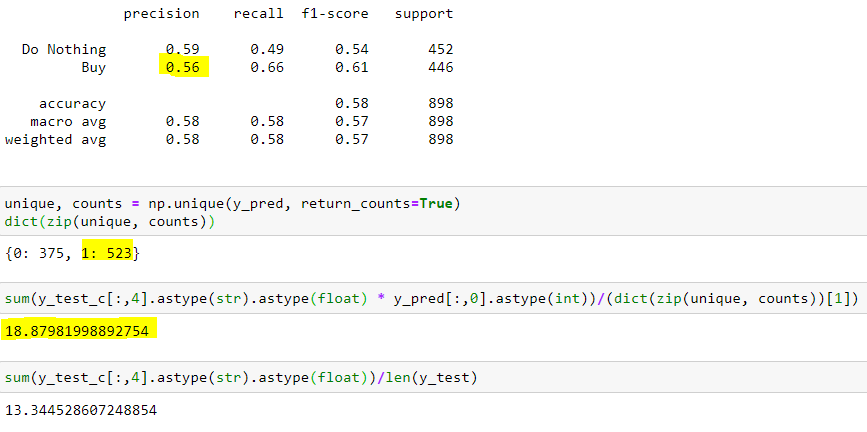


The loss function decreases to its minimal with 15 epochs.

# ERROR ANALYSIS

Primarily, I noted that the algorithm only needed to be precise in "buy" decisions. As an example, if I imagine how it happens in practice: an investor wants to invest 10k euros and there are 100 shares that he can buy, the algorithm would need to indicate as accurately as possible the "buy" type of actions, whether it will indicate 1 or 100 shares does not matter, as long as the algorithm is precise in its "buy" decision.

As the figure below shows in the confusion matrix, I believe that the algorithm performed very Ill.



It got 56% of the purchase decisions right, which generated an average return per share of 18.88%, surpassing the profitability of the naive method (13,34%) by 40.3%.

# CONCLUSION

The objective of this project was to build a model that could forecast stocks with an average annual profitability greater than 13,33%. I used the knowledge acquired during this course in order to develop an algorithm capable of helping us with our decisions of buying stocks. I tried to simplify and I created two categories: "Buy" or "Do nothing", simulating the binary way of thinking in the moment of a buying decision.

I conclude this project understanding the robustness of the Deep learning algorithms, as I overperform our benchmark.

# APPENDIX

Description of the variables of the dataset



