ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

AI in Financial Market

TREND PREDICTION IN FINANCIAL TIME SERIES: A MODEL AND A SOFTWARE FRAMEWORK

CANDIDATE
Alessandro Ghibellini

SUPERVISOR
Prof. Michele Lombardi

Academic year 2020-2021

Session 3st

Ogni giorno sapevo dove ero arrivato

ma non dove sarei arrivato.

Ora sono alla fine ma penso di essere all'inizio.

Invece no è la fine ci sono riuscito!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivations

The financial market is a huge market dominated by the Asset Management Industry which mainly specializes in active funds. A couple of decades ago, investing was not a big deal due to the fact that the price of money was high, so the investment was with a low level of risk and the interest was around 10 % per year. Unfortunately nowadays on the one hand, the most solid states, such as Germany, have negative interest rates. Moreover the average of the P/E indicator considering stocks is extremely high. On the other hand the commission costs w.r.t. active funds are still around 2/3% for year. As a consequence of it, funds are forced to bet on more dangerous or unethical business in order to make a profit for their clients. Due to this condition, today the focus shifts to ETFs, exchange traded funds, which passively follow a set of predefined rules using automatic procedures. In fact, it treats costs well but not as well in terms of risk management. For this reason, AI's goal in this field is to try to build an Active ETF, reducing the costs of asset management funds without losing their performance and their ability to manage risk. Additionally, this cost reduction can be invested in less dangerous and more ethical stocks. Indeed using the enormous amount of data already available, it is possible to build models capable of making decisions, quickly and precisely, in order to maximize profit and minimize the level of risk.

1.2 From AI in Industry to AI in Financial Market

On one hand the Industry 4.0 revolution has completely changed the Industrial sector and the studies in this regard are quite solid . On the other hand studies about Financial Market are not yet mature. In fact, AI in Financial Markets is not well covered mainly due to the fact that it is not clear what drives the market while it is well covered that timing is key and there are many aspects to be considered at the same time. In this scenario, AI can tackle the problem efficiently and accurately, which will drive a revolution similar to the one behind Industry 4.0. So even though these two fields appear to be completely different, they do have similarities. First in both there is the presence of time series and there is already a vast literature on how to deal with it. Second point is the presence of an Expert in the sector. Indeed in the Industrial case the presence of the Expert is very important such as for labeling the data. Same in Financial case, Experts have a number of good practices that can be used to make the input more informative and the output reasonable.

1.3 AIM of the Research

The research aims to build autonomous support for Traders that can be translated into an Active ETF in the future. The first part of the work was the design of the Framework, the foundation of the research. It defines how to create the application and how the application can be used by Experts, structuring the problem into Input, Output and Model parts. Secondly, there was an intense understanding of the financial market in collaboration with Aivolution S.R.L.[2]. Mainly this part of the work aims to collect all the information from the Experts, such as technical indicators that can be used to feed the Input, considerations on the length of the future horizon and definition of market states. Later it was possible to define the Input and Output parts. The last part of the work was the design of a Model that is able to make considerations on the future states of the market having a certain input and a certain length of the future horizon, therefore it is bound by the design choices of the Experts. In the end, the research not only offers a flexible Framework that exploits the presence of Experts in the sector, but offers a solution capable of predicting the future state of the market that can also be used in the wild. Furthermore, this solution can be extended by considering a portfolio analysis and linked to a professional broker such as Interactive Brokers [6] in order to build an effective Active ETF.

1.4 Contributions

My research is characterized for a huge focus on Problem formulation and an accurate analysis on the impact of the input and the length of the future horizon on the results. I will demonstrate that using financial indicators already used by professional Traders every day and considering a correct length of the future horizon, it is possible to reach interesting scores in the forecast of future market states, considering both accuracy, which is around 90% in all the experiments, and confusion matrices which confirm the good accuracy scores, without an expensive Deep Learning approach. In particular I used a 1D CCN. I also emphasize that Classification appears to be the best approach to address this type of prediction in combination with proper management of unbalanced class weights. In fact it is standard to have a problem of unbalanced class weights, to inconsistent trend movements. Finally I proposed a Framework which can be used also for other fields which allows to exploit the presence of the Experts of the sector and combining this information with ML/DL approaches.

1.5 Outline

This is the structure of the Thesis:

- **Chapter 2** It introduces the basic knowledge in order to understand the rest of the work. In particular it begins with the introduction of the Stock Exchange and the Convolutional Neural Network. It ends with a part of literature.
- **Chapter 3** It defines the Problem Formulation, dividing the problem into sub problems with a discussion about possible design choices. It ends with a description of the Solution choices.
- **Chapter 4** In this chapter there is a description of the Framework, in particular it describes its components and how it works.
- **Chapter 5** It is related to the Experimental part. First it defines all the Experimental settings, the model used and the baselines, the evaluation metrics and the list of Stocks and Indexes. Finally for each case it shows the results with a discussion about them.
- **Chapter 6** It draws the conclusions of the presented work. Moreover it specifies the major contributions and introduces possible future improvements.

Chapter 2

Background and related work

2.1 The Stock Market

A stock is a financial instrument that represents ownership in a company or corporation and represents a proportionate claim on its assets (what it owns) and earnings (what it generates in profits). Stocks are also called shares or a company's equity. Stock ownership implies that the shareholder owns a slice of the company equal to the number of shares held as a proportion of the company's total outstanding shares. For instance, an individual or entity that owns 100,000 shares of a company with one million outstanding shares would have a 10% ownership stake in it. Most companies have outstanding shares that run into the millions or billions. Stock Exchange are markets where existing shareholders can transact with potential buyers. It is important to understand that the corporations listed on stock markets do not buy and sell their own shares on a regular basis. Companies may engage in stock buybacks or issue new shares but these are not day-to-day operations and often occur outside of the framework of an exchange. So when you buy a share of stock on the stock market, you are not buying it from the company, you are buying it from some other existing shareholder. Likewise, when you sell your shares, you do not sell them back to the company rather you sell them to some other investor. In few words it follows the Supply and Demand rule. Indeed the Stock Market also offers a fascinating example of the laws of supply and demand at work in real-time. For every stock transaction, there must be a buyer and a seller. Because of the immutable laws of supply and demand, if there are more buyers for a specific stock than there are sellers of it, the stock price will trend up. Conversely, if there are more sellers of the stock than buyers, the price will trend down. The bid-ask or bid-offer spread (the difference between the bid price for a stock and its ask or offer price) represents the difference between the highest price that a buyer is willing to pay or bid for a stock and the lowest price at which a seller is offering the stock. A trade transaction occurs either when a buyer accepts the ask price or a seller takes the bid price. If buyers outnumber sellers, they may be willing to raise their bids in order to acquire the stock. Sellers will, therefore, ask higher prices for it, ratcheting the price up. If sellers outnumber buyers, they may be willing to accept lower offers for the stock, while buyers will also lower their bids, effectively forcing the price down. In addition to individual stocks, many investors are concerned with stock indices, which are also called indexes. Indices represent aggregated prices of a number of different stocks, and the movement of an index is the net effect of the movements of each individual component. When people talk about the stock market, they often allude to one of the major indices such as the Dow Jones Industrial Average (DJIA) or the SP 500. The DJIA is a price-weighted index of 30 large American corporations. Because of its weighting scheme and the fact that it only consists of 30 stocks (when there are many thousands to choose from), it is not really a good indicator of how the stock market is doing. The SP 500 is a market-cap-weighted index of the 500 largest companies in the U.S. and is a much more valid indicator. Indices can be broad such as the Dow Jones or SP 500, or they can be specific to a certain industry or market sector. Investors can trade indices indirectly via futures markets, or via exchange-traded funds (ETFs), which act just like stocks on Stock Exchanges.[12][7]

2.2 Convolutional Neural Network

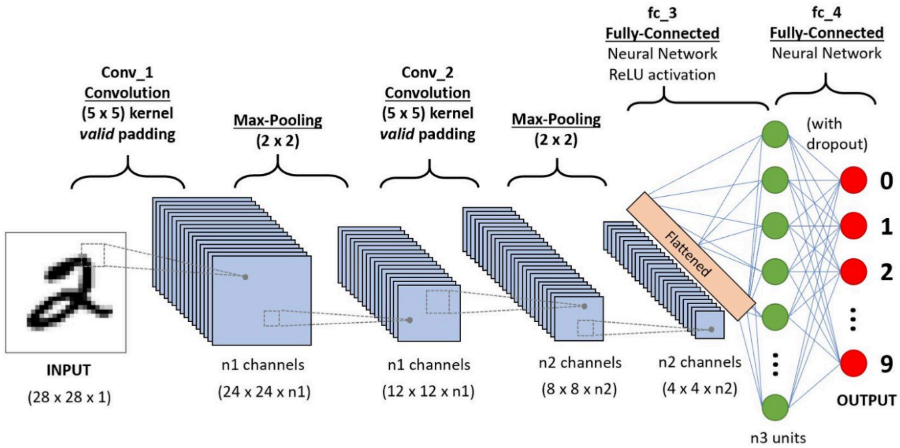

Figure 2.1: CCN Architecture

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter. The filter moves to

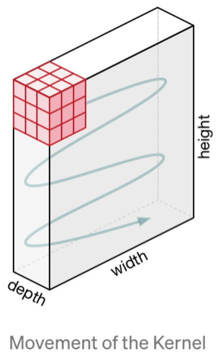

Figure 2.2: CCN Kernel movement

the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed. In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would. There are two types of results to the operation, one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in case of the former, or Same Padding in the case of the latter. Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. Finally adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.[10]

2.3 1D CNN

CNNs share the same characteristics and follow the same approach, no matter if it is 1D, 2D or 3D. The key difference is the dimensionality of the input data and how the kernel slides across the data. In the figure 2.3 there is a practical examples. On one hand considering NLP, a sentence can be seen as a sequence of 9 words. Each word is a vector that represent a word as a low dimensional representation. So, the kernel will always cover the whole word, while the height of the matrix determines how many words are considered.In the example the height is two, so the kernel will iterate 8 time through the data. On the other hand for 2d example we can see a classical Computer vision problem. Considering an image each pixel is represented by its x, y position as well as its RGB value. The kernel in this case slides both horizontal and vertically across the image. [11]

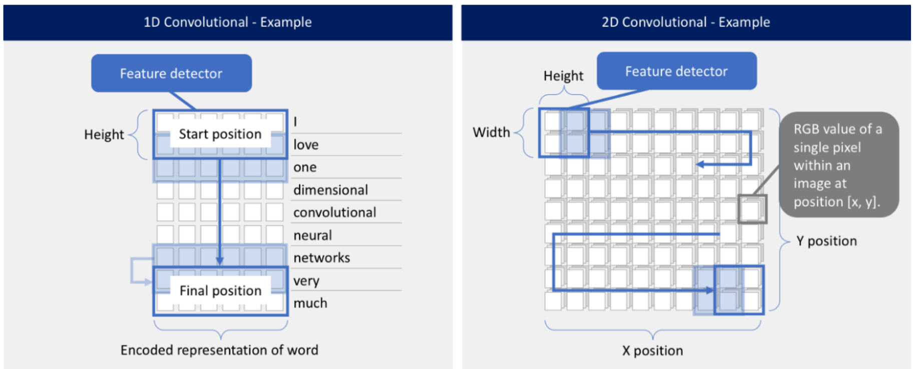

Figure 2.3: CCN1D vs CCN2D

2.4 Literature review

For what concern Literature there a wide variety of solutions. First, the solutions can be splitted into regression or classification approaches. Second, they are different for the kind of techniques which are used. The paper [8] has demonstrated a good performance using a RNN approach based on a regression output. I was impressed on how much closer this model is to the observed price trend. Moreover it is interesting the usage of an informative parameter such P/E. Although it is a good result, in a real case scenario an accurate regression cannot be so useful. Indeed this approach over complicate the situation, of course in the experiment case it works quite well but the price line is quite unpredictable without a combination of time series forecasting and a Sentimental Analysis [1]. On the other hand it is difficult to find a standard way to classify the states of the market. Another point to keep into account is the correlation between price and financial indicators. The paper [4] is focused on the correlation of this factors over the price and it is something in the direction of my thesis. For what concern the decision of using a 1D CNN in my work, it was done because this technique was very effective in this my previous work [5] so, I had some experience in this approach. While in order to do a proper comparison I took the paper [3] because it is quite close to my proposal. Indeed it is based on a classification approach using as a baseline a 1D CCN. The proposed system exploits the GAF imaging approach for encoding time series data as images. The classification phase is carried out by organising in an ensemble a set of CNNs which have the same architecture, but each of them is initialized with a different kernel function for initialization. A majority voting-based policy is adopted in order to take the final decision. I started from this and I noted some problematic aspects:

- Only Close Price as Input.
- Only one day as future time horizon.
- The system can suffer the problem of unbalanced classes. The choice of considering positive/negative day based on the fact that the close price is above or over the previous day does not seem convenient.

My solution will demonstrate a better performance, considering accuracy and confusion matrix, using a 1D CNN with 30 days as future time horizon and input window size. It will prove how much the choice of the correct input, the rules behind the classification and the length of the future horizon are influential in the results.

Chapter 3

Problem formulation and solution techniques

3.1 Problem formulation

In order to deal to Financial Market there are a list of problems which are to be defined. First if it is a regression or a classification problem. Second how much far away the prediction has to be done, because different future time horizons result in different scoring performance, same for what concern the size of the input (temporal size, number of features). Last what kind of ML/DL techniques to use.

3.1.1 Regression or Classification

On one hand Regression allows to check the stock price movements with an high level of details. On the other hand a Classification allows to simplify the problem, as it can reduce the problem into a finite number of classes. Considering pros and cons, Classification seems the best choice because the rules of market are simple, Buy before the price will grow a lot, Sell before the price will decrease in strong way. So knowing the exact amount of % which describes the price movement is not so useful. Indeed what is really important is if in the future the price will be quite the same or it will change of a consistent quantity. For this reason it will be clear that the problem can be splitted into 3 categories which describe the market situation:

- Bear (Decrease situation).
- Stable.
- Bull (Increase situation).

3.1.2 Future time horizons

Another important point is the length of the future prevision because it changes completely the complexity of the problem. Ideally a prevision is harder if it predicts something that is very far away from today and it will be simpler predicting something close to today. It is the case of weather forecasting. Unfortunately the Financial Market does not follow a precise rule and often its behavior it's completely senseless. This is due to the fact that considering intra-day prediction or a couple of days later, the information inside the time series contains a lot of noise. Looking to the pictures it is possible seeing a Bull movement which is clear considering days, while it is cover with a lot of noise considering hours.

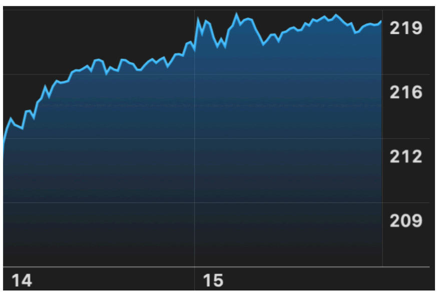

Figure 3.1: NVIDIA price within 2 days,14 and 15 october 2021

Another important aspect is the commission costs. Each movement in the market has a price, a part related to the Broker and another one related to the country where the trading is done in form of taxes. More the future horizon is

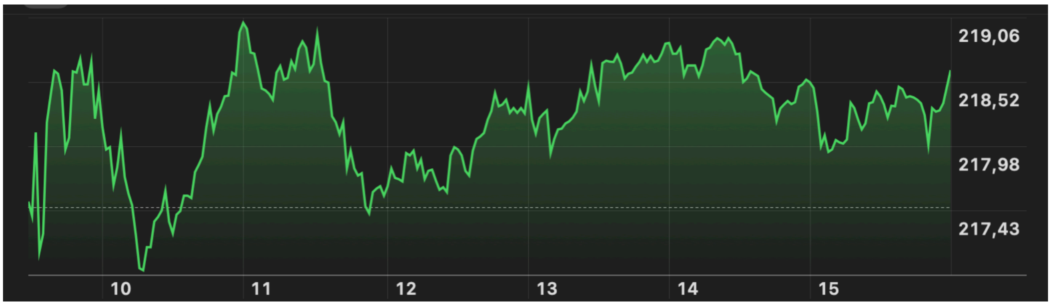

Figure 3.2: NVIDIA price within 1 day, 15 october form 9 am to 4 pm

close, harder will be obtaining good predictions due to noise inside the series and the commission costs will influence more the gain/loss status. So in the experiments is a good design choice does not choose a future horizon window below 15 days, in order to be consistent in term of noise and to make less effective the effect of commission costs inside the final results. Moreover I will considered as minimum interval between element a day, so no intra-day consideration. Finally in order to prove the effect of the noise I will show the different scores I can get using different size of the window in the experimental part.

3.1.3 Characteristic of the Input

Input is characterise by a combination of financial indicators within a certain window. It can be seen as a matrix with n rows and m col (n = number of financial indicators, m = length of the window). In this case the major problem is finding the best combination of indicators and the best length of the window in order to get the best scores. In fact the problem is quite complicate even if with this consideration it is at least reduced.

3.1.4 Model architecture choice

At this point the Model can be seen as a black box, because the Input and the kind of the prediction, the Output are already specified. Considering the

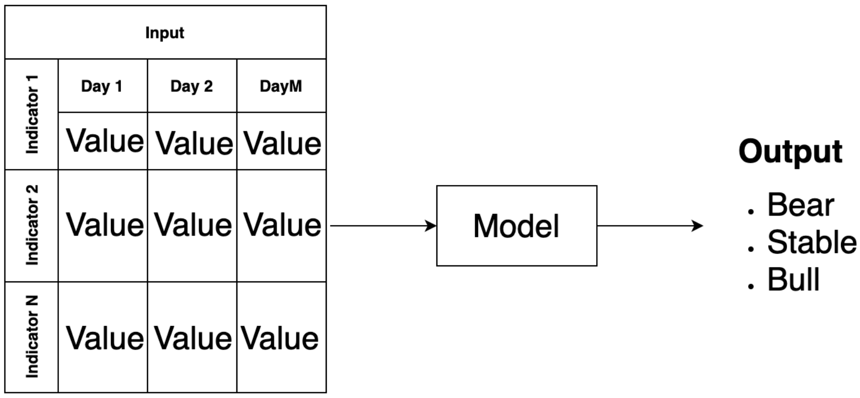

Figure 3.3: Process flow

Input shape and other studies related to applying ML/DL in financial time series prediction, moving the problem in something which can be solved using a Convolution Neural Model seems something which has sense. Although, Convolutional Neural Network models were developed for image classification problems, where the model learns an internal representation of a two-dimensional input, in a process referred to as feature learning. This same process can be harnessed on one-dimensional sequences of data, such as in the case of financial indicators. Indeed the model learns to extract features from sequences of observations and how to map the internal features to different future market states.

3.2 Solution Techniques

Before starting directly on the parts of the solution I will recap some assumptions from problem formulation. First the interval between elements is the day, so no intra-day consideration in order to avoid noise and commission cost problems. This considerations influence the shape of the input as well as all the data which I will use to extract financial indicators which comes from historical daytime series. Second the length of the future horizonis fixed to 30 days which is the best parameter considering the trade off between accuracy and commission costs. While for semplicity I will not change the window size of the input which will be of 30 days.

3.2.1 A Classification problem

I defined 3 classes, Bear, Stable and Bull. Considering % variations of daytime close prices, I split them into positive variations and negative variations, in order to compute the mean of positive variations and the mean of negative variations.

$$j = i - 1$$

$$variation_i = percentage(Close_j - Close_i, Close_i)$$

The formula which describes the distribution of the classes is: Bear if the variation is below negative mean variation, Bull if the variation is over positive mean variation and Stable if the variation stays within positive and negative variations. Finally I encode all the classes into a categorical form using a vector composed by 3 elements, Bear is represented by [1,0,0], Stable is represented by [0,1,0] and Bull is represented by [0,0,1]. Finally a small note about naming, I will use variation_i to describe daily variation while Variation_i is the sum of all variations inside the future horizon window (see next subsection).

$$[1, 0, 0] = Variation_i < MeanNeg$$

$$[0, 1, 0] = MeanNeg <= Variation_i >= MeanPos$$

$$[0, 0, 1] = Variation_i > MeanPos$$

3.2.2 Future time horizons

Choosing the future horizon size, using the variations previously computed for the classification part, allows me to define a Variation as the sum of all variations within the future horizon window. With a future horizon size of 30 days, I sum, considering day i, all the daily variation from i+1 until i+30.

$$Variation_i = \sum_{j=1+i}^{i+30} variation_j$$

After that I will have a set of future Variations, so I can convert them from a real % variations to the categorical form, follow the rules I have already specified before.

3.2.3 Characteristic of the Input

There are two main premises: First the shape of the Input is already defined into the problem formulation. Second there are a huge amount of financial indicators. So here the point is to focus our attention on the indicators most frequently searched by traders and put the emphasis on how much a different combinations of them can increase/decrease the final model score. For sure this is an approximation as there are a huge amount of possible combinations, but on the other hand it demonstrates the flexibility of this approach. Here the set of indicators I used in the experiments:

- **Candle**, it is defined by the combination of Volume and difference between Max and Min price in a specific day. In a graphic plot the shape of this indicator is close to a real candle, this is the reason behind the name.

- **SMAs**, Simple Moving Average. It computes the mean considering a certain window of the closing price for each day inside the window. I will use a combination of 3 SMAs, a short one of 8 days, a medium one of 79 days and a long one of 189 days. The values 8, 79 and 189 can be estimated by a non linear search which can be done using black box optimizer like RBFOPT[9], see future work section.

- **Seasonality**, this indicator is very useful to extract some periodically stock price trends considering the same part of the year, w.r.t. a certain number of years. In order to get the meaning in a simple way, you can think about the price of the wood. In the summer wood price is below the wood price in the winter due to the fact people use it more in the winter w.r.t. the summer. Similarly a company can sell more products in a precise part of the year and its quarterly results can show periodically a better result which is translated by the market usually in an increment of the stock price.

This are the combinations of indicators which I have used in order to investigate the benefit of using more informative input w.r.t. using only the close price. In the experimental part I will use this combinations of indicators:

- Only Close Price.
- Close Price, Seasonality.
- Candle.
- Candle, Seasonality.
- SMAs.
- SMAs, Seasonality.
- Close Price, Candle, SMAs, Seasonality.

In the Experimental part I will show the difference in using one combination or another, while in the Framework I will show you how can be easy for an Expert introduces a new indicator and after that using it as input for the model.

3.2.4 Model

Starting from what I said in the formulation part:

- The Input shape can change, it depends on the number of features (financial indicators) I want to use and the size of the window of historical data I want to considerate. In general it can be described as [samples, timesteps, features].

- The Output shape is a categorical one, so a triplet where the the biggest element represents the prediction class. For example [0,1,0] represents a Stable prediction, so the market will be stable for the next future horizon size.

For what concern the Model architectures it is influenced by Input and Output choices. For this reason as I said in the formulation part I choose a one-dimensional Convolutional approach. A one-dimensional CNN is a CNN model that has a Convolutional hidden layer that operates over a 1D sequence. This is followed by a second Convolutional layer, to deal better with long input sequences, and then a pooling layer whose job it is to distill the output of the Convolutional layer to the most salient elements. The Convolutional and pooling layers are followed by a dense fully connected layer that interprets the features extracted by the Convolutional part of the model. A flatten layer is used between the Convolutional layers and the dense layer to reduce the feature maps to a single one-dimensional vector. The choice about filter and

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=10, activation='relu',
                 input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Conv1D(filters=64, kernel_size=10, activation='relu'))
model.add(Dropout(0.5))
model.add(MaxPooling1D(pool_size=3))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.summary()
model.compile(optimizer='Adam', loss='CategoricalCrossentropy',
              metrics=['accuracy', 'CategoricalCrossentropy'],
              weighted_metrics=['accuracy', 'CategoricalCrossentropy'])
```

Figure 3.4: Model Architecture

kernel size was done after some trials and thinking about the fact that input is always related to 30 days, due to the fact that with a small window I cannot use the same hyperparameters which can work well with a size of a month.

For what concern the activation functions ReLu is what generally is used for 1d Convulutional layers, while in the last layer I choose a Softmax in order to extract the most probable class. Also I choose Adam as optimizator and Categorical Cross Entropy as loss. After that I put an Early Stopping criteria followed by Model Checkpoint mechanism in order to save the best model. Finally I put some attention on the fact that the number of elements for each class is not balanced. Indeed considering on how I classified each element, only few of them can be considered as Bull or Bear element and this is exactly what I want to see in order to get only price movement really interesting.

```
from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint
es = EarlyStopping(monitor='weighted_accuracy', mode='max',
                   verbose=1, patience=500)
mc = ModelCheckpoint('best_model_'+name+'.h5',
                     monitor='weighted_accuracy', mode='max',
                     verbose=1, save_best_only=True)
w_neg,w_neutral,w_pos = define_weight(Y_train)
print(w_neg,w_neutral,w_pos)
model.fit(X_train, Y_train, validation_data=(X_val, Y_val), epochs=10000,
          callbacks=[es, mc],
          class_weight={0: w_neg, 1: w_neutral, 2: w_pos})
```

Figure 3.5: Early Stopping, Model Checkpoint and class weight

Moreover this approach can decrease the commission costs as the Model will underline only few situations. Talking more directly on how to deal with this unbalance problem, I act in two step. First I count the presence percentage for each class inside the training set. Second I use this information in order to penalize the error of a each class in an inversely proportional way w.r.t their presence percentage. In this way I can penalize more the errors related to Bull or Bear classes w.r.t. the Stable one which is the most frequent. I get the penalization factors using the function define_weight which implements what I expressed before. Having the penalization factors TensorFlow allows to put directly this information inside the fit method, defining the class weight parameter. This on one hand can decrease the overall accuracy, on the other hand it increases accuracy related to Bear and Bull classes which are the most important.

Chapter 4

Designing a SW Framework

The SW Framework, the Problem Formulation and the Solution choices are interconnected. The proposed Framework is based on the possibility of integrate the knowledge of an Expert of the field with AI approaches in order to offer a Framework which is able to give useful information about future market states. This part of the work is fundamental for what concern the implementation of Problem Formulation and Solution choices. Moreover it allows to build something that can be more than a simple support for an Expert, but a complete and autonomous solution which can deal with the stock market prediction.

4.1 From an Expert to the Decomposition of the Problem

In this part there is what makes my thesis different than other works. Indeed, before trying to do something new it is important understanding what nowadays works. As I am not an Expert trader I did not have a clear vision on how an Expert acts in order to solve the problem. Thanks to the collaboration with Aivolution S.R.L. I had the opportunity to talk with this kind of Experts and extract their best practises in something which can be codified in a complete and flexible SW. Looking to figure 4.1 it is possible see graphically the rea-

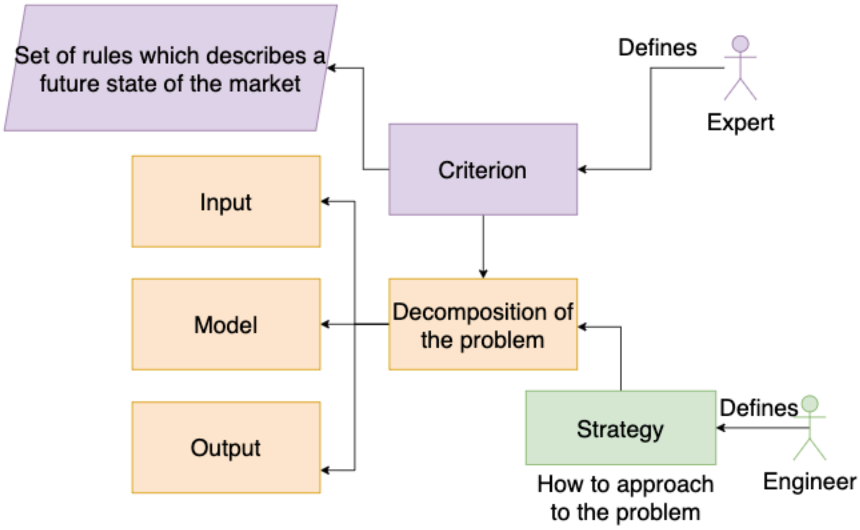

Figure 4.1: From Expert to SW Engineering

soning behind the choices, now I will express it in words. The objective is coding the behavior of an Expert of the field. What I can do as an Engineer is made a Strategy, so a way in order to approach the problem. Starting from a set of rules, or in general a set of best practise, which, for the Expert, are able to descript a future state of the market, a good Strategy can be Divide et Impera, so the decomposition of the problem into three big components: Input, Model and Output which I have described in the previous chapter in detail. A brief recap of them:

- **Input** is related to the choice of window size of the input and the acquisition of the information (financial indicators).
- **Output** is related to the size of future horizon and the kind of prediction he needs, a classification or a regression.
- The **Model** part is something which can be hided to the Expert but it is influenced by Input and Output. Moreover it has to respect the Expert's indications inside the Criterion.

The advantages of this Strategy are:

- The Flexibility, the fact that with the same structure it is possible to deal with problems from different fields not only financial ones. Moreover with the same structure it is possible implementing different Criterions.
- The division in components allows to split the work into different parts which can be implemented in parallel and by different teams.

4.2 SW Framework Components

In this part I will explain in more details the three components. As I said before this structure can be used for different applications which have the aim of exploit the potential of ML/DL approaches. Moreover I will not go into the details about Problem Formulation and Solution choices because I have shown them in the Chapter 3.

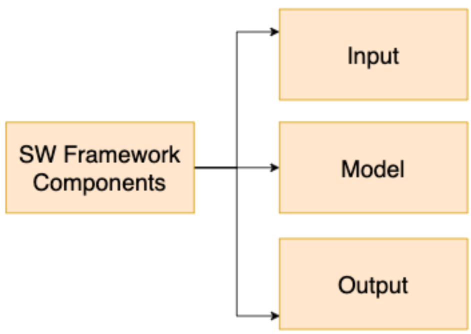

Figure 4.2: SW Framework Components

4.2.1 Output

Starting from the end of the flow I extract from the Criterion the orientation of the problem as the output is the only thing matter at the end. More specifically it means that all other parts are influenced by what kind of output is required.

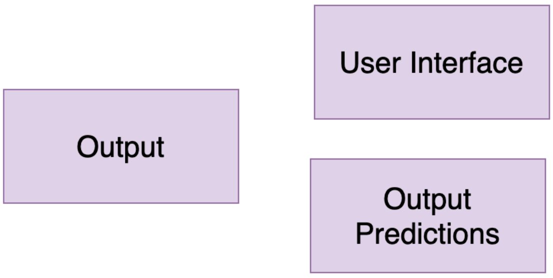

Figure 4.3: Output Components

The Expert can be interested in a classification or in a regression for example. So in this phase, first I have to understand the objective and what is the best Output in order to represent it. It can be a set of class or a real number, it depends on the case. Second I have to represent the prediction in order that it can be easily understandable, so I have to make transparent what is doing behind the scenes. In my application I choose to represent the market states using classes and after that I give to the Expert the output of the prediction in form of Bull, Stable or Bear market state which are term easily understandable by them. In conclusion this part deals with the visualization of the output giving some requirements to the Model and Input parts.

4.2.2 Input

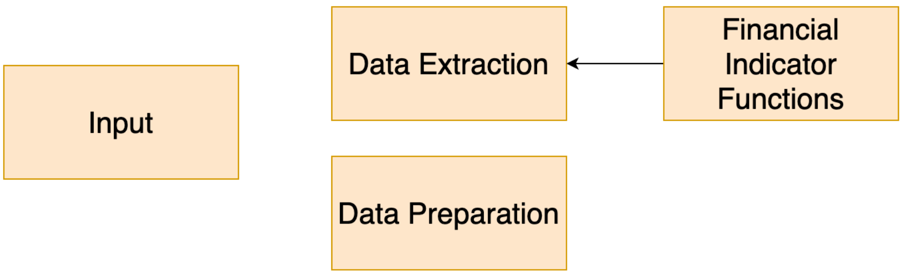

Figure 4.4: Input Components

The Input components has to decode the part of the Criterion related to what are the crucial information to make the prevision. It can be decomposed in two sub parts:

- **Data acquisition**, where it extracts the information which are useful for the prediction. It is done following the indication inside the Criterion, which in this specific case gives information about the name of the stock or index of interest, the temporal window where the model will be trained and the name of financial indicators which are important for the predictions. The output of this sub component will be a DataFrame which contains all the information ready to be pre-processed in order to be the input of the Model.

- **Data Preparation**, in the Criterion are specified the window size of the Input and the percentage of train validation and test set. In this part data where splitted and prepared in order to be in the corrected format w.r.t. the model. In the example of my application the Input shape can be described as [samples, timesteps, features]. For more details see Chapter 3.

For what concern the implementation, I have created a specific function in order to extract the financial indicators giving the Index of the stock and the name of the financial indicators. It uses some predefined functions in order to make the indicators, so the Expert can choose between a fixed set of indicators. Anyway adding a new indicator is very easy as it is a matter of defined a simple function in order to extract it. While for what concern historical data, I used the Yahoo Finance API which returns the DataFrame which I used for the creation of the indicators. After that, the Data Preparation part is done by another function which is linked to the model and also to the output part as it will make the ”X” and the ”Y” which will be used for Training, Validation and Test phases. In conclusion this part deals with data extraction and data preparation, so on one hand it is independent for what concern data extraction, on the other hand it is dependent in part on the model for what concern the data preparation part.

4.2.3 Model

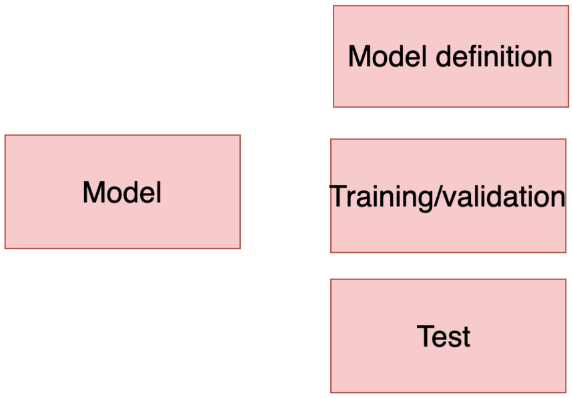

Figure 4.5: Model Components

In this part my objective is making the best model approach in order to solve the problem with the Input and the Output I have already defined. Of course I can change a little some specifications as for example in the Input part I have data preparation in order to set up the Input for the model same for Output where I can use different coding in order to defined it. The best approach is finding a compromise considering the kind of Input and the Output which is required. In my specific case I use a 1D CNN because the input shape was suitable for this kind of approach, similar considering the output I choose a categorical representation using a Cross-Entropy loss. For what concern the implementation I have defined a function which is able starting from a defined ”X” and ”Y” to start the Training, Validation and Test phase. After that it saves the best model and the user can call it in order to make prediction when it needs it (see SW Framework Flow). Finally it will give the answer in an understandable form, giving also the accuracy of the model on the test set. In this way the user can considered how much this model can be reliable. In conclusion in this component there are the heart of the project.

4.3 SW Framework flow

In this section I want to specify how the Framework works and how the components interact each others.

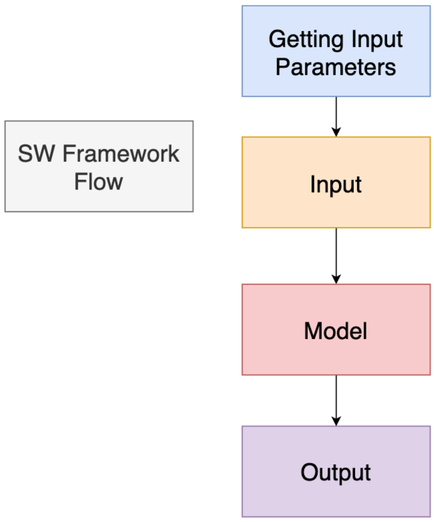

Figure 4.6: SW Framework Flow

4.3.1 Getting Input Parameters

After the design part, the Expert can choose between creating a new model or using one already created in order to get a new prediction. In case of a creation he will be able to select, name of the stock, time future horizon length, window size, name of financial indicators and a period where the model will be trained and tested. Otherwise it has only to select the name of the model. I will show the flow considering the two possibilities.

4.3.2 Input

- Creation phase, It receives the name of the stock, time future horizon length, window size, name of financial indicators and a period where the model will be trained and tested. In the data preparation part it loads the historical data and computes the values for the financial indicators using the support function. After that It splits and prepares the data for the training validation and test phase. Finally it saves all the Input parameters in order to get it in the prediction phase.

- Prediction phase, having the name of the model it gets the Input parameters, it has to make the ”X” considering the already defined window size, the financial indicators which have to be considered and as a period the last window starting from the day of the request (or the last day where the market was open).

4.3.3 Model

- Creation phase, it takes the output of the data preparation functions of the Input component. It starts the training validation phase and finally it saves the best model weights. After that it computes the test and saves the scores in order to be showed by the Output part.

- Prediction phase, it loads the weights and run the prediction using data from the Input component, after that it communicate the result to the Output component.

4.3.4 Output

- Creation phase, it shows the test score at the end of the creation of the Model.

- Prediction phase, it shows the test score to remind the Expert how reliable this model can be and the actual prediction of the actual future market state.

Chapter 5

Experimental results

In this section I will show the experimental results over a list of stocks and an index. These experiments has to two aims. First I want to show how much a different combination of financial indicators can impact on the final results. Second I want to demonstrate that using the correct combination of Input and the length of the future horizon it is possible to achieve good scores in term of Accuracy and Confusion Matrix without an expensive DL approach. In addition to this, I want to add that my solution does not reflect into a complete trading system because the strategy depends too much on external factors like commission costs, place where you trade ect... So if I wanted to show a graph of potential profit following this strategy I should have studied a lot of complex aspects which can change every day and which are not standard. For this reason it is a complete and autonomous system for supporting the trader which has a complete view of his financial situation and trading costs. In conclusion I will not show any potential profit graph because I think that they can not be proved in the real word even if I tried to be more consistent as possible.

5.1 Experimental Setting

The experimental part was conducted using the same the same window size and the same future horizon length. This was done because changing one of

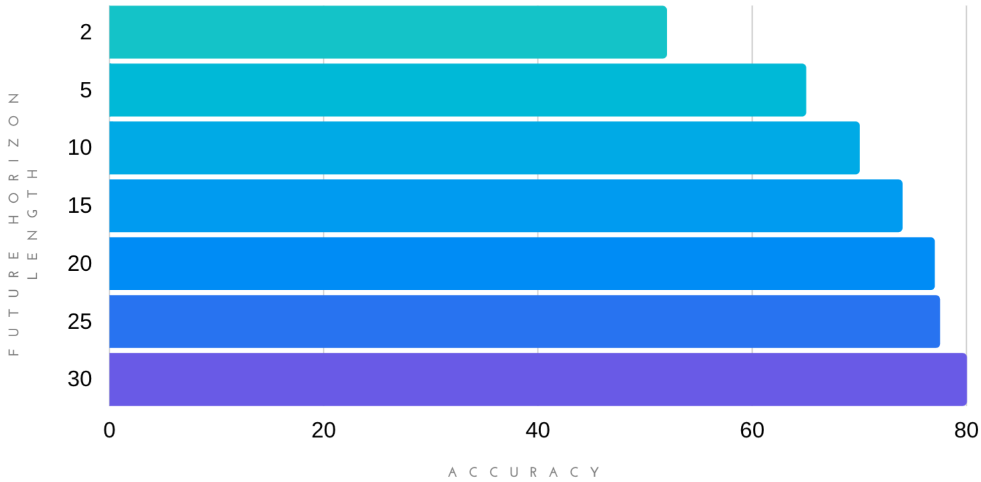

Figure 5.1: length of the future horizon changes w.r.t. accuracy rate

this parameters it will change the type of the problem. Indeed in figure 5.1 there is an example of this problem, considering my Model, Close as input, Apple as stock and Input window equal to 30 days. So 30 days as length of the future horizon seems reasonable, in future can be tested more combinations of them. Finally the combination of financial indicators are:

- Close.
- Close-Sesonality.
- SMA2-1-SMA3-2-SMA3-1.
- SMA2-1-SMA3-2SMA3-1, Sesonality.
- High-Low percentage distance, Volume (Candle Indicator).
- High-Low percentage distance, Volume, Sesonality.

5.1.1 Model and Baselines

For what concern the model I have explained it in detail in chapter 3. As baselines I will use a Persistent technique and another one which will predict always the most frequent class, so the stable one.

5.1.2 Walks' Definition

Considering the fact that I use a period 10 years, a good strategy in order to test a model considering time series is the concept of Walks. Starting from a year date, in my case 2000, I use the period between 2000-2010 and this for me will be a walk. After that I will shift the period to two years later, so the walk will be between 2002-2012. I do the same process until 2020. So in order to recap the walks' period I will consider:

- From 1 Gen 2000 to 31 Dec 2010.
- From 1 Gen 2002 to 31 Dec 2012.
- From 1 Gen 2004 to 31 Dec 2014.
- From 1 Gen 2006 to 31 Dec 2016.
- From 1 Gen 2008 to 31 Dec 2018.
- From 1 Gen 2010 to 31 Dec 2020.

5.1.3 Evaluation metrics

For what concern the evaluation as I said before I will not show the plot of potential profit as it can be not verified in real case scenario. I will focus on the accuracy of the Model, and the type of the errors showing the confusion matrix.The type of the errors is important because accuracy it is not sufficient to determine a good trading support system. Indeed I want to make sure that the Model does not misclassified a Bull state with a Bear one or the opposite. Moreover if it predicts always stable market it will be useless. So looking at the confusion matrix I can show you the potential of this model.

5.1.4 List of Stocks and Indexes

I have chosen a list of stock and an Index which can represent different kind of market situation. The majority of them are tech companies and it is done in order to have a similar market sentiment. Indeed the competition in this field is high and it makes an investment in this area a good choice only if you are able to manage the risk. So I choose companies with a brilliant future and others with a brilliant past in order to evaluate if the Model is able to suggest the correct actions in order to increment the value of the portfolio. Moreover I choose also other companies from other fields in order to have some variability in the experiments. The stocks are:

- Apple AAPL.
- Microsoft MSFT.
- Carnival CCL.
- JPMorgan JPM.
- Nokia NOK.
- IBM IBM.
- Intel INTC
- Amazon AMZN

Finally I have considered also the SP 500 as it is one of the index that better represents the state of the US market.

5.2 Results and Discussion

5.2.1 Apple AAPL

Before starting the analysis of the scores, I express few words about the historical price graph as in different walks the model will have to deal with different situations. The period between 2000 and 2010 was quite stable, even if in 2000 and 2008 there were two big economic events which have influenced the growth of the price. After 2010 Iphone and Ipad had an huge impact in business and Apple became one of the most important tech company. Finally this company has affected by seasonality as its revenue is influenced by the selling of the products.

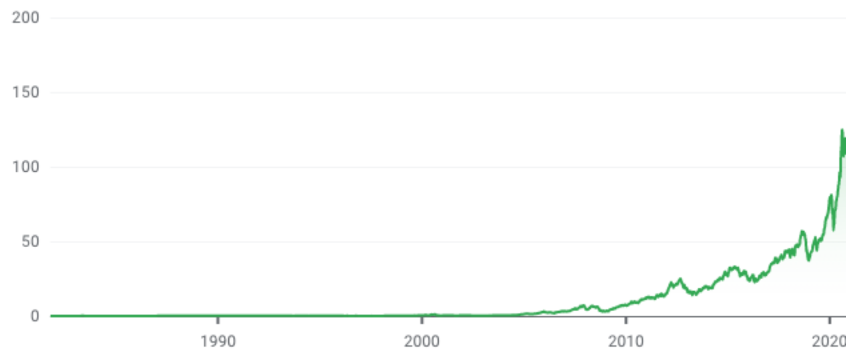

Figure 5.2: AAPL historical price graph

Accuracy Table

As I said previously, Apple is afflicted by seasonality and it can be seen from the table regarding the different scores in the rows were the seasonality is included. What I want to highlight is that the price close parameter which is used in the majority of the papers is the less effective kind of input. For what concern the other scores they are quite close. While considering other two

AAPL accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	59%	89%	89%	89%	68%	87%	80%
Close-Sesonality	80%	89%	86%	88%	86%	87%	86%
SMA2-1-SMA3-2-SMA3-1	86%	90%	91%	87%	88%	87%	88%
SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	90%	89%	91%	88%	87%	89%
High-Low_perc_distance-Volume	85%	90%	88%	90%	88%	87%	88%
High-Low_perc_distance-Volume-Sesonality	86%	88%	90%	90%	88%	87%	88%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	91%	87%	89%	88%	87%	88%

Table 5.1: AAPL Model Accuracy Table

baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

AAPL accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	41%	47%	46%	48%	48%	45%	46%
Most Frequent class	58%	61%	60%	62%	58%	59%	60%
Model with SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	90%	89%	91%	88%	87%	89%

Table 5.2: AAPL Models Accuracy comparison Table

Confusion Matrix

In this subsection I want to demonstrate that the model not only is able to have a good accuracy, but also that is able to make only manageable errors. The idea is: it can make mistake but it is not permitted that it classifies a Bull market as a Bear one and the opposite case. Considering the Best Input I will show how much it is accurate. The matrix is computed summed all the Confusion Matrices of each walk.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	318	48	1
Stable	75	1637	134
Bull	1	86	729

Table 5.3: AAPL Model Confusion Matrix

5.2.2 Microsoft MSFT

Before starting the analysis of the scores, I express few words about the historical price graph as in different walks the model will have to deal with different situations. The period between 2000 and 2010 was quite stable with only a peak due to tech bubble. Microsoft is less affect by seasonality due to the fact they offer services. After 2015 with the introduction of cloud services the price has increased a lot. The direction of the curve is less influenced by seasonality with respect to Apple.

Figure 5.3: MSFT historical price graph

Accuracy Table

As I said previously, Microsoft is less affected by seasonality and it can be seen from the table regarding the different scores in the rows were the seasonality is included. What I want to highlight is that the price close parameter works better because it follows a simple trend. While considering other two

MSFT accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	85%	86%	86%	86%	85%	86%	86%
Close-Sesonality	88%	86%	85%	86%	83%	89%	86%
SMA2-1-SMA3-2-SMA3-1	88%	81%	83%	85%	84%	84%	84%
SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	85%	86%	88%	87%	86%	87%
High-Low_perc_distance-Volume	85%	80%	83%	88%	83%	87%	84%
High-Low_perc_distance-Volume-Sesonality	88%	86%	86%	82%	83%	87%	85%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	85%	86%	83%	85%	85%	85%

Table 5.4: MSFT Model Accuracy Table

baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

MSFT accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	42%	44%	47%	38%	42%	44%	43%
Most Frequent class	61%	59%	61%	55%	57%	57%	58%
Model with SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	85%	86%	88%	87%	86%	87%

Table 5.5: MSFT Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	449	66	1
Stable	93	1567	134
Bull	3	123	588

Table 5.6: MSFT Model Confusion Matrix

5.2.3 Carnival CCL

Carnival is a completely different company w.r.t. Apple and Microsoft, indeed it is related to cruise trips. This sector is more sensitive to economic crisis as its services are considered not essential, so with a small amount of money, people do not buy a cruise trip. On the other hand it remains a valuable stock when the economic period is good. This can explain the historical plot until last year where with the world in a pandemic situation the impact on this sector was shooking.

Figure 5.4: CCL historical price graph

Accuracy Table

Carnival is affected by seasonality but less than Apple. Indeed people can go in any period of the year considering the fact that the cruise trip can be done in different parts of the world. Even if this periods are related to longest holiday periods. It can be seen from the table regarding the different scores in the rows were the seasonality is included. Also in this case the Close parameter is the less effective. While considering other two baselines, the margin considering

CCL accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	72%	63%	89%	87%	88%	83%	80%
Close-Sesonality	81%	88%	88%	85%	85%	81%	85%
SMA2-1-SMA3-2-SMA3-1	89%	85%	86%	85%	84%	86%	86%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	87%	88%	87%	84%	90%	87%
High-Low_perc_distance-Volume	87%	87%	89%	87%	84%	87%	87%
High-Low_perc_distance-Volume-Sesonality	85%	86%	88%	89%	86%	85%	87%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	90%	87%	87%	89%	84%	87%	87%

Table 5.7: CCL Model Accuracy Table

accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

CCL accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	47%	45%	43%	45%	45%	52%	46%
Most Frequent class	62%	64%	60%	61%	58%	67%	62%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	87%	88%	87%	84%	90%	87%

Table 5.8: CCL Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	361	67	1
Stable	78	1758	134
Bull	1	106	546

Table 5.9: CCL Model Confusion Matrix

5.2.4 JPMorgan JPM

JPMorgan is a bank, so it is involved in a completely different field w.r.t. previous cases. Indeed the historical trend allows us to see how much are ”stable” as business instead of huge Bear or Bull situation, which can be happened but are always related to a particular situation.

Figure 5.5: JPM historical price graph

Accuracy Table

In this case it very hard understand the trend has it is not so clear. Indeed the best combination was the one with more details. While considering other two

JPM accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	88%	86%	88%	86%	88%	88%	87%
Close-Sesonality	87%	86%	86%	86%	87%	88%	87%
SMA2-1-SMA3-2-SMA3-1	86%	86%	87%	88%	88%	88%	87%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	85%	84%	84%	89%	90%	87%
High-Low_perc_distance-Volume	89%	84%	85%	85%	86%	87%	86%
High-Low_perc_distance-Volume-Sesonality	89%	88%	88%	85%	87%	87%	87%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	91%	86%	87%	87%	88%	90%	88%

Table 5.10: JPM Model Accuracy Table

baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

JPM accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	48%	50%	48%	42%	59%	47%	49%
Most Frequent class	68%	65%	63%	59%	67%	61%	64%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	91%	86%	87%	87%	88%	90%	88%

Table 5.11: JPM Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	361	67	1
Stable	78	1758	134
Bull	1	106	546

Table 5.12: JPM Model Confusion Matrix

5.2.5 Nokia NOK

Nokia is a company based on products like Apple. Until the launch of the Iphone it was one of the bigger player into the cellular market which was reflected into the price of 2000/2001 and 2008. After that the plot is quite stable.

Accuracy Table

As I said previously, Nokia is afflicted by seasonality and it can be seen from the table regarding the different scores in the rows were the seasonality is

Figure 5.6: NOK historical price graph

included. What I want to highlight, as it happened for other examples, is that the price close parameter which is used in the majority of the papers is the less effective kind of input. For what concern the other scores they are quite close. While considering other two baselines, the margin considering accuracy is

NOK	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	80%	83%	90%	90%	85%	83%	85%
Close-Sesonality	90%	85%	88%	88%	88%	87%	88%
SMA2-1-SMA3-2-SMA3-1	89%	87%	89%	86%	88%	90%	88%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	88%	90%	90%	88%	90%	89%
High-Low_perc_distance-Volume	88%	89%	89%	90%	88%	89%	89%
High-Low_perc_distance-Volume-Sesonality	91%	90%	88%	89%	89%	92%	90%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	89%	90%	88%	91%	90%	90%

Table 5.13: NOK Model Accuracy Table

quite interesting. Moreover I want to confirm this looking at the confusion matrix.

NOK accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	42%	43%	42%	46%	44%	46%	44%
Most Frequent class	57%	58%	58%	62%	62%	61%	60%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	89%	90%	88%	91%	90%	90%

Table 5.14: NOK Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	539	77	0
Stable	67	1671	93
Bull	0	82	500

Table 5.15: NOK Model Confusion Matrix

5.2.6 IBM

IBM is a company more focus on the services instead of products. Looking on the historical graph price, we can see an huge peak due to tech bubble, after that the price was quite stable in the sense that the trend was nor a clear Bull nor a clear Bear. Finally with the pandemic situation, tech field has raised the scene so also IBM has reached a price near 60$ stock.

Figure 5.7: IBM historical price graph

Accuracy Table

IBM is not so affected by seasonality, indeed using it does not allows to improve the scores. Moreover the trend is quite complicate so the model need all the information I can feed in. Indeed the best input was the most complete one. While considering other two baselines, the margin considering accuracy

IBM accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	
Close	87%	88%	86%	87%	86%	83%	86%
Close-Sesonality	85%	87%	85%	88%	85%	81%	85%
SMA2-1-SMA3-2-SMA3-1	90%	87%	86%	85%	85%	85%	86%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	88%	86%	86%	83%	86%	86%
High-Low_perc_distance-Volume	88%	78%	83%	83%	83%	87%	84%
High-Low_perc_distance-Volume-Sesonality	89%	88%	88%	84%	84%	84%	86%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	86%	87%	86%	86%	85%	87%

Table 5.16: IBM Model Accuracy Table

is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

IBM accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	49%	50%	42%	45%	41%	46%	46%
Most Frequent class	63%	61%	62%	60%	60%	62%	61%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	86%	87%	86%	86%	85%	87%

Table 5.17: IBM Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	305	81	0
Stable	56	1346	106
Bull	0	105	526

Table 5.18: IBM Model Confusion Matrix

5.2.7 Intel INTC

Intel business is based on HW products. At the beginning of the 2000s it was one of the big player for what concern HW components production. It is affect to seasonality as it is dependent on the selling of the components. The big peak is due to the tech bubble, while the rest of the graph is quite stable. Finally the pandemic situation allowed it to grow fast, until it has to face with a new competitor on the field, Apple Silicon, and the chip crises.

Accuracy Table

As I said previously, Intel is affected by Seasonality. Indeed you can see the effect of it on the accuracy table. Again in this situation seems that close price indicator is not enough to reach the best result. While considering the other two baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

Figure 5.8: INTC historical price graph

INTC accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	
Close	89%	80%	87%	88%	87%	82%	86%
Close-Sesonality	87%	85%	88%	81%	86%	84%	85%
SMA2-1-SMA3-2-SMA3-1	86%	88%	89%	86%	86%	84%	87%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	90%	87%	86%	88%	87%	88%
High-Low_perc_distance-Volume	85%	86%	88%	84%	83%	83%	85%
High-Low_perc_distance-Volume-Sesonality	88%	88%	87%	85%	85%	88%	87%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	86%	89%	86%	86%	84%	83%	86%

Table 5.19: INTC Model Accuracy Table

INTC accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	44%	44%	49%	48%	45%	46%	46%
Most Frequent class	61%	60%	64%	62%	61%	63%	62%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	90%	87%	86%	88%	87%	88%

Table 5.20: INTC Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	466	66	0
Stable	88	1609	129
Bull	1	99	571

Table 5.21: INTC Model Confusion Matrix

5.2.8 Amazon AMZN

Amazon price is under a continuous growth until it was born. It is related to seasonality even if it principally sells services for an indirect consequence. Moreover it is in a monopoly situation w.r.t. online retail market and maybe in the future governments will deal with it in order to allow more competition in this field.

Figure 5.9: AMZN historical price graph

Accuracy Table

As I said previously, Amazon is afflicted by seasonality and it can be seen from the table regarding the different scores in the rows were the seasonality is included. What I want to highlight, is that not always add information is useful. Indeed in this case the input which is composed by all the indicators does not perform well. In fact it is always important finding the correct combination of financial indicators in order to find the correct compromise otherwise it will generate only noise. While considering other two baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

MSFT accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Close	85%	86%	86%	86%	85%	86%	86%
Close-Sesonality	88%	86%	85%	86%	83%	89%	86%
SMA2-1-SMA3-2-SMA3-1	88%	81%	83%	85%	84%	84%	84%
SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	85%	86%	88%	87%	86%	87%
High-Low_perc_distance-Volume	85%	80%	83%	88%	83%	87%	84%
High-Low_perc_distance-Volume-Sesonality	88%	86%	86%	82%	83%	87%	85%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	87%	85%	86%	83%	85%	85%	85%

Table 5.22: AMZN Model Accuracy Table

AMZN accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	43%	41%	47%	47%	45%	45%	45%
Most Frequent class	58%	55%	60%	59%	60%	60%	59%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	86%	86%	90%	88%	85%	88%	87%

Table 5.23: AMZN Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	389	76	1
Stable	70	1623	130
Bull	0	121	619

Table 5.24: AMZN Model Confusion Matrix

5.2.9 SP500 Index

SP500 is an Index, it is composed by first 500th most capitalized companies in US. The historical graph shows how much it was reliable in the past. Indeed it allows to follow the direction of the market with less risk w.r.t. choosing only a single company.

Figure 5.10: SP500 historical price graph

Accuracy Table

In this case as the trend is difficult, a long Bull period follows to another long Bear. As in other cases I have analyzed the only way to improve the accuracy it is the acquisition of as much useful information as possible. Indeed in this case using all the indicators allows me to acquire the best result. While

S&P accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	
Close	88%	86%	88%	88%	84%	80%	86%
Close-Sesonality	87%	89%	87%	87%	86%	83%	87%
SMA2-1-SMA3-2-SMA3-1	88%	85%	88%	86%	88%	87%	87%
SMA2-1-SMA3-2-SMA3-1-Sesonality	88%	87%	87%	88%	86%	86%	87%
High-Low_perc_distance-Volume	88%	85%	89%	86%	83%	87%	86%
High-Low_perc_distance-Volume-Sesonality	87%	89%	84%	88%	86%	88%	87%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	87%	88%	89%	84%	90%	88%

Table 5.25: SP500 Model Accuracy Table

considering other two baselines, the margin considering accuracy is quite interesting. Moreover I want to confirm this looking at the confusion matrix.

S&P500 accuracy	2000-2010	2002-2012	2004-2014	2006-2016	2008-2018	2010-2020	mean
Persistence	43%	47%	45%	43%	45%	47%	45%
Most Frequent class	63%	60%	62%	59%	60%	61%	61%
Close-High-Low_perc_distance-Volume-SMA2-1-SMA3-2-SMA3-1-Sesonality	89%	87%	88%	89%	84%	90%	88%

Table 5.26: SP500 Models Accuracy comparison Table

Confusion Matrix

Here I make again the confusion matrix and again the score shows that the Model is able to identify with a good accuracy the different classes.

Confusion Matrix Best Model configuration	Bear	Stable	Bull
Bear	330	57	1
Stable	49	1651	139
Bull	0	127	675

Table 5.27: SP500 Model Confusion Matrix

Chapter 6

Conclusion

6.1 Conclusions

Considering the Experimental results I can say that:

- The classes are not balanced, this can be seen on the performance of the baseline which predicts always stable market. The model has to deal with it in order to perform well.
- The model even if it is simpler w.r.t. other proposal can achieve a good accuracy, which is confirmed also looking into the confusion matrix. So it can deal with the fact that the classes are not balanced.
- The input window size and the length of the future horizon parameters are able to change the complexity of the problem, so it has to be investigate in an accurate way in future contributions.
- Close price is not a great technical parameter considering the result it has achieved in different situations, while looking for a more complex combination of financial indicators can not only improve the accuracy but also the kind of errors. Indeed in all the experiment results, using a more complex combination of financial indicators, the number of miscalssification between Bull and Bear market are very close to 0.

- SMAs, Candle and Seasonality are indicators which allow the Model to reach good scores. They have to be considered instead of taking only the close price.

6.2 Contributions to AI in Finance research

The major contributions of my work in the AI in finance research is:

- A SW Framework which contains in the design phase the presence of an Expert of the field. This allows to combine ML/DL techniques to best practise in the sector. Moreover the application will be easier to be used and also it allows to use simpler techniques in order to obtain good scores.
- The study on the impact of the Input on the results allows me to say that using a more informative resources it is possible to reach better results. It can be implemented in an easier way without a huge change in the current state of the art.
- My work highlight the importance of taking into account the complexity between the choice of the input window size and the length of the future horizon parameters. Often researchers take more attention on the input window size instead of the length of the future horizon which can change completely the complexity of the problem. Moreover the majority of the studies considering a 1 day length which in my opinion is too short to make an accurate prediction and also the result will be affected to commission costs.
- The choice of focusing on a classification approach, considering only important movements of the price and a future window quite large is something which can be useful for other studies in order to build application which can work out of the high frequency trading field.

6.3 Future research

The Major future research points are:

- Applying non linear solver techniques in order to find the correct values for input window size and the length of the future horizon parameters. I am trying to use RBFOPT library in order to find the best combination of them. The problem now is that this search requires an huge amount of resources and time to be completed. The same approach can be done to tune the financial indicators, I already tested it in tuning SMAs lengths.
- Improving the Model, starting from adding an ensemble technique which can use different combinations of input window sizes and the future horizon lengths. Moreover adding NLP techniques related to Sentimental Analysis and a technical analysis of the intrinsic value of the stock can make a quite effective revolution in this field.
- Creating a complete autonomous trading system, adding a portfolio view which allows the system to know the current situation of the trader, the related costs and the possible correlations inside it. It will use the Criterion mechanism to extract information for each stock, while the portfolio view can be done using a reinforcement learning techniques which can learn on how to manage all the information. Finally there are already present APIs to connect the system to a professional Broker like Interactive Brokers.

Bibliography

[1] A. Agarwal. Sentiment analysis of financial news. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 312–315, 2020. DOI: 10.1109/CICN49253.2020.9242579.

[2] Aivolution. Aivolution srl. URL: http://www.aivolutionsrl.com (visited on 10/10/2021).

[3] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3):683–692, 2020. DOI: 10.1109/JAS.2020.1003132.

[4] S. Chen. Correlation analysis of financial indicators and stock price fluctuations based on artificial intelligence system. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 43–46, 2021. DOI: 10.1109/ICAIS50930.2021.9395944.

[5] B. Ghibellini Di Felice. Intelligence at the iot edge: activity recognition with low-power microcontrollers and convolutional neural networks, 2022.

[6] interactivebrokers. Ib api python. URL: https://www.interactivebrokers.eu/it/index.php?f=5041 (visited on 10/10/2021).

[7] Investopedia. The stock market. URL: https://www.investopedia.com/terms/s/stockmarket.asp (visited on 10/10/2021).

[8] G. Li, M. Xiao, and Y. Guo. Application of deep learning in stock market valuation index forecasting. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pages 551–554, 2019. DOI: 10.1109/ICSESS47205.2019.9040833.

[9] G. Nannicini. Rbfopt a non linear optimizer. URL: https://github.com/coin-or/rbfopt (visited on 10/10/2021).

[10] M. Nils. Intorduction to 1d ccn. URL: https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf (visited on 10/10/2021).

[11] M. Sumit Saha. Intorduction to ccn. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (visited on 10/10/2021).

[12] UsGov. What are stocks ? URL: https://www.investor.gov/introduction-investing/investing-basics/investment-products/stocks (visited on 10/10/2021).

Acknowledgements

E' stato un lungo percorso, sapevo quello che avevo fatto il giorno prima ma non quello che avrei scoperto il giorno dopo. Ringrazio il prof Lombardi per aver creduto in quello che volevo portare avanti senza mai frenarmi, lasciandomi esplorare questo mondo ignoto e i miei fedeli compagni Arcangelo, Gianluca e Matteo per avermi accompagnato nei vari progetti ed esami. Grazie anche a tutto il team di Aivolution, sono cresciuto tanto in questi mesi. Ho commesso un sacco di errori imparando un sacco di aspetti nuovi che mi hanno reso una persona diversa, non solo a livello tecnico ma soprattutto a livello umano. Non è stato semplice, ma se dicessi che è solo merito mio mentirei. Partendo dai miei genitori senza i quali: primo non sarei nato, secondo non saprei stare al mondo. In modi differenti sono i miei due grandi punti di riferimento quando le cose si fanno difficili, oltre a Paolo e Barbara per la loro infinita disponibilità. Secondo, i nonni e gli zii per i loro appuntamenti fissi dove mi fanno sentire sempre a casa. Vi ricordo di non invecchiare troppo in fretta, ho ancora bisogno di voi! Terzo, Piero per le lunghe chiacchierate e per i tanti consigli. Quarto, Nicole che mi sopporta da quando è nata. Quinto per finire in bellezza lato famiglia, i Tati, che con i loro scherzi e la loro allegria mi fanno sempre sfuggire un sorriso anche quando stressato! Un capitolo a parte gli amici che ci sono sempre nonostante tutto, per le cene, le vacanze ed ogni altro momento che mi ha riempito la vita di altro oltre che questi maledetti libri:) Ognuno di voi mi ha aiutato a suo modo a raggiungere l'obbiettivo. Non faccio nomi ognuno sa quanto è stato importante dentro il suo cuore! Finisco con un sospiro pensando che me la dovrò godere un po',

conoscendomi sapete che sarà molto difficile, ci proverò. Grazie di cuore a tutti!