POLITECNICO DI MILANO

Facoltà di Ingegneria

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e Bioingegneria

Master of Science in

Computer Science and Engineering

A supervised learning approach to swaption calibration

Supervisor:

PROFESSOR MARCELLO RESTELLI

Assistant Supervisor:

DOTTOR MATTEO PIROTTA

Master Graduation Thesis by:

LEONARDO CELLA
Student Id n. 838074

Academic Year 2015-2016

A chi c'è sempre stato, mi ha supportato e sopportato
Para quem sempre esteve, me apoiou e suportou

ACKNOWLEDGMENTS

Aiuta gli altri, e gli altri aiuteranno te.
L.Cella,A.Celli,S.Cereda.(RS)

Finalmente, ma sopratutto inaspettatamente, sono arrivato a descrivere l'ultimo passo di questa fantastica e indimenticabile carriera accademica.

Passando ora ai ringraziamenti veri e propri, vorrei iniziare ringraziando il Professor Restelli per avermi concesso l'onore di svolgere un progetto al suo fianco. Sono stati mesi di crescita personale oltre che professionale. Non bastano due righe per un ringraziamento degno di nota. Mi limito a sottolineare che mi ritengo fortunato e onorato ad essere stato scelto per tale progetto. Volevo ringraziare poi il Dottor Pirotta. Dal primo giorno l'ho ammirato per la sua leadership, le sue competenze e il suo essere sempre presente. Penso che la cosa migliore che possa augurarmi sia affermare il desiderio di diventare, un giorno.., competente quanto lui. Infine hanno contribuito in questo progetto Edoardo Vittori, Giorgio Facchinetti e Roberto Da Luiso. Persone di estrema disponibilità e competenza che hanno alimentato la mia passione nel mondo finanziario.

Ci sono poi dei ringraziamenti dovuti al supporto morale e affettivo che spesso si è rivelato più importante di quello professionale, ringrazio quindi la mia famiglia e in particolare i miei genitori Valerio ed Elena. Senza il loro totale appoggio e i loro insegnamenti, non sarei sicuramente qua.

Un grazie va poi ai miei affetti, che ci sono sempre stati. Tra i vari ricordo Daniele, Davide, Enrico, Joel, Martina ci sarebbero altri che però non riesco a mettere in questo elenco, ma di cui non mi dimentico.

Vorrei concludere ringraziando di cuore i compagni di questa avventura, presenti in esperienze indimenticabili e in momenti irripetibili, ringrazio quindi Alessandro, Stefano, Andrea, Valerio, Marco e Francesco. Grazie perché senza il vostro supporto, non sarei mai riuscito ad arrivare dove sono arrivato e con questi risultati.

Un ultimo ringraziamento va ad Andrea Gervasoni, un professore senza tale titolo, ma che mi ha fatto compiere il primo passo in questo mondo facendomi scoprire le mie potenzialità e passioni.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

MPT	Modern Portfolio Theory
HFT	High Frequency Trading
GFC	Global Financial Crisis
IRS	Interest Rate Swap
OOP	Object Oriented Programming
ANN	Artificial Neural Networks
FWD	Forward Rate Curve
DSCT	Discounting Rate Curve
EUR	Euribor Curve
EUR6M	Euribor curve with tenor six-months
OIS	OIS discounting curve
FRA	Forward Rate Agreement
LMA	Levenberg–Marquardt algorithm
HC	Hard Calibration
SC	Soft Calibration
ML	Machine Learning
PCA	Principal Components Analysis
UL	Unsupervised Learning
SL	Supervised Learning
SSL	Semi-supervised Learning
RL	Reinforcement Learning
LF	Loss Function
EPE	Expected Prediction Error
MSE	Mean Squared Error

GP	Gaussian Process
EURO	Euro
MLP	Multi-Layer-Perceptron
USD	US Dollar
GBP	British Pound
NZD	New Zealand Dollar
CAD	Canadian Dollar
JPY	Japanese Yen
CHF	Swiss Franc
NOK	Norwegian Krone
SFS	Supervised Features Selection
IFS	Iterative Features Selection
SVD	Singular Values Decomposition
ATM	At The Money
DTR	Decision Tree Regressor
RMSE	Root Mean Squared Error

ABSTRACT

The study of new techniques for pricing options and more in general derivative contracts, is gathering increasing attention on both the Financial community and the Applied Statistic community. It is interesting and enterprising to research new models capable of describe financial market behaviours.

This thesis addresses the question of how financial curve and derivative contract price predictions can be improved using machine learning techniques. The focus is on the modelling of discount and interest-rate curves leveraging on Vasicek model, with swaption prices as unique feedback.

In this work we will focus on performances obtained by offline and online models, in addition we will evaluate the performances with respect to the currently used analytical models. Along the model development, there will be several arguments that consist of reliable analyses about the analytical model. In particular we will focus on relationships between peculiar values that constitute curves and relative contract prices. These analyses are an additional knowledge base that is usually missing, this is due to either contract practitioners and composers know only how to use them better or their financial mathematical details.

Previous research has focused on using time series, or Machine Learning techniques directly applied to predict contract variables that define instrument state. Our diversity relies on the ability of predicting curves that define several contracts.

Besides the achieved results, there is an accurate analysis of our approach limitations and future work suggestions.

SOMMARIO

Lo studio di nuove tecnice di prezzamento delle opzioni e più in generale di contratti derivati, sta ricevendo notevole considerazione sia nella comunità di Finanza che in quella della Statistica Applicata. Risulta infatti particolarmente interessante e intraprendente la ricerca di nuovi modelli in grado di descrivere i comportamenti futuri dei mercati finanziari.
Questa tesi si occupa di rispondere alla domanda di come migliorare la predizione di curve finanziarie e in particolare prezzi di contratti che ne derivano, andando a utilizzare tecniche di Machine Learning. In particolare ci concentreremo sul modellizzare l'andamento delle curve di sconto e dei tassi di interesse tramite un modello Vasicek, avendo come unico riscontro il prezzaggio di contratti swaption definiti a partire da tali variabili.
In questo lavoro ci concentreremo sulle prestazioni ottenute da modelli offline e online, infine valuteremo le prestazioni ottenute rispetto ai modelli analitici che vengono utilizzati attualmente. Durante la costruzione di tali modelli, saranno presentate varie argomentazioni che consistono in analisi robuste circa i modelli analitici utilizzati. In particolare ci focalizzeremo sui legami di dipendenza tra valori caratteristici componenti le curve e i prezzi dei relativi contratti. Queste analisi costituiscono una base di conoscenza aggiuntiva che spesso viene a mancare, in quanto gli utilizzatori dei contratti come anche chi li definisce, conosce solo o le proprietà che ne consentono un utilizzo migliore, o i dettagli matematico finanziari.
I precedenti lavori di ricerca consideravano o l'utilizzo di serie temporali, o l'utilizzo di tecniche di Machine Learning per predirre direttamente variabili caratterizzanti lo stato di particolari contratti. La nostra diversità risiede nell'abilità di predirre curve presenti nella definizione di svariati tipi di contratti.
Oltre i risultati conseguiti, è presente una analisi dettagliata delle limitazioni e degli scenari futuri suggeriti.

1 INTRODUCTION

It takes you 500,000 microseconds just to click a mouse. But if you're a Wall Street algorithm and you're five microseconds behind, you're a loser.
Kevin Slavin (July 2011)

I have always been interested in the financial markets. In particular I have always asked myself how it works, and how it can change our everyday life.
From what I have learnt about finance, this complex world was completely revolutionized with the introduction of Modern Portfolio Theory (MPT) five decades ago. MPT was pioneered by Harry Markowitz (Markowitz, March 1952). MPT is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given level of risk, defined as variance. The year 1952 can be considered as the advent for development of increasingly sophisticated financial instruments.
This growth catches on thanks to *deregulation*. Deregulation states that market economy should be supervised by supply and demand, without any government intervention. This phenomenon takes place for the first time in the 1980s with the deregulations of U.S. airlines and AT&T.

The increases in complexity, required significant investments in automation. Nowadays the number of financial transactions, as well as the speed at which they are handled, would not be possible without the support of computers.
A big field that is strictly related to this topic is the High Frequency Trading (HFT). In this world computers take less than a microsecond to evaluate a single transaction; losing just a microsecond, would lead to the loss of hundreds of millions of dollars in a year (M. Malvaldi, 2014).
For this reason today cash movements require significant development of models for predicting and pricing, that are subjected to restrictive time and accuracy constraints.
This field incorporates us to one of the aspect that most fascinates me, that is the conjunction between financial and IT domains. As a lot of IT people, I continuously wonder myself about the old but still present competition between humans

and algorithms, thus, reliability versus speed. Since the introduction of computers, we have wondered what their limitations and potentialities would be. Now we are dealing with computers that learn from their own errors. The effects carried by this auto-learning approach, would be dramatic. In particular, the insights gained and the efforts spent over the last years in both the financial and the artificial intelligence fields, have substantially contributed to the increase of efficiency in modern world economies and the business operating in them.

As I mentioned there are lots of benefits for such a progress, however, there are also some drawbacks.

During the Global Financial Crisis (GFC) of 2008-2011, there were a lot of cases of problems occurred due to computers and HFT. A clamorous episode known as flash crash, happened on the May 6, 2010, when several concurrent algorithms stumbled upon each other causing loss of almost 10% of Dow Jones value (M. Malvaldi, 2014).

The last recession as well as all other financial catastrophes, always involves low and medium classes. I am half Brazilian and in this country the gap between different class of people is very high and marked. In particular, financial recessions caused the widening of this gap to the detriment of the middle and lower classes. This is what likely ignites my interest.

This thesis is about swaps, a particular kind of derivative. Derivatives are a category of contracts that occupy the biggest slice of the current financial market, and are also the main actors in the GFC. With the growth of the market in financial derivatives, the pricing of instruments for hedging positions on underlying risky assets and optimal portfolio diversification have become major activities in international investment institutions (McNelis, 2005).

1.1 RESEARCH DOMAINS

1.1.1 *Finance: swap and swaption*

A swap is a contract between two parties agreeing to exchange payments on regular future dates for a defined period of time, in which one stream of future interest payments is exchanged for another based on a specified principal amount (Chisholm, 2004).

The first swap contracts were negotiated in the early 1980s between IBM and World Bank (Hull, 2011). Since then the market has seen phenomenal growth of the number of contracts. Swaps now occupy a central position in the market of derivatives. As mentioned, they are derivatives, this means that their value is derived from another asset called *underlying*, in the specific case an interest rate.

The most common type of Interest Rate Swap (IRS) is a fixed-floating deal in which the payment made by one party is based on a fixed interest rate, and the return payment is based on a variable or floating leg. *Leg* is a technical term that denotes the cash flows exchanged in a swap.

Swap are usually used by corporations, by investing institutions and by banks with the purpose of mitigating and transfer risk between parties that wants to reduce it, to those who want to increase it. In the financial domain this operation is referred as *hedging*.

SWAP CHARACTERISTICS. The most common type of interest rate swap is the fixed/floating swap, often referred to as "plain vanilla deal " (Chisholm, 2004). The characteristics of a vanilla IRS contract are:

- The nominal amount of the swap is used to calculate the interests. This amount is notional, that is to say it is never exchanged;
- One party agrees to pay a fixed rate of interest applied to the notional principal on regular future dates, this rate is called strike price;
- The other party agrees to make a return payment on regular future dates based on a variable rate of interest applied to the same notional principal. The referred index can be LIBOR, EURIBOR etc.;
- The frequencies between consecutive payments either for the fixed or the floating leg are specified by the tenor τ. It denotes how long takes between consecutive payments.

SWAP EXAMPLE Let us suppose that two parties Charlie and Sandy contract a vanilla IRS between themselves. For example, assume that Charlie owns a €1,000,000 investment that pays him EURIBOR + 1% every month. As EURIBOR goes up and down, the payment Charlie receives changes. Now assume that Sandy owns a €1,000,000 investment that pays her 1.5% every month. The payment she receives

never changes. Charlie decides that that he would rather lock in a constant payment and Sandy decides that she would rather take a chance on receiving higher payments. So Charlie and Sandy agree to enter into an interest rate swap contract.
Under the terms of their contract, Charlie agrees to pay Sandy EURIBOR + 1% per month on a €1,000,000 principal amount (called the "notional principal" or "notional amount"). Sandy agrees to pay Charlie 1.5% per month on the €1,000,000 notional amount.

The terminology for swaps always refers to the fixed leg, so the holder of a *receiver swap* will receive the fixed leg and pay the floating one (Bjork, 2003). Conversely, for a *payer swap* the payments go in the reverse direction. Now that I have described what a swap is, I can go further and describe the swaption, that has the leading role of this project.
Swap options, or swaptions, are options on interest rate swap and are the most popular type of interest rate option. They give the holder, the right to enter in a certain interest rate swap at a certain time in the future, at the swaption expiry time (Hull, 2011).

SWAPTION VARIETIES. According to the rights given to the swaption holder, we can distinguish among a few varieties of contracts .

- Bermudan swaption, in which the owner is allowed to enter the swap on multiple specified dates;
- European swaption, in which the owner is allowed to enter the swap only on the expiration date;
- American swaption, in which the owner is allowed to enter the swap on any day that falls within a range of two dates; (Robert Kolb, 2002)

We are not interested in all these kinds of options, but only on the European ones. This kind of option is the standard in the marketplace, and is the one involved in our case study. All the described instruments are analytically defined in the second chapter of this document.

1.2 RESEARCH OBJECTIVE

The framework in which we are operating is called *Calibration*. It consists of find the optimal parameters for a dynamic model, that describes the behaviour of financial curves that are necessary to define the swaption leg values, thus the contracts' prices.

Currently, this problem is approached as an iterative minimization task over the differences between real contract prices and the ones derived by the so far estimated parameters according to analytical model (Black's formula).

The research presented in this thesis aims to overcome current gaps. The principal problems involved in this task are those of determining the right prices through complex mathematical models and formulas that require high computational time. A second obstacle is given by the presence of attractive local optimal results for the dynamic model.

The hidden assumption behind the use of ML techniques is that by following data-driven optimization, is expected to have advantages in terms of complexity, produced outcomes and adaptation to changing environments. In fact they are suitable to recognize trends in data. That are the reasons why introduction of ML techniques should be appropriate in the face of its advantages and the presented problems.

As we will be discussed and motivated in the next chapters, the approach followed in this project is based on Supervised Learning (SL) (e.g. Multi-Layer-Perceptron (MLP)). As a brief introduction, I only highlight that according to the current state of the art, neural networks are particularly well-suited to the type of data found in financial markets, such as high dimensional data sets of noisy data with apparent non-linear relationship between the variables (McNelis, 2005).

1.2.1 *Calibration*

As was described in the previous chapter, we are dealing with an iterative minimization problem. It consists in fit the behaviour described by the European swaption prices with an arbitrarily pre-established model. The parametric model only predicts curves that are necessary to compute the contract prices.

Within the project, only one problem may derive from erroneous prediction, errors may lead to wrong investments over swaptions and creation of non optimal portfolios. By reasoning more in detail on how these prices are computed, we comprehend that the biggest controlled assets involved are the predicted curves, in fact with those we can price several other financial instruments. More qualitative speaking, knowing those

curves allows us to get a good prediction over the future state of several financial contracts that depend on them. We are definitely not restricted to swaptions as a single instrument.
This means that the spectrum of application of the provided algorithm are wide-ranging and takes a leading role in the financial domain.
Currently, contract prices are computed by exploiting the Black's formula that takes as input the implied Black's volatility for computing the prices besides the the mentioned curves. So far we are not interested in the mathematical aspects, we have just to know that the second parameter, thus the implied volatility is not provided by the market, in fact it only gives the instrument prices and the curves. Anyway, volatilities are usually preferred by traders since they give a quick idea of the instrument state and behaviour. Output parameters, thus the ones of the model used to describe curves, are derived with a gradient descent approach over a loss function that is defined over the predicted and the real prices. Predictions are made with the Black's formula upon the estimated curves and the given implied volatilities.
The mathematical details behind the prices computation and the minimization algorithm is described in detail during the next chapter.

1.3 RESEARCH CONTRIBUTION

The classical approach to the calibration problem, that I will call Hard Calibration (HC), is capable of pricing all the instruments and it was used to make our database. However bank, cannot exploit HC due to the time constraints. They instead leverage on a weighted calibration that selects only a subset of instruments. I will call this approach Soft Calibration (SC). The SC drawbacks are:

- It is able to consider only a limited subset of the swaption prices and totally overlooks the leftovers. In particular the looked subset is composed of only 7 instruments from a total of $\backsimeq 200$;
- The compromised version produces results that are currently not evaluated with respect to the ones given by the optimal method.

The provided model counteracts both the phenomenons. It follows a data-driven approach where the data are derived directly from the complete calibration version, this means that avoids the local optimal problem that inflicts the compromised method. It also reduce the computational time required from the optimal calibration by exploiting machine learning techniques. Finally for what concerns the scalability limitation, the data-driven model learn how to best price all the available instruments together. The fourth chapter of this thesis contains a detailed version of the obtained results.

1.3.1 *Computer Science techniques*

From a computer-science point of view there are several different fields involved during the project, they are not only strictly related to the data-driven algorithm:

- Object Oriented Programming (OOP) was used to create the original dataset from which the machine learning task starts. It was also necessary for the comprehension of state of the art code, and to the development of the functionalities required for the dataset writing. Moreover, it was necessary in the evaluation phase for the developed machine learning model. This task was done by exploiting a domain-based function that was in a hidden tier of the object-oriented software architecture.

- Data analysis and visualization: was necessary for the first interaction with data. It allows me to study data distributions and shapes, and analyse features correlations. These tasks was fundamental for reducing the original features space and also to get in touch with data.
  These skills were necessary also in a second phase in order to make outlier identification and to get a quick qualitative feedback on the prediction made by the developed model.

- Machine Learning::
  - Dimensionality reduction: dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration, via obtaining a set of principles variables. It can be divided into feature selection and feature extraction. It can also be divided into supervised and unsupervised dimensionality reduction.

This technique as a leading role also in the financial domain. Often large amounts of data are summarized with averages, medians, or trimmed means. For instance, the Dow-Jones Industrial Average is just an average of industrial share prices (McNelis, 2005).
The techniques covered by this field were massively used for a preprocessing of the initial dataset, in order to lead it in a structured reduced version and to give us an empirical vision of feature importances and correlations. All the experimental details about these phases are addressed by the fourth chapter of this document.

- Supervised Learning: there are two types of problems known as supervised and unsupervised learning. It is called “supervised” when there is the presence of the outcome variable to guide the learning process. In the "unsupervised learning" case, we observe only the features and have no measurements of the outcome. Our task is rather to describe how the data are organized (T. Hastie, 2009).
  As it will be widely described in the next chapters our project is a classical example of multi target supervised learning problem.

- Deep learning: (also known as deep structured learning, hierarchical learning or deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations. This structure grew out of the cognitive and brain science disciplines for approximating how information is processed and becomes insight.
  This framework usually involves Artificial Neural Networks (ANN) for forecasting. "*Forecasting* simply means understanding which variables lead or help to predict other variables, when many variables interact in volatile markets" (McNelis, 2005).

1.4 DOCUMENT PRESENTATION

During the next chapters I will present the problem from the financial and the machine learning perspective, by explaining the set of instruments involved. Only in a second phase there will be the description of the experiments, the developed models and their results.
I will now give a division by chapters in order to facilitate the reader. In the first chapter we have just got an introduction to the problem from a high-level description. In the next chapter is defined all the maths behind the financial contracts and the minimization problem. In the third chapter, I will formally introduce all the techniques necessary in the development of the data-driven algorithm; here there will be an investigation of the current state-of-the-art solutions for this specific problem. The main contribute of this document, is the fourth chapter, it discusses all the steps taken in building the final algorithm. It also describes the obtained results from a financial perspective.

The project is made in collaboration with IMI Bank. This implies that all experiments are made on real scenarios and not on simulations.

2 SWAPTION: A THEORETICAL VIEW

Non perdere tempo con la finanza, anche facendoci un corso di dottorato ci sarebbero cose che non capiresti.
A. Prampolini (12/05/16)

This chapter is set to provide a sufficient financial knowledge necessary to reach the research objectives outlined in the introduction. It follows a bottom-up direction, from the simplest element to the coarse-grained object.
As a first step I define what is a derivative, that is the most popular contract in the document and in the market.
With the word derivatives, we refer to contracts that are completely defined in terms of an underlying asset, which makes it natural to call them *derivative* or *contingent claim* (Bjork, 2003) .

Definition 2.1. Derivative is any instruments whose price depends on, or is derived from, the price of another asset called *underlying* (Hull, 2011) .

2.1 SWAP

In this section we will define the simplest of all interest rate derivatives, the Interest Rate Swap (IRS). This is a contract in which you exchange a payment at a fixed rate of interest, known as swap rate, for a payment stream at a floating rate, in our case denoted with a forward rate curve (FWD). In the European marketplace the standard forward curve is the Euribor Curve (EUR). These derivatives are rarely treated directly between parties, unless they are financial institutions. The Figure 2.1 specifies the surrounding context in which we are operating. There are basically three actors involved in swaps: the bank, the Street and the Corporate. Figure 2.1 shows two different swap instances: A and B. Swap A is exchanged between the bank and the Corporate and contains a gain for the bank. Swap B is exchanged between the bank and the street and it is evened out.

A more formal representation of a swap model is provided in Figure 2.2.

A swap is completely specified by the following elements:

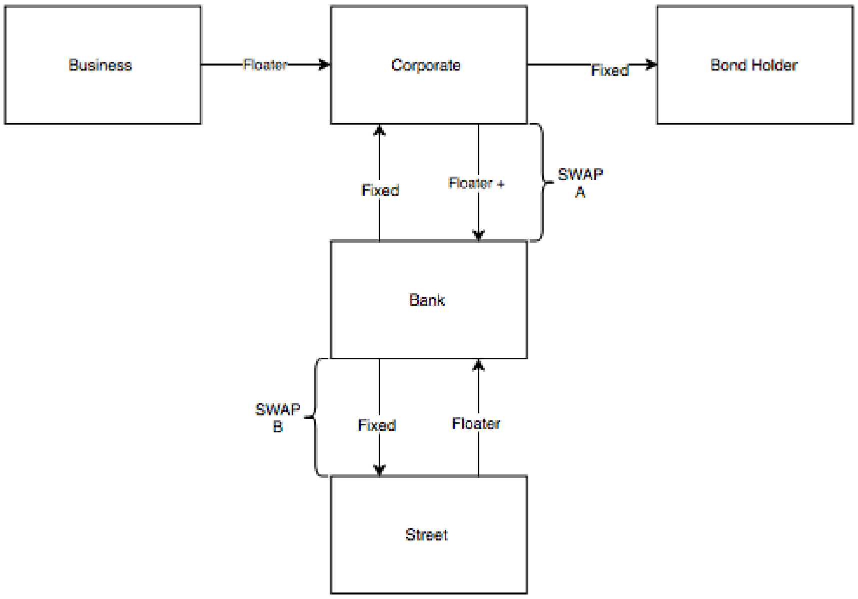


Figure 2.1: Context in which our project takes place. It represents exactly the environment in which these contracts are applied.

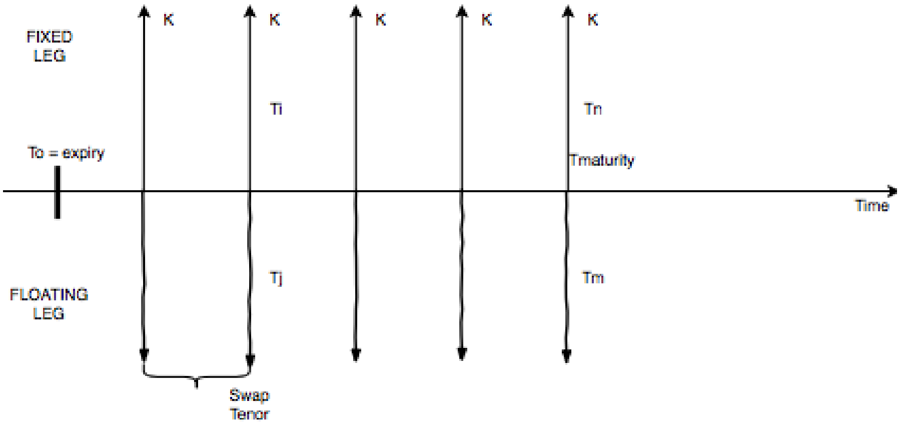


Figure 2.2: Abstract representation of swap contract.

- nominal value: N, thus the amount to be deposited at T_{exp};
- strike price: K, that defines the interest rate paid by the *fixed leg*;
- float rate: FWD, that is indexed by a market defined curve for example: EURIBOR, LIBOR. It defines the interest rate paid by the *floating leg*;
- tenor: τ, that defines the elementary period for the legs payment. In particular specifies the interval between the payments;
- expiry date: T_{exp}, that specifies when the contract will start;
- maturity date: T_{mat}, that defines when the contract will end.

We assume to have a number of equally spaced dates $T_0, .., T_n$ and payments happen at time $T_1, ...T_n$. Equal reset dates, thus payment dates, on both sides are assumed in the rest of what follows without loss of generality, this is often not true in the reality.
If we swap a fixed rate for a floating one, and by denoting with τ the time interval between consecutive dates, then, at time T_i we will receive the amount:

$$N \; \tau \; FWD(T_{i-1}, T_i) \tag{2.1}$$

that defines the *floating leg*. The term $FWD(T_{i-1}, T_i)$ is the Forward rate, it will be published only at time T_{i-1} for the interval (T_{i-1}, T_i) . This lack of knowledge of the Forward rate at time $t < T_{i-1}$, is the reason for the name floating leg.
At time T_i you will pay the amount:

$$N \; \tau \; K \tag{2.2}$$

that is usually known as *fixed leg*.
The net cash flow at time T_i is thus given by the difference of the two legs:

$$N \; \tau \; [FWD(T_{i-1}, T_i) - K] \tag{2.3}$$

2.1.1 *Actualization with Discount Curves*

Since our interest is to evaluate the swap value, we need a further function to actualize at time $t < T_i$ (today) the cash flows that will be paid at time T_{i+1}. This will allow us to compute the value of the entire leg.
Such a task is accomplished thanks to the discount rate curve, in the European market place, that is the main case dealt by this project, we will treat with OIS discounting curve (OIS). The discount curve indicates the expected value of the amount required at time t, to have at time T the cash flow equal to 1.
If we have the short rate $r^{dsct}(u)$ for the infinitesimal interval $[u, u + du]$, we can define the long rate curve as:

$$DSCT(t, T) = e^{-\int_t^T r^{dsct}(u)du} \tag{2.4}$$

This definition is derived in the proof below.

short-long discount rates relation.

$$DSCT(r^{dsct}(t)dt + 1) \tag{2.5}$$
$$DSCT(r^{dsct}(t)dt + 1)(r^{dsct}(t + dt)dt + 1); \tag{2.6}$$
$$DSCT(1 + r^{dsct}dt)^2; \tag{2.7}$$
$$DSCT(1 + \frac{r^{dsct}T}{n})^n; \tag{2.8}$$
$$DSCT(T) = e^{\int_t^T r^{dsct}(u)du} DSCT(t); \tag{2.9}$$
$$DSCT(t) = DSCT(T)e^{-\int_t^T r^{dsct}(u)du}. \tag{2.10}$$

In the first step we consider $t = t + dt$. From (2.6) to (2.7) we assume r^{dsct} constant in dt. From (2.7) to (2.8) we consider $dt = \frac{T}{n}$. Finally from(2.8) to (2.9) we move n to infinite $\lim_{n\to\infty}$. ■

Now that we have built a framework to actualize future values, the only missing element specified in the contract is the rate to be paid for the derived floating leg.

2.1.2 *Floating leg rates: FRA and forwarding*

In our simplified context, we will see the floating rate as composed by multiple Forward Rate Agreement (FRA) instances. FRA is another type of contract with an abstract representation depicted in Figure 2.3.
It has a single payment: a fixed rate K for a float rate $L(t, T_{fix}, T_{pay})$

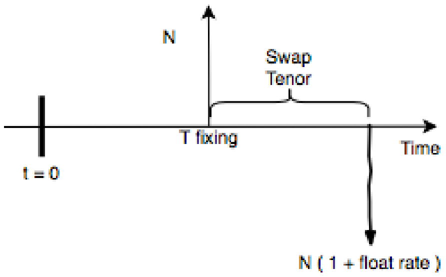


Figure 2.3: Forward Rate Agreement contract schema.

with respect to a nominal value, in this case N. The floating payment occurs at $T_{pay} = T_{fix} + \tau$. When we consider the FRA as a contract per se, τ is usually referred as *year fraction*, only in this overlapping situation it coincides with the swap tenor.
In our case we can consider the FWD curve with an opposite role with respect to the discount function, in fact its aim is to determine the forward rate to be paid as an interest. More formally, we should consider the fixed rate equilibria K_{eq} such that the FRA value seen today, thus at time t for the FRA at (T_{fix}, T_{pay}), is zero. The FRA value is computed by following the same approach seen for the swap contract and subtracting the leg value at the same date. This specific K_{eq} is known as FWD *forward rate*, and it will be denoted as $L(t, T_{fix}, T_{pay})$.
Now it should be clear why is called forward, the reason is that it is defined today for a contract specified for (T_{fix}, T_{pay}). K_{eq} has also the property that the unknown flaw $FWD(T_{fix}, T_{pay})$ seen at t is equal to the known index $L(t, T_{fix}, T_{pay})$.
$L(t, T_{fix}, t_{pay})$ is than fixed at time t, it refers to a deposit that will start only in the future at time T_{fix} and will terminate at time T_{pay}.
Given the forward rate defined above, we compute the swap price for each i only with $L(t, T_{i-1}, T_i)$.
As I have already mentioned above, float rate is a stochastic quantity, it is not granted on the contract in fact it is not known at $t = 0$, that is why it is called float leg. As before, we have to determine a fair value for this rate.
A useful way to express the index $L(t, T_{fix}, T_{pay})$ exploits the *pseudo-discount curves* $FWD(t, T)$. We will use two different non-linearly dependent indices to refer to the forward rate, that are the $L(t, T_{fix}, T_{pay})$ index and the pseudo-discount curve $FWD(t, T)$.
Thanks to the introduction of pseudo-discount we can get a suitable definition of $L(t, T_{fix}, T_{pay})$.

$FWD(t, T)$ is called "pseudo" since it can be seen as a discount curve, but conversely from the $DSCT$ that is given day by day, it contains also risk factors, so it is not given explicitly as a discount curve form but just as an input required for calculate the float rate index $L(t, T_{fix}, T_{pay})$, as shown below.
Further more, since the pseudo-discount curve contains risk factors, it will be necessary the introduction of an expected value operator over a probability defined over this risk, such a probability is known as: *risk neutral*.
In the demonstration below, the left-hand side denotes the amount to be paid at time t in order to have the nominal N at time T_{fixing}. The right-hand side, that contains the unknown index $L(t, T_{fix}, T_{pay})$, is the actualization of the nominal plus the interest rate to be paid on it.

Forward rate and pseudo-discount factors.

$$N\ DSCT(t, T_{fix}) = N\ DSCT(t, T_{pay})(1 + \tau\ L(t, T_{fix}, T_{pay}))$$

$$\frac{DSCT(t, T_{fix})}{DSCT(t, T_{pay})} = 1 + \tau\ L(t, T_{fix}, T_{pay})$$

$$\frac{DSCT(t, T_{fix})}{DSCT(t, T_{pay})} - 1 = \tau\ L(t, T_{fix}, T_{pay})$$

$$L(t, T_{fix}, T_{pay}) = \frac{1}{\tau}(\frac{DSCT(t, T_{fix})}{DSCT(t, T_{pay})} - 1)$$

■

In the European market place, we will deal only with EURIBOR curves, with tenor equal to six-months, such a curve it will be denoted as $EUR6M(T_{fix}, T_{pay})$ and defines the interest rate for the index $L^{eur}(t, T_{fix}, T_{pay})$ with tenor τ equal to six months. The point is that, from the EURIBOR we get a pseudo-discount curve $EUR(T_{fix}, T_{pay})$.
Actually there is a tiny difference between what the market gives us and the FWD, in fact the market give us only the expected value over the *risk neutral* probability, I leave to the reader this financial topic, so the following relationship holds:

$$FWD(t = 0, T_{pay}) = \mathbb{E}_{riskneutral}[L(t, T_{fix}, T_{pay})] \qquad (2.11)$$

2.1.3 *Swap Price*

We now have all the elements to calculate the swap value, that it will be denoted as Net Present Value^{IRS}, and then we will finally move to the swaptions.

Definition 2.2 (Swap price).

$$NPV^{IRS} = NPV^{\text{fix leg}} - NPV^{\text{float leg}} \tag{2.12}$$

$$NPV^{\text{fix leg}} = N \sum_{i=1}^{n} K\, \tau_i \, DSCT(t, T_i) \tag{2.13}$$

$$NPV^{\text{float leg}} = N \sum_{j=1}^{m} L(t, T_{j-1}, T_j)\, \tau_j \, DSCT(t, T_j) \tag{2.14}$$

The equation (2.12) defines the IRS net present value. The two equations below formally define the two legs, since in general the legs may be defined on different dates of exchange, I use two distinct time horizon indices.

The previous definition is enough for what concerns the swap price, but since we deal with swap at the money (ATM), we restrict to swap whose strike price K leads to a NPV equal to zero:

Definition 2.3 (Swap at the money).

$$NPV^{IRS}(K_{eq}) = 0 \tag{2.15}$$

$$NPV^{\text{fix leg}} = NPV^{\text{float leg}} \tag{2.16}$$

$$NPV^{IRS}(K \neq K_{eq}) = (K - K_{eq}) \sum_{i=1}^{n} N\tau_i DSCT(t, T_i) \tag{2.17}$$

I will close this section with just a brief explanation of the relationship between the described curves, thus OIS and EUR6M. These functions are linked by a shift that can be derived given their definition in terms of short rates. As already mentioned short rates are stochastic quantities that is why we will use expected value operator:

Definition 2.4 (Shift structure).

$$\begin{aligned}
P^{OIS}(t,T) &= \mathbb{E}[e^{-\int_t^T r^{OIS}(u)du}] && (2.18)\\
P^{EUR6M}(t,T) &= \mathbb{E}[e^{-\int_t^T r^{EUR6M}(u)du}] && (2.19)\\
&= \mathbb{E}[e^{-\int_t^T (r^{OIS}(u)+\varphi(u))du}] && (2.20)\\
&= P^{OIS}(t,T)\Phi(t,T) && (2.21)\\
& && (2.22)
\end{aligned}$$

The variable P is used to denote in a common way, long term rates.

So we have define a new quantity, that will be called *curve-shift*:

$$\Phi(t, T) = \frac{P^{EUR6M}(t, T)}{P^{OIS}(t, T)} \tag{2.23}$$

The existence of this relation will be extremely useful during the dataset analysis phase since explains a correlation between the curves.

2.2 VASICEK PRICING MODEL

Now that we have described all the elements specified in the swap contract, we will describe how the short rate behaviour is modelled using stochastic dynamic equations.
One of the first model devised to capture the short rate behaviours, was proposed by (Vasicek, 1977).

2.2.1 *Model definition*

In his model the short rate $r(t)$ has to satisfy the following stochastic differential equation:

$$dr(t) = k(\theta - r(t))dt + \sigma dW(t) \tag{2.24}$$

where k, θ, σ are positive constants (Hull, 2011) . The term k is the mean reversion velocity, θ is the mean interest rate level or long term mean, σ is the volatility and $W(t)$ the Wiener process. The main characteristic of this model is the exhibition of the mean reversion phenomenon. For instance, let us consider the case in which the current interest rate r is lower than the long term value θ so we have $(\theta > r)$, the drift will be positive and the rate will increase on average.
Conversely, when $(\theta < r)$, the drift will be negative and the rate will decrease on average. The role of the coefficient k is to control the velocity of movements towards the long term mean θ of the rate $r(t)$.

2.2.2 *Wiener Process*

For what concerns the Wiener process, I will briefly explain it since it is a mileston in finance. Wiener (1931) did the mathematician work for Brownian Motion. Brownian motion was a model built to describe the random movements of particles in a fluid. It is also used to describe random movements that have similar behaviours as financial curves. Besides the Vasicek model, it appears in the definition of the Block-Scholes option pricing model, that is a de-facto standard in the financial domain for pricing options.

Definition 2.5 (Wiener process)**.** A *Wiener Process* W(t) with $t \geqslant 0$ is a real valued, continuous, stochastic process with the following properties:

- $W(0) = 0$;
- W has continuous paths with probability 1;
- W has Gaussian independent increments.
  Thus, if $r < s \leqslant t < u$ then $W(t) - W(u)$ and $W(s) - W(r)$ are independent stochastic variables.
- For $s < t$ the stochastic variable $W(s) - W(t) \sim N(0, \sqrt{t-s})$

(Bjork, 2003)

2.3 SWAPTION

Now we have all the elements required to move to the main contract involved in this project, the swaption. As for the swap, I firstly make an example and then I will formally define its details.

2.3.1 *Swaption Contract*

Definition 2.6 (swaption)**.** A $T_{exp} : (T_{mat} - T_{exp})$ payer *swaption* with swaption strike price K, is a contract which at the exercise date T_{exp} gives the holder the right but not the obligation to enter into a T_{exp} $(T_{mat} - T_{exp})$ swap with the fixed strike price K and swap tenor δ (Bjork, 2003).

This contract is totally defined by two times variables:

- T_{exp}: that is the time at which the holder may decide to exercise its right to enter in the underlying swap contract, thus the *expiry*;
- T_{mat}: that is the expiration date of the contract and corresponds, without loss of generality, with the last payment date of both the legs.

Over these quantities is possible to define a tenor for the swaption, that is equal to the difference of the two dates. It defines the option duration and it will be denoted as δ.
If a swaption gives the holder the right to pay fixed and receive floating, is called *put option* on the fixed-rate bond with strike price equal to the principal. Conversely, if a swaption gives the holder the right to pay floater and receive fixed, it is called *call option*.

2.3.2 *Swaption price*

As shown in Figure 2.4 there are different possible scenarios for the K_{eq} value at time T_{exp}. This will introduce the expected value operator in the calculation of the swaption payment. We must not forget that K_{eq} is directly derived starting from the forward rate.

$$NPV^{option}_{T_{exp}} = ((K_{eq} - K)^{+} DVO1(T_{exp}) \tag{2.25}$$

$$NPV^{option}_{t} = \mathbb{E}_t[((K_{eq} - K)^{+} DVO1(T_{exp})] \tag{2.26}$$

where DVO1 is the sum of discount factors over the summed payment dates:

$$DVO1(T_{exp}) = \sum_{i=1}^{n=T_{exp}} \tau \, DSCT(T_{i-1}, T_i) \tag{2.27}$$

. The meaning of the positive part is that, if the return is negative the swaption holder will never exercise it.

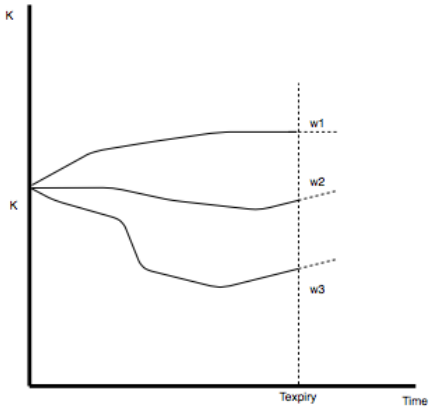


Figure 2.4: Swaption stochasticity over the strike value.

2.4 BLACK FORMULA

I will now introduce a new framework known as Black-76, that is an option pricing model. The name comes from the fact that it was first presented in a paper by Fischer Black in 1976.
This formula involves two main variables that are *swaption price* and *implied black volatility*. These will be analytically defined in the formula definition. In particular we refer to a shifted version that introduces an additional variable known as: log-normal shift.
From now on we will use the Black's formula as a mask for the evaluation of the swaption prices. Thanks to this model we can refer to prices as direct numbers, or as implied volatilities plus log-normal shifts. Thus, Blacks' formula defines a non linear relationship between volatilities and prices.
Even if the implied volatilities are an input of Black-76 to compute the swaption prices, usually the market provides the prices. The *implied Black volatility* σ_{Black} shapes the stochasticity over the strike K_{eq}; for this reason volatilities are usually preferred by traders, and in general by users, since they transmit more information regarding the instruments' status. First I define the formula inputs, and then I will move to its analytical details.

$$\text{Black-76}(K, T_{exp}, T_{mat}, \sigma_{t,T}, FWD(t,T), DSCT(t,T)), lnshift)$$

where:

- K, is the strike price defined on the contract;
- T_{exp},T_{mat}, are the described dates that are defined on the contract;
- $\sigma_{t,T}$, is a constant known as implied black volatility;
- the accrual factor and the PAR swap rate that are derived starting from the $FWD(t,T)$ and $DSCT(t,T)$ curves as shown in the following intermediate paragraph;
- $lnShift$ is the log normal shift and it is a necessary input for compute the contract value.

As anticipated I present now some easy steps in order to define useful quantities that will be exploited later by the Black-76.

It is easy to see that the arbitrage free value, at $t < T_{exp}$, of the *floating payment* is given by (Bjork, 2003):

$$\begin{aligned} & \sum_{i=T_{exp}}^{T_{mat-1}} [FWD(T_{i-1}, T_i) - FWD(T_i, T_{i+1})] && (2.28) \\ = \; & FWD(T_{exp}, T_{exp+1}) - FWD(T_{mat-1}, T_{mat}) && (2.29) \end{aligned}$$

The discount does not appear in the formula since all the terms are discounted at the same time t today.
The total value at time t of the *fixed leg* equals (Bjork, 2003):

$$\sum_{i=T_{exp}}^{T_{mat-1}} DSCT(t, T_i)\, \tau\, K = K\, \tau \sum_{i=T_{exp}}^{T_{mat-1}} DSCT(t, T_i) \tag{2.30}$$

Given these results we can reformulate the *swap net present value* as:

$$\begin{aligned} NPV^{swaption}(K, T_{exp}, T_{mat}) = FWD(T_{exp}, T_{exp+1}) - \\ - FWD(T_{mat-1}, T_{mat}) - K\, \tau\, DSCT(t, T_i) \end{aligned} \tag{2.31}$$

Definition 2.7 (PAR swap rate)**.** (Bjork, 2003) I can now define the *par swap rate* or *forward swap rate* as:

$$R_{T_{exp}}^{T_{mat-1}}(t) = \frac{FWD(T_{exp}, T_{exp+1}) - FWD(T_{mat-1}, T_{mat})}{\sum_{i=T_{exp}}^{T_{mat-1}} \tau\, DSCT(t, T_i)} \tag{2.32}$$

From the previous definition, we derive another variable that refers to the PAR denominator.

Definition 2.8 (Accrual Factor)**.** For each pair n, u, with $n < u$, the process $S_n^u(t)$ is defined by:

$$S_n^u = \sum_{i=n}^{u} \tau\, DSCT(t, T_i) \tag{2.33}$$

S_n^u is referred to as the *accrual factor* or as the present value of a basis point. It should be noticed from this definition, that it coincides with the DVO1 variable that was introduced in the first attempt of definition of the swaption price.

Definition 2.9 (Black-76 formula)**.** *Black's formula for Swaptions.* The Black-76 formula for a $T_{exp} : (T_{mat} - T_{exp})$ payer swaption with strike price K, is defined as :

$$\begin{aligned} NPV_{T_{exp}, T_{mat}}^{swaption} = S_{T_{exp}}^{T_{mat}}(t)[(R_{T_{exp}}^{T_{mat}}(t) + lnShift)\, N(d_1) - \\ -(K + lnShift)\, N(d_2)] \end{aligned} \tag{2.34}$$

where,

$$d_1 = \frac{1}{\sigma_{T_{exp},T_{mat}} \sqrt{T_{exp} - t}}[\ln(\frac{R^{T_{mat}}_{T_{exp}}(t)}{K}) + \frac{1}{2}\, \sigma^2_{T_{exp},T_{mat}}\, (T_{exp} - t)] \tag{2.35}$$

$$d_2 = d_1 - \sigma_{T_{exp},T_{mat}} \sqrt{T_{exp} - t} \tag{2.36}$$

and $N(.)$ is the cumulative distribution function of a standard gaussian (Bjork, 2003) .

The constant $\sigma_{T_{exp},T_{mat}}$ is known as the Black volatility. Given a market price for the swaption, the Black volatility implied by the Black formula is referred to as the implied Black volatility.

2.5 CALIBRATION

2.5.1 *Calibration method*

To conclude this part, I combine together all the theoretically defined variables in order to explain the task that will be substituted by the machine learning algorithm, thus the *Calibration*. We denote the bootstrapped curves $DSCT_{boot}(t, .)$, $FWD_{boot}(t, .)$, as the discount and forward curves calculated at time t, that is equal to the date of interest. The name of these curves is derived by the bootstrap phase. It consists of the curves computation starting from the more fine granular variables.
Calibration aim is to find the Vasicek's optimal parameters, the ones that give us the best curves $DSCT_{predicted}$, $FWD_{predicted}$, where best means the closest approximation to the bootstrapped ones. I will define below the error metric.
Before introducing it, I need to make a clarification. We use a shifted version of the Vasicek model, this means that the predicted curves will be corrected with a time dependent shifting phase. The shift aims to completely overlap the market curves at least for $t = T_{referenceDate}$. In this way, the only error will regard the future predictions.
It is important to remark that this time-dependent shift has nothing to do with the log-normal shift, since as already defined, this one is a required input in the Black's formula for computing the instruments prices.
Given the predicted curves, and imposing that the swap is at the money, we can derive the strike price K_{eq}.

We have now computed K_{eq} and the predicted curves, so we can derive the swaption price and we will look to minimize the difference with respect to the market one.
In practice we will never work on a single swaption prediction, but we will have a set of instruments to deal with. For this reason with the calibration phase, we focus our attention to a specific set of instruments.
The error E can be analytically defined as below:

$$\begin{aligned} E = \sum_{t_{exp}=1}^{m} \sum_{t_{mat}=1}^{n} vega(t_{exp}, t_{mat}) \| NPV^{real}_{t_{exp},t_{mat}} - \\ NPV^{predicted}_{t_{exp},t_{mat}}(k, \sigma) \| \end{aligned} \tag{2.37}$$

In the equation above, the term NPV^{real} is the one approximated by leveraging on the Black model, $NPV^{predicted}$ is the one approximated by our model. The sum is made over all the swaptions, that are identified by the couple (t_{exp}, t_{mat}). The full set of instruments is usually represented as a matrix. This holds for all the informations: prices, log-normal shifts, implied volatilities. Given the error E the problem that we are trying to solve is:

$$\min_{k \in \mathbb{R}, \sigma \in \mathbb{R}^+} E(k, \sigma) \tag{2.38}$$

where k and θ are the discussed Vasicek parameters. I will not provide you all the matrices, in Figure 2.1 is displayed the general scheme to refer to when we will speak about matrix of instruments, independently from the underlying variable.

	T_{mat1}	T_{mat2}	...
T_{exp1}	swaption(t_{exp1},t_{mat1})		
T_{exp2}			
T_{exp3}			
T_{exp4}			
T_{exp5}			
...			swaption(T_{exp-m},T_{mat-n})

Table 2.1: Contracts displayed in a matrix way. Rows are the instrument expiry dates and columns swaption maturity dates.

Currently this minimization is made by Levenberg–Marquardt algorithm (LMA) over the absolute error of swaption price; in

particular the sum is weighted on *vega*, that is the measurement of an option's sensitivity to changes in the volatility of the underlying asset.
Our goal is to replace this minimization over the swaptions prices, with a Machine Learning algorithm that predict the optimal parameters given the bootstrapped curves and the implied Black's volatilities.
LMA is used to solve non linear least square minimization problems and least square fitting, it can be seen as an interpolation between the gradient descent approach and the Newton algorithm (Marquardt, 1963).
It acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value.
It is a iterative procedure that requires an initial guess to start the minimization. In case of a single minimum the algorithm will converge to it independently from the guess, in cases of multiple minima, the algorithm will converge to the global one only if the initial guess is already close to the final solution.
So we need to run the LMA exponential times with different initialization in order to find global minima.
An alternative may be to use another method such simulated annealing that grants global minima, the problem is that such an algorithm cannot be run before each pricing operation since takes long time to run.

Finally, I present a flow chart of the current calibration to quickly resume all the phases and variables in one shot.

2.5.2 *Calibration: Feedback function*

The bank has provided us two own feedback functions in order to evaluate our results: $eval()$ and $\overline{eval}()$. Their followed approach consists of evaluating the predicted results by replacing the contract market prices. I will denote each input market price as p_i, and its respective predicted price with $\overline{p_{ij}} = g_{ij}(k, \sigma)$. This last price depends from (k, σ). The first feedback function is defined starting from the term:

$$\epsilon_{ij}(k, \sigma) = \|p_{ij} - g_{ij}(k, \sigma)\| \tag{2.39}$$

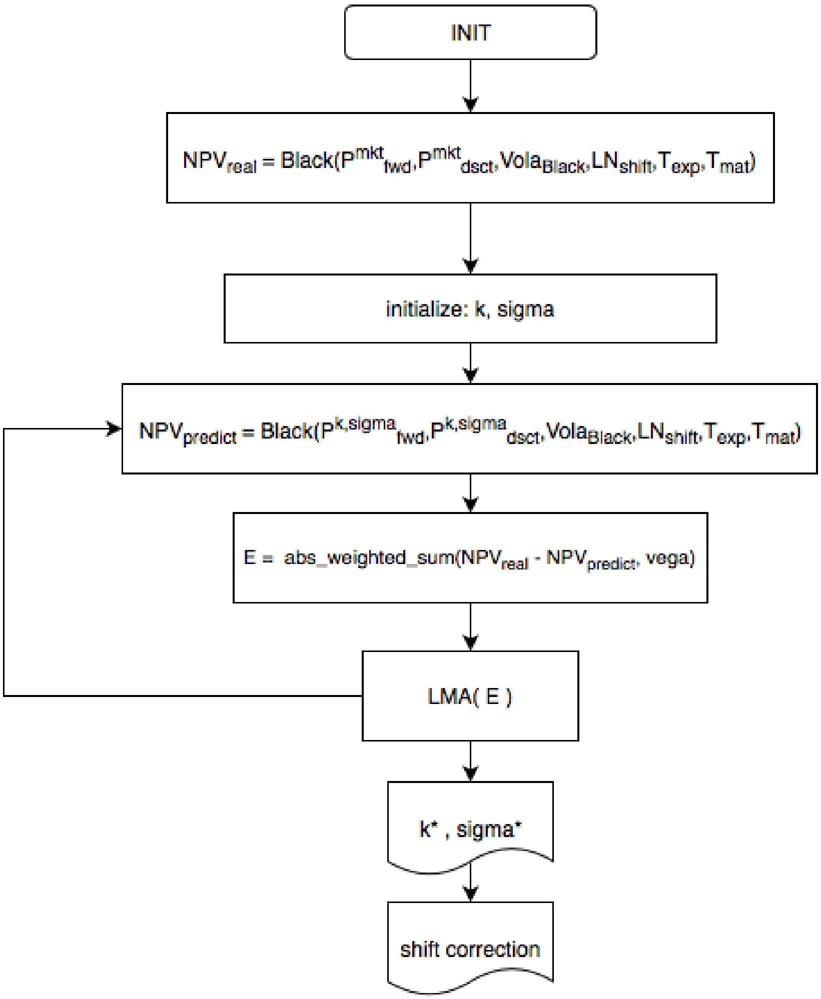

INIT
NPV_real = Black(P^mkt_fwd,P^mkt_dsct,Vola_Black,LN_shift,T_exp,T_mat)
initialize: k, sigma
NPV_predict = Black(P^k,sigma_fwd,P^k,sigma_dsct,Vola_Black,LN_shift,T_exp,T_mat)
E = abs_weighted_sum(NPV_real - NPV_predict, vega)
LMA(E)
k* , sigma*
shift correction

Figure 2.5: Calibration Flow-chart.

It reduces the set of feedbacks related to each single contract to a scalar value by exploiting a norm. This norm was introduced to evaluate the full vector of feedbacks and lead us to the first feedback function provided to us:

$$eval(k,\sigma) = \sqrt[2]{\sum_{i=1}^{m}\sum_{j=1}^{n}\epsilon_{ij}(k,\sigma)^2 w_{ij}^2} \tag{2.40}$$

where w_i are the respective elements in the weight matrix (Section 2.5.1).

This function has a potential flaw: it does not account the fact that the underlying Hard Calibration HC from which our model learn from has an intrinsic error that cannot be removed. This error should not be treated as a penalty since it cannot be deleted. Starting from this argumentation the bank has offered us the second feedback metric: $\overline{eval}()$. This takes as additional input the couple of parameters obtained with the hard calibration $(k*, \sigma*)$. It is defined as:

$$\overline{eval}(k, \sigma, k*, \sigma*) = \sqrt[2]{\sum_{i=1}^{m} \sum_{j=1}^{n} \bar{\epsilon}_{ij}(k, \sigma, k*, \sigma*)^2 w_{ij}^2} \tag{2.41}$$

$$\bar{\epsilon}_{ij}(k, \sigma, k*, \sigma*) = [\epsilon_{ij}(k, \sigma) - \epsilon_{ij}(k*, \sigma*)]^+ \tag{2.42}$$

I will use both these feedback metrics in order to have a better evaluation of our final regression model.

3 MACHINE LEARNING AND CALIBRATION

Machine Learning is not magic, we have to derive new information and not to invent it.
M. Restelli (07/04/16)

This chapter starts from the formalized calibration problem, it then defines a new version based on the data. Besides defining this new model, this chapter also formally describes all the techniques used to handle the encountered problems. Finally it presents the current state of the art for machine learning models and environments.

3.1 A NEW WORLD: MACHINE LEARNING

Let's make a step back before introducing the Machine Learning (ML) world. Since computers introduction, we have wondered what would be computers limitations, and if they would be adapt to learn. If we would be able to set them to learn, thus to acquire knowledge automatically and independently with their own experience, the effects would be huge.
The data driven model, as its name says, is about learning from data. In a typical scenario, we have an outcome measurement, usually quantitative (such as model parameters), that we wish to predict based on a set of features (such as curves and matrix cells). We have a training set of data, in which we observe the outcome and feature measurements for a set of objects (such as calibrations). Using this data, often called experience E, we build a prediction algorithm, usually known as learner whose goal is to predict outcomes of previously unseen objects.
This prediction is based on historical information. In particular we could give more importance on recent data and less to the old. By doing so, we can easily capture trends, this is not possible with the analytical model. In order to keep update this property, we have to consider future training of the learner according to trend changes.
Another useful perspective of machine learning is that it searches in a very huge space of possible hypothesis to best fits the hidden relationship in the data. Clearly this search is made ac-

cording to the structure in the training set. Many of the algorithms search the best fit within a hypothesis space defined by an underlying data representation (e.g. linear functions, decision trees). Different representations are necessary in order to learn several kinds of target functions.

As already mentioned we have a large dataset known as *train set* which is used to tune the parameters of an adaptive model. The result of running the developed algorithm once ready, might be expressed as a new function $g(x)$ where x is a general input vector (Bishop, 2006). For the moment just denote the generated vector of prediction as r, thus $r = g(x)$ and let us try to build-up a solid structure of how ML works.

The way along which the function $g(.)$ is computed, is totally and only dependent from the *training phase*, thus the learning is only based on the training data. That is the reason why train phase is affected more by train dataset then the used algorithm. To be clearer I have to introduce here an additional phase that is used to defined model hyper-parameters, thus the *validation phase*. We consider as model hyper-parameters all the variables that define its structure (e.g. layer in an MLP). This phase comes next to the train stage and applies the last modification to the model.

Once the model will be defined in all its parameters, it will be used to work with new input. This phase is known as *test phase*, and it usually done over a set of records known as *test set*.

One of the most important ability is to correctly behave with unseen input, thus those there are not in the train set. This property is called *generalization* and we will return on it later since it is the hottest topic to look on in this framework.

For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve (Bishop, 2006). This phase will massively used in our project as it will be described in Chapter 4.

Now that we have taken a glance on the most important phases, thus preprocessing, training and testing, we can move to see some dichotomies in machine learning. These introduce a specific terminology that will be widely used from here on.

There are applications in which the training records consist of input vector coupled with their target vector t, in this case we will deal with Supervised Learning problems (Bishop, 2006). The relation between these variables can be expressed by the target function $f(x)$, thus the function that the ML model will

try to approximate.
Conversely, the opposite situation occurs when we have no measurements of any outcome but just features, this is called Unsupervised Learning (UL) (Bishop, 2006). Here the tasks involve to understand how data are organized, if there are hidden connections or structures.
There is an additional field that can be seen as an interpolation between these two, it is known as Semi-supervised Learning (SSL) and is characterized by the fact that in the train set co-occur supervised and unsupervised records. This environment is suitable for financial applications since banks usually have only small dataset of labelled data to its disposal (Sven Sandow, 2007).
Finally there is a third big sphere in ML, that is Reinforcement Learning (RL). RL aim is to learn a policy, thus understand which action have to be taken according to the different situations with the objective of maximize the expected future reward (Bishop, 2006). The main difference with respect to SL is the absence of a set of labelled samples, in fact labels have to be discovered during the learning.

3.2 SUPERVISED LEARNING

It is called "supervised" due to the presence of the target vector in the train set that heads the learning phase. This is exactly the context in which we are, in particular our target vector will be the couple of Vasicek model parameters.
Within this context is used a suitable algebraic representation of the dataset. Let a dataset D consists of n records over d features. Thus, a n $\times$ d matrix. Each record l_i= (l_{i1}, ...l_{id}) can be seen as a vector within the d-dimensional vector space, that is spanned by the d orthonormal basis vectors: e_1,e_2,...,e_d , where e_i is a zero d-dimensional vector with a single one that corresponds to the i-th attribute. Recall that the standard basis is an orthonormal basis for the data space, that is, the basis vectors are pairwise orthogonal, $e_i^T e_j = 0$ *for all i, j*, and have unit length $\|e_i = 1\|$ (Mohammed J. ZakiI, 2014). In particular when we deal with supervised learning problems, the matrix D can be seen as divided by columns in input and target matrices that are respectively represented by X and Y. This means that each record l=(X_{l1},...,X_{ld_1},Y_{l1},...,Y_{ld_2}) can be seen as a vector within the d_1 input dimensional space and the d_2 target dimensional space.

3.2.1 *Learning with a teacher, a density estimation problem*

The SL case is metaphorically known as "learning with a teacher". The interpretation of this appellation is that a student presents an answer r for each input vector x in the training set, and the teacher that is the supervisor, evaluate the student's answer (T. Hastie, 2009). Final grade is computed with a Loss Function (LF) that conversely from the student case, it is not maximized but minimized. The idea, as the name "loss" says, is that it does not define a grade but how many mistakes the student has done. The most common example of LF for quantitative variables is:

$$L(Y,R) = (Y-R)^2 = (f(X) - g(X))^2 \tag{3.1}$$

Now we will look to this minimization problem from a different prospective. If we consider the input X as a real valued random vector, and the target Y as a real valued random variable, with joint probability $P(X,Y)$, then we can review the SL problem as a density estimation problem. Thus instead of approximating the real function $f(.)$, we want to discover the joint probability $P(X,Y)$ or more intuitively causal relationship between input and outcome, thus conditional probabilities over $P(Y|X)$.
This heads us to a new criteria to choose $f(.)$:

$$\begin{aligned} EPE(f) &= \int \mathbb{E}[(Y - f(X))^2] && (3.2) \\ &= \int [y - f(x)]^2 P(dx, dy) && (3.3) \end{aligned}$$

the Expected Prediction Error (EPE).
By factoring the joint density with respect to X, we rewrite the EPE as:

$$EPE(f) = \mathbb{E}_X \mathbb{E}_{Y|X}([X - f(x)]^2 | X) \tag{3.4}$$

and we see that is suffices to minimize EPE point by point as:

$$f(x) = argmin_c \mathbb{E}_{Y|X}([Y-c]^2 | X = x) \tag{3.5}$$

that has the following solution:

$$f(x) = \mathbb{E}(Y|X = x). \tag{3.6}$$

The conditional expectation is known as *regression function*. Thus the best prediction of Y at any point X is given by its conditional mean, where the word best refers to the average squared error.

From this statistic introduction we can resume that our goal is to find an approximation $g(.)$ of the function $f(.)$ that underlies the predictive relationship from input to output variables.
Let us suppose that the generative model behind our data is:

$$Y = f(X) + \epsilon \tag{3.7}$$

where the random error ϵ has $\mathbb{E}[\epsilon] = 0$ and is independent from X. In this case the effect given by the expected conditional value over $P(Y|X)$ leads us to: $f(x) = \mathbb{E}(Y|X = x)$ since the conditional probability over the error is zero thanks to independence assumption over the error ϵ.
The ϵ variable collects all the measurements errors, perturbation and noise that may occur in the data. It is reasonable to think that it does not depend on the specific value of the input. Anyway, this assumption can be easily removed to the dependent case. Since is not the case of this project, I will not focus on this extension.
As I have previously introduced, SL attempts to approximate the function $f(.)$ by example through a teacher. In particular we try to use the information in *training set* to learn the function $f(.)$ hoping to learn a valid approximation for all the possible values of the input x.
Many of the approximation we will encounter are specified by a set of parameters, that I denote with θ, and totally define the learnt approximate function $g(.)$.
What we will do, is find a set of parameters θ^* that minimizes the residual sum of squares:

$$RSS(\theta) = \sum_{i=1}^{N} (y_i - g_\theta(x_i))^2 \tag{3.8}$$

where N is the number of records present in the training set.

3.2.2 *Qualitative and Quantitative Targets*

There might be distinguished two main types: quantitative and qualitative variables, this distinction will define another dichotomy in machine learning.
A first class of problems involves qualitative output. Usually they define classes and so it does not make sense to introduce distance relation between values, thus there is not a metric notion, in this case we will speak about *categorical variables*. There

might be also sub-case where holds an order relationship between elements, in this case we will speak about *ordinal variables*.
For example, if our aim is to predict a temperature in: hot-normal-cold, we are in the ordinal case; instead when we have to predict an image colour there is no sense to think on an order between target values.
A second class of problems involves quantitative measurements, where measurements are bigger than others, and close in values means close in nature.
This distinction in the target types conducts to a naming convention of two main tasks in supervised learning: *regression* for quantitative target variables and *classification* for qualitative ones.
It is always possible to pass from numerical to qualitative representation. This can be achieved by representing categorical variables with numerical codes thus by coding them. It is often preferred a one-shot codification, except when the translation is made by an expert that is able to handle with induced relations.
The one shot case is technically known as *dummy variables*, if we have a qualitative variable with K possible values, we will represent it with a vector of K binary variables or bits, and only one per time can be set to one. Also the reverse is possible, in particular we might need to translate numerical targets to categorical, this task is made with discretization.
This possibilities are explained since some algorithms may work only with numerical values and others only with qualitative.

Our problem can finally be defined as a supervised regression task.

3.2.3 *Online and offline learning*

I will now briefly introduce another dichotomy in machine learning. Algorithms can in fact be distinguished in *online* and *offline*.
At a first glance the distinction is based on the availability of data, in reality it depends on how the algorithm uses them.
We are dealing with an *online algorithm* when data are used sequentially one per time, as in a stream, thus the training phase is not made in a single step on the whole block of records, but each new record contributes in the learning by updating the

current best predictor. This means that the model continues to evolve, updates are not made just once.
The opposite is the batch or *offline learning* technique in which the learning is made on the entire training dataset at once.
Generally speaking, online approach is used when is not computationally feasible, or is too expensive, to train over the entire dataset. It is also used when there are time dependent patterns within the data, as it might be in stock price prediction.
In our project we will introduce both these approaches and we will end with some considerations on their results.

3.3 PARAMETRIC AND NON PARAMETRIC MODELS

To be more specific according to the evaluations and comments that will be described in the following chapters, I must mention two different families within the machine learning algorithms: *parametric* and *non parametric* models.
As I have already mentioned, machine learning algorithms try to best approximate the underlying function $f(.)$ with $g(.)$.
Qualitatively speaking, the target function is unknown, so a machine learning scientist has to consider and make different assumptions regarding the function shape and the learning process.
Assumptions usually help in simplifying the learning process, even if they may limit the research area. Algorithms that make assumptions on the function form are called *parametric algorithms*.

> *A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model. No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs.*
> *(Stuart Russel, 2009).*

The learning task involves two phases:

- select a specific parametric function;
- learn the function parameters that best fit the dataset;

The advantages in using this class of models range from the simplicity either in their use or in the comprehension of the results, to the high speed of the learning process and last but not

least, in general, they do not required as much training data even if they the fit to the data is not perfect.
Limitations instead range from the most obvious constraint of the function form, to the low unlikely in fitting complex data structures.

Let us now focus on *non parametric models*.

> *Non-parametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features.*
> *(Stuart Russel, 2009).*

This class of algorithms seek to best fit the training records in approximating the function $f(.)$. This observation will be fundamental to discern the empirical results, in particular we will state that *non parametric model are free to overfit the data*. As such, they are able to fit a large number of map target functions.
The advantages of such models are the flexibility on fitting different mapping functions, the power and the performance. Limitations are the need of more data with respect to the parametric case, and the higher time required to train them.

3.4 OVERFITTING AND GENERALIZATION

The ability to correctly classify unseen examples that differ from the ones already seen during the training phase is known as *generalization* (Bishop, 2006). Often, the variability described by the training examples is only a tiny subset of all possible input vectors. For this reason *generalization* is an usual central goal in pattern recognition.
This property is achieved when the model does not learn all the details present in the training set, thus it does not mirror all the data with their noise, but only approximates the real function $f(.)$.
There are also cases in which the dataset does not contain any noise, and cover a wide manifold of the input space. In such a case, our goal is not to generalize but to mirror, technically speaking this goes under the name *overfitting*.
In our context we are dealing with dataset without noise, for this reason we are not worried about overfitting.
Usually this apparently "bad" property is avoided by introducing some countermeasures known as *regularization*. Those try to force generalization by avoiding directly manipulation of

model parameters but by controlling the learning phase in order to avoid mirroring. Qualitatively speaking this is usually translated in avoid to have too complex models or model that are too unstable. The most known techniques are *Lasso* and *Ridge Regression*. I leave to the reader the task to deepen these aspects since they will not use in this project.

3.5 DIMENSIONALITY REDUCTION

All machine learning algorithms are affected by negative effect of irrelevant attributes. As expected this occurs frequently when dealing with high-dimensional data, in which only a subset of the features contains useful features. For this reason in the preprocessing phase it is important to check whether the dimensionality can be reduced while preserving the essential properties of the original features space (Ian H. Witten, 2005).
The best way to perform this step would be manually, based on a deep understanding of the learning problem and on the features meaning and definition. Clearly, this cannot be perform according to the assumption of high-dimensional dataset. For this reason were developed automatic methods.
As all machine learning algorithms also these may work in a supervised or in an unsupervised manner, both aim in removing unsuitable attributes, thus, they improve the machine learning algorithms performances and speed up them.
As a second effect, they help the comprehension of underlying problems by let the attention focus only on the most relevant variables. This as the direct consequence of facilitate the patterns and results visualization.
Robert Collins appeared to believe that model accuracy and feature relevance are two sides of the same coin, when they stated that feature selection "can improve classification performance by discarding irrelevant or redundant features" (Collins RT, 2005).
I will now present the most known dimensionality reduction techniques focusing more on the ones used during the project development.

3.5.1 *Principal Component Analysis*

Principal Components Analysis (PCA) is an unsupervised learning technique that looks for a q-dimensional basis of the original dataset matrix that best captures the dataset information

that is described by the variance in the data (Joliffe, 2014).
The direction with the largest projected variance is called *first principal component*, the second largest *second principal component* and so on. PCA converts the original possibly correlated variables into linearly uncorrelated principal components, that define an orthogonal basis set.
This technique can be run under a parametric way either by specifying a threshold on the number of future dimensions of the reduced matrix, or on the desired captured variance percentage. Finally, PCA replaces the d original variables with a smaller number, q, of derived variables which are linear combination of the original variables. In this sense it can be seen as a dimensionality reduction technique.
I will now make some considerations regarding practical implementations based on their computational cost. Based on a algebraic theory, it can be guess that this technique has a high computational cost, since it is necessary to compute an orthonormal transformation. In particular by referring to a $n \times d$ dimensional matrix l. We can assume without loss of generality that is *centred*, thus column means has been subtracted and are now equal to zero.
We want to find a $q \times d$ orthonormal transformation matrix P so that PX has a $d \times d$ diagonalizable covariance matrix[1] (thus all its distinct components are pairwise uncorrelated). Calculating P matrix means solving a Singular Values Decomposition (SVD) problem:

$$C = VLV^T \tag{3.9}$$

where V is a matrix of eigenvectors (each column is an eigenvector) and L is a diagonal matrix of eigenvalues.
Assuming that P were unitary yields:

$$\begin{aligned} var(PX) &= \mathbb{E}[PX(PX)^T] && (3.10)\\ & \mathbb{E}[PXX^TP^T] && (3.11)\\ & Pvar[X]P^{-1} && (3.12)\end{aligned}$$

This transformation is possible only if $var(X)$ were diagonalisable by P. The good part is that thanks to the variance definition, $var(X)$ is guaranteed to be a non-negative definite matrix and thus is guaranteed to be diagonalisable by some unit matrix.

[1] The covariance matrix is given by $C = \frac{X^TX}{n-1}$. It is symmetric and so it can be diagonalized.

When we are dealing with high dimensional space, the computation of this transformation is too expensive for this reason were developed *iterative computation* that does not require the calculation of its covariance matrix known as *power iteration*.

3.5.2 *Curve Fitting*

We can use the regression definition defined in Section 3.2.1 for dimensionality reduction purposes. In particular by fitting a curve data with a specific parametric function $h(\mathbf{x}, \mathbf{w})$ where $\mathbf{w}$ is the set of parameters that defines the curve approximation and $\mathbf{x}$ is the input vector space (Bishop, 2006), we can exploit the function $h(.)$ parameters to reduce the original dimensional space.

Parameter values will be defined during the training phase by fitting the parametric function to the train data. Once more we can easily seen as a minimization problem over a properly defined *error objective function*. As mentioned, one of the most used error function is the Mean Squared Error (MSE) between the prediction $h(\mathbf{x}_n, \mathbf{w})$ and the corresponding target values t_n, so that we minimize:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} (h(\mathbf{x}_i, \mathbf{w}) - t_i)^2 \tag{3.13}$$

where the factor $\frac{1}{2}$ has is introduced for a mathematical convenience that we will not see during this project.

After the use of this algorithm the original data will be substituted by the vector of parameters $\mathbf{w}$. In this sense this method can be seen as a dimensionality reduction technique.

3.5.3 *Supervised feature selection*

As I have already mentioned, the feature extraction task can be accomplished in both supervised and unsupervised approaches. The previous described techniques such as: PCA and curve fitting are both unsupervised since they do not care on the data targets.

We will now focus on the supervised features selection algorithms. In many pattern recognition problems, identifying the most characterizing features, is critical to minimize the regression error. We are trying to select a subset of features that "optimally" characterize the target. The "optimal characterization condition" often means "minimum regression error". The authors also indicated that: if a regressor is not specified, "minimal error usually requires the maximal statistical dependency" and the task becomes "selecting the features with the highest relevance to the target class" (Huang, 2015).

As in the supervised learning case, there are a bunch of techniques to accomplish this. In our project we will use two different approaches. The first one, that is also the simplest is based on a threshold, the second one follows an iterative pattern. In particular, in the first case we are dealing with an estimator that basically fits the data and assigns to each feature a score. This specifies the feature importance in predicting the target variables. Given the list of scores associated with all the features we will select only the ones whose score is above the scores average.

One of the most common used metrics is the *coefficient of determination*, denoted as R^2, it is a key output in regression analyses and indicates the proportion of variance in the dependent variable that is predictable from the independent ones. Where, given the real and the predicted lists of samples, respectively, Y and Y^*, if $\bar{y}$ is the mean of the observed data:

$$\bar{y} = \frac{1}{n}\sum_{1}^{n} y_i \tag{3.14}$$

we can measure the variability of data using two variables:

$$SS_{res} = \sum_{i=1}^{n}(Y[i] - Y^*[i]) \tag{3.15}$$

$$SS_{reg} = \sum_{i=1}^{n}(Y[i] - \bar{y}) \tag{3.16}$$

starting from the we define:

$$R^2 = (1 - \frac{SS_{res}}{SS_{reg}}) \tag{3.17}$$

where n is the number of samples.

The second used approach is the Iterative Features Selection (IFS) approach (Castelletti A., 2011). Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of IFS is to select features by recursively consider bigger sets of features. Given the targets to be explained

and the set of candidate features, the IFS algorithm first globally ranks the features according to a statistical measure of significance, e.g. the R2 score. To account for feature redundancy, only the most significant features are then added to the set of selected features, which will be used to fit the estimator to explain the targets. The algorithm proceeds by repeating the ranking process using as new output feature the residuals of the model built at the previous iteration. The algorithm iterates these operations until the best features returned by the ranking algorithm are already in the select set, or, the accuracy of the model built upon the selected features does not significantly improve (e.g, the feature scores are too small).
This approach is also used in reinforcement learning problem, since it strongly simplifies the learning of good control policies and can highlight interesting properties of the systems under control. In our case we will use this specific approach in order to understand which features are more relevant in describing the target parameters.

3.6 DATA-BASED MODEL AS A BLACK BOX

Before seeing in the details all the empirical steps I present a first glance on the main steps according to the explained machine learning instruments.
I will start by defining the developed model as a black-box for specifying its input and output variables.
The algorithm gets as input the following elements:

- $Swap_{prices}$, matrix of swaption prices;
- $Swap_{volatilities}$, matrix of swaption volatilities;
- $lnShif$, matrix of logNormal shifts;
- P_{fwd}, bootstrapped forward curves;
- P_{dsct}, bootstrapped discount curves.

It then provides as outcome the Vasicek model parameters:

- k, mean reversion speed;
- σ, volatility.

It is important to remark that the original data have a too high dimensional space, for this reason these quantities are filtered

and reduced during the learning process according to the so far presented methodologies.
It is also useful to underline that all the defined inputs have to be coherent each other, thus they all are generated at the same reference date.
Originally these quantities consists of: full matrices of $\simeq 230$ elements, one per each instrument, where the rows denote the swaptions expiry dates and the columns specify maturity dates. The curves are described by time-series defined as list of couples (date,value), their lengths differ according to the currencies and are established by the market.

3.7 STATE OF THE ART

In the past years many efforts were spent in the research of data-driven models for pricing derivatives. I will now speak about *analytical models* referring to traditional closed-form pricing formulas, and to *statistical model* referring to data-driven method for pricing.
Initially many efforts were spent in using time series models for predicting instruments behaviour. The first main benchmark was carried out with the introduction of statistical models, such as ANN for hedging derivatives (James Hutchinson, 1994). This research opened an alternative framework for pricing derivative assets. Although not a substitute for the traditional pricing formulas, statistical models may be more accurate and computationally more efficient when the underlying asset's price dynamics are unknown or cannot be captured by analytical models, or when the pricing equation associated with the no-arbitrage condition cannot be solved analytically according to external constraints, such as the timing one (James Hutchinson, 1994).
The gap left by this research is given by the period of analysis. In fact, the research was performed before the Global Financial Crisis GFC, that introduced additional cases in the data that were not be taken in account by the presented study. Moreover, as I will discuss, there are some additional drawbacks given by the use of ANN.
After this first step, several works have focused in pricing options. Many attempts were made on pricing options, both statistical and analytical models were be taken into account. What we can conclude is that: ANN models significantly improve the

prediction accuracy of option pricing compared with no-arbitrage analytical and time-series option pricing models (Hyejin Park, 2012; James Hutchinson, 1994). However, statistical models such as a ANNs has some drawbacks. They provide only a point forecast of option pricing, which is less useful to traders than, the distribution prediction of option prices from the point of view of a practitioner (James Hutchinson, 1994).

According to the describe issues, two solution have been proposed: introduction of Gaussian Process (GP) and forecasting of useful correlated variables, such as volatilities instead of prices. For the first path the main results were achieved with the introduction of GP models that not only recommends a solution for overcoming overfitting problems using a variety of mixed kernels for learning, as a byproduct they can also provide a predictive distribution of option prices (Hyejin Park, 2012;Hyejin Park, 2014).

The second approach (J. Beleza Sousa, 2012) refers to the use of GP to calibrate dynamic models, e.g the Vasicek. During this research the calibration task was carried out on zero coupon bond, that are a different type of contract. This research has as main advantage the fact that needs only zero coupon bond prices and all the other parameter are obtained in the risk neutral measure. Its drawback, as it is for all the GP strategies, is that its applicability is restricted to Gaussian models only, or models that can be transformed into Gaussian as it is for the log-normal distribution that occur for zero coupon bond prices.

What we can conclude is that GP are pretty-fine although they are restricted to Gaussian-like models, and *non-parametric models* such as ANN have some drawback since they were used to predict only prices.

By taking in consideration also the efforts spent with the time-series approach, we can state that there is a long and persistent research interest in the option pricing prediction problem. This was expected given the competitive environment of the industries involved and the delicacy of decisions that have to be faced by decision makers.

There are still some unanswered questions so far:

- no prior research was be conducted specifically on multi-currency swaptions, even if the available literature for option pricing is extensive. The questions that may arise is: is the previous work generalizable even to our swaptions case?

- during our project a lot of work was developed in the direction of finding features correlations. This is an interesting area, since a lot of practitioners have an intrinsic knowledge and an own idea about the underlying feature correlation but with no empirical feedbacks. In the literature there was not an work that points out these aspects;

- previous works usually predict prices or volatilities, or their manifold and probability space. The point is that an analysis of the conjunction of the two variables was never made. In our project we are predicting the most fine granular variable, thus the curves. We use the swaption prices just as a feedback for the prediction, but our range of application is so wide that we can predict prices, volatilities of all the contracts that depend on the predicted curves.

4 PRACTICAL ANALYSIS

Niente è difficile, si tratta solo di entrarci in confidenza.
M. Pirotta (September 2016)

In this chapter I will describe all the phases occurred during the development of the final algorithm. I will start from the dataset generation until the production of the algorithms and their results. During the phases description I will massively comment all the considerations that I have taken during the development of the final algorithm.

4.1 DATASET CONSTRUCTION

As I have mentioned in Chapter 1.3, the currently used soft-algorithm, thus the one that has a low cost from a time perspective, has as main limitation the capability of pricing only a small number of instruments. For this reason in developing the machine learning algorithm we have worked towards this lack by learning on the full set of instruments and not on a small subset of them. This practically means that the dataset contains records that refer to an instruments selection matrix (Chapter 2.5.1) of only ones.

The dataset composition is realized by calling the hard-calibration for each reference date, thus the one most complex from a time point of view, that is also the only capable of best pricing all the set of instruments. Clearly it cannot be used in practice for trading purposes due to the high computational time required to solve the problem, and it has to be settled out on a grid architecture, otherwise there would not be any reason to carry out this research. Here it should be clear the second gain leaded by our algorithm, thus the lower computational time.

Given the grid computing[1] definition it should be clearer why the hard-calibration cannot be used before each pricing prediction. I resume here the advantages leaded by the dataset creation phase. We have information referred to all the instruments, in this way the developed algorithms that learn from

[1] we intend a collection of computer that all works to achieve a common goal. In particular this action may be done in a distributed system, thus, involving computers that are physically on different locations.

Currencies abundance		
currency	# artificial samples	# original samples
EURO	17598	2298
USD	191032	3172
GBP	16953	2837
NZD	13125	2187
CAD	13272	2212

Table 4.1: Dataset composition by currency. In the first columns there are the currency names, in the second the number of perturbed data and in the last the number of original data.

currency	first date	last date
EURO	2013-06-28	2016-09-12
USD	2013-06-28	2016-09-12
GBP	2013-06-28	2016-06-08
NZD	2013-06-28	2016-06-08
CAD	2013-06-28	2016-06-08

Table 4.2: Samples interval ranges divided by currency.

them will best price all the swaption contracts. In particular this operation will be made by respecting the time constraints imposed by the soft-calibration model.

This task was realized by exploiting the *calibrateMarketRatesforAI()* functionality that was offered by the bank. The function parameters required by the calibration method are: the reference date and the currency. Starting from them the cited function derives the required bootstrapped curves that are used for computing the target parameters. This function follows the same approach of the soft-calibration with only multiple initializations in order to avoid local-optimal results.

The main currencies that we have handled during these preliminary phase are: EURO, US Dollar (USD),British Pound (GBP), New Zealand Dollar (NZD), Canadian Dollar (CAD).

As accurately described at the last part of this chapter, original data were augmented by exploiting on a perturbation approach based on a static schema. In the table table 4.1 is represented the dataset composition from the currencies perspective.

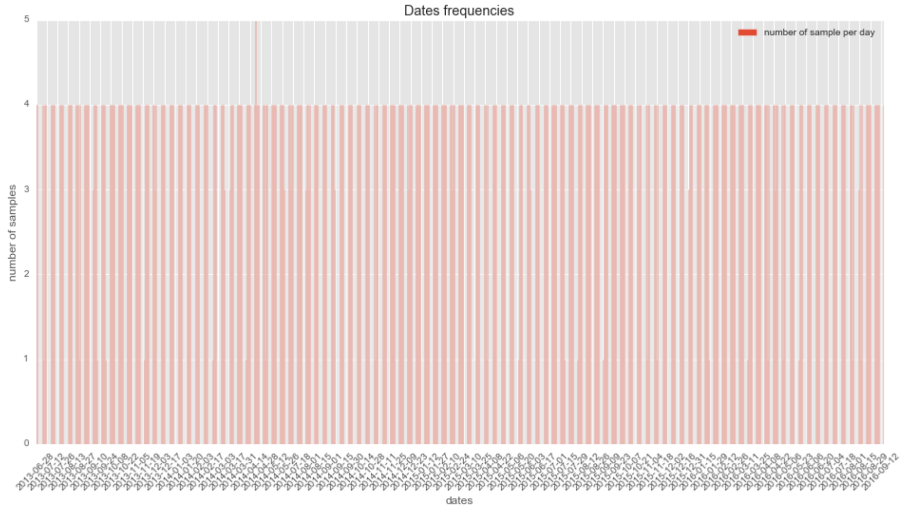


Figure 4.1: Sampling frequency for single currency EURO. In the x—axis there are the dates from June 2013 until September 2016 and in the y—axis the number of sample per day.

4.2 Data Exploration

To simplify the presentation, I will preliminary focus on the EURO currency since it is the one with most data and the one more relevant for the bank.

4.2.1 *Data abundance and samples analysis*

During these analyses I will rely only on the original data.
As observable in Figure 4.1 the data were recorded starting from August 2013 until September 2016. We have roughly 500 records for the year 2013, 900 records for years 2014 and 2015 and 700 records for 2016. This means that on average we have 80 records per months and 4 records per day. Not all the days were suitable for recording, e.g. data are not available on holidays, that is the cause for mathematical mismatch in the numbers.
In regards of data augmentation, perturbation were applied to each of the main features. The volatility matrix was perturbed with a 1% parallel shift (both positive and negative). The interest rate curve was perturbed in two different ways. First it was subdivided into three time buckets: from present time to 5 years, from 5 years to 15 years and from 15 years until the end of the series. Then shifts with an absolute value of 10 basis points were applied to each of these buckets. The results was an augmented dataset consisting of both original and per-

reference date	string that specifies the calibration date
currency	string that specifies the record currency
swaption tenors	list of swaption tenors as interval
swaption expiries	list of swaption expiries as interval
swaption rows	number of expiries
swaption columns	number of tenors
discount dates	list of dates for the discount curves
forward dates	list of dates for the forward curves
discount values	list of values of the discount curve ordered by dates
forward values	list of values of the forward curve ordered by dates

Table 4.3: dataset original feature names and descriptions.

turbed records. The lack of this perturbation procedure is the absence of an additional new information, in fact the described perturbation does not add new cases to learn from but just reproduce a noisy version of the original records.
For what concern the other currencies, we have analogous results for USD, CAD, GBP and as expected little less samples for NZD and Japanese Yen (JPY).
I present in Table 4.2 the first and the last samples' dates, in order to give a quick comprehension of the years under analysis
.

4.2.2 *Features exploration*

Let us focus on interest and discount curves. In the EURO scenario we have the OIS curve as discount and the Euribor curve with tenor six-months (EUR6M) as interest rate. As discussed in Chapter 2.4, they have a key role in pricing the instruments. In this section I will present a summary of the curve trends for the EURO scenario.
They consist in a time series of sixty points ordered by date, as a transformation we have converted this absolute date into delta-days. Delta-days are integer values that indicate how many days occur from the first point in the series until a specific one. By looking on Figures 4.2, 4.3 two considerations can be carried out. The first one is the high similarity between discount and forward curves shape, this will be really important and in particular have to be reminded during the reduction and the supervised regression analyses. The second aspect is the strange

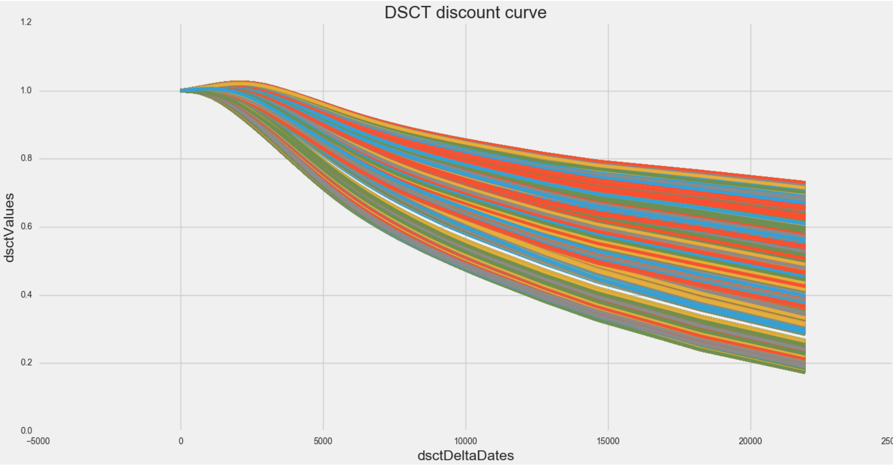


Figure 4.2: Discount curves plot over delta days for teh single currency dataset EURO.

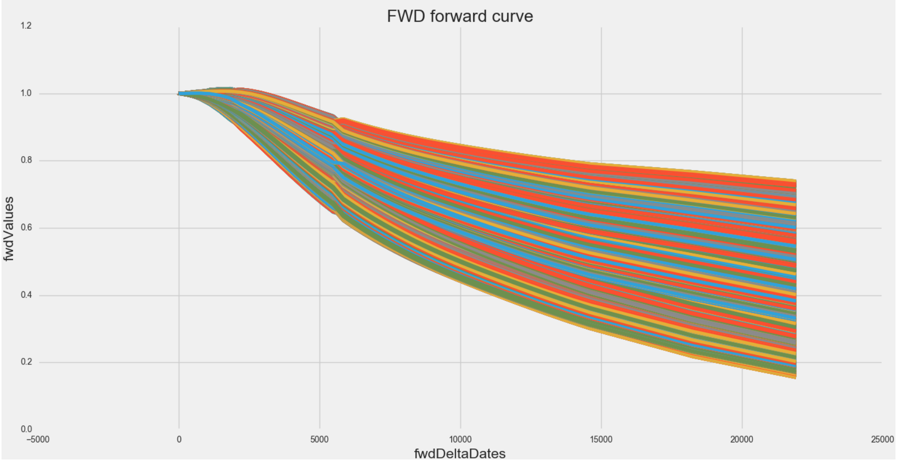


Figure 4.3: Forward curves plot over delta days for the single currency dataset EURO.

interweaving occurred only for the forward curves between the five thousand and six thousand delta-days. This phenomenon is highlighted in the Figure 4.4. We have deeply investigate this issues and no errors have been found, probably is caused by the generation model.

In regards to the log-normal shift variable, is a matrix which has the same shape of the matrix of weights (Section 2.5.1) for the instruments selection and its domain has two distinct values and it is shared to all the instrument available at the same reference date. This is just a convention adopted by our client and it is not formally defined in the financial literature. I present in Figure 4.5 a plot of its behaviour on the reference dates.

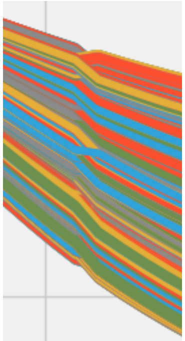


Figure 4.4: Forward interweaving in single currency dataset EURO.

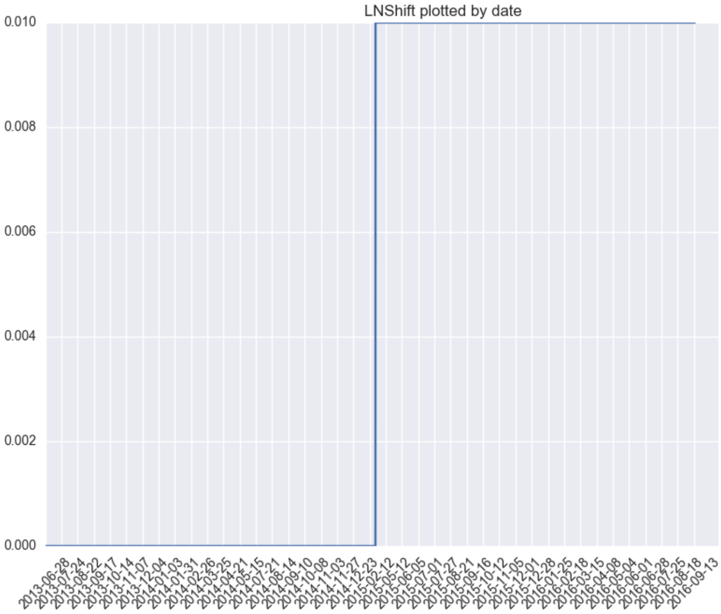


Figure 4.5: Reduced to scalar log-normal shift for single currency dataset EURO.

The last couple of features to be examined are the two-dimensional arrays of prices and volatilities. For the EURO dataset they are 14×17 matrices of float values. In order to get a quick feedback about matrix volatilities and prices variability I present in Figures 4.6, 4.7 the heatmaps of the correlations over the list of matrices for prices and volatilities over all the records. In this representation the repeated small squares within the heatmaps 4.8 corresponds to the single matrix of prices or volatilities referred to a single calibration date. The main feedback that can be derived from the single calibration volatility correlation matrix is that the correlation decreases with the increase of time distance of tenors or expiries, this can be translated into an uncertainty on long-time horizon. This phenomena can be seen by

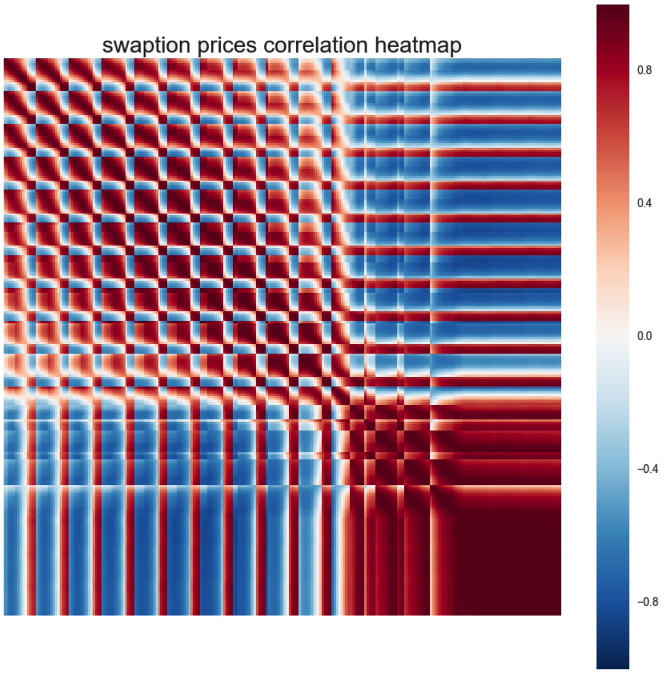


Figure 4.6: Prices correlation heatmap for dataset EURO. It represents the correlation between all the possible couples of prices expiry-maturity. The first row describes the correlation of the first expiry with respect to all the maturities.

focusing on the last fourth tenors at which the correlation become negative. This uncertainty causes high and constant values for the volatilities and a high variability on prices, these explain both the negative correlations for prices and zero correlations for volatilities. What can be further deduced given the just described results is an expected primal-dual relation between prices and volatilities. When volatilities are constant and high prices have much variation, conversely when volatilities are low they vary a lot causing prices variation within the same range of values.

4.3 DIMENSIONALITY REDUCTION

Given the high dimensional space, that consists of the previous describe features and the low amount of original data that we have, we decided to apply two strategies before any learning task: reducing the dimensional space, augmenting the number of records. I will now present the procedures involved in the construction of the reduced features space.

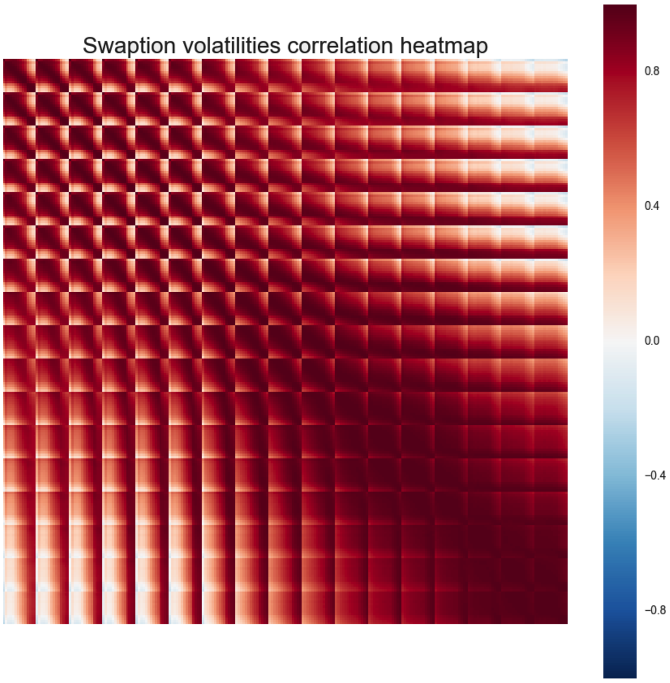


Figure 4.7: Volatilities correlation heatmap for dataset EURO. It represents the correlation between all the possible couples of volatilities expiry-maturity. The first row describes the correlation of the first expiry with respect to all the maturities.

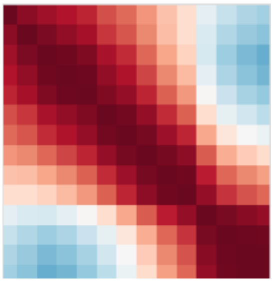


Figure 4.8: Volatilities correlation heatmap frame. It refers to the contracts identified by the first expiry with respect to all the tenors. In particular it computed by referring on the first calibration date.

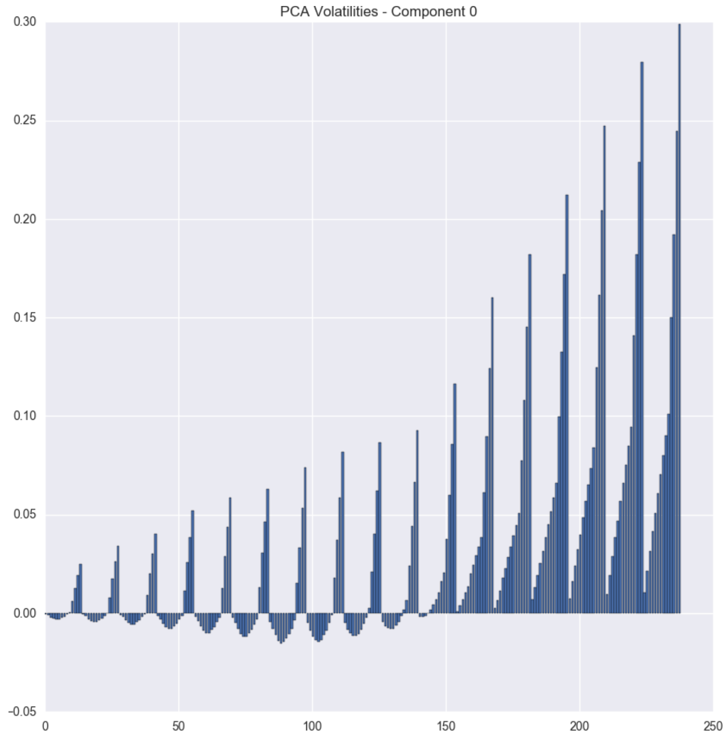


Figure 4.9: Price eigenvector composition of first pca component for single currency dataset EURO.

4.3.1 *Features construction*

Recall that the dataset contains the following features: forward curves, discount curves, matrices of prices, matrices of volatilities, reference date and log-normal shift. This space of features was too high, and was not uniform in particular there are curves with different dimensions. For these reasons we have built up a new reduced and uniform space of features starting from the original one.

MATRIX OF PRICES AND VOLATILITIES: PCA. As already mentioned, each record contains these matrix referred to its respective day. In order to reduce it with the minimum loss of information, we have exploit the PCA procedure. In particular the objective was to explain 99% of the variance with a new lower dimensional space. This procedure was called over the $m \times k$ matrix, where m are the number of samples and k the number of cells of the single matrix. In this way was possible to analyse all the correlations between all the samples' matrices contained in the dataset. In regards to prices the 99% of variance is represented by four new features. What it can be

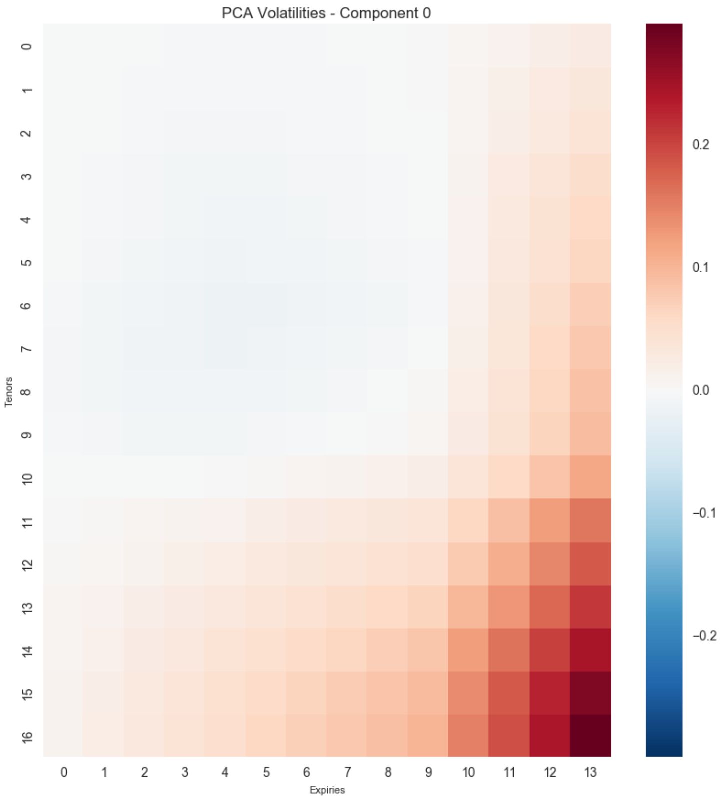


Figure 4.10: Price eigenvector correlation heatmap of first pca component for single currency dataset EURO.

noticed from Figures 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, is that as expected the component are orthogonal. Moreover, it can be noticed what was pointed out in the previous chapter, thus, the first component collects the most variable part that is given by the temporary most far instruments,thus the one with highest maturities. The same holds for all the other components respecting the orthogonality property. In this way, as observable in Figures 4.15, 4.16 of the eigenvector composition, the last component captures the variance implied by the closest contracts. The same behaviour can be noticed by referring to the matrices of volatilities and leveraging on the duality relationship between the two variables. In this second case the same amount of variance is described by six components. I will refer just to the first Figures 4.17, 4.18 and to the last Figures 4.19, 4.20 in order to let you able to see the same behaviour and the duality relationship. In particular you should notice the component composition, in fact the first component is described by the instruments which expiries are at most two months, as opposite to the last component that is described by temporally distant instruments.

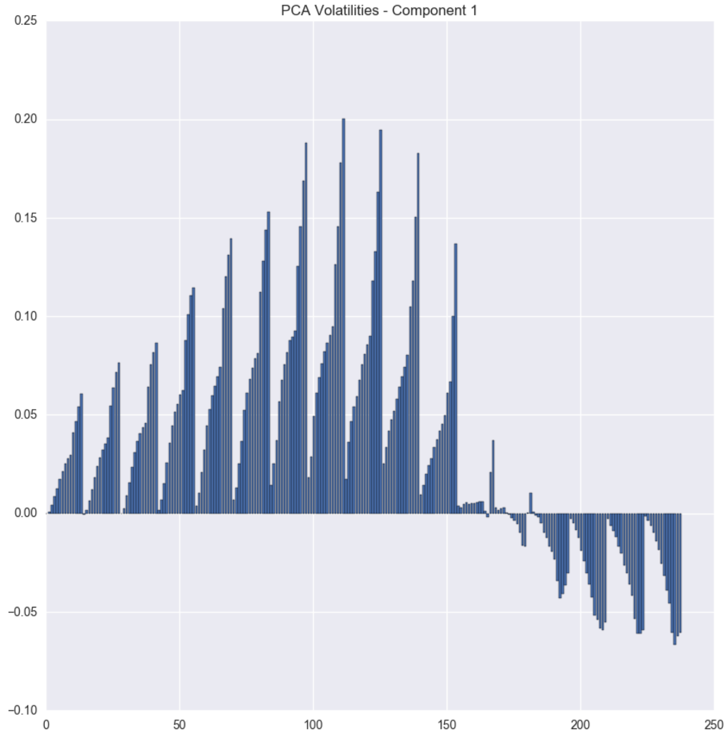


Figure 4.11: Price eigenvector composition of second pca component for single currency dataset EURO.

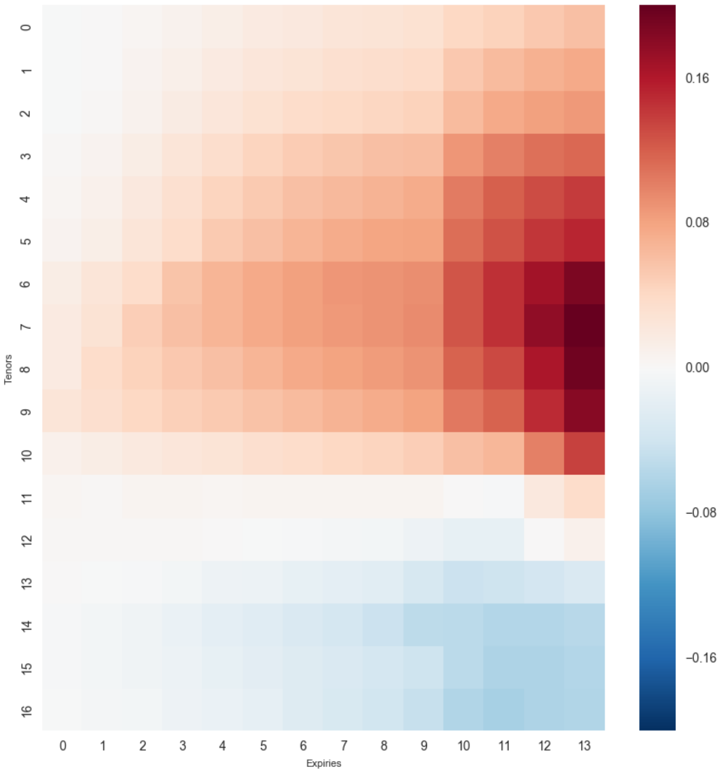


Figure 4.12: Price eigenvector correlation heatmap of second pca component for single currency dataset EURO.

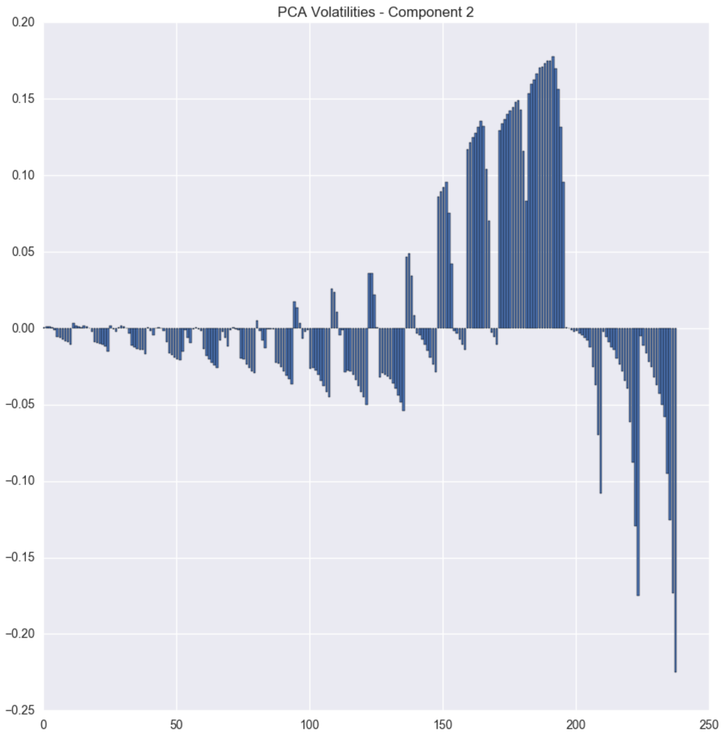


Figure 4.13: Price eigenvector composition of third pca component for single currency dataset EURO.

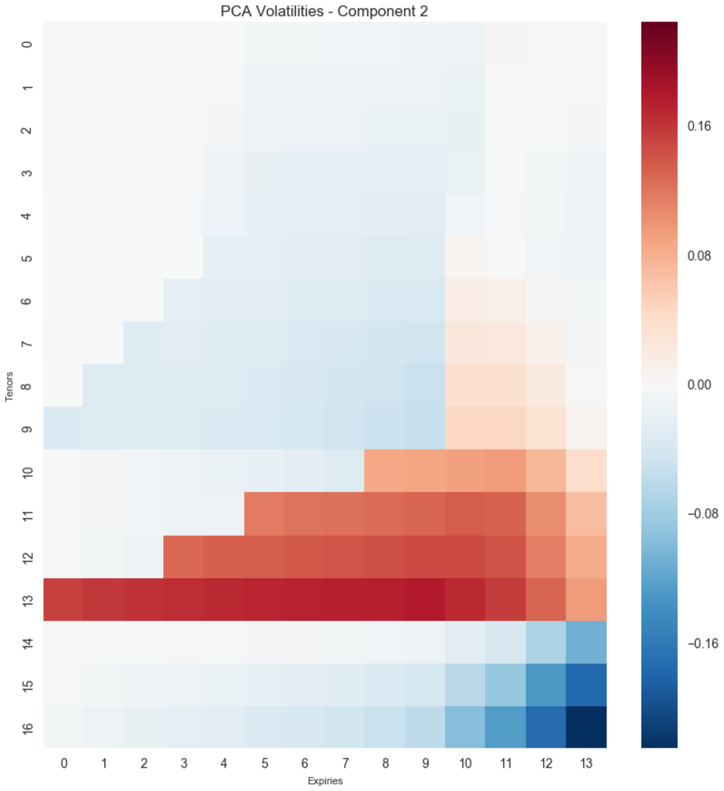


Figure 4.14: Price eigenvector correlation heatmap of third pca component for single currency dataset EURO.

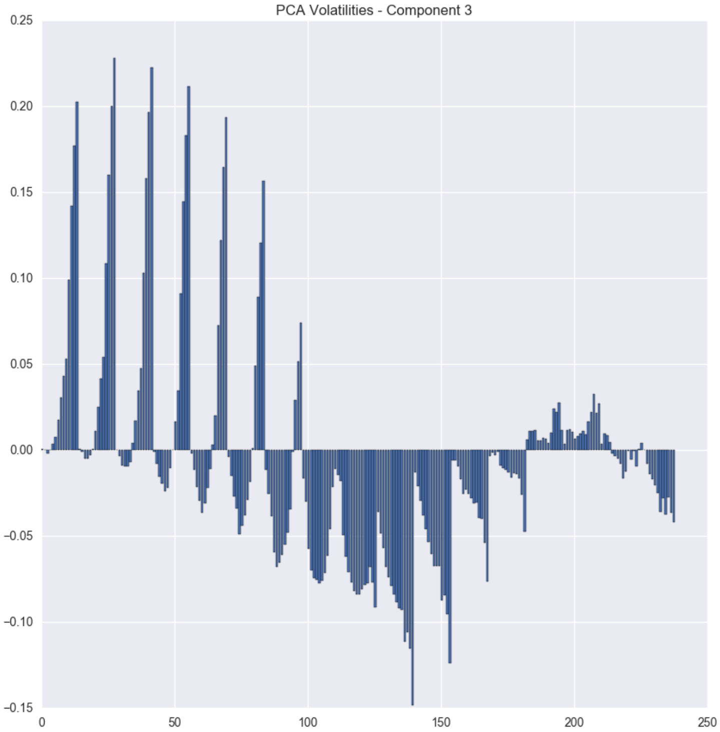


Figure 4.15: Price eigenvector composition of fourth pca component for dataset EURO.

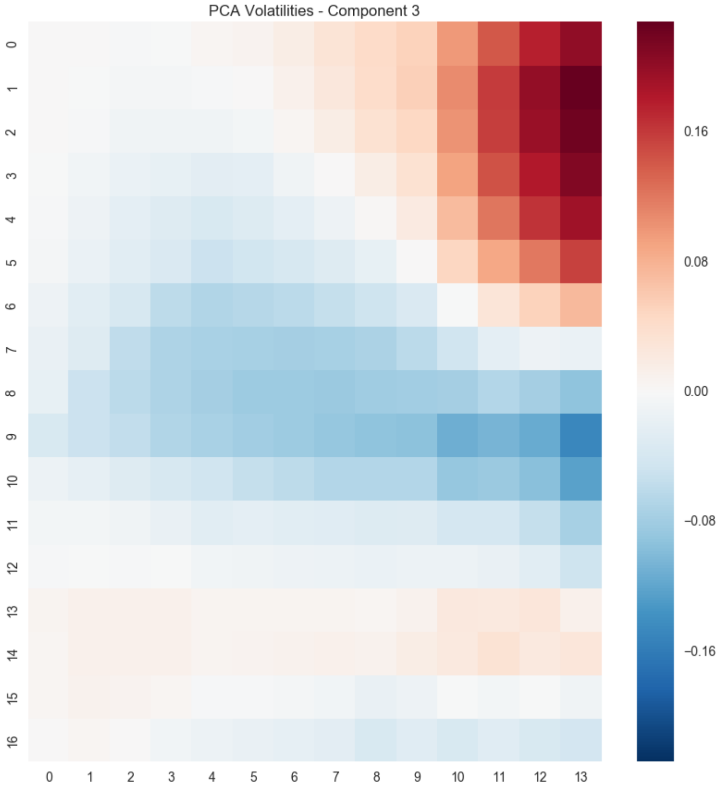


Figure 4.16: Price eigenvector correlation heatmap of fourth pca component for dataset EURO.

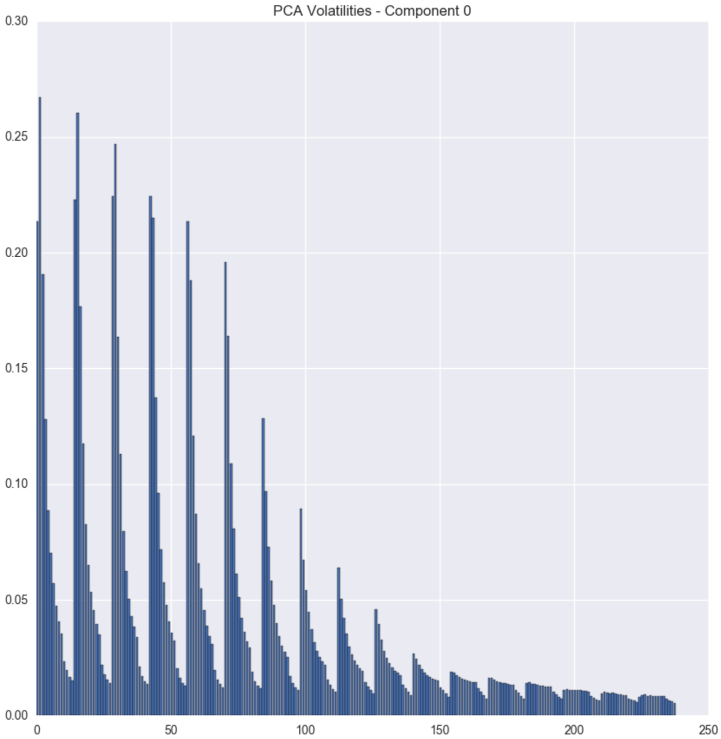


Figure 4.17: Volatility eigenvector correlation heatmap of first pca component for dataset EURO.

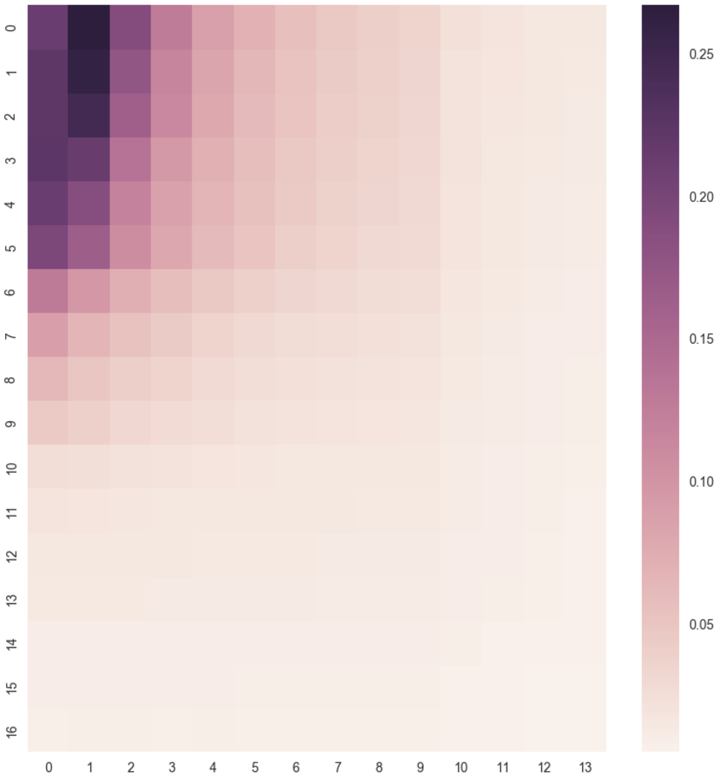


Figure 4.18: Volatility eigenvector correlation heatmap of first pca component for single currency dataset EURO.

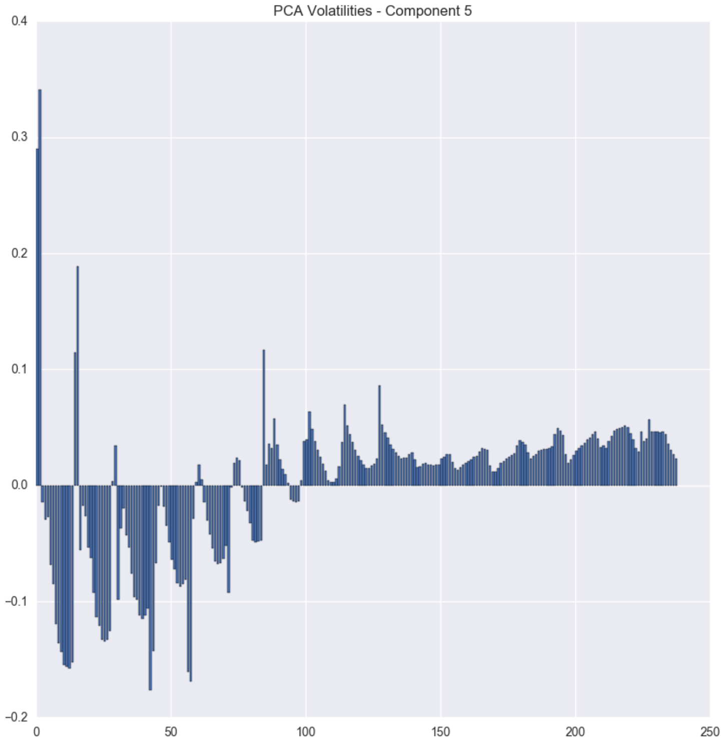


Figure 4.19: Volatility eigenvector correlation heatmap of sixth pca component for single currency dataset EURO.

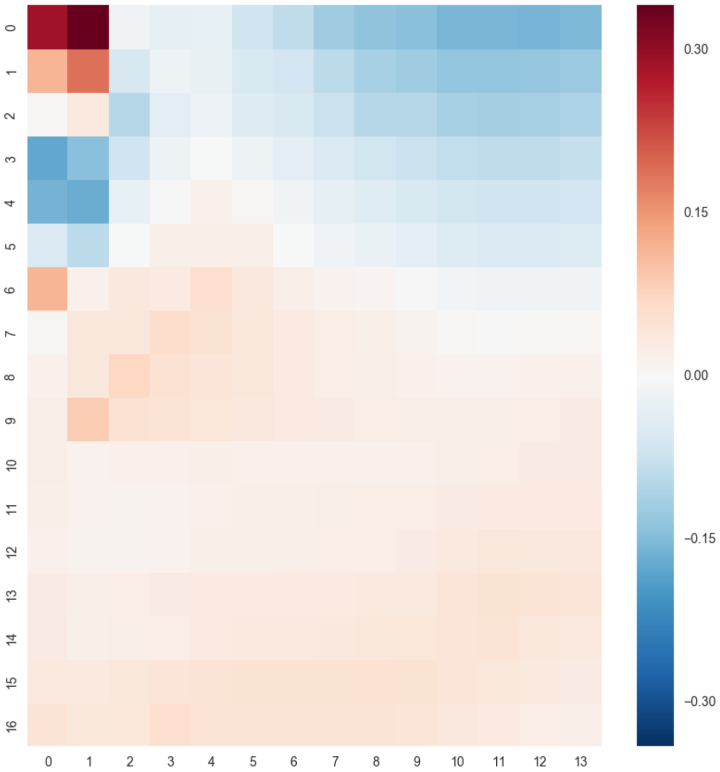


Figure 4.20: Volatility eigenvector correlation heatmap of sixth pca component for single currency dataset EURO.

LOG-NORMAL SHIFT: REDUCTION TO SCALAR. A brief comment should be open here for the other matrix attribute, thus the log-normal shift. As discussed in the previous section and shown in Figure 4.5, by convention it has a value shared by all matrix elements, and a domain of only two possible values. For this reason we have reduced it to a scalar value, which values is the same of the matrix elements. With this last shortcut we have reduced the number of features without loss of information.
By concluding I want to highlight that after the PCA reduction, our dataset is composed by this new list of features: PCA prices (4 components), PCA volatilities (6 components), log-normal shift(1 component), reference date, discount and forward curves (120 components)

INTEREST RATES AND DISCOUNT CURVES: NELSON-SIEGEL. The last couple of features that we have reduced are the interest rates and the discount curves. Concerning the interest rates curves, we have exploited a parsimonious model called Nelson Siegel(Charles R. Nelson, October 1987). Given the apparently correlation assumption discussed in the previous section, and the similarity in their shapes we have adopted this model also for the discount curves, even if it was not originally developed for discount curves. I will now formally present the model, and then I will show the results given by the model fitting over the curves. The Nelson-Siegel model is a sum of exponential terms:

$$y(t) = a + b \times (\frac{1 - \exp^{\phi(t)}}{\phi(t)}) + c \times (\frac{1 - \exp^{\phi(t)}}{\phi(t)} - \exp^{-\phi(t)}) \quad (4.1)$$

where,

$$\phi(t) = 1 + \frac{1}{d} \quad (4.2)$$

And a,b,c and d are the parameter to be learnt.
Given this model, we have fit it over each curve in the dataset. After the curve-fitting process, all the original curves were substituted by the four Nelson-Siegel parameters. I present in Figures 4.21, 4.22 two examples of fitting, one per type of curve. After the described transformation, the new reduced dataset consists of the following features: PCA prices (4 components), PCA volatilities (6 components), log-normal shift (1 components), discount curves (4 components) and forward curves (4 components).

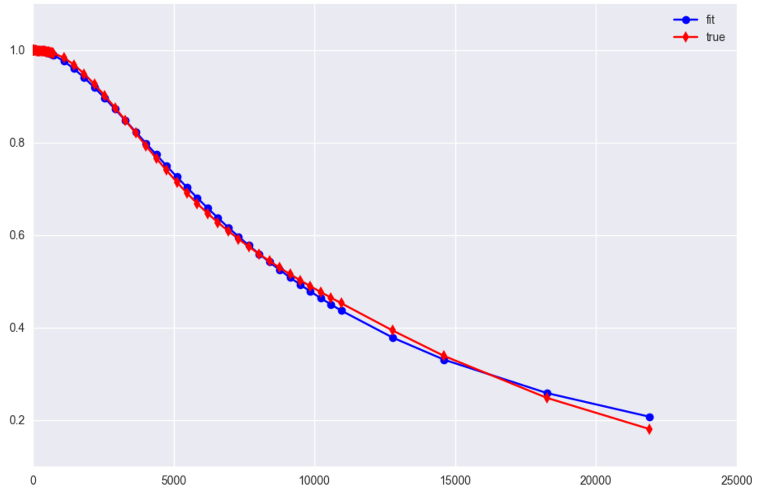


Figure 4.21: Nelson-Siegel fit for discount curve for a record within the single currency dataset EURO.

In order to let you be more confident with the reduced dataset, I conclude the feature construction section by presenting in Figure 4.23 a heatmap of the reduced parameters correlations. What can be observed in Figure 4.23 is that as expected, the discount and forward curve parameters are highly correlated. In particular, this holds for the first three Nelson-Siegel parameters, that appear to be the most important in the model definition. What is reasonable to expect starting from this consideration is that, during the supervised analysis not all the curves parameters will be selected at the same time. Another interesting aspect that I want to point out, is that the log-normal shift is correlated with both the curves and with the principal eigenvector fo the pca prices matrix. This could be derived from the fact that the curves with the log-normal shift are all the input necessary to compute the contract price. Finally, something similar can be noticed also for the second principal eigenvector of the pca volatilities matrix even if in this case there is less correlation.

Given these results over the features construction, I will now move to the features selection phase. In particular we will see two approaches one iterative and one based on a threshold.

4.3.2 *Features selection*

Also in this case we are dealing with an estimator that basically fits the data and assigns to each feature a score. This specifies the feature importance in predicting the target variables

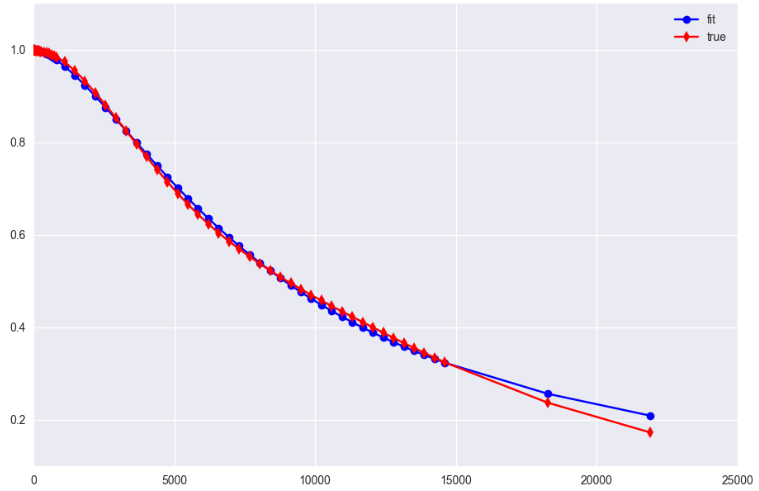


Figure 4.22: Nelson-Siegel fit for interest rate curve for a record within the single currency dataset EURO.

depicted in Figures 4.26, 4.27. Also in this case we restrict our models to the ones able to perform a multivariate fit in order to exploit the correlation between the targets as displayed in Figure 4.25. Given the list of scores associated with all the features we will select only the ones whose score is above the average. The results are displayed in Figure 4.24, where the y-axis is the dependent from the regressor method used in the supervision. In this example the selected features are the second and fourth NS parameter of the forward curve and the first for the discount curve. Moreover, there were selected the first and the third PCA component for the price matrix and finally the third PCA component for the volatility one.

The second algorithm that we have used is IFS. Even in this case we have made the fitting phase in a multivariate approach since we are dealing with two targets and we want to exploit their correlation. The results are described in Table 4.4 and Table 4.5; they are ordered by iteration. During the first iteration the first feature selected was the fourth component of the pca volatility matrix. In the second iteration the selected feature is the first principal component of the pca matrix of volatilities. Another main aspect that can be noticed from both iterations is the relation implied by the Black's formula, in fact no prices and volatilities are selected in the same time since they explain the same concept. Further more, we can notice that in the first iteration they are far in the ranking, the same holds with respect to the log-normal shift as can be observed in the second iteration, see Table 4.5.

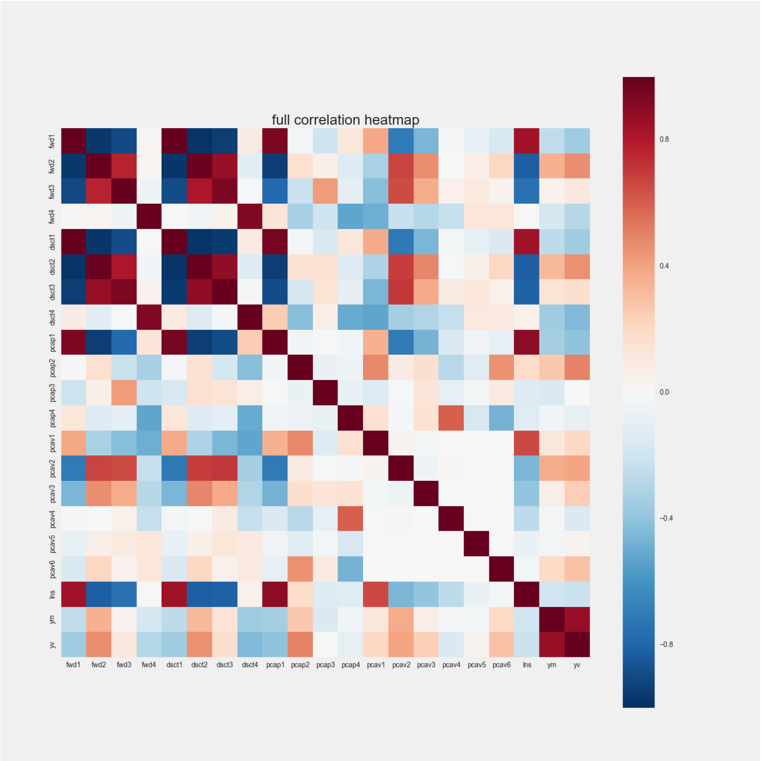


Figure 4.23: Heatmap of the correlation of the reduced features for single currency dataset EURO.

These aspects can be also easily noticed in Figure 4.23. Here we observe an high correlation between the target parameters, also their correlations with respect to the input parameters are similar. In particular, they have a low and positive correlation with the PCA component of the volatility matrix. They also have either a positive or an inverse correlation with the NS parameters of the discount and forward curves.

4.4 ADDITIONAL FEATURES EXPLORATION

Even if now we are more confident on the features, I want to point out an important aspect from a different perspective. So far we have not look on the parameters behaviour according to the presence or absence of trends. In particular this aspect has a key role in deciding which approach to apply between offline and online learning.
What we have done was a plot of the reduced features with respect to records calibration dates, I will not provide here all the feature plots, but I will display the most interesting one.

First Iteration			
Rank	Index	Score	Feature Name
#0	15	0.271339	4th component PCA-volatilities
#1	14	0.195478	3rd component PCA-volatilities
#2	12	0.193172	1st component PCA-volatilities
#3	17	0.165025	6th component PCA-volatilities
#4	5	0.033712	1st discount NS
#5	1	0.021080	1st forward NS
#..	...	...	...
#18	11	0.002112	4th component PCA-prices

Table 4.4: First iteration of IFS algorithm over the single currency dataset EURO.

Second Iteration			
Rank	Index	Score	Feature Name
#0	12	0.215391	1nd component PCA-volatilities
#1	14	0.210749	3rd component PCA-volatilities
#2	15	0.202462	4ht component PCA-volatilities
#3	17	0.175555	6th component PCA-volatilities
#4	5	0.027314	1st discount NS
#5	13	0.019949	2nd component PCA-volatilities
#..	...	...	...
#18	18	0.002581	log-normal shift

Table 4.5: Second iteration of IFS algorithm over the dataset EURO.

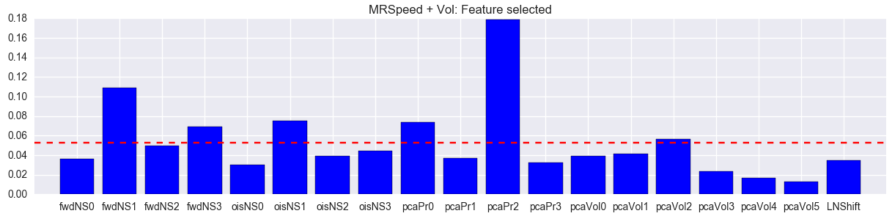


Figure 4.24: Feature scores wrt the average, for dataset single currency dataset EURO. In this case the selected features are: second and fourth NS parameter for the forward curve; the second for the discount curve; the third component for both the pca matrix of volatilities and prices.

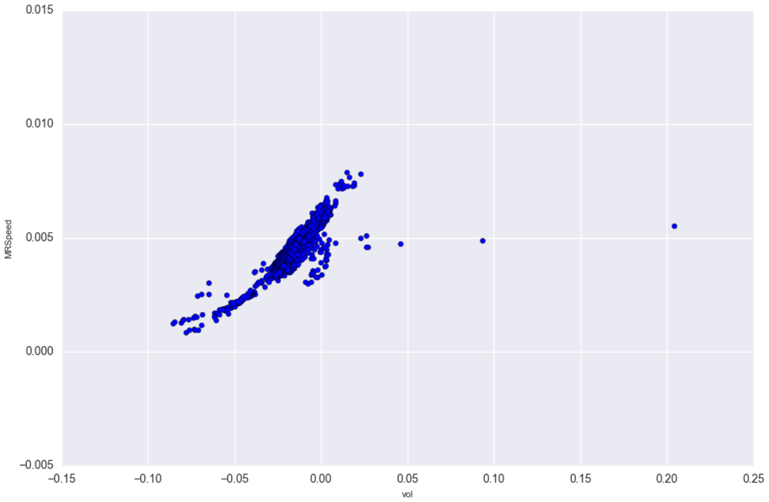


Figure 4.25: Scatter of the target parameters for single currency dataset EURO. On the x-axis there is the volatility parameter, on the y-axis the MRSpeed.

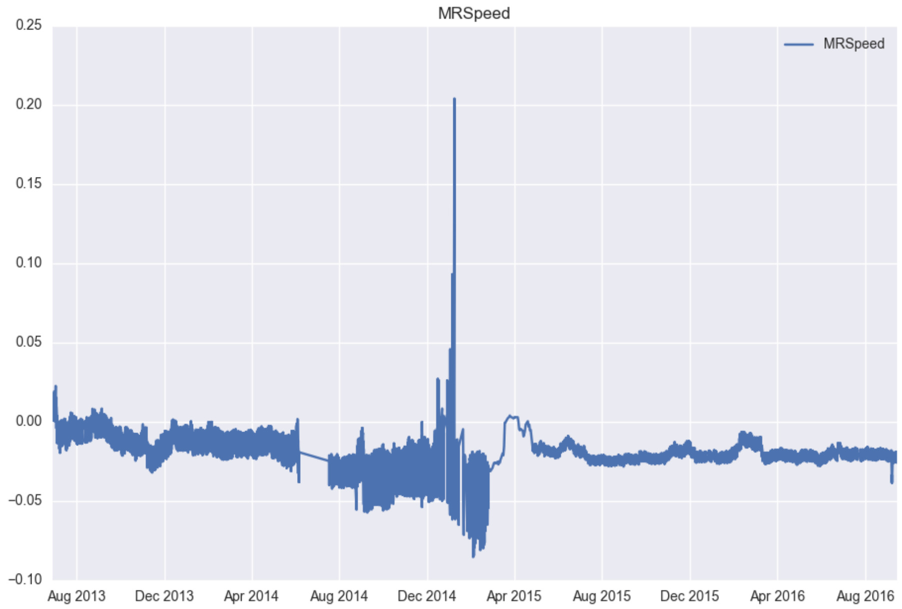


Figure 4.26: Vasicek MRSpeed plot ordered by calibration dates for single currency dataset EURO.

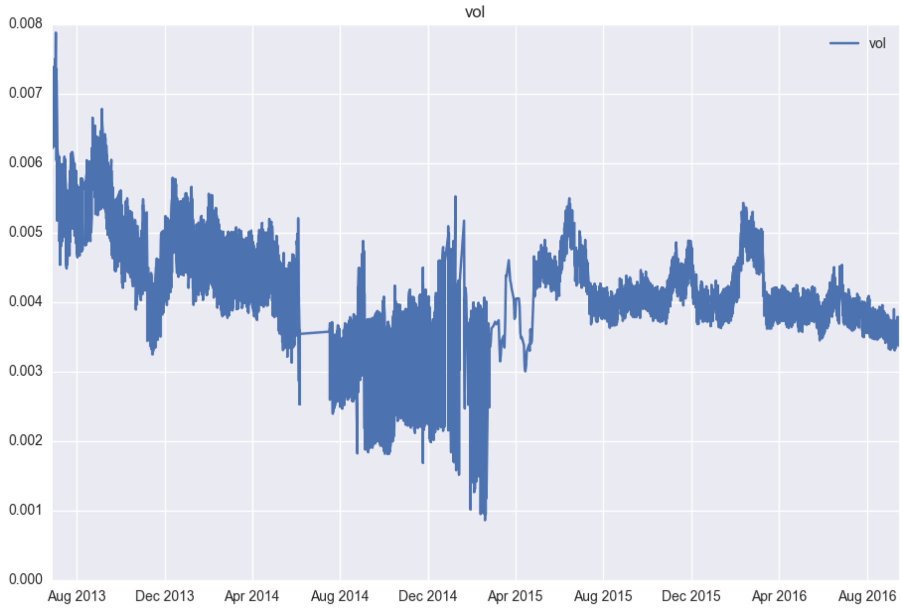


Figure 4.27: Vasicek volatility plot ordered by calibration dates for single currency dataset EURO.

4.4.1 *Curves: discount and forward*

From Figure 4.28 and Figure 4.29 we can notice that the process is not stationary, at least in our window of available data, we cannot ensure this for future or past data. Moreover, we have a further concrete proof of the intuition made in the previous section about the high correlation of the discount and forward curves. The main aspect that we have to notice is that the process is not stationary, from this assumption, neither a time series approach nor an offline approach are suitable for our specific problem. Moreover, we should think in using the already seen samples for predicting the trends behaviour, thus the future Nelson-Siegel parameter values for both the discount and forward curves.

4.4.2 *Matrices: prices and volatilities*

In regards to the pca matrices of prices and volatilities, I take advantage of this section for proving the previously discussed duality relationship between these variables. In order to do this, I will once more take advantage of the plots of the variables with respect to time. In particular, in Figures 4.30, 4.32, 4.31, 4.33 you should notice that the volatilities are stationary conversely the prices vary a lot. This occurs since by definition, volatilities denote the price variations. As a consequence when volatilities are high and constant, prices vary a lot, conversely when volatilities are small. There is no interest in analysing the variability of volatilities since it should be small, if it is not may mean that something strange or unseen it is happening on the market. The plotted couple of principal components express the duality relation among these parameters. They are a further confirmation of the results obtained during the analysis of the heatmap of the reduced features correlation for what concerns the pca parameters of both volatility and price matrices.

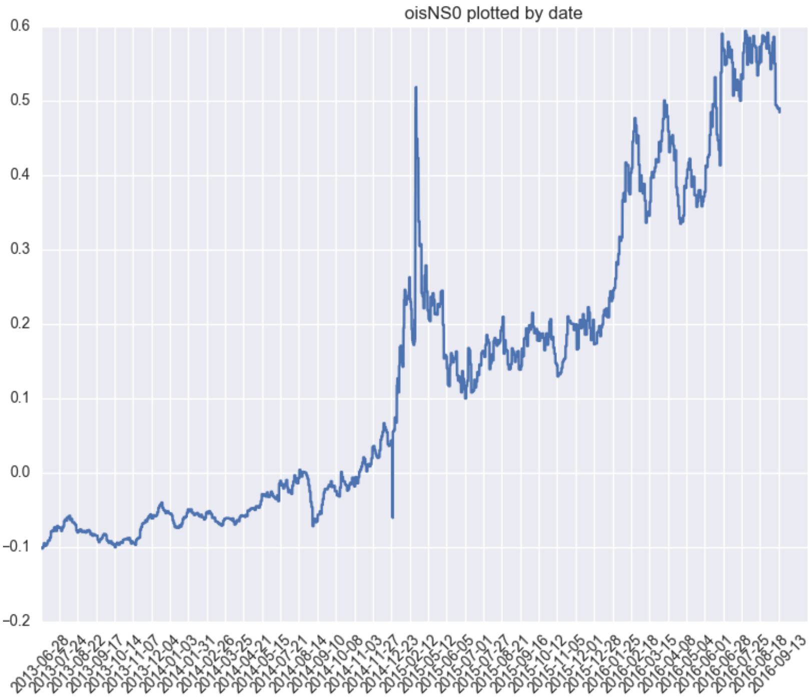


Figure 4.28: First NS parameter for discount curve plotted over calibration dates for single currency dataset EURO.

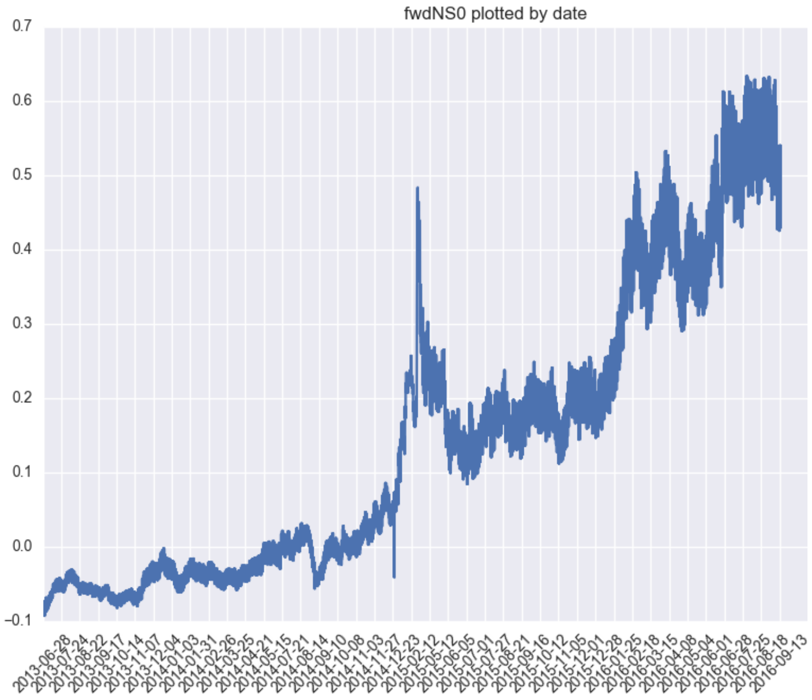


Figure 4.29: First NS parameter for forward curve plotted over calibration dates for single currency dataset EURO.

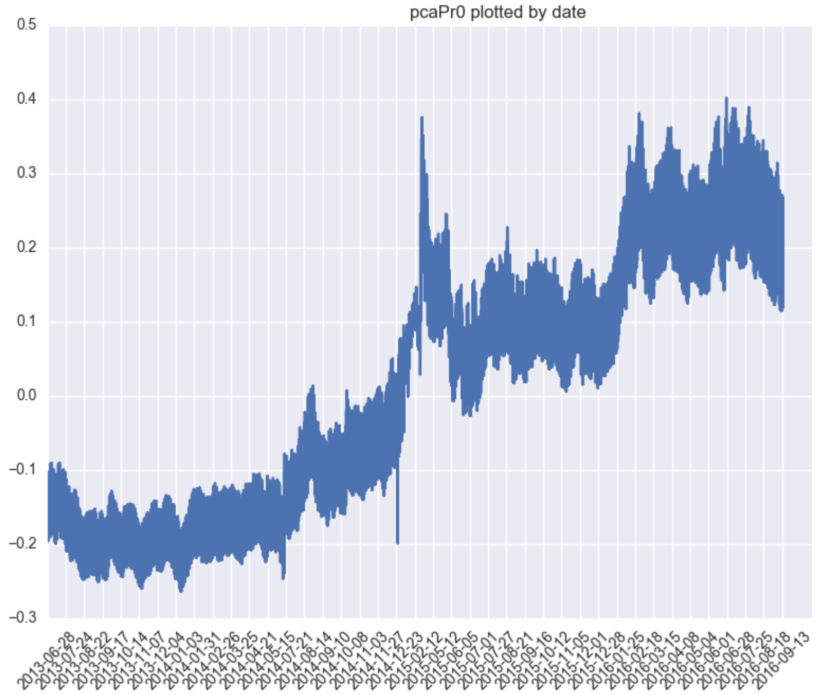


Figure 4.30: First PCA price parameter curve over calibration dates for single currency dataset EURO.

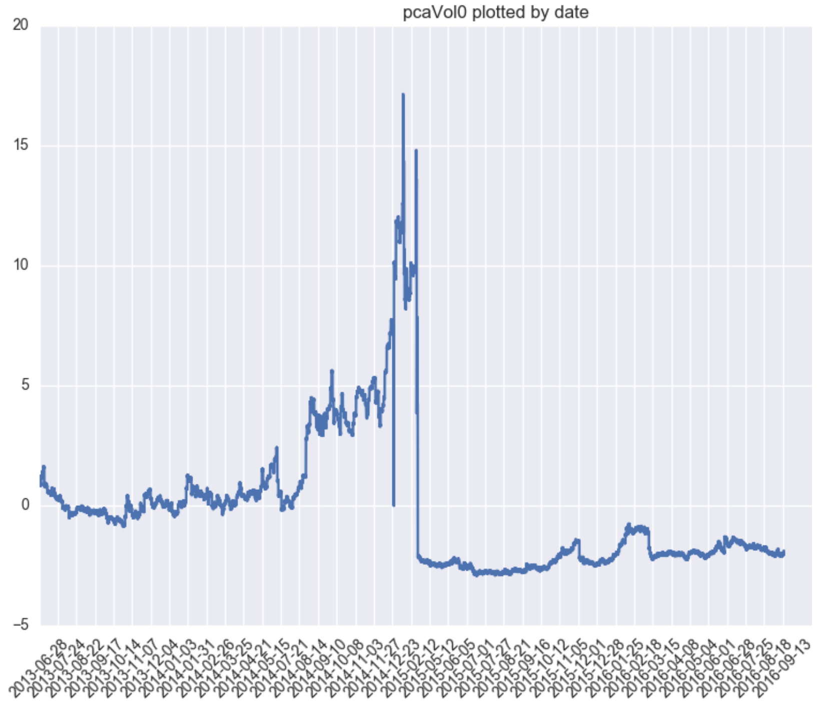


Figure 4.31: First PCA volatility parameter curve over calibration dates for single currency dataset EURO.

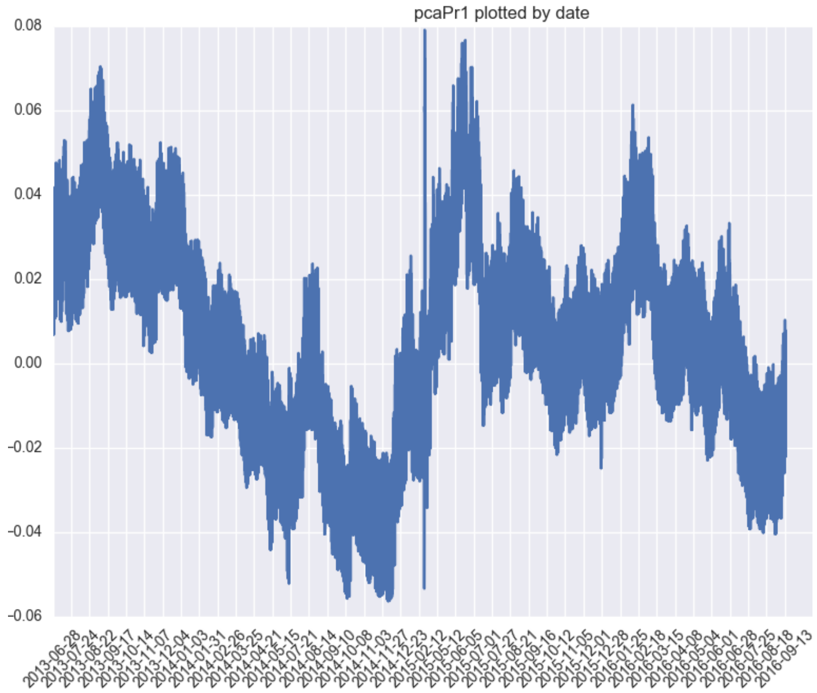


Figure 4.32: Second PCA price parameter curve over calibration dates for single currency dataset EURO.

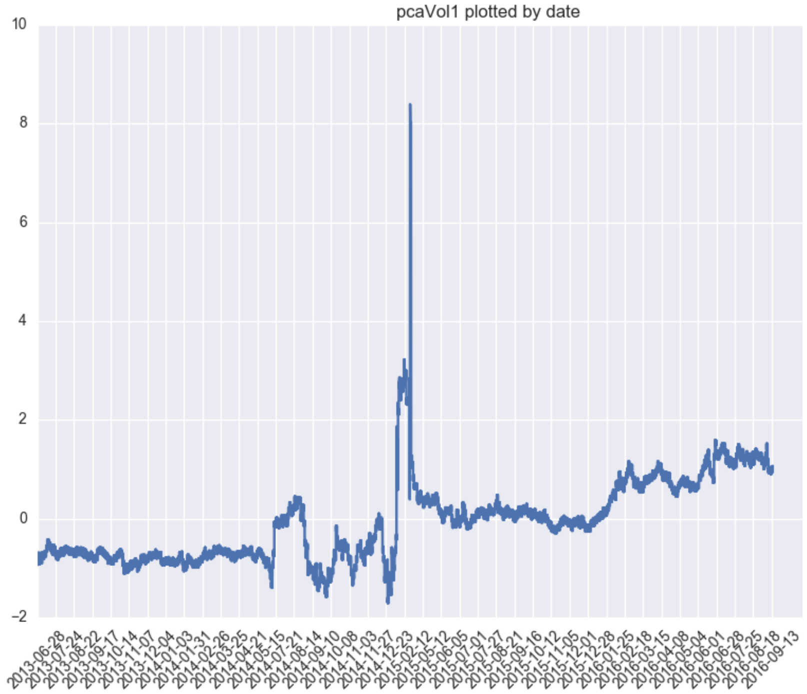
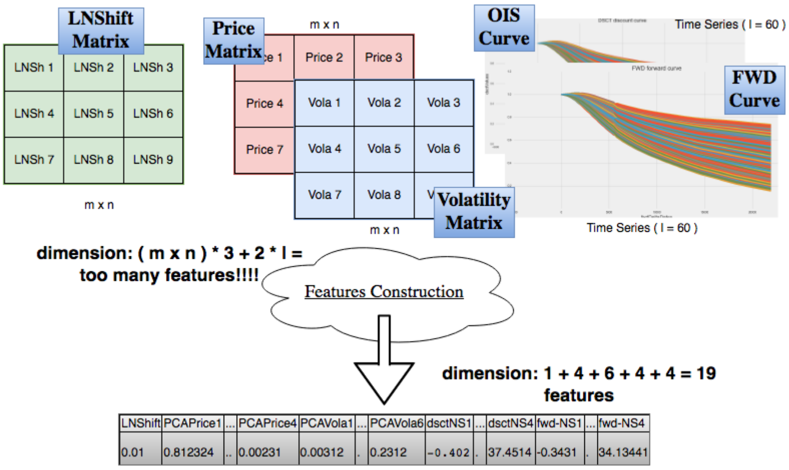


Figure 4.33: Second PCA volatility parameter curve over calibration dates for single currency dataset EURO.



LNShift	PCAPrice1	...	PCAPrice4	PCAVola1	...	PCAVola6	dsctNS1	...	dsctNS4	fwd-NS1	...	fwd-NS4
0.01	0.812324	..	0.00231	0.00312	.	0.2312	-0.402	.	37.4514	-0.3431	.	34.13441

Figure 4.34: Schema of all the features construction techniques starting from the original ones.

4.5 ONLINE LEARNER

Given the previously considered argumentations, we have decided to adopt an online approach. In particular with the objective of get more feedback possible on data, we have used both the original and the artificially generate samples. The online algorithm structure consists of: a batch training phase and an online test phase. Before entering in the details regarding each step, we have to define the used learner. This model is fully defined with an estimator used to fit the data and an estimator used to perform Supervised Features Selection (SFS). Our experiments were computed by using a Decision Tree Regressor as model to fit the data. It is important to state that this developed learner requires as input the already Nelson–Siegel fitted dataset, that is the reason why this phase does not occur during the training stage; the final features space construction is displayed in Figure 4.34. Moreover, according to the theory presented in Section 3.1 we will speak about: train, validation and test datasets. They are disjoint portion of data that define the knowledge respectively used by: train, validation and test stages. It is important to remark that during our experiments we used outliers cleaned data.

TRAINING: This first stage follows a *batch* approach, thus, it works on the whole dataset to perform learning. In particular here is only computed the fitting procedure over the training set of data, no predictions are made.

Set name	number of record	computational time [s]
Train	1186	18.5659
Validation	891	184.1214
Test	890	270.8483

Table 4.6: In the first column is shown the dataset division between train-validation-test sets. In the last the computational time required expressed in seconds. All the information refer for the single currency dataset EURO.

VALIDATION: This middle stage is implemented according to the online pattern. Its objective is to tune the model hyper-parameters (e.g. the number of layers of a MLP).

TEST: This final part follows an online approach. The idea is that each new sample contributes in updating the model parameters.

In our specific case, according to empirical results, we have noticed that there is no needed of a validation phase, no further contributions are given from it. In particular, we have verified that in this problem it is not necessary to constrain the decision tree but we can let it to over fit when needed. Given these argumentations, the validation phase to which I will refer to in the following experiments, is an intermediate light online stage.

4.5.1 *Original data only*

In this first version of our online model, we have exploited only the original records of the single currency dataset EURO. The original dataset was divided into: train, validation and test as specified in the Table 4.6. The dataset division is time-dependent. In particular all the data in the train set precede the records in the validation set, the same holds between validation and test sets. According to the general results presented in Section 4.3.2, the selected features are:

- second parameter ois NS fitted;
- sixth component pca matrix of prices;
- third component pca matrix of volatilities;

metric	MRSpeed	Volatility
Train phase		
RMSE	1.0156 e^{-7}	7.6225 e^{-9}
RMSE [/range]	1.2703 e^{-6}	1.2570 e^{-6}
R2	1.	1.
Validation phase		
RMSE	0.0074	00003
RMSE [/range]	0.0258	0.0765
R2	0.7704	0.7687
Test phase		
RMSE	0.0025	0.0002
RMSE[/range]	0.0774	0.1044
R2	0.6187	0.7053

Table 4.7: In the first column is shown the adopted error metric. In the second the errors referring to the MRSpeed variable. In the last the errors referring to the volatility variable. All the information refer for the single currency dataset EURO.

The results are displayed in the Table 4.7. In particular the notation [/$range$] means that the error was divided by its range of values in order to have a more useful information. We may get a similar result by computing the relative error, but in this case there may occur divisions by zero. What we can state by looking on the errors is that the chosen model structure seems to be appropriate. During the train phase the model fully overfits the data as can stated by the R2 metric that is equal to one for both the targets. The error is still low in both the validation and the test phases, this can be observed by focusing on the percentage error that was roughly equals to 2.58% for the MRSpeed and 7.65% for the Volatility during the validation phase. The same holds in the test phase, where the target errors are respectively 7.74% and 10.44%. In order to get a more suitable idea on the behaviour of our model, we have computed some plots that show the tracking capability. Firstly we can observe in Figure 4.35 the selected features behaviour during the train phase. The first observation we can make is that the process is not stationary and there are some trends. In order to get a clear vision I plot in Figure 4.36 the targets behaviour. From this window of observation it seems that the targets follow the

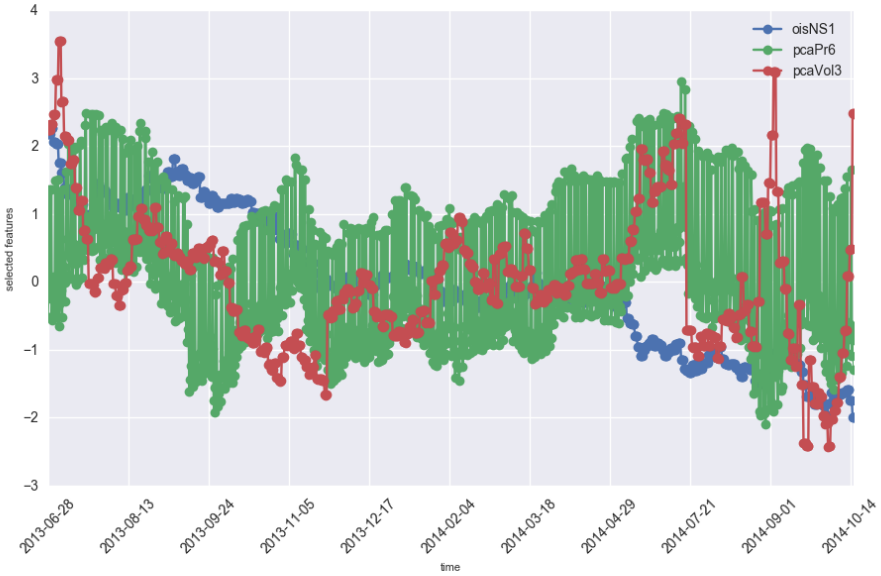


Figure 4.35: Online learner: selected features curves in train stage for single currency dataset EURO.

same behaviour, we may think in substitute the couple of targets by a single one in order to exploit univariate models. The point is that, as we will seen during the test phase analysis, this is not always true. As an empirical proof of the obtained results, we may notice how the curve peaks of the selected features closely reproduce the target curve shapes, in particular this aspect can be noticed by focusing on the third component of the pca volatility matrix. Given this results for the train stage, we can move to the images that represent the behaviour of the parameters during the test stage. I will start by presenting the just displayed figures in the train stage for the test phase. In Figure 4.37 we can observe the selected features behaviour for the test stage. What we can notice is that, differently from the train phase, here the features are more stationary and in particular for the sixth component of the pca matrix of prices, its curve has a lower amplitude with respect to the train window of observation. By looking on the Figure 4.38 that shows the targets behaviour, we can notice how, in particular starting from the points that corresponds to the end of June 2016, the two targets assume different behaviours and in general they are less overlapped than the train case. As last step, I will comment the tracking capability and the implied relative error results for both the target parameters. Starting from the volatility, in Figure 4.39 we can notice how our model predicts its values. In particular we can notice its ability in reproducing the original

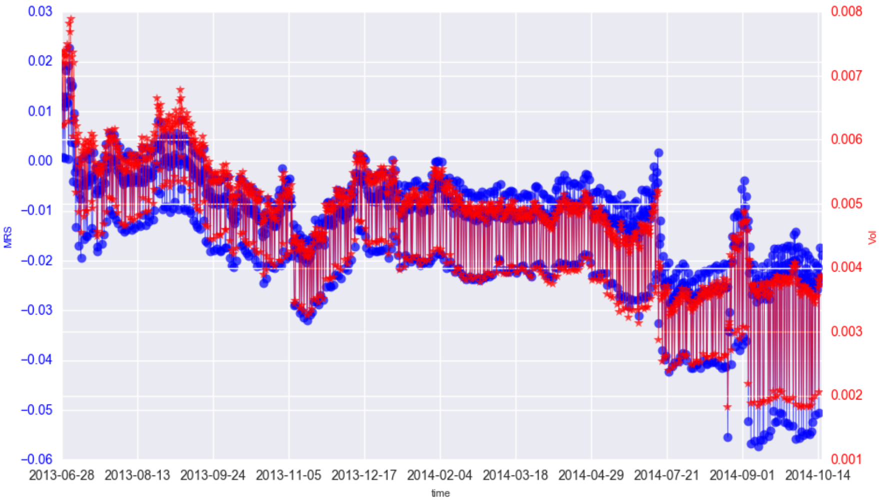


Figure 4.36: Online learner: targets behaviour in train stage for single currency dataset EURO.

curve shape starting from the first unseen sample until the last, this means that either the train and validation stages and the online updates, contribute in helping it catching the manifold. The volatility relative error is displayed in Figure 4.40 from which we can observe that it is almost always stationary on the zero value with some exceptions within a range of 10%. The same results hold for the second target parameters, thus the mean reversion speed, as it is displayed in Figures 4.41, 4.42. There are no big differences with respect to the volatility, the only one is given by the presence of a set of points that are totally mispredicted by our model, they can be seen easily since they are out of the manifold with a MRSpeed value equal to zero. Given this observation we can state that their contribution in the error outcomes is bigger than the contribution given by other test points. The MRSpeed relative error is within a range of 3% on average.

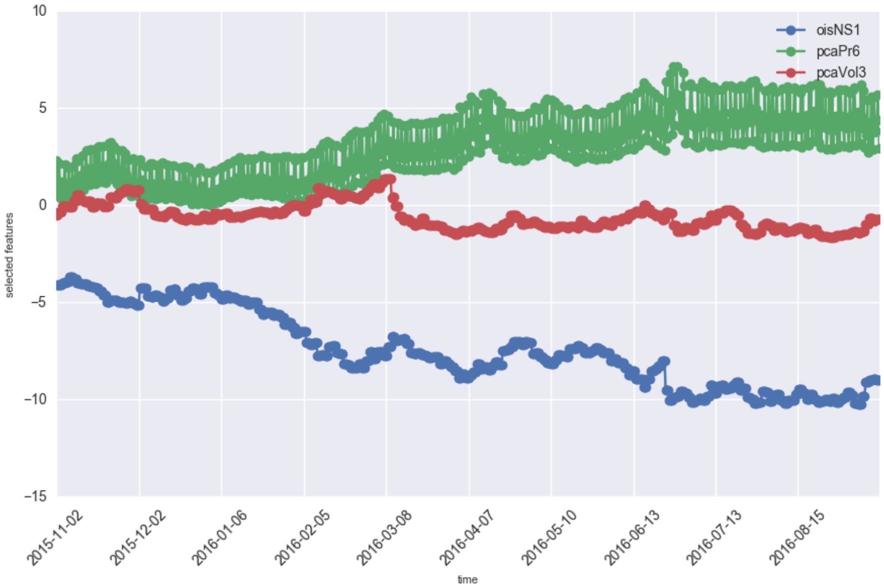


Figure 4.37: Online learner: features behaviour in test stage for single currency dataset EURO.

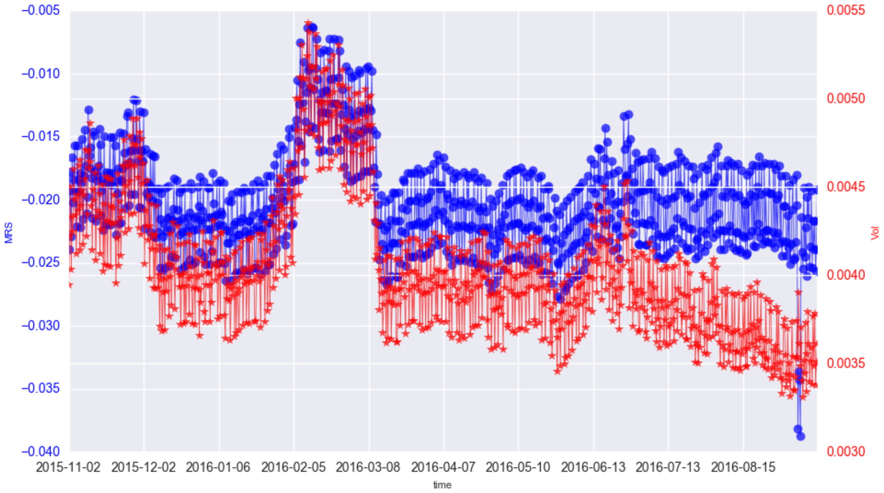


Figure 4.38: Online learner: targets curves in test stage for single currency dataset EURO.

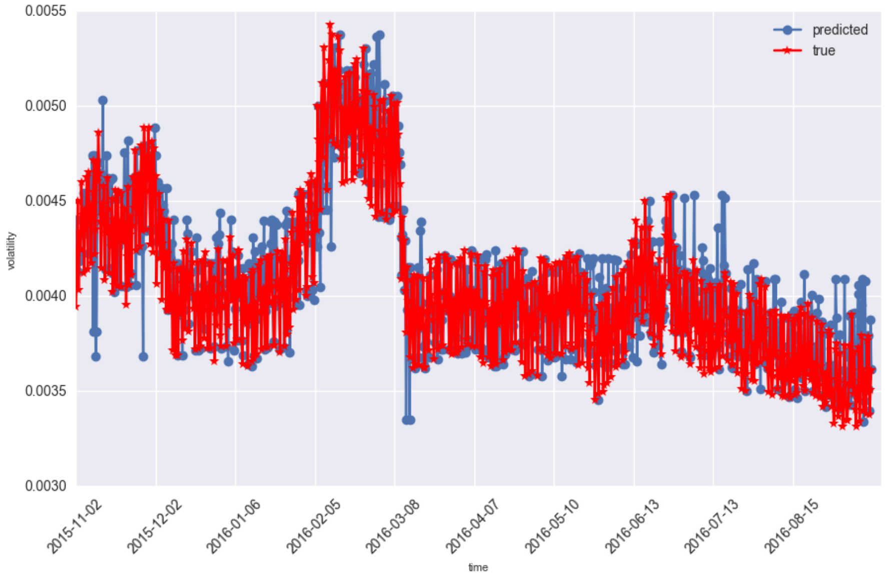


Figure 4.39: Online learner: Volatility prediction during test stage for single currency dataset EURO.

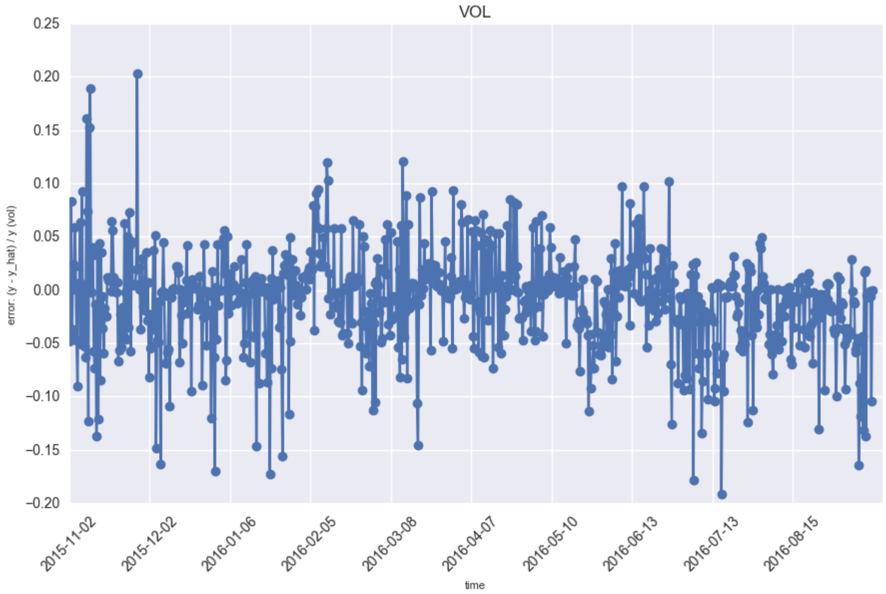


Figure 4.40: Online learner: Volatility relative error during test stage for single currency dataset EURO.

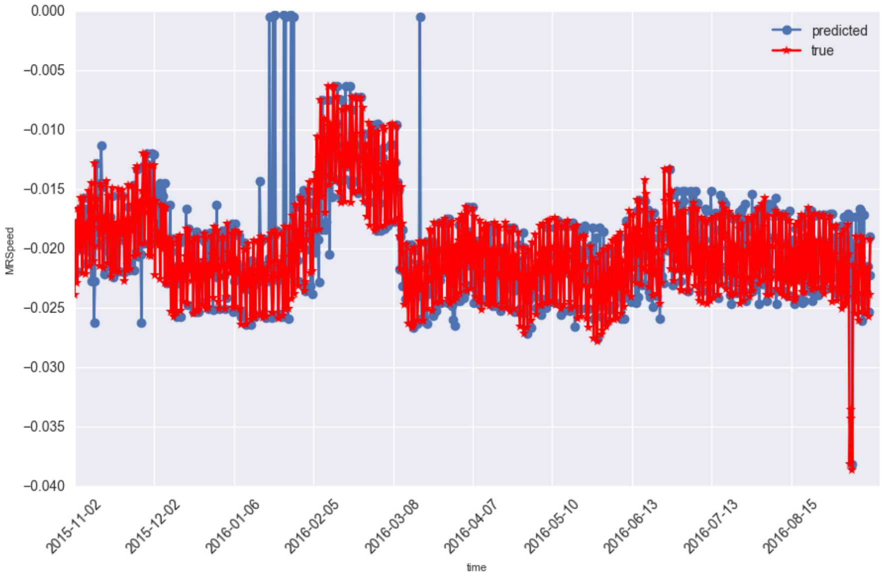


Figure 4.41: Online learner: MRSpeed prediction during test stage for single currency dataset EURO.

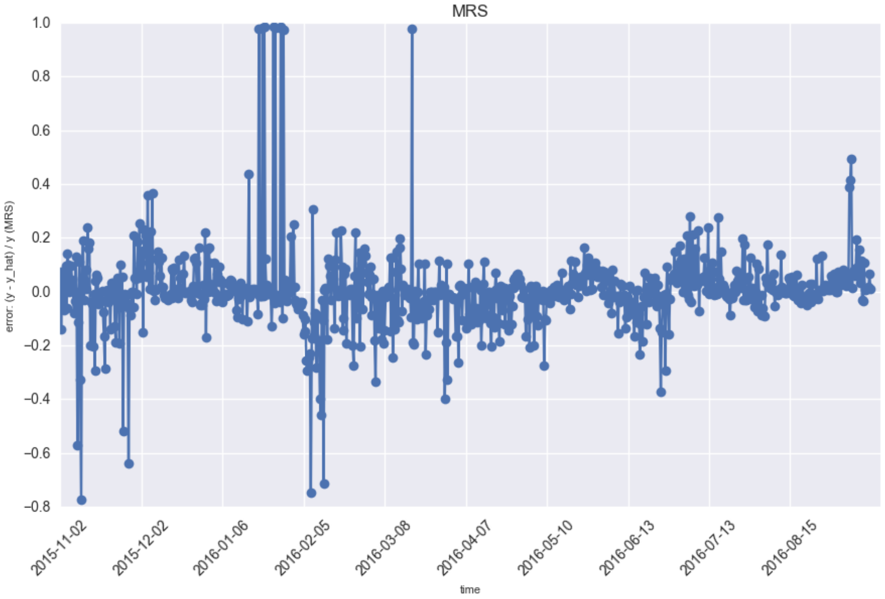


Figure 4.42: Online learner: MRSpeed relative error during test stage for single currency dataset EURO.

Set name	computational time [s]
Train	43.6210
Validation	1118.1204
Test	1635.9449

Table 4.8: In the first column is shown the considered phase between train-validation-test sets. In the second the computational time required expressed in seconds. All the information refer for the single currency dataset EURO.

After these results we have wondered if there would be some improvements in using also the perturbed data. For this second experiment, we have follow the same approach taken by the previous model; the only difference is that now for each predicted point in the test and validation phases we use as knowledge base to fit on, the full set of previous records, here both original and artificially generated are considered. In this second experiment, the obtained results are similar to the previous ones.

4.5.2 *Artificial contribute*

In this second version there is no a clear division of the dataset, in fact starting from the three sets described in Table 4.6 for each point in the validation and in the test datasets, the model is fitted on a mix of previous original and artificially generated record. This implies that each point as an own learning set. The computational time required by each stage is changed drastically and is represented in Table 4.8. Even in this case, we have used a Decision Tree Regressor as underlying model. The selected features are in agreement with the theory presented in Section 4.3.2, now they are:

- first parameter ois NS fitted;
- second parameter ois NS fitted;
- sixth component pca matrix of prices;
- third component pca matrix of volatilities;
- fourth component pca matrix of prices;

The performances are displayed in the Table 4.9. What we can state by looking on the obtained results is that: the new data

metric	MRSpeed	Volatility
Train phase		
RMSE	$1.1506\ e^{-7}$	$1.0663\ e^{-8}$
RMSE [/range]	$1.3068\ e^{-6}$	$1.4976\ e^{-6}$
R2	1.	1.
Validation phase		
RMSE	0.0159	0.0004
RMSE [/range]	0.0550	0.0901
R2	−0.0433	0.6792
Test phase		
RMSE	0.0021	0.0002
RMSE[/range]	0.0823	0.1222
R2	0.5692	0.5945

Table 4.9: In the first column is shown the adopted error metric. In the second the errors referring to the MRSpeed variable. In the last the errors referring to the volatility variable.

do not introduce further cases to learn from, so we can conclude that they just augment the computational time without any gain in terms of results. We analyse once more the capability of the model in tracking trends, I will no display the plots neither for the train nor for the validation set since there is not a single significant figure. From Figures 4.43, 4.44 we observe that there are additional selected features, this may occur given the augmented variability in particular in the matrix of volatilities and prices. The targets follow exactly the behaviour seen for the original records; also the results are approximately the same except for the computational time. For both the target parameters we can notice with Figures 4.47, 4.48, and Figure 4.45, 4.46, that in general we reach the same results of the previous case. There are still mispredictions in data that increase the error values, no advantages were leaded with respect to the previous scenario. As a conclusion, we can state that augmenting the number of samples lead a real benefit only if the new data contain new information. In our case this does not occur, for this reason we have discarded a dataset of 17598 perturbed records.

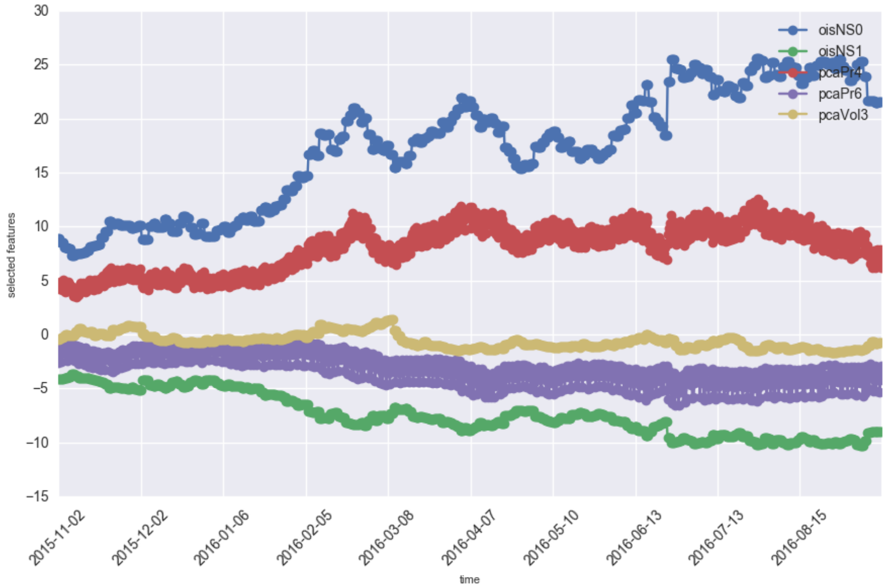


Figure 4.43: Online model with artificial data: features behaviour in test stage for single currency dataset EURO.

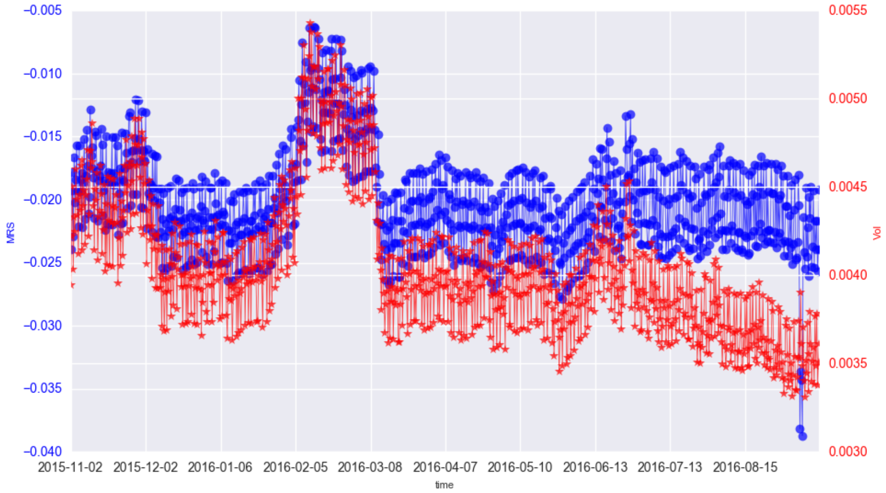


Figure 4.44: Online model with artificial data: targets curves in test stage for single currency dataset EURO.

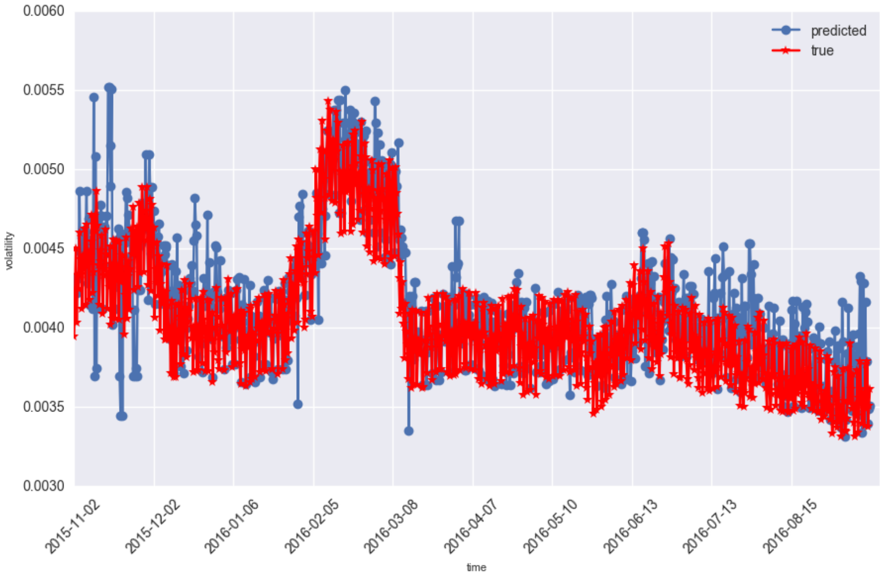


Figure 4.45: Online model with artificial data: Volatility prediction during test stage for single currency dataset EURO.

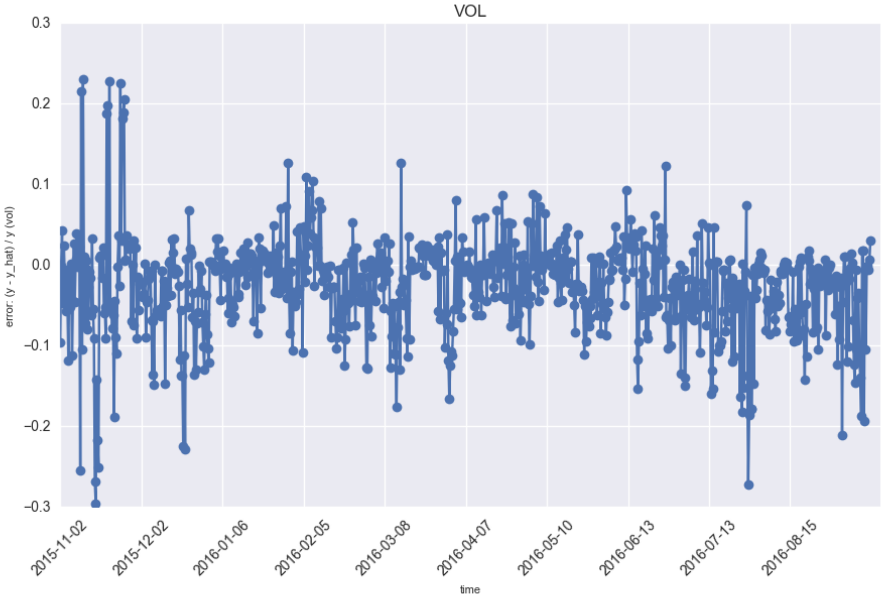


Figure 4.46: Online model with artificial data: Volatility relative error during test stage for single currency dataset EURO.

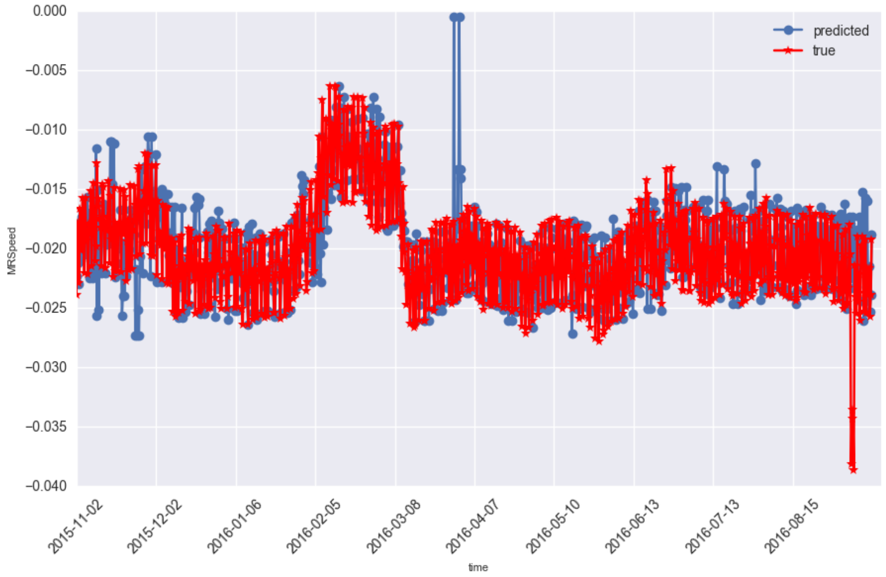


Figure 4.47: Online model with artificial data: MRSpeed prediction during test stage for single currency dataset EURO.

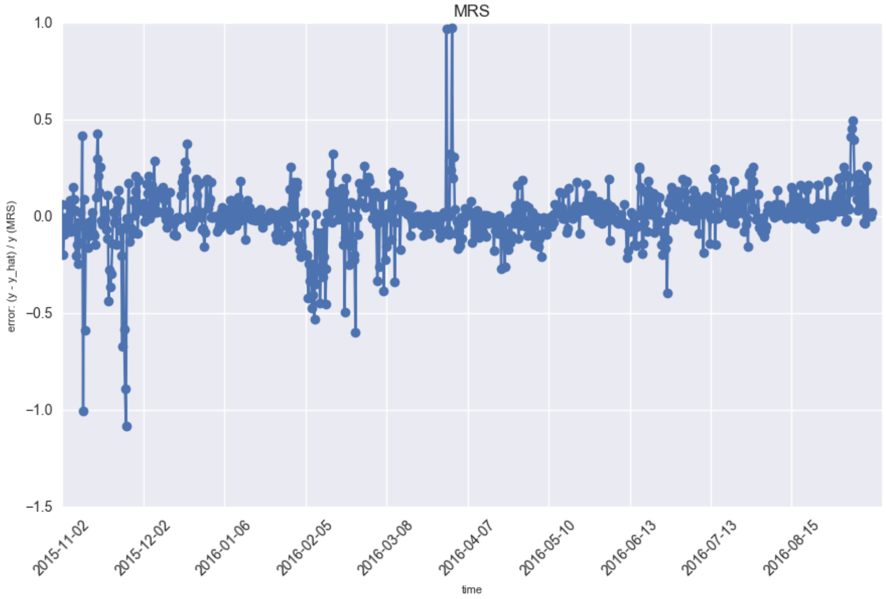


Figure 4.48: Online model with artificial data: MRSpeed relative error during test stage for single currency dataset EURO.

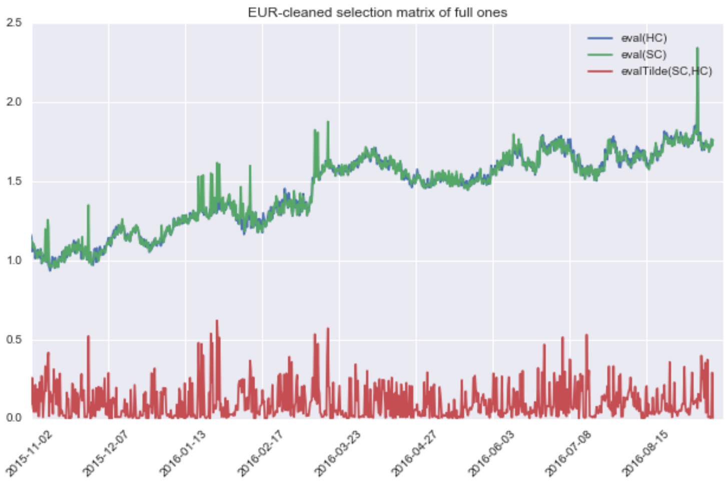


Figure 4.49: Feedback function of online model developed on original data only, for single currency dataset EURO. In the figure $evalTilde$ denotes $\overline{eval}$.

4.5.3 *Evaluation and matrices generalization*

To summarize the previous results and by looking at Tables 4.7 and 4.9, we have stated that both developed models achieve good performances. In particular by focusing on the R2 metric we can say that both the online approaches are able to describe a good portion of the variance information. By referring to the computational times shown in Tables 4.6 and 4.8, we have discarded the second version since it takes too long without any improvement on the performances. With these numbers, we were not able to get a concrete and real feedback about our predictions, to do this we need to leverage on a domain-feedback function. In our experiment we have used the one described in Section 2.5.2. As first trial we have compared our model with the hard-calibration; this comparison was made by using a weighting matrix of ones (Chapter 2.5.1). What we expect to see is that: the errors of our model and the hard calibration are close to each other. For the moment we do not take into account the range of numbers taken by these metrics. We have just to focus on the $\overline{eval}()$ function and check if it is close to zero, and if the $eval()$ curve peaks of the HC and our ML model refer to the same calibration instances. The results are depicted in Figure 4.49, in this plot there were removed two mispredictions that were out of the manifold and do not allow to see properly the error curves. What we can see is that in general the performances obtained by our model are close to the ones obtained with the analytical one. The same conclusion can be derived by focusing on the $\overline{eval}()$ function that is always behind 0.5 and in general close to zero. Given this positive result, we have wondered if there is an advantage in learning on the weight selection unitary matrix, in particular we have wondered if there are some generalization relations over sparse matrices. To check this property we should compare the $eval()$ curves obtained by the analytical model computed over the sparse matrix and the one obtained by our model referred to the unitary matrix. If the two curves would be similar we could state that this property holds. Since this check requires the development of a new dataset computed over the sparse matrix, we did a first experiment to get a first proof of validity. From the previous results we know that our model is capable of learning the optimal parameter referred to an unitary matrix with an optimal error $eval^{*ML}_{unitary}$. What we can do with the available data, is get the feedback obtained over the sparse matrix $eval^{ML}_{sparse}$ by the same parameters, and check if these curve have similar shape. Clearly, from the feedback function definition (Section 2.5.2), we may consider the introduction of rescaling transformations since errors do not have any normalization according to the weights. The results are displayed in Figure 4.50, here a multiplicative rescaling factor of 0.5 was applied to the error that refers to the unitary matrix. What he can conclude from this first analysis is that the curves have similar shapes, so there might be generalization opportunities that have to be checked as discussed at the beginning of this paragraph. Given these argumentations we have deepened our understanding behind this phenomenon. In particular we have performed the analyses by following a statistical approach. For the already discussed algorithms, thus:

- Hard Calibration applied to the unitary case, that I will denote as HCu;
- Hard Calibration applied to the sparse case, that I will denote as HCs;
- Soft Calibration applied to the unitary case, that I will denote as SCu[2].

We have perform an evaluation based on the $eval()$ function, by using both the sparse and the unitary weight selection matrix. As indicators we have computed the sample mean with a

[2] In this case we have learnt over a dataset computed with HCu, thus, the hard calibration over an unitary weight selection matrix.

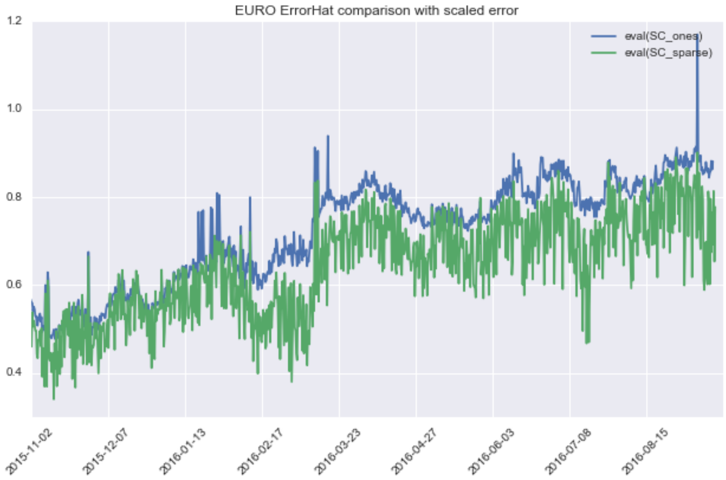


Figure 4.50: In the figure _$sparse$ and _$ones$ refer to the weight matrix used in the feedback. It can be or sparse or unitary.

$eval()$ selection matrix:	unitary	sparse
HCu	1.4477 ± 0.0156	0.6517 ± 0.0077
HCs	5.7973 ± 0.7676	$0.0049 \pm 7.82\ e^{-7}$
SCu	1.4505 ± 0.0154	0.6423 ± 0.0075

Table 4.10: In the first column there are the feedback errors obtained by using an unitary selection matrix. In the second the ones referring to a sparse selection matrix.

confidence interval of 95%. The results are displayed in Table 4.10. Given these outcomes, what we can state is that: the introduced data-driven approach reaches performances that are similar to the ones obtained by the analytical model. In particular its error is below 1% with respect to the one obtained by the analytical model from which it learns from. As expected this relation holds for both the selection matrices used during the feedback computation. This means that our model behaves as the one from which it learns from independently from the weight configurations. By referring to HCs we can state that it minimizes a different objective, in fact its error with respect the sparse feedback is smaller than the one achieved by HCu, conversely for the unitary feedback the opposite is true. In particular the error achieved by HCs with respect to the $eval()$ function is 5.7973, with respect to the sparse weight selection matrix is 0.0049.

Set name	number of record	computational time [s]
Train	2076	0.9127
Test	891	0.0994

Table 4.11: In the first column is shown the dataset division between train-test sets. In the last the computational time required expressed in seconds.

4.6 BATCH LEARNER

In order to provide a complete analysis we have decided to investigate also the batch approach. What we want to show with these analyses is that: a batch approach is not appropriate for our problem and that perturbed data do not add significant information with respect to the original ones.

4.6.1 *Time dependent predictions*

In this first scenario we test the batch model capability of following trends, in particular we expect to have low performances since a batch model does not distinguish old and recent samples. This first test as in the online case, was performed with a Decision Tree Regressor (DTR) and by considering only original records. We have cleaned them from outliers and they were divided according to their reference date, the percent in the train was the 70% and in the test the other 30% of original data. Given this scenario, the dataset composition is the one described in Table 4.11, what we can state from this first information is that the batch model requires lower computational time than the online one. By moving to the analysis of its performances, in general we expect to have much bigger error values. They are displayed in the Table 4.12. What we observe is that there is an high overfitting phenomena during the train phase, in particular the R2 scores are equal to one for both the targets. The same is not true during the test phase, we can get a first feedback of the incapacity of following trends from the R2 scores that is negative for both variables. As in the online case, I provide some useful pictures in order to get a full comprehension on the results. For the train stage, we can observe the overfitting phenomena in the scatter depicted in Figure 4.51, we can notice that the manifold of the predicted points is completely overlapped to the one of original samples. This occurs

metric	MRSpeed	Volatility
Train phase		
RMSE	8.0372 e^{-8}	1.1764 e^{-8}
RMSE[/range]	2.7768 e^{-7}	1.6772 e^{-6}
R2	1.	1.
Test phase		
RMSE	0.0074	0.0006
RMSE[/range]	0.0253	0.0894
R2	−2.2564	−1.5879

Table 4.12: In the first column is shown the adopted error metric. In the second and in the third the errors by parameters.

due to the overfitting capability that characterizes non parametric models. For the test phase this does not occur with the same results, as it is depicted in Figure 4.52. In particular we can observe the disadvantages leaded by the usage of a batch approach by focusing on the three points out of the manifold, all of them seem to be mispredicted. This cannot be guaranteed since with scatter plots we can understand just the general behaviour with analyses on manifolds and not results on each single predicted record. From this first analysis we can just state that the manifolds of old and recent samples are different and a batch approach is not the appropriate choice in this scenario. In order to investigate more this issue I move to the plots of the test error with respect to each predicted sample and each single target as displayed in Figures 4.53, 4.54. Here we can notice two main aspects, the first one is the fact that the error is not always close to zero, in particular it seems to be negative for the first half of records and positive for the second one. Moreover, there is a set of points, those corresponding to the days between 11th and 16th of November 2015, that are totally mispredicted since their values are out of the manifold range. From these two observations we can confirm the expected incapacity of following trends by batch model. From this conclusion we moved to a further test. We have took advantage of batch models to get an empirical view on the artificially generated samples with respect to the original records.

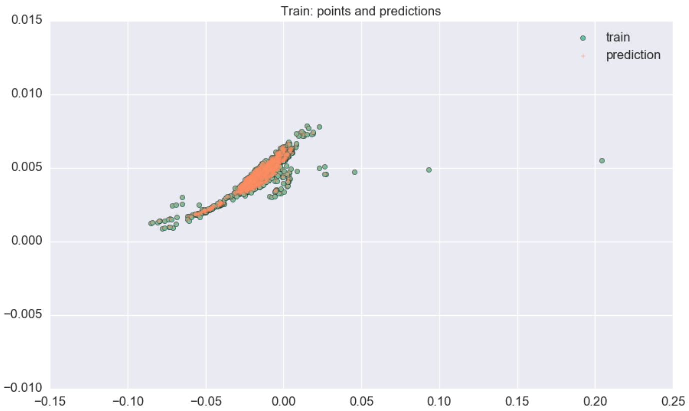


Figure 4.51: Scatter plot for the manifolds described by the original train data and their predictions for the single currency dataset EURO.

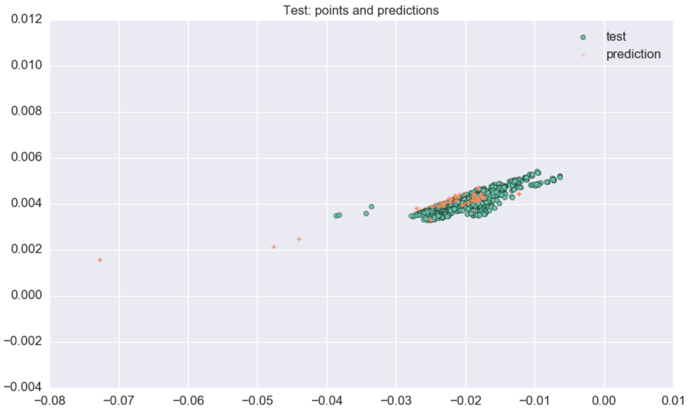


Figure 4.52: Scatter plot for the manifolds described by the original test data and their predictions for the single currency dataset EURO.

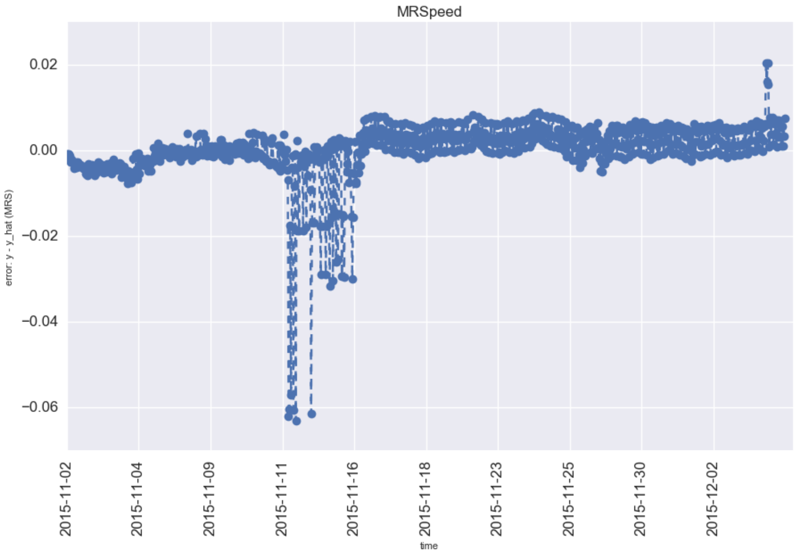


Figure 4.53: MRSpeed plot for original test data for the single currency dataset EURO.

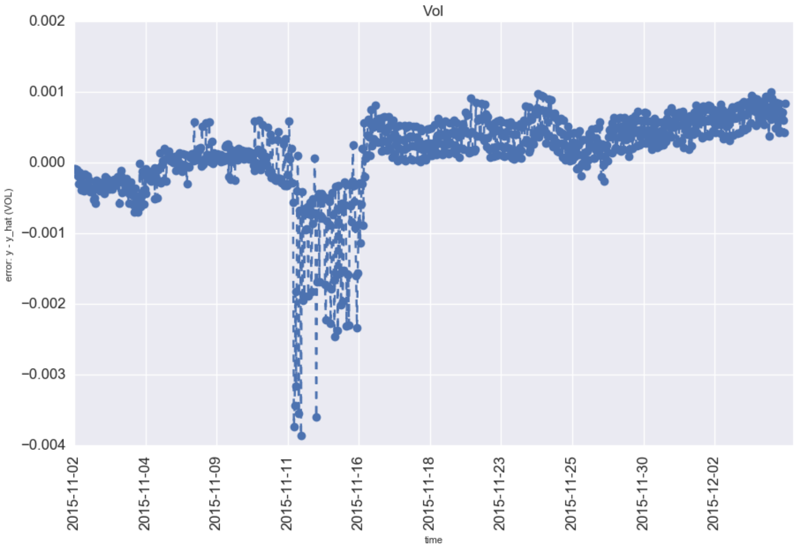


Figure 4.54: Volatility plot for original test data for the single currency dataset EURO.

metric	MRSpeed	Volatility
Train phase		
RMSE	$1.2986\ e-^{7}$	$1.5730\ e^{-8}$
RMSE[/range]	$1.3390\ e-^{7}$	$1.9491\ e^{-6}$
R2	1.	1.
Test phase		
RMSE	0.0138	0.0005
RMSE[/range]	0.0142	0.0653
R2	−0.2958	0.6088

Table 4.13: In the first column is shown the adopted error metric. In the second and in the third the errors by parameters.

4.6.2 *Using perturbed data to predict original*

We have decided to use the information contained in the perturbed data to make inference on the original ones. This task was accomplished by using the perturbed records during the learning phase, and the original records during the test phase. Even in this case we have performed our analysis over the single currency EURO, and by using as an underlying model a Decision Tree Regressor. I present in Table 4.14 the dataset shapes and in Table 4.13 the results obtained by following this approach. As in the previous case we can notice the overfitting phenomena during the training phase by observing the R2 values that are still equal to one.

In the test stage we expect to have an high overfitting since we are learning from the perturbed data that are a noisy and augmented version of the original data, here the test set. Even if the outcomes are better than the previous experiment performed with the batch model, they cannot be translated into overfitting. As before, I will start by analysing the manifolds and secondly the target curves. The scatter plots are displayed in Figure 4.55 for the train phase and in Figure 4.56 for the test phase. By analysing them we can notice that the perturbed data define exactly the same manifolds of the original ones for a big portion of them but not for the whole set of points. In particular, there is a subset of non artificial points whose perturbation is projected on a different manifold with respect to the target parameters. This may occur due to the perturbation phase.

In order to better analyse if the overfitting is present in the test

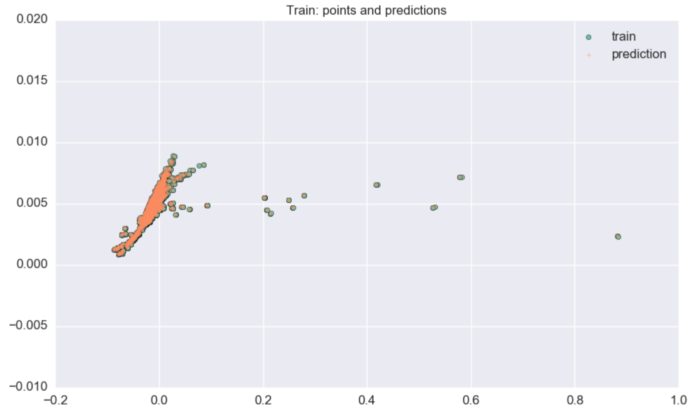


Figure 4.55: Scatter plot for the manifolds described by the perturbed train data and their predictions for the single currency dataset EURO.

Set name	number of record	computational time [s]
Train	17420	5.4332
Test	2967	0.6997

Table 4.14: In the first column is shown the dataset division between train-test sets. In the last the computational time required expressed in seconds.

dataset, I will move to the target curve plots. What we expect is to see an almost zero error for all the samples except for this subset of points whose projection does not reflect the original data manifold. In regards to target plots, with Figures 4.57, 4.58 we can notice that there is the expected portion of mispredicted point, but among all the points, the majority of them have a zero error that evidence the overfitting phenomena. This can be noticed by analysing both the plots. They contain a single point that may be considered as an outlier, a group of points that are mispredicted and may correspond to the ones with different manifold and the rest of them are overfitted.

To conclude the analysis for the EURO scenario, we can state that the perturbed data, even if they have a small subset of different points, they do not contain enough significant information. This was not an ideal case, but thanks to visualization techniques we were able to highlight the presence of an hidden overfitting phenomena.

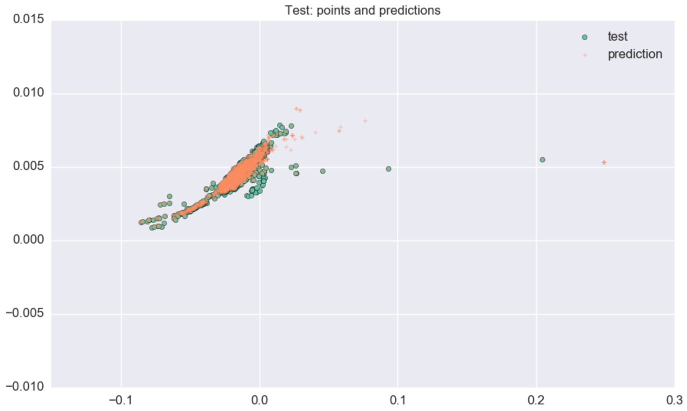


Figure 4.56: Scatter plot for the manifolds described by the original test data and their predictions for the single currency dataset EURO.

4.7 DATA AUGMENTATION AND MULTI-CURRENCIES

The only problem that we still did not handle, that also did not let us end our experiments here, is that we were dealing with a not enough big amount of data. The perturbation approach, as proved with the previous experiments, did not lead us the expected improvements. For this reason we had to consider a different way to enlarge our basis of knowledge. We have then decided to move to a multi-currencies dataset in order to have more significant performance measures and at the same time learn on more real cases. In order to exploit the information contained in the dataset of all the currencies we had to bring the data in a common features space.

At first glance, the format mismatches are related on curves and matrix features. Given the NS reduction discussed in Section 4.3.1 the transformed curves are already in an uniform format, each curve will be substituted by the four Nelson-Siegel parameters independently from the currency. The same does not hold for the matrix features, thus: prices and volatilities. The first strategy would be to try to directly apply PCA by fixing as a parameter the number of future features. This would be not correct, in particular we should remind the principal component definition (Section 4.3.1). The point is that having the same number of eigenvectors does not imply referring to the same state space, in general different eigenvectors may define different directions. The solution that we have adopted consists of using a common sub-matrix between all the currencies. The

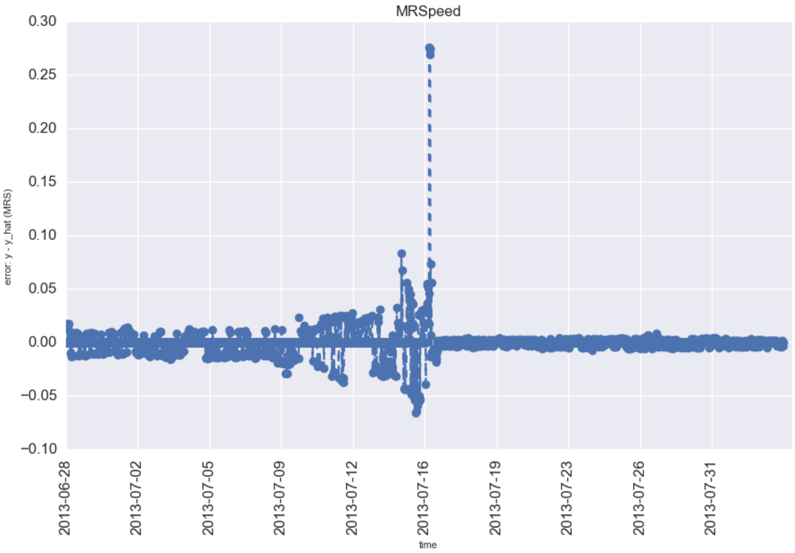


Figure 4.57: MRSpeed plot for original test data for the single currency dataset EURO.

Set name	number of records	computational time[s]
Train	10270	103.5854
Validation	4401	7326.3118

Table 4.15: In the first column is shown the dataset division between train-validation-test sets. All the records are originally generated.

sub-matrix is computed by considering cells that refer to the same couple of tenor and expiry dates. After these data transformation phase, we have a set of cross-currencies records that have an uniform reduced features space. In our experiment we have considered the following currencies: NZD, GBP, EURO, CAD, USD, Swiss Franc (CHF) and Norwegian Krone (NOK). In particular in computing the cross-currencies dataset we have considered data referring only to original calibration since as already discussed, adding the perturbed samples would mean just increasing the computational time without any gain in terms of performances. The used dataset division is described by Table 4.15; the computational time values displayed are computed with a different and with higher performances machine. The model is composed of only a batch train phase and an online test phase and leverages on a Decision Tree Regressor. As for the EURO case we have tried to use a validation stage but for the same disadvantages discussed for the single currency model

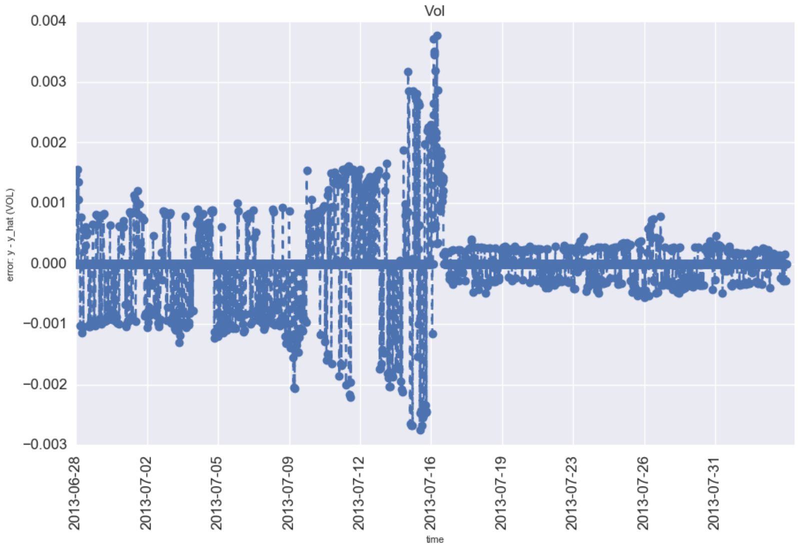


Figure 4.58: Volatility plot for original test data for the single currency dataset EURO.

(Section 4.5) this phase was omitted.
Here the amount of data is consistently bigger and the online phase has an high computational cost. In this way the portion of dataset that would define the validation set was part of the train set, thus was learnt in a batch mode. Finally, IFS was used as supervised feature selection algorithm. The features selected by this approach are:

- first component pca matrix of prices;
- second component pca matrix of prices;
- third component pca matrix of prices;
- fourth component pca matrix of prices;

In figure 4.59 we can see their behaviour during the test phase. From this image is difficult to make proper considerations since there is a lot of overlap between their curves. The only aspect that we can see is that: both the first and the fourth pca component seem to behave as the targets parameters of Figure 4.60. From this last image we may see that the two variables are highly correlated in fact their curves have similar shapes.

I present in Table 4.16 the results obtained with the online approach. What we can observe is that as in the single currency case, the model fully overfits the data during the train stage. In regards of the test, the errors are low, in particular the percentage Root Mean Squared Error (RMSE) is: 8.35% for the mean

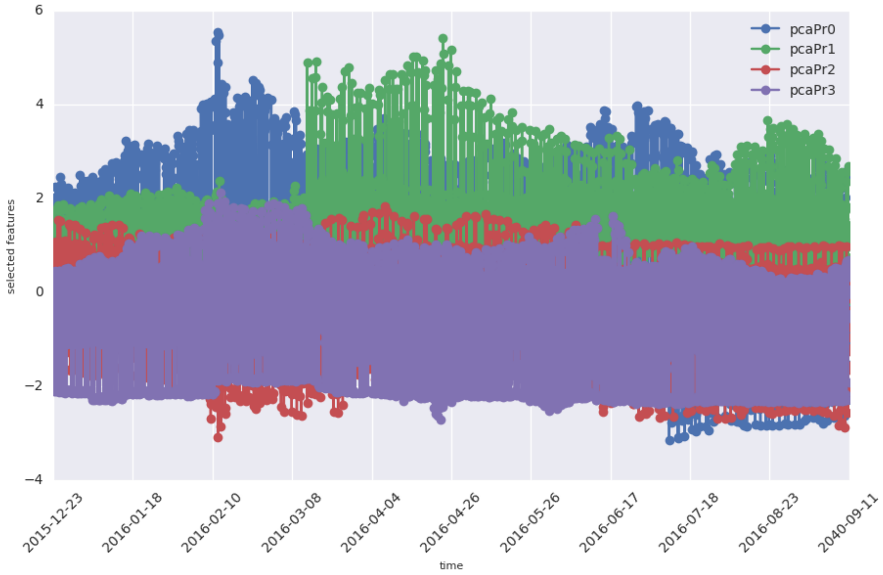


Figure 4.59: Online model cross-currencies features behaviour in test stage. The selected features are the first, the second, the third and the fourth principal component of the PCA matrix of prices.

reversion speed, and 2.97% for the volatility. These results can be also displayed with: the plots of the targets relative error, and their values with respect to the original ones. With this images we are able to evaluate the model tracking capability. In particular with Figures 4.61, 4.62 we can notice that our model predictions are almost fully overlapped with respect to the original values, this is in agreement with the performances of Table 4.16. In particular, the only mispredicted point is one of the outliers discovered in the previous analysis for the single currency dataset. The same results holds for the volatility parameter as can be observed in Figures 4.63, 4.64. In both the cases the RMSE is close to zero for all the points within the parameters manifold, thus our model is tracking their trends.

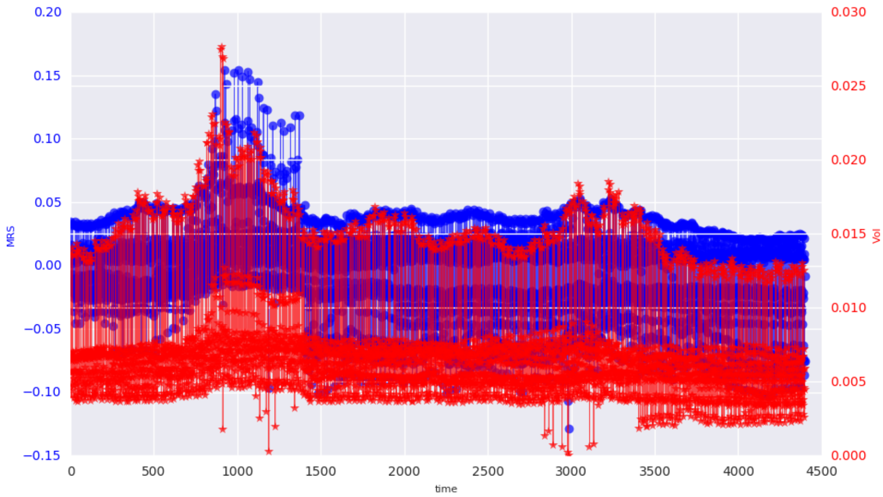


Figure 4.60: Online model cross-currencies targets curves in test stage. It is important to notice that they are plotted on different y-axis.

metric	MRSpeed	Volatility
Train phase		
RMSE	8.1829 e^{-7}	8.896 e^{-8}
RMSE [/range]	7.7486 e^{-7}	3.1369 e^{-6}
R2	1.	1.
Test phase		
RMSE	0.0236	0.0008
RMSE [/range]	0.0835	0.0297
R2	0.5914	0.9519

Table 4.16: In the first column is shown the adopted error metric. In the second the errors referring to the MRSpeed variable. In the last the errors referring to the volatility variable. All the information refer to the cross-currencies dataset with only original data.

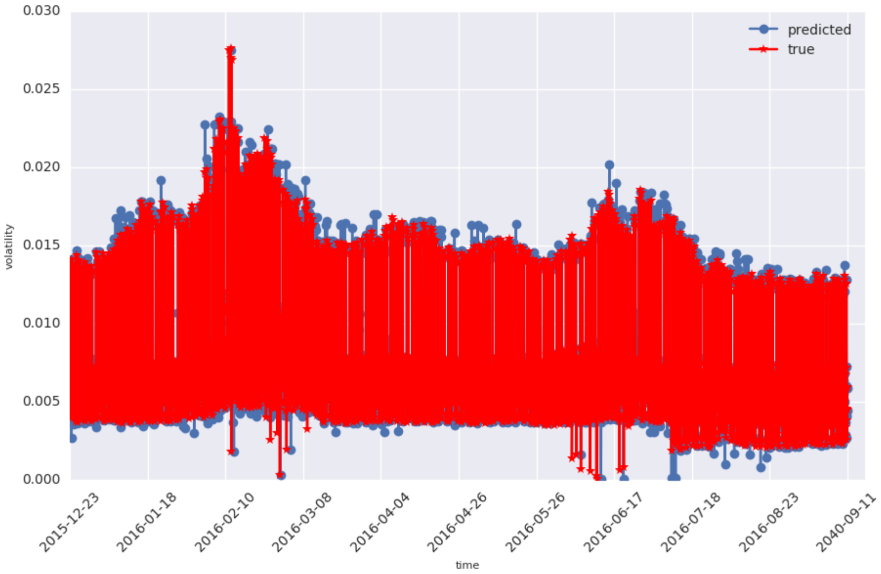


Figure 4.61: Online cross-currencies model Volatility prediction during test stage.

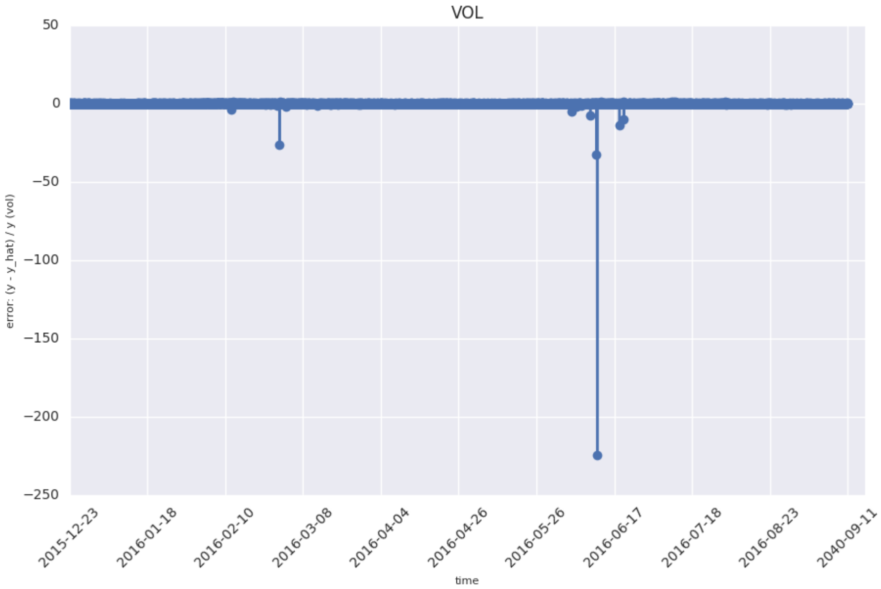


Figure 4.62: Online cross-currencies model Volatility relative error during test stage.

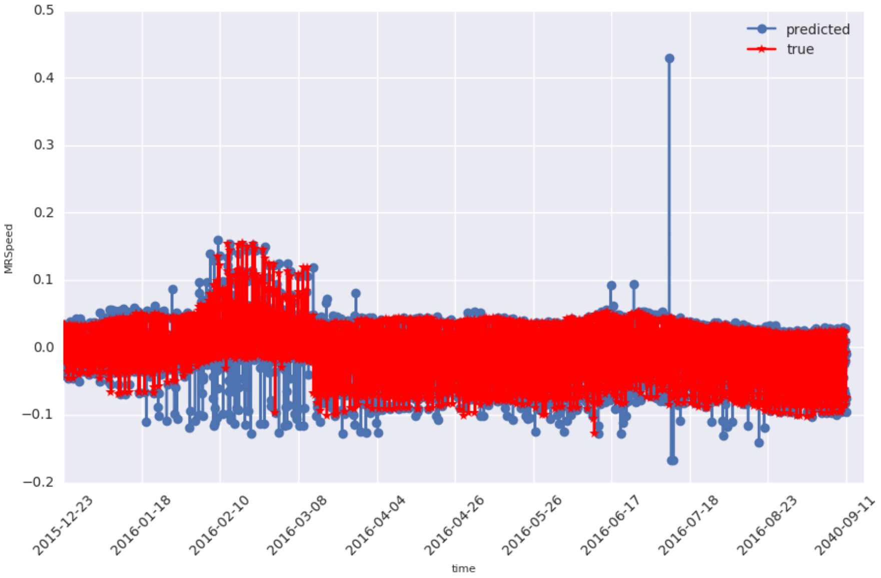


Figure 4.63: Online cross-currenceis model MRSpeed prediction during test stage.

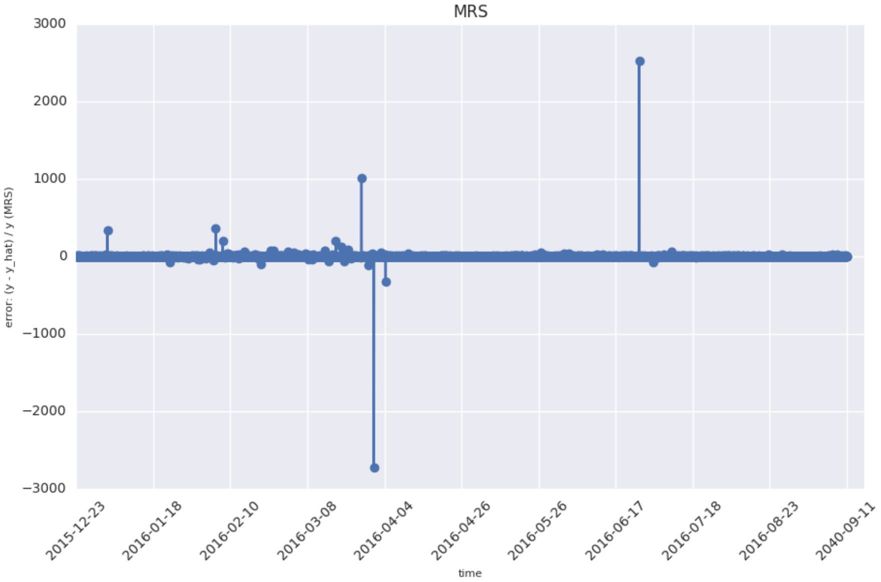


Figure 4.64: Online cross-currencies model MRSpeed relative error during test stage.

4.7.1 *Evaluation and matrices generalization*

As we have discussed for the single currency scenario (Section 4.5.3), from the previous results, we can state that from a machine-learning point of view our model have good performances. The point is that with these values we can only make some general considerations, we are not sure about the real contribute given by our model. In order to have a proper feedback we have used the feedback function provided by the bank (Section 2.5.2).

In this way we can understand how well our model predicts. In Table 4.17 we can see that our model error is close to the analytical one. In considering the $\overline{eval}()$ value of our model we have to remind that it takes in consideration only the cases in which it predicts worst than the analytical one. The contribute given by the other cases are a zero term in the error and not a negative one, thus, the better predictions do not compensate the mis-prediction. The same behaviour is displayed in Figure 4.65, what we can notice is that the main contribute in the feedback error is given by the bias of the analytical model from which we learn from.

As for the single currency case (4.5.3), we have evaluated the generalization property over different weight selection matrices. Here the computational costs associated to the development of a cross-currencies dataset referring to a sparse matrix are much higher that for the single currency case, given that this implies the computation of a single currency dataset that refers to the sparse weight selection matrix for each single currency that appears in the cross-currencies dataset. The only test that we can make is the one already discussed, that compares the errors obtained by our model with respect to the unitary matrix and the one over the sparse matrix. The results are displayed in Figure 4.66, as before a rescaling factor of 0.5 was applied to the error computer over the unitary weight selection matrix. As before, we can make the same positive conclusion, more work has to be spent to check this generalization property over different selection matrices.

	eval()	$\overline{\text{eval}}()$
HC	0.8211	0.0
ML	0.8346	0.1201

Table 4.17: Feedback function mean errors for Hard Calibration and data-driven calibration.

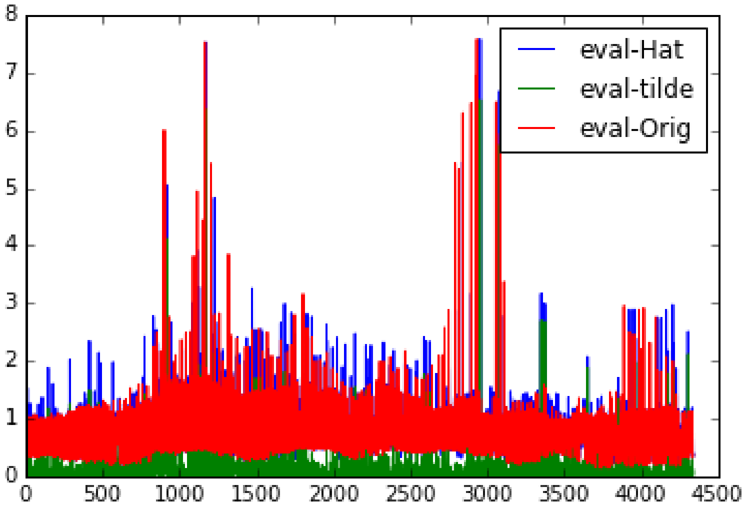


Figure 4.65: Feedback function results for cross-currencies online model.

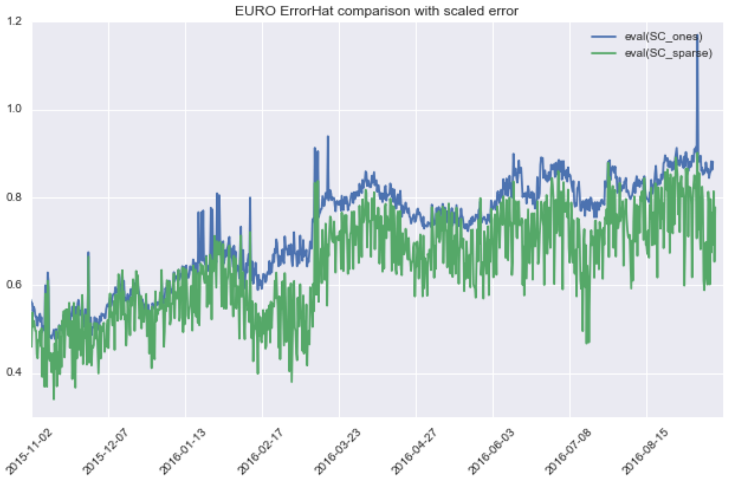


Figure 4.66: Feedback function comparison over different selection matrices, as a first proof of generalization capability for cross-currencies data. In the figure _sparse and _ones denote the weight selection matrix used for the feedback. It can be either sparse or unitary.

5 CONCLUSIONS

Non devi odiare il sole solo perché tu non puoi vederlo.
F.

This final chapter provides an overview of the possible future works regarding Swaption pricing and interest rate and discount curves prediction. Then, possible future developments of the pricing framework are suggested, including the extension to cross contracts models. Finally, the current limitations of the proposed approach are discussed along with the conclusions of the overall work presented in this thesis.

5.0.1 *Summary of Results*

The results presented during the previous chapters can be resumed in several summary conclusions.

DATASET. The evidence presented in the previous chapters offers a support to the fact that the performed perturbation was not a good solution, instead the choice of move to a cross-currencies approach proved to be appropriate.

ANALYTICAL MODEL. The analyses performed over the analytical model outcomes, provide an additional feedback. Now, we have a clear vision about the curve variable relationships and contributions in pricing contracts. Further more, we know which are the variable that mostly contribute in determining Vasicek parameters. Thus, the use of data analysis and visualization techniques show structural and functional relationships behind the analytical formulas.

LEARNER APPROACHES. In regards to the considered learning approaches, we have proved that online learning has several benefits with respect to follow an offline learning approach. Even if the online approach takes longer in training, it is capable of following trends that is not possible with an offline model.

5.0.2 *Future Research*

I will start by highlighting some issues and limitations in our approach; starting from them I will propose some solution strategies that can be followed as future work.

ONLINE IMPROVEMENTS. Starting from the proof that online learning has several benefits, we would like to strengthen its ability of following trends. To do this, we suggest to introduce a samples space derived from the previously handled data. This new space will be the starting point to derive a set of possible future samples that may follow in the current trend. With this addition, we expect to have better prediction regarding future points. What it should be evaluated is the trade-off given by the increasing of model complexity with respect to the new obtained performance.

CONTRACT RESTRICTION. Given the good results obtained with this uncommon learning schema of feedbacks given by real prices, we can improve it by moving to an extended space of contracts. In particular it could be evaluated the possibility of considering also swaption not At The Money (ATM). This extension may augment the number of features since it may require more information concerning the contract state; what should be discussed is the trade-off given by the new information with respect to the augmented dimensional space.

FIXED SELECTION MATRIX. The last research that we suggest, is the usage of machine-learning techniques with the objective of develop algorithms that starting from the proposed regressor are able to define optimal portfolios. In this way we expect to augment the number of real applications. As a first version we may restrict only to swaptions, in this case all the results obtained over the generalization property over different selection matrices hold. More in general, we should enlarge also to different contracts that refer to the same discount and forward curves, that as discussed during the introduction, are the most important assets involved.

BIBLIOGRAPHY

Bishop, Christopher (2006). *Pattern Recognition and Machine Learning*. Springer. ISBN: 978-0387-31073-2 (cit. on pp. 30, 31, 36, 39).

Bjork, Tomas (2003). *Arbitrage theory in continuous time*. Oxford University Press. ISBN: 978-0199-27126-9 (cit. on pp. 4, 11, 19, 22, 23).

Castelletti A. Galelli S., Restelli M. Soncini-Sessa R. (2011). "Tree-based variable selection for dimensionality reduction of large-scale control systems." In: (cit. on p. 40).

Charles R. Nelson, Andrew F. Siegel (October 1987). "Parsimonious Modeling of Yield Curves." In: *The Journal of Business* (cit. on p. 60).

Chisholm, Andrew M. (2004). *Derivatives demystified*. A Step-by-Step Guide to Forwards, Futures, Swaps and Options. West Sussex PO19 8SQ,England: Wiley. ISBN: 978-0470-09382-5 (cit. on pp. 2, 3).

Collins RT Liu Y, Leordeanu M. (2005). "Online Selection of Discriminative Tracking Features." In: (cit. on p. 37).

Huang, Samuel (2015). "Supervised feature selection: A tutorial." In: *Artificial Intelligence Research* 4 (cit. on p. 40).

Hull, John (2011). *Options, Futures and Other Derivatives Options, Futures and Other Derivatives*. Pearson Education. ISBN: 978-0273-75907-2 (cit. on pp. 3, 4, 11, 18).

Hyejin Park, Jaewook Lee (2012). "Forecasting nonnegative option price distributions using Bayesian kernel methods." In: (cit. on p. 43).

Hyejin Park Namhyoung Kim, Jaewook Lee (2014). "Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over KOSPI 200 Index options." In: (cit. on p. 43).

Ian H. Witten, E. Frank (2005). *Data Mining. Practical Machine Learning Tools and Techniques*. Elsevier (cit. on p. 37).

J. Beleza Sousa M. L. Esquivel, R. M. Gaspar (2012). "Machine Learning Vasicek Model Calibration with Gaussian Processes." In: *Communication in Statistic Simulation and Computation* (cit. on p. 43).

James Hutchinson Andrew Lo, Tomaso Poggio (1994). "A non-parametric approach to pricing and hedging derivative securities via learning networks." In: *The Journal of Finance* (cit. on pp. 42, 43).

Joliffe, Ian (2014). "Principal Component Analysis." In: *Wiley StatsRef: Statistics Reference Online* (cit. on p. 38).

M. Malvaldi, D. Leporini (2014). *Capra e calcoli, L'eterna lotta tra gli algoritmi e il caos*. i Robinson/Letture. Laterza Editori. ISBN: 978-8858-11192-5 (cit. on pp. 1, 2).

Markowitz, Harry (March 1952). "Portfolio Selection." In: *Journal of Finance* 7, pp. 77–91 (cit. on p. 1).

Marquardt, Donald W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." In: *Journal of the Society for Industrial and Applied Mathematics* (cit. on p. 25).

McNelis, Paul D. (2005). *Neural Network in Finance: gaining predictive edge in the market*. Elsevier Academic Press. ISBN: 978-0124-85967-8 (cit. on pp. 2, 5, 8).

Mohammed J. ZakiI, Wagner Meira Jr. (2014). *Data Mining and Analysis. Fundamental Concepts and Algorithms*. Cambridge University Press. ISBN: 978-0521-76633-3 (cit. on p. 31).

Robert Kolb, James Overdahl (2002). *Financial Derivatives*. Wiley. ISBN: 978-0471-23232-2 (cit. on p. 4).

Stuart Russel, Peter Norvig (2009). *Articial Intelligence: A modern approach*. Pearson. ISBN: 978-0136-04259-4 (cit. on pp. 35, 36).

Sven Sandow, Xuelong Zhou (2007). "Data-efficient model building for financial applications. A semi-supervised learning approach." In: *The Journal of Risk Finance* (cit. on p. 31).

T. Hastie R. Tibshirani, J. Frieman (2009). *The Elements of Statistical Learning. Data mining, Inference and Prediction*. Springer. ISBN: 978-0387-84858-7 (cit. on pp. 8, 32).

Vasicek, Oldrich (1977). "An equilibrium characterization of the term structure." In: *Journal of Financial Economics* (cit. on p. 18).