POLYTECHNIC UNIVERSITY OF TURIN

Master's Degree in Computer Engineering

Politecnico di Torino

Master's Degree Thesis

A Machine learning approach to anomaly detection on derivative pricing

Supervisors

Prof. Francesco VACCARINO

Prof. Luca CAGLIERO

Alessandro GRIBALDO

Candidate

Alberto RIVA

March 2021

Abstract

Risk management has become an essential part of the process for any company that trades derivatives. Banks and corporations own risk-management offices dedicated to the development of risk models involving market, credit and operational risk, assure controls are operating effectively, and provide research and analytical support. Their models and operations have to be compliant to current European Central Bank legislation, which supervises the process. In the thesis we provide a new machine learning approach to detect anomalies in option pricing models. The research focuses on a preliminary analysis to assess the applicability of the method, starting with an analysis of call options quoted on the S&P500 index, with historical data from 2010 to 2021.
The model consists in forecasting the value of implied volatility and using it to check the results provided by other pricing models used in companies. This is achieved by using three different machine learning models for regression: artificial neural network, random forest and gradient boosting regression. The best results were given by the application of the first model. The analysis of the results shows that our approach can accurately solve the problem and be applicable in a real-world working application, with an integration in the workflow of risk-management offices, providing real-time alerts on errors of various nature that can occur in the pricing process.
In the future, this approach could be extended to work on other types of financial derivatives, by collecting the necessary data to train the machine learning algorithms, while another field of research could instead focus on trying out other machine learning models, looking for an improvement in the accuracy of forecasted implied volatility values.

Acknowledgements

It is a pleasure to thank all the people who have supported me through my academic career, with their constant help and encouragement, from elementary school to University. I would not be here without them.

A special thanks to Prof. Francesco Vaccarino and Prof. Luca Cagliero for their help in this thesis, their availability and their kindness. Thanks to the company *Myrios* for making the project possible and giving me the chance to explore new interesting and fascinating domains.

The biggest thank you goes to my parents, Anna and Bruno, for teaching me how to approach life and face the hardest moments that life throws at us. Thanks to my whole family, my grandparents, uncles and aunts, that have always believed in me and surrounded me with love.
This journey would have been terrible and impossible without the support of my friends Gabriele and Cristian, with whom I shared 20 years of my life and made everlasting memories.
Thanks to all the friends that I made during this fundamental stage of my life, I want and I will thank you one by one in person.

"To Scottie Pippen, from way downtown."
Moxous Morris

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

IV
Implied volatility

ML
Machine learning

NN
Neural network

BS
Black-Scholes

OTC
Over The Counter

CBOE
Chicago Board Options Exchange

VIX
Cboe Volatility Index

SP500
Standard and Poor's 500 Index

MAE
Mean Absolute Error

MSE

Mean Squared Error

RFR

Random Forest Regressor

GBR

Gradient Boosting Regressor

MLPR

Multi-layer Perceptron Regressor

LSTM

Long Short-Term Memory

Chapter 1

Introduction

1.1 Problem and Background

The financial crisis of 2008 attracted a lot of attention and fear around the derivatives market. In order to avoid another collapse of the global economy, the major financial institutions and governments introduced new regulations to which corporations, banks and all the players in the derivatives market need to oblige to. These regulations focus on the aspect of credit-risk management, which is a challenging task in the world of financial derivatives because of the complex mathematical models needed to keep it under control. Before the crisis this was an often neglected aspect, with the conviction of investors that the market could not collapse.

Nowadays, credit-risk management has become an essential part of the process for any company that trades derivatives: entire *risk-management* offices are active and under constant control of institutions such as the European Central Bank. Their task is to provide development of risk models involving market, credit and operational risk, assure controls are operating effectively, and provide research and analytical support. This is achieved by constantly creating new models to analyze risk and compare them with standard models that are used to take business decisions in the companies [1]. The final price of a derivative can be wrong for numerous reasons: wrong market data, inefficient model, a mistake made when registering contract information in the accounting system. This could lead to bad financial decisions by the trading company and fines by the authorities that supervise accounting books. This is why it is essential to continuously check working models with new models and compare results to track down mistakes.

In derivatives pricing, the value of a financial asset is typically measured via its return. It represents monetary profit or monetary loss in an investment and

it is characterized by randomness that evolves with time. This random term is captured in a variable called volatility, which expresses a statistical measure of the dispersion of returns for a given security or market index. In price evaluation of such assets, volatility plays a key role, because it provides a variable that determines the order of magnitude of change that we expect in an observed underlying financial indicator. There exist two different measures of volatility: historical and implied. Historical volatility, also called realized volatility, is usually computed by looking at the standard deviation of one to six months of previous returns, so it is clearly a backward looking measure. Implied volatility (IV) is instead a theoretical value, which needs an option price model, for example the Black-Scholes model, in order to be calculated. Said model is usually expressed as $OptionPrice = BS(volatility)$: when we observe the option price that is provided by the market, we can invert the equation and iteratively solve for implied volatility.

The thesis strongly relies on the importance of this parameter, with it being the most influencing factor in the determination of the price of the derivative. We propose a new method for derivatives pricing that exploits machine learning models, with the final objective of deploying an automated framework to monitor derivative prices and ensure their correct evaluation.

1.2 Goal

In the thesis we want exploit IV in order to detect anomalies in price changes overnight. This is achieved by forecasting the next day value of IV with machine learning algorithms. This approach leads to two results: a price-checking feature and an accurate one-day forecast of IV.

Artificial neural networks (ANNs) are universal function approximators and they have proved themselves successful at extracting features and detecting patterns from large data sets. This ability is used to model implied volatility through a data-driven approach in order to be able to forecast its value. By training an ANN on historical option data, we try to build a model that is able to provide us the fundamental block of a larger framework that can solve the aforementioned problems.

Our approach consists in a framework that can forecast implied volatility for the next day, given today's data, and compare it with the observed implied volatility of the next day: depending on the magnitude of the difference between the two implied volatilities, it detects whether an anomaly as occurred or not, alerting the

user that some mistakes are probably present in market data gathering, in the pricing model or in its input values.

The framework workflow schema is illustrated in figure 1.1

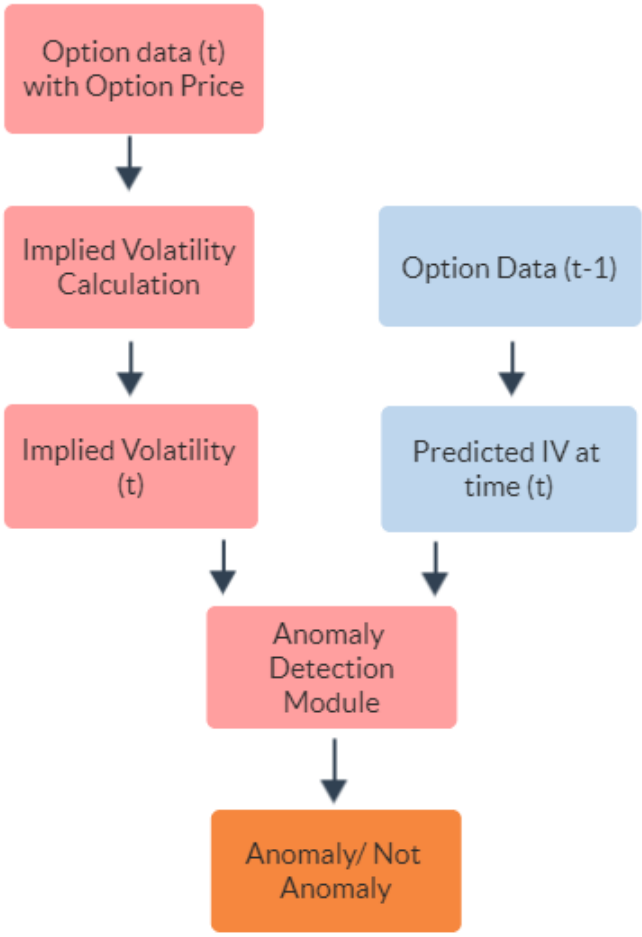

Figure 1.1: Anomaly Detection Framework Schema

Chapter 2

Literature Review

In this chapter we present an overview of related works on the topic of derivative pricing using machine learning algorithms and implied volatility forecasting techniques. We then introduce derivatives, discuss some theoretical background on the basics of option pricing with Black-Scholes model and expose its limitations. I then present volatility in its two forms (historical and implied), by providing mathematical background and focusing on its important properties.

2.1 Related Works

The problem of derivative price monitoring is commonly approached by forecasting volatility. Volatility forecasting can be grouped into two main categories: option-implied volatility and historical time-series models, such as historical volatility, autoregressive conditional heteroscedasticity (ARCH), and stochastic models [2]. A general schema of existing approaches to the problem is presented if figure 2.1.

The two main approaches present some very important differences:

- *Historical time-series volatility forecast*: mathematically expresses, in a previous time interval that could be days, months or years, the magnitude of price variations for a given financial product. This volatility form is a backward looking measure and exclusively based on historical observations, so it does not include any expectations on future price evolution.

- *Option-implied volatility*: it is harder to compute but it is the form of volatility forecast used by practitioners. Its uses, as stated in Bacha [4], are mainly two: (1) compare two option prices and ascertain their value; (2) determine options mispricing, by comparing forecasted implied volatility with actual volatility: if implied volatility is higher than observed volatility, the options might be overpriced.

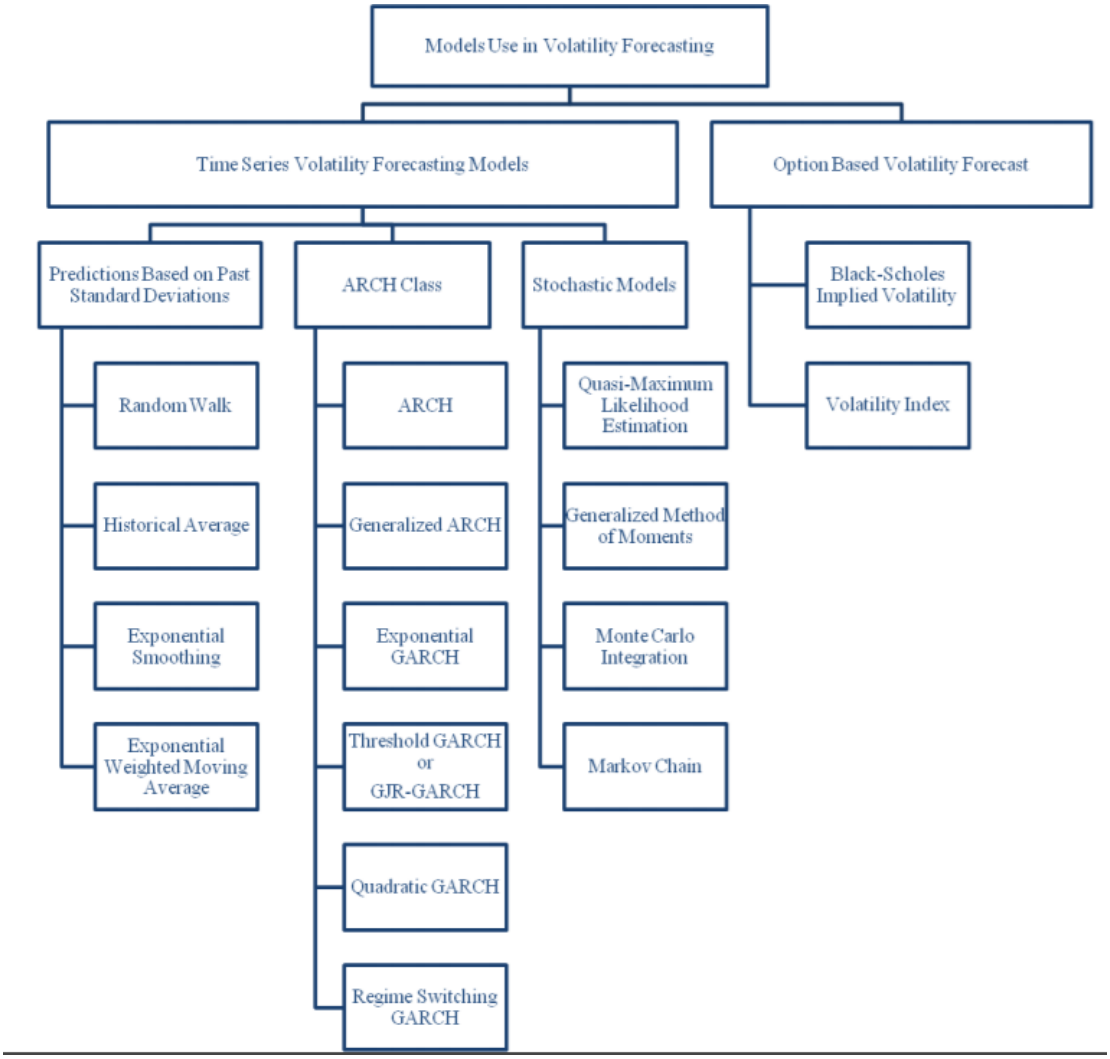

Figure 2.1: Studies on volatility forecasting

Source: Poon and Granger (2003, 2005) [3]

In addition, the implied-volatility of an option includes in its value the expected average volatility of the options' underlying, as showed in Hull [5].

In figure 2.1, we see the results of the study conducted by Poon and Granger [3], showing the models used to predict volatility in 93 different empirical studies. Given these studies, results from different models were compared to evaluate which model could be the best one to solve the forecasting task. Let's present the different approaches for time-series volatility forecasting models:

- *Past Standard deviation*: expected standard deviation is calculated from

historical returns of the underlying, assuming the Black-Scholes conditions, therefore considering volatility a constant in time. The two other models try to solve this issue, since it is empirically evident that volatility is non constant and it is a well-known problem in the Black-Scholes pricing model. Poon and Granger [3] found that the most common model, employed by practitioners, is the GARCH model, by Bollerslev [6], since it uses only three parameters that allow for an infinite number of time-series terms.

- *ARCH Class*: this model relies on an autoregressive time-series approach, considering the output as linearly dependent on previous values of the same time-series (Engle [7]). This model suffers from an inefficiency caused by the heteroscedasticity of the observed process.
- *Stochastic Models*: the underlying of the financial product is considered a random variable, improving flexibility but preventing a closed form formula to solve the problem.

Recently, solutions to the problem that rely on machine learning techniques started being explored. A first approach, by Ke et Yang [8], uses deep learning models, exploiting LSTM to directly forecast future price. This approach shows results better than the Black-Scholes pricing model, but directly estimating price can be limiting when we need to compare prices between models, since errors could be due to many factors in the model equation.

Other studies focus on forecasting implied volatility: Chen *et al.* [9] use LSTM with an attention mechanism, Hull *et al.* [10] use a simple 3-layer neural network to solve the task; both studies show good results in their ability to predict IV. In this thesis we deploy three different ML models (Gradient Boosting Regressor, Artificial Neural Network, Random Forest Regressor), to conduct a preliminary analysis on the applicability of the IV forecast approach to an anomaly detection system for derivative pricing.

2.2 Financial instruments

First of all options are a *financial instrument*. A financial instrument is an asset that may be traded and that is defined by a contract, in order to provide a flow of cash, commodities or other assets between two parties, under a contractual right [11].

There exist two categories of financial instruments:

- *Cash Instruments*: their value is directly influenced and determined by the markets, so it is regulated by the supply and demand law. Some common examples are loans, deposits and bonds.

- *Derivative Instruments*: their value and characteristics are based on the vehicle's underlying components, such as assets, interest rates, or indices. Underlyings on which derivative instruments' value depends on are often represented by *tradable assets*, usually a stock price, but their nature can be a great variety of anything uncertain about the future, from meat price to amount of snow that will fall in a certain season in a given location. Examples of this type of contract are *forward, futures* and *options*.

Cash instruments constitute the vast majority of financial operations that non-insiders perform in their lives: subscribe for a loan, deposit money at the bank, lend money to buy a new car. These instruments are widely used and known to the majority of people, but they only represent the surface of the derivatives world. On the other hand, derivative instruments are widely used by corporations, countries, investment funds and trading companies, to address many problems caused by the uncertainty of the future.

A simple but significant example of why derivatives are essential in the global market is given by the following common problem for companies and farmers. A European sweets factory needs to buy tons of South American coffee for the next year and forecast an expense for this purchase. How can it deal with the uncertainty of weather? Who ensures the company that the commodity prices will be the same next year? Another problem arises for the producer: will he be able to produce enough coffee? What will it cost him? A derivative contract can solve the problem for both parties: the farmer and the company make a deal to sell a certain amount of product at a certain price in the future. The sweets factory is now able to write the expense in its accounts today, without worrying about commodities' price fluctuation, and the farmer is ensured that he will be able to sell its product at a fair price, even if the following year there is a huge offer of its commodity. This is a typical *forward* contract [12].

Forward contracts The contract in the example is called *forward* because it is a deal that formalizes and obliges two parties in buying or selling some asset at a certain date in the future, at a certain price. The buyer is said to assume a *long position*, while the seller a *short position*.
Forward contracts can be stipulated on different underlyings, in the example it was a commodity, but another crucial reason for such contracts is for multinational corporation to protect itself from currency rates fluctuation, stipulating a currency forward.

Future contracts Like *forward* contracts, a *future* contract is a deal between two parties to buy or sell an asset at a certain future date, at a given price. The difference between the two typologies of contract lies in their regulamentation: forwards are usually contracted between two parties in private, in what is called a *OTC (over the counter)* contract, while futures are standardized by some Market *e.g.* Chicago Board of Trade (CBOT), which ensures to the two parties that their contract will be honoured, since they probably do not know each other. The underlying assets on which contracts are stipulated are the most various and they all fall into two categories: *commodities* and *financial assets*. Some examples of commodities are gold, pork bellies, lumber, copper; while financial assests are tipically stock indexes, currencies, Treasury bonds.
The price of a future contract is determined by the law of supply and demand: if there are more investors that want to sell (long position) than investors that want to buy (short position), the price goes up. When the opposite is the case, price goes down.

Options Options are traded on both public stock exchange, referred to as *Exchange Traded Options* (ETO's) and in OTC market *Over The Counter Options* (OTC's).
There exist two types of options: *calls* and *puts*. *Call options* give the buyer the right to buy a certain asset by a certain date, at a given price. On the other hand *put options* give the right to sell a certain asset by a certain date, at a given price. In the contract, they are both described by three factors:

- Exercise (Strike) Price: the determined price in the contract for the option's underlying
- Expiration Date (Maturity): date at which the option expires
- American or European: American options can be exercised anytime before maturity, while European options can only be exercised at maturity date. In general, European options are easier to analyze and less traded than American's, whose properties are often derived by their corresponding European option.

Figure 2.2 shows profit changes for call and put options as function of stock price. *Calls* price decreases as strike price increases, while *puts* price increases. They both increase their price when maturity increases.

A fundamental concept of options is that their owner has the right to buy/sell and it is not necessarily obliged to exercise it. This is what distinguishes options from forwards and futures, along the fact that entering an option contract has a cost, which is not true for the two other types of derivatives.

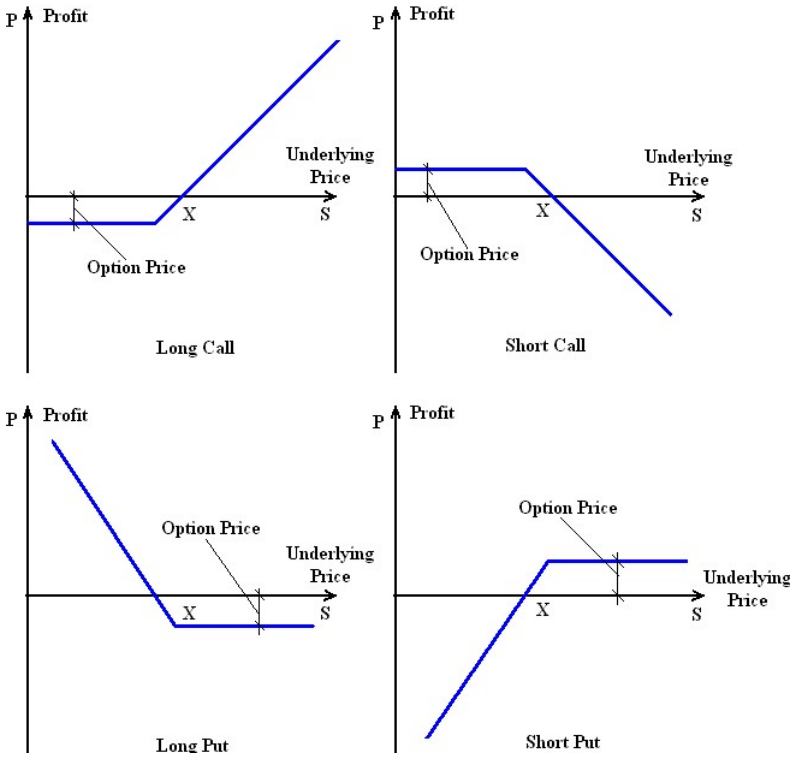

Figure 2.2: Long call, long put, short call and short put price variations as function of underlying variations.

Source: www.briandcolwell.com (May 2021)

2.3 Options

In this section we illustrate the fundamental properties of options on stocks, which hold for options on indexes as well, since our project relies on options on the S&P500 index [12].
There are six factors that influence the price of an option on an index:

- current option price: S_0
- strike (exercise) price: K
- time-to-maturity: T
- volatility of underlying stock or index: σ
- risk-free interest rate: r
- expected dividends during option's life, with the current value: D

Stock price and strike price The value of a *call* (P_{call}) at current time t is equal to the difference between stock price (underlying) and strike price:

$$P_{call} = S_0 - K \tag{2.1}$$

This means that *calls* gain value if option's price grows and lose value if strike price decreases. The value of a *put* option is instead given by:

$$P_{put} = K - S_0 \tag{2.2}$$

so they behave in the opposite manner. Figure 2.3 shows *calls* and *puts* price changes as function of stock price, exercise price and time-to-maturity.
This price is referred to as the *intrinsic value* of the option and describes what would be the profit or loss of the position if the option was exercised immediately. There are three possible situations for a *call* and the opposite holds for a *put*:

- *In-the-money*: $S_0 > K$, the option results in profit
- *At-the-money*: $S_0 = K$, the option makes no profit or loss
- *Out-the-money*: $S_0 < K$, the option would result in a loss of money if exercised immediately

This is the so called *moneyness* of an option.

Time to maturity We now take into consideration how time influences price. American options decrease in value as they get closer to maturity date, since the opportunities of exercise decrease. If we consider two options that only differ in their maturity date, the option with a greater time to maturity must be at least as expensive as the one with less time-to-maturity, since it is given more exercise opportunities than the latter.
The same concept holds for European options, with one exception. Imagine two European options: one has a time to maturity of 1 month and the other one expires in 2 months; if a large dividend is paid in 6 weeks, with the consequence of stock price going down, we have that the 2-months expiry option is worth less that the 1-month one [12].

Volatility Volatility is described in greater details in sections 2.4 and 2.5, since it is the fundamental parameter of this research, but let's illustrate here some important concepts.
This variable describes a measure of uncertainty on the future behaviour of a stock, or index price. If volatility increases, the probability that the *performance* of the underlying improves or decreases by a considerable amount increases. The upside

and downside movements balance each other for whomever holds the underlying, while it affects in different ways the owners of *calls* and *puts*:

- *Call Holder* benefits of rises but it has a *downside risk* because it cannot lose more than the premium that was paid to enter the option contract
- *Put Holder* benefits of falls but has a *downside risk* in the case of rises of the underlying's value

Therefore *calls* and *puts* value increases as volatility increases.

Risk-free rate The risk-free rate represents the interest an investor would expect from an absolutely risk-free investment over a specified period of time [13]. Every investment has a risk but instruments like US Treasury Bonds or German Bonds can be considered secure investments and their yields taken as benchmark.
The link between this variable and option price is less clear than the above mentioned relations. If risk-free interest rates increase, the market expects the growth rates of stocks to increase as well. For options' holders, the interest-rates growth decreases the current value of future cash-flows. The combination of these two factors results in:

- *call prices* increase because of the increase of the value of stocks
- *put prices* decrease

But this is not always the case, since an increase of risk-free rates does not always and directly imply an increase in stock prices. The opposite holds for a decrease in risk-free rates, which does not imply a fall in stock prices.

Dividends A dividend is the distribution of some of a company's earnings to a class of its shareholders, as determined by the company's board of directors [14]. Is is substantially a reward paid to the shareholders for their investment in a company, which usually comes from the company's profits, even if, in general, a lot of the earnings are kept within the company for future investments and to maintain current activities.
Dividends have an impact on company's share value since the action of paying dividends means that money that is part of the company's assets will be spent as a regular expense, without an actual investment, if not a reward for investors that trusted the company. This has an impact on shares price: it goes up on the announcement date of a certain amount, to then go back to roughly the original amount after the pay date. Companies can decide to pay either regular or irregular dividends, depending on their financial situation and operating sectors. The consequence of a dividend is that option prices on the company's underlying

shares decrease during the day of dividend's distribution, resulting in a negative event for a *call owner* and a positive event for a *put owner*. So there exist a positive relation between *put and dividends* and a negative one between *call and dividends*.

2.3.1 Options pricing: Black-Scholes-Merton Model

In the early Seventies, Fischer Black, Myrion Schole and Robert Merton, provided a fundamental contribute to option pricing theory, developing the famous Black-Scholes-Merton model [15]. The consequences of this model were so huge for the global market, changing the way traders operate and setup finance covers, that the group won a Nobel Prize for the economy in 1997.
The model consists of a differential equation that solves for option prices.

Fundamental concepts of Black-Scholes-Merton Model The differential equation of Black-Scholes-Merton has to be satisfied by the price of every financial derivative that depends on the price of a variable underlying that does not pay any dividends. In order to evaluate options we need to create a risk-free portfolio[1] with options and stocks, where the rate of return is equal to the risk-free interest rate r. The reason why it is possible to build a risk-free portfolio is because both option and stock prices are influenced by the same source of uncertainty, which is the stock price itself. In a short time interval the price of a *call* is positively correlated with the price of the underlying, while a perfect negative correlation exists with *put* price. On that note, any loss in options is a gain in stocks and vice-versa, so that we can know with certainty, is a short time interval, the total amount of our position. Let's stress that this is true only for an instantaneous time interval.

Assumptions Black-Scholes-Merton model relies on some very important assumptions:

- The option is European and can only be exercised at expiration
- No dividends are paid out during the life of the option
- Markets are efficient, meaning that market movements cannot be predicted
- There are no transaction costs in buying the option
- The risk-free rate and volatility of the underlying are known and constant
- The returns on the underlying asset are log-normally distributed

[1] A portfolio is a collection of financial investments like stocks, bonds, commodities, cash, and cash equivalents

Some of these assumptions can actually be mitigated, for example we could have stochastic risk-free rates and σ and r known functions of t.

Risk-free valuation Another fundamental principle in the evaluation of derivatives is the *risk-neutral valuation* principle. It states that when we evaluate derivatives we can assume that investors are *risk-neutral*, meaning that investors do not ask an higher return rate as a premium for the risk they are taking.
It is obvious that we do not live in a risk-free world, because the higher the risk, the higher return rates are required by investors. Surprisingly, derivative's prices are the same in both risk-free and real world. If derivatives are very risky products, why does not the attitude against risk count? When we evaluate the price of a derivative in terms of price of its underlying, risky attitudes do not matter, since the relationship between derivatives' prices and underlying stocks is unchanged. This has two implications that simplify the valuation of a derivative:

- The rate of return that is expected from any asset is equal to the risk-free rate
- The rate used to discount an expected payoff is equal to the risk-free rate

In other words, the probability p that a certain stock increases its value, grows in average with the risk-free yield. So the expected value of an increase in stock price is equal to the risk-free rate and this can be shown for every model that describes the evolution of a stock price.
To summarize, when evaluating a derivative with this principle:

1. Calculate probabilities associated to stock prices in a risk-free world
2. Calculate the expected value of the derivative with these probabilities
3. Discount the expected value of the derivative with risk-free rate

Merton solved the problem with a different and more general approach than Black and Scholes and it is here presented. *Price* refers to the price of an option.

Log-normal distribution of option prices In the construction of the model of an option price, we can assume that the immediate rate of return of an option price is normally distributed. Adopting the following symbols:

- μ: expected annual rate of return
- σ: volatility
- ΔS: price variation in time interval Δt
- $\varphi(m, v)$: normal distribution with mean m and variance v

the rate or price variation in the interval Δt has mean $\mu\Delta t$ and standard deviation $\sigma\sqrt{\Delta t}$. So we can state:

$$\frac{\Delta S}{S} \sim \varphi\left(\mu\Delta t,\ \sigma^2\Delta t\right) \tag{2.3}$$

by using Ito's Lemma to study the log process $\ln(S)$.
The model implies

$$\ln(S_T) - \ln(S_0) \sim \varphi\left[\left(\mu - \frac{\sigma^2}{2}\right)T,\ \sigma^2\,T\right] \tag{2.4}$$

Follows that

$$\ln\left(\frac{S_T}{S_0}\right) \sim \varphi\left[\left(\mu - \frac{\sigma^2}{2}\right)T,\ \sigma^2\,T\right] \tag{2.5}$$

and

$$\ln(S_T) \sim \varphi\left[\left(\ln\left(S_0\right) + \mu - \frac{\sigma^2}{2}\right)T,\ \sigma^2\,T\right] \tag{2.6}$$

where S_T is option price at time T and S_0 is price at time 0.
The equation shows that $\ln(S_T)$ is normal, so S_T is log-normal.
$\ln(S_T)$ has mean: $\left(\ln\left(S_0\right) + \mu - \frac{\sigma^2}{2}\right)T$ and standard deviation $\sigma\sqrt{T}$.

A log-normal distribution can assume any value in the interval $(0, +\inf)$. It is also asymmetric, implying that mean, median and mode are different from each other. Given the above equation and using log-normal distribution properties, we can show that the expected value of S_T is given by

$$E\left(S_T\right) = S_0 e^{\mu T} \tag{2.7}$$

And its variance by

$$var\left(S_T\right) = S_0^2\, e^{2\mu T}\left(e^{\sigma^2 T} - 1\right) \tag{2.8}$$

This justifies the definition of μ as expected rate of return.

Rate of Return Distribution Log-normal properties of stock prices can be used to analyze the distribution of stock's rate of return (continuously compounded) from time 0 to T. With η continuously compounded annual rate of return from 0 to T:

$$S_T = S_0\, e^{\eta T} \tag{2.9}$$

$$\eta = \frac{1}{T}\ln\left(\frac{S_T}{S_0}\right) \tag{2.10}$$

And using equation (2.5):

$$\eta = \varphi\left[\left(\ln\left(S_0\right) + \mu - \frac{\sigma^2}{2}\right),\ \frac{\sigma^2}{T}\right] \tag{2.11}$$

This result shows that annual rate of return follows a normal distribution with mean $\mu - \frac{\sigma^2}{2}$ and standard deviation $\frac{\sigma}{\sqrt{T}}$.
When time to maturity T increases, the standard deviation of the distribution decreases: we are more confident that the yield for the next 20 years will be closer to the mean than the yield of next year.

Black-Scholes equation Given the following boundary conditions:

- *European Call*: $f = \max(S - K, 0)$ when $t = T$
- *European Put*: $f = \max(K - S, 0)$ when $t = T$

The Black-Scholes formula for European *calls* and *puts* evaluation, for options that do not pay any dividends, is the following:

$$c = S_0 N(d_1) - K e^{-rT} N(d_2) \tag{2.12}$$

$$p = K e^{-rT} N(-d_2) - S_0 N(-d_1) \tag{2.13}$$

with

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \tag{2.14}$$

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \tag{2.15}$$

$N(x)$ represent a normal distribution with mean = 0 and standard deviation = 1, so it measures the probability that a standardized normal variable assumes a value less that x. The other variables are:

- S_0: stock price at $t = 0$
- K: strike price
- r: risk-free rate
- σ: option's price volatility
- T: time to maturity

We can obtain the BS equation by exploiting the concept of risk neutral evaluation. In a risk-free world, the expected value of a *call* is

$$\hat{E}\left[\max(S_t - K, 0)\right] \tag{2.16}$$

From this equation we can get the current value of the *call* option by discounting its expected value at expiration

$$c = e^{-rT}\hat{E}\left[\max(S_t - K, 0)\right] \tag{2.17}$$

From equation 2.17 we can obtain equation 2.12, which can be rewritten as

$$c = e^{-rT}\left[S_0 N(d_1)e^{rT} - KN(d_2)\right] \tag{2.18}$$

The expression $N(d_2)$ is a cumulative normal distribution, which represents the probability that an option is exercised, with $KN(d_2)$ being the exercise price multiplied by the probability of the option being exercised. $S_0 N(d_1)e^{rT}$ is the expected value of a variable that is equal to S_T when $S_T > 0$, and that is otherwise equal to zero.

Black-Scholes equation: properties In the case of extreme values, we show that the behaviour of BS equations returns the expected results for *call options*.

- *High value of stock option price, high* S_0. Here the *call* is most certainly going to be exercised, so that the options starts to look like a *forward* contract with exercise price K. Expected option value is $S_0 - Ke^{-rT}$, if we treat the option as a *forward*; this result is actually the one that equation 2.12 returns, since for an high value of S_0, d_1 and d_2 assume large values and as a consequence $N(d_1)$ and $N(d_2)$ are close to 1. The opposite holds for a *put* option, with $N(-d_1)$ and $N(-d_2)$ close to zero.

- *Volatility close to 0,* $\sigma \to 0$: option price at maturity S_t can now be considered risk-free, since it will not move, and this term can be replaced with risk-free discounted price $S_0 e^{rT}$, resulting in a price of $\max(S_0 e^{rT} - K, 0)$. This still needs to be discounted with the expected rate r, giving a current value for the option of

  $$e^{-rT}\max(S_0 e^{rT} - K, 0) = \max(S_0 - Ke^{-rT}, 0)$$

  How does this verify equation 2.12? If $S_0 < Ke^{-rT}$ we get that $\ln(S_0/K) + rT < 0$. With $\sigma \to 0$, then d_1 and $d_2 \to -\infty$, causing the distributions $N(d_1)$ and $N(d_2)$ tend to 0 and equation 2.12 gives a price equal to zero. So, the price of a *call* with $\sigma \to 0$, is always $\max(S_0 - Ke^{-rT}, 0)$.
  The same holds for a *put*, resulting in $\max(Ke^{-rT} - S_0, 0)$

2.4 Historical Volatility

As introduced in paragraph 2.3, volatility describes a measure of uncertainty on the future behaviour of a stock price, or index price. Typical volatility values of options fall between 15% and 60% for a whole year.
Equation (2.11) can actually give us a sense of volatility's definition: *the volatility of an option price is the standard deviation of rate of return, continuously compounded, over a period of one year.*
In equation (2.3) we can see that $\sigma\sqrt{\Delta t}$ is the standard deviation of the variation rate of a stock price $\frac{\Delta S}{S}$.

Volatility estimation on historical data The volatility of a stock price, just like the volatility of an index, can be estimated by the time-series of its price variations.
Let's take an option whose price is taken at regular intervals, which could be daily, weekly, monthly or yearly. Let:

- $n+1$: number of observation
- S_i: stock price at time interval i
- τ interval length expressed in years

and

$$u_i = \ln\left(\frac{S_i}{S_{i-1}}\right) \tag{2.19}$$

for i = 1, 2, 3,..., n.
The standard deviation of the log of variations u_i is s, denoting $\bar{u}$ as the mean of all u_i

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i - u)^2} \tag{2.20}$$

and we get

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}u_i^2 - \frac{1}{n(n-1)}\left(\sum_{i=1}^{n}u_i\right)} \tag{2.21}$$

Given equation (2.5), the standard deviation oh u_i is $\sigma\sqrt{\tau}$. This means that s is an estimate of $\sigma\sqrt{\tau}$ and that σ can be estimated with $\hat{\sigma}$, where

$$\hat{\sigma} = \frac{s}{\sqrt{\tau}}$$

With an error of approximately $\hat{\sigma}/\sqrt{2n}$. The greater the number of samples, then better the approximation. This statement holds mathematically but we need to

take into account that σ actually changes over time and old data is probably irrelevant, if not dangerous, to forecast the future. Empirically, a range from 90 to 180 days seems to be work pretty well. Another rule of thumb is to use as much data as the number of days we want to forecast, for example a year of data for an option that expires in a year.

Day-count convention It is important to have a convention that describes how time to maturity is calculated. Do we want to include all days in a year or just the working days? Empirical studies like [16] have shown that volatility is much higher when markets are open than when they are closed. There are many theories about why volatility behaves in this way, many believe that it is because of the information that reaches the markets is greater during the weekend, but this has been proved wrong. The accepted interpretation is that volatility changes are actually caused by trades themselves.
When calculating time to maturity for an option, we need to choose a day-count convention, that may be as simple as dividing by 365 (Actual/365 Fixed), or more sophisticated strategies that take into account leap years like Actual/Actual ISDA, or divide by number of working days in a year, generally assumed to be 252.

2.5 Implied Volatility

The only parameter than cannot be directly observed in the Black-Scholes formulas is volatility. In section 2.4 we showed how volatility can be estimated from the time-series of option prices, but there is another approach that is fundamental and worth understanding very well, especially for in the scope of this thesis.
These two types of volatility differ in their meaning and usage. *Historical volatility* is a *backward looking* measure, since it refers to past prices and it could be hazardous to use it as a feature for forecasting option price. *Implied volatility* is instead a *forward looking* measure, used by traders to estimate market's sentiment about the fluctuations of a certain underlying. This method is so extensively used that traders often quote implied volatilities instead on option prices, especially because variations in volatility are steadier than option prices.

IV calculation *Implied volatility* is calculated starting from option prices quoted on the market, by using an option pricing model. Since it is the only factor in the model that isn't directly observable in the market, we must invert the pricing model and solve for IV. Unfortunately, the BS model cannot be inverted to obtain a closed formula to calculate IV, so an iterative process is needed; given option price, time-to-maturity, risk-free rate, option strike and current stock price, we can

determine IV with, for example, the *Newton-Raphson* method. But how do we get option price? This method is only applicable to options that are quoted on the market, so that the price is known and dependent on the law of supply and demand. Of course, another factor that influences IV is the time value of an option: a short-dated option is more likely to have a low IV, whereas a long-dated option tends to have an high value of IV.
The pros of this parameter are:

- Helps quantifying market sentiment and the uncertainty related to it
- Helps in setting an option price
- Helps in determining a trading strategy, because of the sense of variation that it shows

On the other hand:

- It is only based on prices, since it is calculated by reverting the pricing model
- It is sensitive to external factors like news
- It gives a sense of the movement but not of the direction

CBOE Volatility Index (VIX) The CBOE Volatility Index (VIX) is a real-time index that represents the market's expectations for the relative strength of near-term price changes of the S&P 500 index (SPX). Because it is derived from the prices of SPX index options with near-term expiration dates, it generates a 30-day forward projection of volatility. It is an important index in the world of trading and investment because it provides a quantifiable measure of market risk and investors' sentiments [17]. VIX is often referred to as the *fear index* and it easy to understand why it is so by looking at figure 2.4. As we can see, during the economic crisis of 2008, with the failure of Lehman Brothers, the "fear" in the markets and the uncertainty was skyrocketing, resulting in very high volatility expectations.

2.6 Greeks

"Greeks" is a term used in the options market to describe the different dimensions of risk involved in taking an options position. These variables are called Greeks because they are typically associated with Greek symbols. Each risk variable is a result of an imperfect assumption or relationship of the option with another underlying variable. Traders use different Greek values, such as delta, theta, and others, to assess options risk and manage option portfolios [18].

Let's briefly introduce them and in the specific the *practitioner's Black-Scholes delta*, used in this work.

- *Delta*: Δ represents the rate of change between the option's price and a \$1 change in the underlying asset's price. In other words, the price sensitivity of the option is relative to the underlying asset. Delta of a call option has a range between zero and one, while the delta of a put option has a range between zero and negative one. For example, assume an investor is long a call option with a delta of 0.50. Therefore, if the underlying stock increases by \$1, the option's price would theoretically increase by 50 cents.

- *Theta*: Θ represents the rate of change between the option price and time, or time sensitivity - sometimes known as an option's time decay. Theta indicates the amount an option's price would decrease as the time to expiration decreases, all else equal. For example, assume an investor is long an option with a theta of -0.50. The option's price would decrease by 50 cents every day that passes, all else being equal.
  Theta increases when options are at-the-money, and decreases when options are in- and out-of-the money. Options closer to expiration also have accelerating time decay. Long calls and long puts will usually have negative Theta; short calls and short puts will have positive Theta. By comparison, an instrument whose value is not eroded by time, such as a stock, would have zero Theta.

- *Gamma*: Γ represents the rate of change between an option's delta and the underlying asset's price. This is called second-order (second-derivative) price sensitivity. Gamma indicates the amount the delta would change given a \$1 move in the underlying security. For example, assume an investor is long one call option on hypothetical stock X. The call option has a delta of 0.50 and a gamma of 0.10. Therefore, if stock X increases or decreases by \$1, the call option's delta would increase or decrease by 0.10.
  Gamma is used to determine how stable an option's delta is: higher gamma values indicate that delta could change dramatically in response to even small movements in the underlying's price.Gamma is higher for options that are at-the-money and lower for options that are in- and out-of-the-money, and accelerates in magnitude as expiration approaches.

- *Vega*: ν represents the rate of change between an option's value and the underlying asset's implied volatility. This is the option's sensitivity to volatility. Vega indicates the amount an option's price changes given a 1% change in implied volatility. For example, an option with a Vega of 0.10 indicates the option's value is expected to change by 10 cents if the implied volatility changes by 1%.

Because increased volatility implies that the underlying instrument is more likely to experience extreme values, a rise in volatility will correspondingly increase the value of an option. Conversely, a decrease in volatility will negatively affect the value of the option. Vega is at its maximum for at-the-money options that have longer times until expiration.

- *Rho*: ϱ represents the rate of change between an option's value and a 1% change in the interest rate. This measures sensitivity to the interest rate. For example, assume a call option has a rho of 0.05 and a price of \$1.25. If interest rates rise by 1%, the value of the call option would increase to \$1.30, all else being equal. The opposite is true for put options. Rho is greatest for at-the-money options with long times until expiration.

Delta Hedging As mentioned before, the Δ of an option is defined as the derivative of option price with respect to the price of the underlying, so it is equal to the slope of curve that represents the relation between option price and underlying price.

$$\Delta = \frac{\partial c}{\partial S}$$

Delta hedging is an option trading strategy that aims to reduce, or *hedge*, the directional risk associated with price movements in the underlying asset. By looking at the delta of an option, traders try to reduce risks associated with the option by buying a certain amount of the option's underlying.
Let's see an example of how this can be done exploiting Black-Scholes' results on an European call option. It can be proved that the delta associated to such an option is equal to $N(d_1)$, where d_1 is defined in equation 2.14. In order to cover a short position on a call, the trader has to buy $N(d_1)$ stocks of the underlying, so that its position reaches a *delta neutral* state where there cannot be a loss.
This cover operation is usually done not only on a single option but on an entire portfolio, and the position is *delta neutral* only for a short amount of time, so this cover strategy is said to be *dynamic hedging scheme*, where the portfolio needs periodic rebalancing [2].

Moneyness and Black-Scholes Practitioner's Delta The moneyness of an option describes the current value of an option, as explained in paragraph 2.3. *Practitioner's Delta*, just like the Greek delta, can be used to measure moneyness since it relates underlying price and strike price. The difference with the common delta lies in its calculation, where the volatility parameter in BS equation is replaced by implied volatility. Instead of looking at strike price and underlying

price difference, as in paragraph 2.3, we look at practitioner's delta values, for a *call*:

- $\delta_{BS} > 0.5$: *in-the-money*
- $\delta_{BS} = 0.5$: *at-the-money*
- $\delta_{BS} < 0.5$: *out-the-money*

while for a *put*:

- $\delta_{BS} > -0.5$: *in-the-money*
- $\delta_{BS} = 0.5$: *at-the-money*
- $\delta_{BS} < -0.5$: *out-the-money*

Practitioner's BS delta can also be used to estimate a minimum variance delta for hedging. This is a hedge ratio that takes account of expected volatility changes as well as the change in the underlying asset price. Its value is given by the formula:

$$\Delta = \delta_{BS} + \nu_{BS}\frac{\partial \sigma_{imp}}{\partial S}$$

Where practitioner vega (ν_{BS}) is defined just like the traditional greek vega but the volatility term is is substituted with implied volatility. The term $\frac{\partial \sigma_{imp}}{\partial S}$ is estimated from empirical results.
In another research that exploits machine learning in order to model implied volatility changes, Nian et al [19], results show that such approach can lead to improvements in delta hedging strategies.

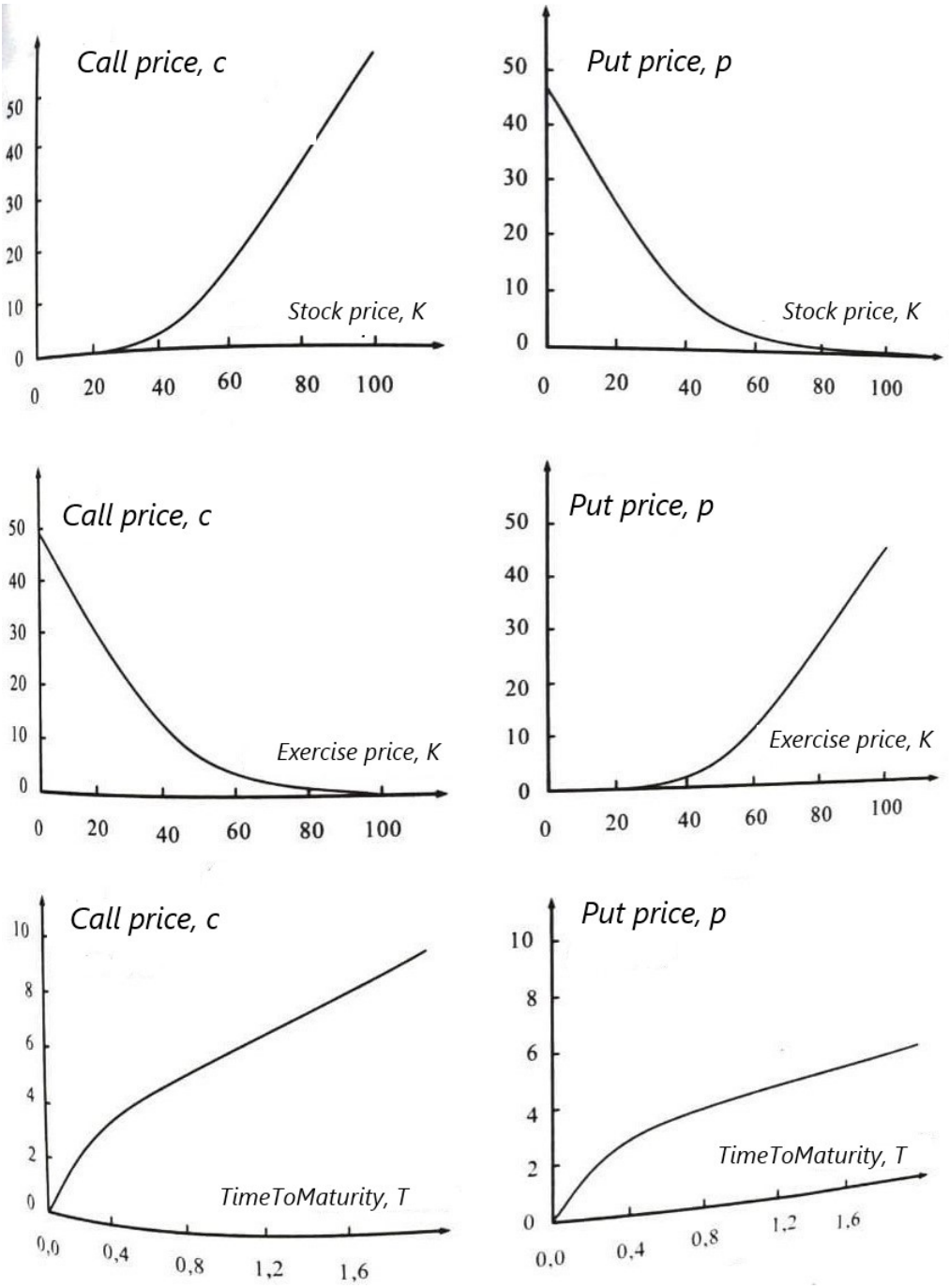

Figure 2.3: Option price variations with respect to strike price, underlying price and time-to-maturity.

Source: Options, futures and other derivatives. John Hull, 2012 [12]

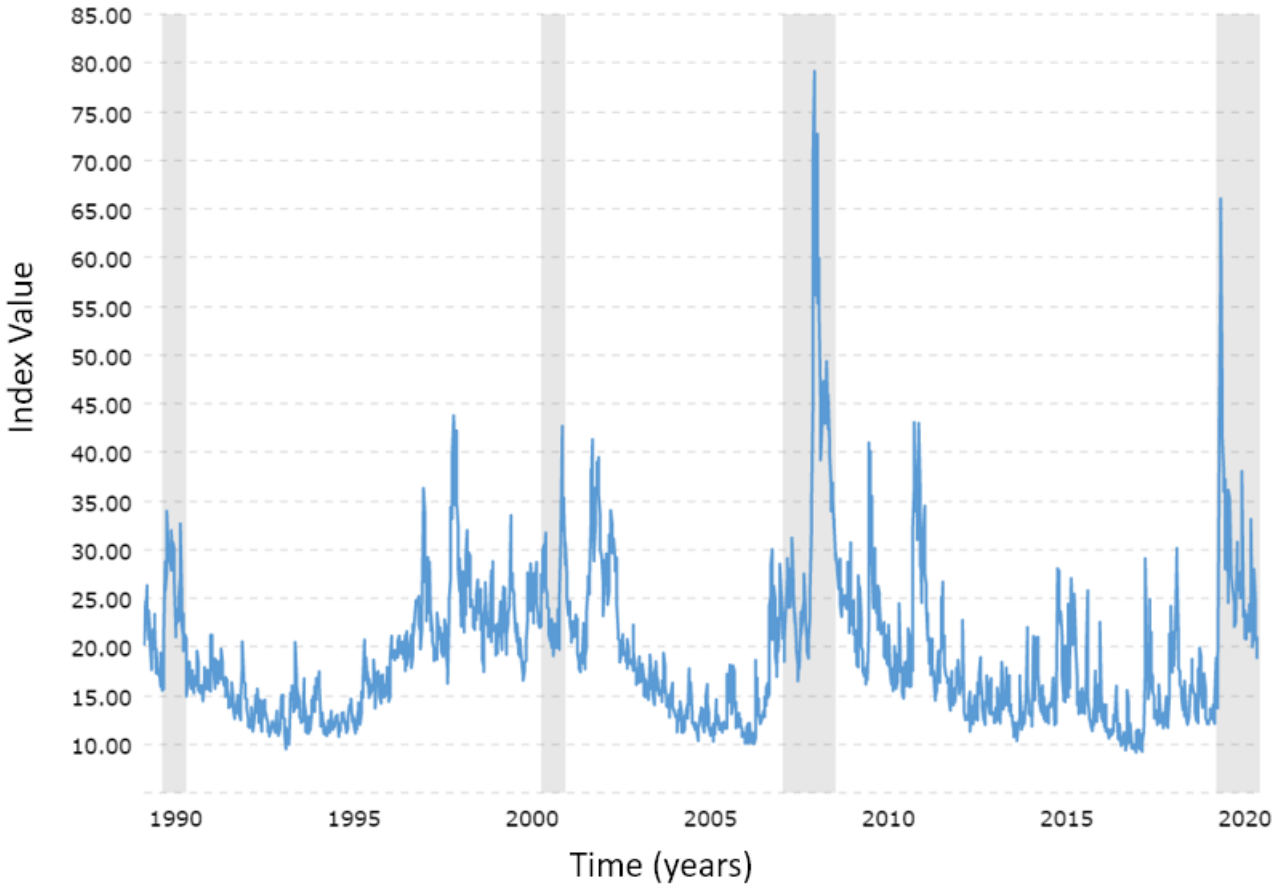

Figure 2.4: VIX index from 1990 to 2020

Source: www.macrotrends.net

Chapter 3

Data and methodology

3.1 Dataset

In this section we present the various steps involved in the creation of the dataset and provide a detailed description of the resulting data. This is a fundamental part of the thesis since many considerations have been done under the financial point of view, along with a lot of preprocessing work needed to obtain the desired final dataset.

3.1.1 Data Preprocessing

Data used in this thesis has been kindly bought by the company *Myrios*, which has collaborated in the problem's definition and it has supported us in the specific financial aspects of the work.
Data was bought from the website *discountoptiondata.com* but it still needed a lot of preprocessing in order to obtain the final dataset that we wanted. It included historical options data from 2019 to 2021 on options traded on roughly 10000 different underlyings.
Data was provided via *.csv* files, with fairly consistent column names across the 16 years (2005-2021), with one file for each trading day, including all tickers. Once downloaded, the total file size consisted of 33.8GB zipped and 215GB once unzipped. The included features were: Option Key, Ticker Symbol, ExpirationDate, AskPrice, AskSize, BidPrice, BidSize, LastPrice, Put or Call, StrikePrice, Volume, Underlying Price, Detection Date.

The preprocessing stage of the project transformed the initial data to:

- Only options on index S&P-500
- Time interval: April 2010- February 2021

- Features:
  - Option Moneyness, measured with practitioner's Black-Scholes delta (δ_{BS})
  - S&P-500 daily return
  - Time-to-maturity
  - Implied Volatility

The chosen features are the result of an analysis done by Hull (2017) [5] and Hull (2019) [10], where he conducted a research on the most important parameters that influence the price of an option and its characteristics, confirming a well consolidate theory by the financial community. Time-to-maturity is essential because the further in time an option expires, the more uncertain the value of the underlying will be. S&P-500 daily return concretely represents the current changes in the underlying and together with δ_{BS}, they express the expected changes of the underlying. All this information should be captured by the target variable IV, given by some function learnt directly from data.

Figure 3.1 summarizes the preprocessing pipeline, which was divided in many transformation steps, each one with some important considerations that have been done under the financial point of view, along with some additional steps to get the data needed to calculate the final features. This is an outline of the work that has been done:

1. Filter out all options that are not on the S&P-500 index
2. Merge all *.csv* files together
3. Drop redundant features and useless features for our use case, like bid and ask prices, since we work on close prices only
4. Build new features from the previous ones like practitioner's BS delta and implied volatility, calculated using another dataset containing risk-free rates of US dollar currency.

Since the thesis is based on the work of [10], we preprocessed the data in order to obtain a dataset that is close to the one used in Hull's work. This includes:

- Retain options where the information is complete
- Remove options with remaining lives less than 14 days
- Remove options where practitioner Black-Scholes delta is less than 0.05 or greater than 0.95

Aggregation and filter stage First of all, we had to merge together 33.8 GB of *.csv* files and filter them according to the aforementioned parameters. To accomplish this task, we used the python library *pandas*, which provides great filtering and merging capabilities, in a reasonable amount of time, while other common instruments to deal with *.csv* files such as *Excel* simply could not handle the huge amount of data. In addition, the whole project has been carried out in the Python programming language, the most common tool nowadays to deal with big data and machine learning, especially because of the many freely available ML libraries, its simplicity and its portability .
Since the original dataset was so massive, we suspected that some errors could be present. Right after doing some basic preprocessing, like standardizing column name across all the dataset, we checked for option price values that could not be possibly quoted on the market. For some options the price was zero during their whole existence, so we removed them. Another important check was the evaluation of options' intrinsic value. For *call options*, their intrinsic value is calculated as underlying price minus strike price:

$$P_{intrinsic} = P_{underlying} - P_{strike}$$

This value cannot be negative since the maximum amount of money that can be lost when buying a call option is the *premium*. Let's keep in mind that an option gives the *right* to exercise it and not the obligation to do so. Therefore, options with an intrinsic value less than zero were marked as mistakes and discarded from the dataset.

Risk-free yield curve computation A risk-free rate of return is the theoretical return form an investment that is assumed to have no risk associated with it. In practice, it does not exist, because every investment carries a risk associated with it. We can obtain an estimate of such value by looking at the yields of Treasury bond for the market that we are taking into consideration. In this project we are dealing with options quoted on the S&p500 index, which includes the top 500 companies quoted on Nasdaq; since it is an American market, the associated risk-free rates must be taken by looking at Treasury rates emitted by the government of the United States of America.

The dataset of risk-free rates for *US dollar* is made out of two *.csv* files kindly provided by the company *Myrios*. The two files only differed for their columns but the conceptual steps to obtain a risk-free curve were the same:

- Define estimation points to use for curve interpolation: we choose to interpolate the risk-free curve using yield values at 1M, 3M, 6M, 1Y, 2Y. Since options' life usually spans over a couple months at best, this values should give us a good interpolated curve.

- Choose an interpolation type: we interpolated the curve using a cubic spline, with the method *scipy.interpolate.CubicSpline* from the python library *SciPy* [20].

- For each day we interpolated the curve and saved the results in a dictionary containing the pairs (Date, Interpolation Coefficients)

- Saved the results to use them later to calculate implied volatility and practitioner's BS delta with another python library *Mibian* [21].

An example of an interpolated yield curve is provided in figure 3.2.

3.1.2 Data Description

The resulting dataset is made out of 272 thousands observations on daily volatility changes for 11003 options during the time interval from April 2010 to January 2021. Each option time-series has been filtered as described in the previous section (3.1.1). In average, each option has been observed for 29 days, considering that the last 14 days of the options have been truncated, it is a good time window for our monitoring, which allows us to analyze the trends and behaviour of implied volatility across time. The average value of implied volatility for the options in the dataset is consistent with the average value calculated on quoted call options on the market over the last 10 years, which is 18.1%.

S&P500 index The S&P500, or the Standard & Poor's 500 Index, is widely regarded as the best single gauge of large-cap U.S. equities and serves as the foundation for a wide range of investment products. The index includes 500 leading companies such as Apple, Microsoft, Google and captures approximately 80% coverage of available market capitalization [22]. Depending on the markets or website, its ticker symbol can assume different names including $SPX, GSPC and INX.
The S&P 500 is one of the most widely quoted American indexes because it represents the largest publicly traded corporations in the U.S. The S&P 500 focuses on the U.S. market's large-cap sector and is also a float-weighted index, meaning company market capitalizations are adjusted by the number of shares available for public trading [23].
Figure 3.3 shows the index from 1990 to 2020. It is interesting to notice how the index reflects the economy in its fluctuations: figure 3.3 shows a drastic drop in 2008, concomitantly with the global financial crisis caused by the excessive risk-taking by banks and the bursting of the United States housing bubble.

3.2 Machine learning models

In this section we illustrate the different machine learning algorithms that can solve the implied volatility forecast problem, under a theoretical point of view, focusing on their specific regression capabilities.

Machine learning models can be used with two main different objectives:

- *Classification*: the algorithm aims at predicting whether a given input belongs to a certain output class. A classical example of an application is a spam detector, which given an email, detects if it is a spam or not, in this case we talk about a *binary classification* problem; more generally, such algorithm can be extended to detect many different output classes, like in the image classification domain.

- *Regression*: aims at predicting a continuous output value given a set of input predictors. The resulting model is some function that approximates a function which maps inputs to outputs with an estimate coming from training data.

All the models used in the thesis are *regression models*, since what we aim to obtain is a continuous value for implied volatility.

3.2.1 Artificial Neural Networks

Firstly introduced by McCulloch and Pitts in 1943 [24], ANN represented the first effort to model the way in which the human brain can perform computations; after another wave of success in the 80's when artificial intelligence gained a lot of attention, we now face a dramatic increase in this research field due to the powerful computational capabilities of modern hardware. The success of the applications of this technology has been raising in the last decade, revolutionizing an incredible amount of fields, from advertising to medicine and biology, including the financial sector.
This growth is not only noticeable in the more specific field of *deep learning* but more generally in the *machine learning* approach to problem solving. This approach has always been employed to build parametric models that learn from data, but with the restriction of approximating functions at a given order and with the need of a strong competence is the application domain by practitioners.
For example, forecasting implied volatility is a problem that has already been faced in 2017 by Hull and White [5], using a classical machine learning parametric approach. It consists of a simple analytic 3 parameters model that can be estimated using linear regression:

$$E(\Delta\sigma_{imp}) = \frac{\Delta S}{S}\frac{a + b\delta_{BS} + c\delta_B^2 S}{\sqrt{T}}$$

Since we already know that the relationship between the features and the implied volatility is a non-linear function, we do not expect the model to be able to capture all the variance in the dataset.
We expect ANN to perform better since the very core difference between traditional machine learning models and deep learning is that in deep learning we can build universal function approximators without imposing constraints on the order and complexity of what we are modeling.
A fundamental concept is that as the depth of the artificial neural network grows, the more complex functions can be modelled, with the drawback that more data is needed to learn this representation. This property always holds, but as stated in the Universal Approximation Theorem, Hornik [25], a single hidden layer is enough to approximate any function arbitrarily closely, even if the number of nodes might be very large and having a deeper network could be more practical.
By employing deep learning methods, we can also move from one representation space of data, given by the original inputs, to a latent representation of data, that can show us hidden features and patterns in the phenomenon under investigation. This holds with the concern that this representation is learnt from data and it often has to be treated as a *"black-box"*, that we exploit but find very difficult to give meaning to.

ANNs general structure provides for a determined number of inputs, referred to as features, and one or more outputs, referred to as targets. Depending on the architecture, one or more hidden layers can be present between input and output and different types of transformations can occur in the network. Each hidden layer has a certain number of nodes that perform calculations, propagating and transforming the initial input through the net, until a final function is applied to the output layer, which contains the output targets. A simple and general ANN architecture is shown in figure 3.4.

In general, in each hidden layer the nodes perform transformations to the signal that comes from the previous layer, by applying an *activation function*. If the network has m_0 features, m_i nodes per hidden layer and m_{n+1} targets, the input layer would be 0 and the output layer $n + 1$. The formula to calculate the final value of the $\nu_{0J} = j$th feature, for each layer, can be expressed as:

$$\nu_{i,j} = f_i(b_{i,j} + \sum_{k=1}^{m_{i-1}} w_{i-1,k,j}\ \nu_{v-1,k})$$

This equation is the *activation function*, the function used to propagate values from layer n to layer $n+1$. There exists several types of activation functions, figure 3.5 shows the most common ones. The other parameters that define the transformation are:

- w: the *weight* of each node at each layer, creating a matrix representing the network.
- b: this a constant term referred to as *bias* and it is added to the weighted value of each node before its value is passed to the next layer. It is used to adjust the weighted sum of inputs, just like an intercept is added to a linear equation.

For a network made out of H hidden layers, M nodes, F features, T targets, the total number of parameters is given by:

$$(F+1)M + M(M+1)(H-1) + (M+1)T$$

The high number of parameters can easily lead to *overfitting*: the trained model works very well on training data but does not perform well on new inputs, basically creating a model that is not flexible and as a consequence not useful. For example, a relatively shallow network that consists of 3 hidden layers with 80 nodes in each layer, needs 13361, and in the context of deep learning applications, especially in computer vision domain, this number is in the order of millions.
To monitor the overfitting problem, some *model validation techniques* can be applied, such as *cross-validation*, presented in paragraph 3.3.1.Cross-validation consists in training a model multiple times on different partitions of the dataset, scoring the model with some metric on changing test data and averaging the results.

Feed-Forward Neural Networks Feed-forward neural networks constitute the easiest architecture that a neural network can adopt. Their name derives from the flow of information in the network: it only flows in one direction, without creating any cycles among different layers. This architecture is commonly used in classification problems, but without building very deep networks with layers of this type, since they require a very high number of parameters and a lot of data would be necessary for their training, along with the problem of overfitting. This is why these layers are usually applied after some convolutional layers, presented in paragraph 3.2.1. For example, in image classification, fully-connected feed-forward layers are only applied at the end of the network to obtain a classification score, while the previous layers are applied convolutions to pixel inputs, in order to extract new features.

Backpropagation Algorithm Deep neural networks learn by modifying their internal parameters in order to approximate some function. This is achieved by minimizing a *loss function*, which describes the error that the network is making when performing a given task. Minimizing a function corresponds to an optimization problem, which is solved by calculating the gradients of the loss function, the target of the minimization, and to update the weights of the ANN with respect to the parameters, trying to moving them closer to the ideal function, which is the one that actually minimizes the error. Calculating such gradients with respect to the ANN parameters is not an easy task and we can simplify the problem by looking at the network as a composition of functions:

$$f = f^{(1)} \circ f^{(2)} \circ .. \circ f^{(L-1)}$$

where $f(i) = g^{(i,j)(z^{(i,j)}())}$, with $g(i,j)$ activation function of the *j-th* hidden layer and the *i-th* node of the hidden layer. $z^{(i,j)}$ is a usually affine transformation of the type $z = W\dot{x} + b$. In order to efficiently calculate the gradient, the *chain rule* can be applied recursively on the composition of functions that describe the neural network.

In the scalar case, the *chain rule* states that, given $x \in \mathbb{R}$ and two functions $f, g : \mathbb{R} \to \mathbb{R}$, differentiable and for which $y = g(x)$ and $z = f(y) = f(g(x))$, we can say:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y}\,\frac{\mathrm{d}y}{\mathrm{d}x}$$

This statement is extensible to vectors, where $x \to \mathbb{R}^n$ and $f, g : \mathbb{R}^n \to \mathbb{R}^m$. The backpropagation algorithm consists in the recursive application of the chain rule through the network, where each tensor, which mathematically represents a layer, is transformed into a vector, chain rule is applied and then the resulting vector is converted back into a tensor, providing updated values for the network's weights.

Convolutional Neural Networks The purpose of the hidden layers is to learn non-linear combinations of the original inputs; this is called *feature extraction* or *feature construction*. This approach is particularly useful for problems where the original input features are not very individually informative, like pixels in an image, which do not provide a lot of information as single points.What gives meaning to the image is the combination of pixels, which tells us what objects are present. In contrast, for a task such as document classification, using a bag of words representation, each feature (word count) is informative on its own, so extracting features of a higher order is less important. This is why the field of computer vision is the one more motivated in exploring this kind of architecture. Here the hidden units have local receptive fields (as in the primary visual cortex),

and in which the weights are shared across the image, in order to reduce the number of parameters. The resulting effect of such spatial parameter tying is that any useful features that are "discovered" in some portion of the image can be re-used everywhere else without having to be independently learned [26].

3.2.2 Other machine learning models

Decision Trees and Random Forest Regression

Decision tree is a hierarchical construct that looks for optimal ways to split the data in order to provide classification or regression. The model approach is top-down, recursive and needs training in a supervised setting.
Decision trees main advantages are:

- interpretable results that can be graphically displayed to make sense of decision process (very important in the industry context)
- can handle numerical and categorical data equally well
- a stastical validation is possible
- good performance with large data sets

Decision tree is constructed by scanning each feature and finding a value for the x_j feature that gives the best prediction for y_j . The prediction accuracy is compared for all possible split values and the best one is selected as the split for the tree. Here we have created two new branches and the process is repeated recursively on them, until a stopping criteria is met, such as only one class belonging to the node or a minimal number of samples in the node.
In order to choose the best partitions, ideally we would like to minimize the residual sum of squares (RSS) between the training observations and the mean response of training observations, denoted here by $\hat{y}_{R_j}$, where R_j is the region of space delimited by the partition split. So, for J regions:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

It is practically infeasible to consider every possible partition of the feature space into J regions, so we approach the problem in a greedy way: we look for the best split at a particular step, without looking ahead and picking a split that will lead to a better tree in some future step.

Major drawbacks of decision trees are the non-robustness (small change in the data can cause large change in the constructed tree) and a generally lower prediction

accuracy compared to other models. These weaknesses are well-compensated by another model that actually relies on trees, called Random Forest.

Random forest or random decision forests, are an ensemble learning method for both classification and regression. The fact that it's an ensemble model is fundamental because it means that it can provide robustness and also avoid the decision's tree habit to highly overfit the data.
The Random forest model relies on *bootstrap aggregation* (or bagging), which is a technique to reduce the variance of a statistical learning method, so it is extremely useful in the context of decision trees, which suffer from high variance. This method is explained in the following paragraph 3.2.2.

Bootstrapping Bootstrapping consists of creating many training sets from the population, by selecting elements with replacement. This is done because averaging a set of observations reduces variance: if we have a set of n independent observations $Z_1, ..., Z_n$ each with variance σ^2, the variance of the mean $\overline{Z}$ of the observations is $\frac{\sigma^2}{n}$.

Using this technique we can then reduce the variance simply by bootstrapping many training sets, building a separate decision trees on each one of them and averaging out the predictions. (Mean value of predictions in case of regression, majority vote in case of classification).

What distinguishes bagged trees from random forest is the step used to decorrelate the generated trees. This is achieved by limiting the number of features that are considered for each split; it might seem odd but this method will prevent a strong predictor to be the one that's always included in the top split and therefore avoid all trees looking very similar. A typical number for m (subset features number of p features) is $\sqrt{p}$, but its value depends on the problem's features number and importances.

Gradient Boosting Regression

Gradient boosting is a machine learning technique for both regression and classification, which produces a prediction model using an ensemble of weak prediction models, which are usually basic decision trees. The algorithm works by applying the weak learner, an algorithms that perform slightly better than random guessing, *sequentially* to weighted versions of the data, where more weight is given to examples that were misclassified by earlier iterations. The difference with random forest algorithm is that weak learners are added sequentially, so that the last learner is trained on the results of all previous learners.

The goal of boosting is to solve the following optimization problem:

$$\min_{f} \sum_{i=1}^{N} L(y_i, f(X_i))$$

If we use squared error loss, the optimal estimate is given by

$$f^*(\mathbf{x}) = \underset{f(\mathbf{x})}{\operatorname{argmin}} = \mathbb{E}_{y|\mathbf{x}}[(Y - f(\mathbf{x}))^2] = \mathbb{E}[Y|\mathbf{x}]$$

This cannot be computed in practice since it requires knowing the true conditional distribution $P(Y|\mathbf{x})$. Hence this is sometimes called the population minimizer, where the expectation is interpreted in a frequentist sense and its value is approximated by boosting technique [26]. This algorithm has several advantages:

- A very accurate prediciton accuracy
- Flexibility: different loss functions can be optimized with many different hyperparameters
- Preprocessing of features can often be skipped, even with categorical values
- Automatically handles missing values in the dataset

On the other hand, the approach brings some disadvantages:

- Continuing to minimize errors can lead to overfitting on training data, so techniques like cross-validation are required
- Computational time can be a burden since a great number of trees might be required
- High number of hyperparameters, that allows flexibility, allow brings a great search space to be explored when performing grid-search

3.3 Parameter Tuning and Models Generation

In order to predict the implied volatility value of call options on S&P500 index, we applied three different machine learning techniques for regression, described in section 3.2: neural network, random forest, gradient boosting regressor.

3.3.1 Model Validation

Model Evaluation Metrics

In the thesis we used Mean Absolute Error (MAE) as error metric to measure the performance of implied volalitity's predictions. This metric is defined by the equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.1}$$

Where y_i is the real value of implied volatility and $\hat{y}_i$ is the predicted value by the regression model taken into consideration. This metric calculated the mean of the absolute difference between predictions and real values. In our use case MAE is preferable to Mean Squared Error (MSE), because errors are not weighted, resulting in less sensibility to outliers. MSE is instead calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

So the predictions errors are squared, giving more relevance to errors of greater magnitude. This sensitivity to outliers is not desirable our study because of the various market crashes that have been included in the dataset. A single market crash event can dramatically throw off the prediction model by the order of 250 for several days of observations, but we are not interested in modeling such events, whereas we want to predict IV in a normal context scenario.

k-Fold Cross-Validation To validate models and estimate their performance in terms of prediction error, we used *k-fold cross-validation* technique.
This approach consists in randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds. The mean absolute error, MAE_1, is then computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set [27]. This process results in k estimates of the test error, $MAE_1, MAE_2, ..., MAE_k$. The k-fold CV estimate is computed by averaging these values:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MAE_i$$

Figure 3.6 shows the described k-fold process.

3.3.2 Parameter Tuning

Artificial Neural Network

A general introduction of artificial neural networks is presented in section 3.2.1. Since Hull's work [10] on a similar task showed good results, we trained a simple neural network with a three hidden layer architecture, with 80 nodes per layer.
The specific implementation of the network is the *MLPRegressor* in package *sklearn.neural_network*. It consists of a multi-layer perceptron regressor, implementing a simple neural network with customizable number of layers and nodes. In order to train it we performed a grid search with cross-validation, trying out different architectures with different depths and hyper-parameters. In this process we used a *Pipeline* object, which allowed us to include different feature normalization techniques in the grid search, in a simple manner.
The parameters of the grid-search are shown in table 3.3.2.

Grid-search for MLPRegressor						
Normalization	Hidden Layer Size	Activation function	Batch size	Learning Rate	Solver	Alpha
MinMaxScaler	(120,120)	ReLu	1	0.01	sgd	0.001
RobustScaler	(80,80)	tanh	128	0.05	Adam	0.05
	(80,80,80)		512	0.1		0.01
	(80,120,80)			0.5		

Table 3.1: Grid-search parameters for MLPRegressor

For each parameter configuration a model was trained and scored and a *cross-validation test* was performed.

Random Forest Regression

The theoretical basics of the algorithm are presented in section 3.2.2. The implementation is the one provided by *sklearn*: *sklearn.ensemble.RandomForestRegressor*. We performed a grid-search in order to estimate the best parameters with parameters' values shown in table 3.3.2.

Gradient Boosting Regression

For GBR, the implementation is, just like the other algorithms, taken from the *sklearn* library. The class is *sklearn.ensemble.GradientBoostingRegressor*.
We performed a grid-search with parameter shown in table 3.3.2.

Grid-search parameters for Random Forest Regressor				
Normalization	Estimators	Max Depth	Max Features	Bootstrap
MinMaxScaler	100	None	auto	True
RobustScaler	500	50	log2	False
	1000	5000		

Table 3.2: Grid-search parameters for Random Forest Regressor

Grid-search parameters for Gradient Boosting Regressor				
Normalization	Estimators	Learning Rate	Loss	Criterion
MinMaxScaler	100	0.1	Least Squares	friedman-mse
RobustScaler	1000	0.5	quantile	mse
		0.05		mae

Table 3.3: Grid-search parameters for Gradient Boosting Regressor

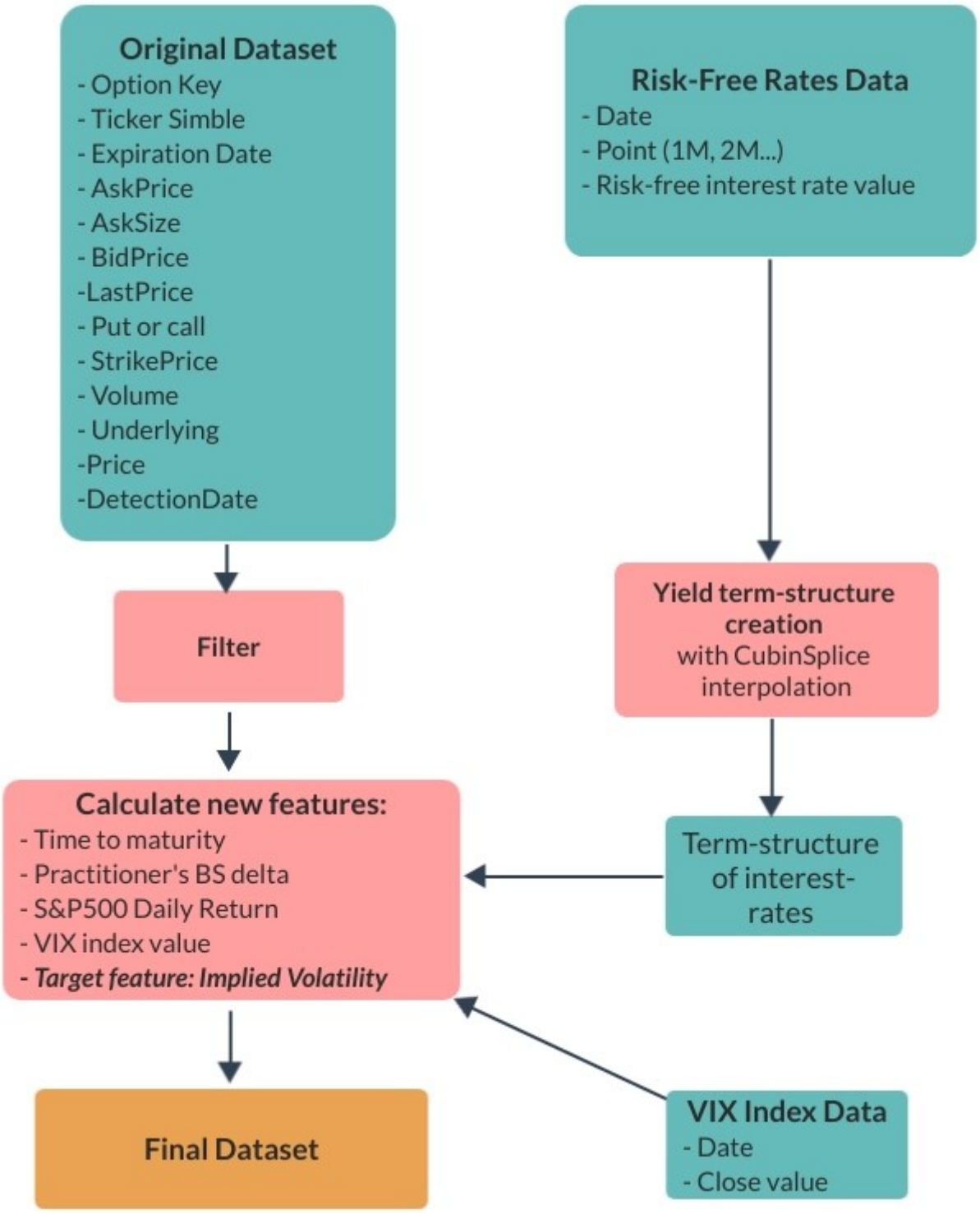

Figure 3.1: Dataset creation pipeline

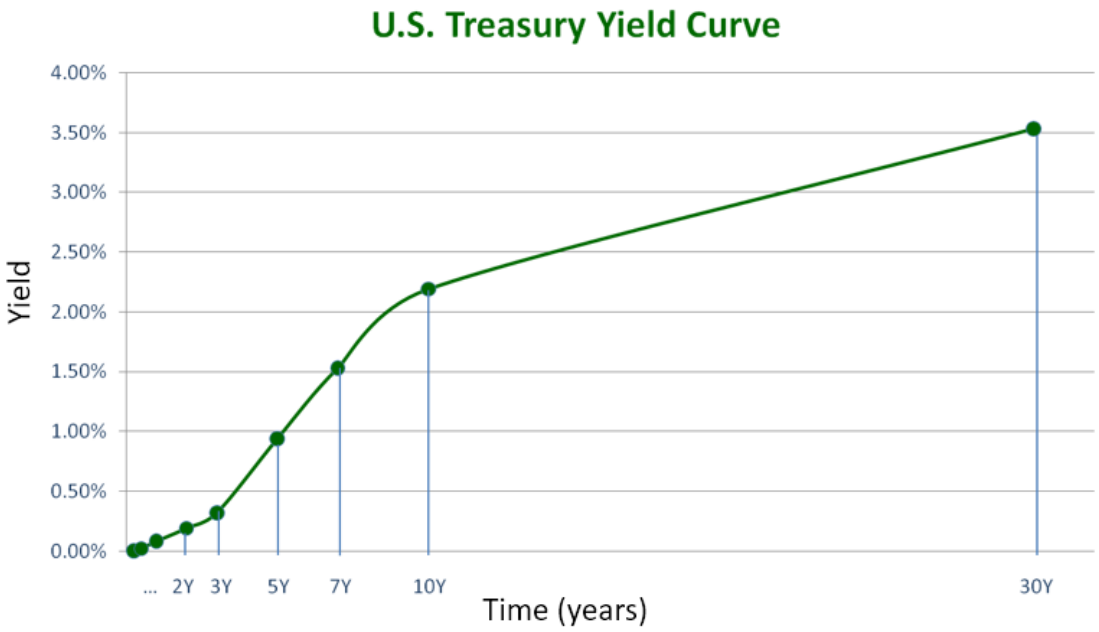

Figure 3.2: An example of a risk-less yield curve

Source: https://eiptrading.com/yield-curve (July 3rd, 2021)

Figure 3.3: S&P 500 index from 1990 to 2020

Source: www.macrotrends.net (July 3rd, 2021)

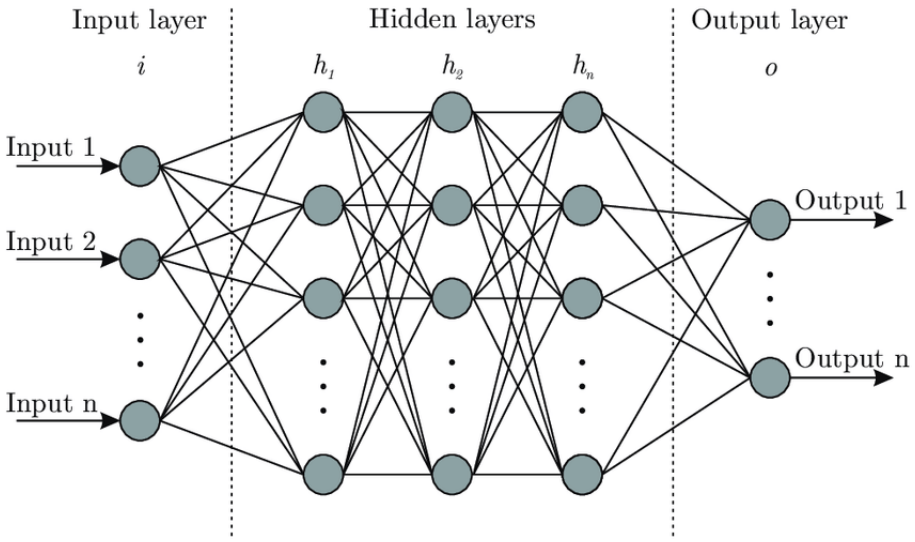

Figure 3.4: General Artificial Neural Network Architecture

Source: Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. Bre, Facundo and Gimenez, Juan and Fachinotti, Víctor (2017)

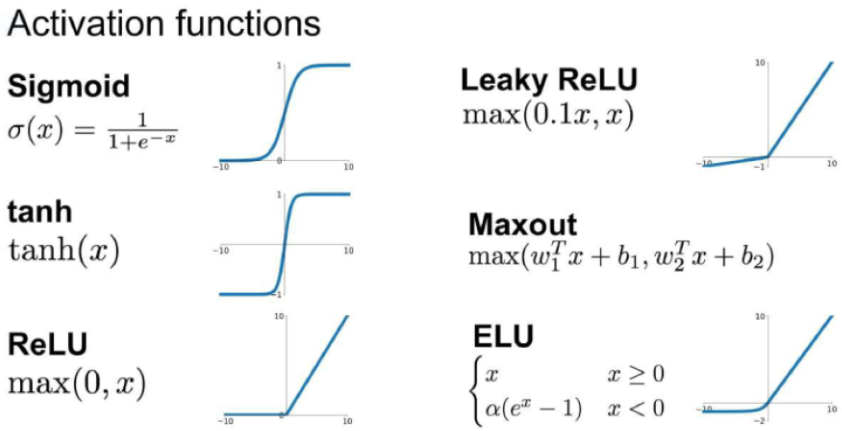

Figure 3.5: ANN activation functions

Source: Reconstruction of porous media from extremely limited information using conditional generative adversarial networks. Junxi Feng (2019)

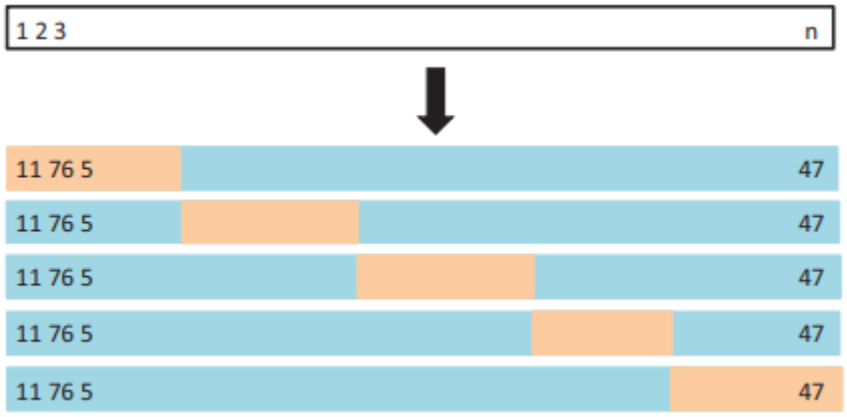

Figure 3.6: A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups.

Source: An Introduction To Statistical Learning, Robert Tibshirani (2013) [27]

Chapter 4

Results

In this chapter we illustrate the experimental setup under which we trained and deployed machine learning models to predict implied volatility, we then show their capabilities and discuss statistical validation of the results.

Two machines with different characteristics have been used to run calculation for this thesis, this is because we need two different kind of computational power in two stages of the work:

- *Dataset creation*: the calculation of implied volatility is solved with the Newton–Raphson method, which is a CPU-intensive work load. For this stage we used a powerful machine provided by the company *Myrios*.
- *Model training*: machine learning methods, especially neural networks, need to be able to perform many operations in parallel on a GPU. This computational power was provided by a Nvidia T4 GPU, through Google's cloud computing platform *Google Colaboratory Pro*.

Hardware specifications for the two machines are illustrated in table 4.

Hardware Specifications							
Machine	CPU model	CPU frequency	CPU cores	RAM size	GPU model	GPU performance	GPU memory
MyriosPC	Intel i9-9900K	3.60GHz	8	64 GB	Intel UHD Graphics 630	230.4 GFLOPS	None
Colab	Intel Xeon(R)	2.30GHz	2	12 GB	Nvidia T4	8.1 TFLOPS	16 GB

Table 4.1: Hardware Specifications of utilized machines.

All the applied machine learning models and prediction generations have been carried out using the python library sklearn [28], while the dataset creation phase mainly used two other python libraries:

- *numpy*: an open source package specialized in scientific computing [29]
- *pandas*: open source data analysis and manipulation tool [30]

In the following sections we illustrate the specific configurations of used models, experimental design choices, model performances and the practical usability of the approach.

4.1 Experimental Design

Data As mentioned in section 3.1.2, our dataset is made out of 271.814 thousand observations on S&P500 call options on data from April 2010 to January 2021 with a 1-day granularity. The training partition of the dataset is set to 70% while the test set is made of out the 30% of the dataset. Given the size of the dataset, each model is trained on 190.270 trading days and tested on 81.544 samples.
The size of the dataset can be considered good for a machine learning problem, so it does not necessitate the application of any data augmentation techniques.

We investigated the forecast problem in two different scenarios:

- Average volatility swing: we only took into consideration periods of time where volatility changes are in the interval $\pm 1.5\%$, which is the mean swing interval as reported in [31].
- Moderate volatility swing: volatility swings in the range $\pm 5\%$ are considered.

In addition, as suggested by Hull [10], we later included daily VIX index values in the dataset and re-trained all models with this new feature. At the end of the training process we have 4 different sets of results:

- Volatility swing = $\pm 5\%$, VIV not included
- Volatility swing = $\pm 5\%$, VIV included
- Volatility swing = $\pm 1.5\%$, VIV not included
- Volatility swing = $\pm 1.5\%$, VIV included

Algorithms In the experiments, 3 different machine learning regression models have been used. Theoretical specifications of the inner workings of the algorithms are explained in section 3.2, while parameter tuning and implementations are presented in section 3.3. Here, we show each model with its parameters after having performed a grid-search:

- MLPRegressor (MLPR): activation function = ReLu, alpha = 0.01, batch size = 512, hidden layer = (80,80), learning rate = 0.01 adaptive, solver = Adam, early stop = True

- Random Forest Regressor (RFR): bootstrap = True, max depth = None, max features = 'sqrt', max leaf nodes=None, n estimators = 100, min samples split=2

- Gradient Boosting Regressor (GBR): learning rate=0.1, loss = 'ls', n estimators = 100, min samples leaf = 1, min samples split = 2

4.1.1 Model Validation

As mentioned in Section 3.3.1, the model validation has been carried out applying cross-validation to each model with parameter $k = 10$. The used scoring function is Mean Absolute Error (equation 3.1) and tables 4.2, 4.3 show results of cross-validation with MAE score and R2 score for each trained model, in the 4 different setups. IV MAE error is expressed as percentage and execution times for single fit and cross-validation tests are expressed in minutes.
Figures 4.1, 4.2 and 4.3 show MAE distribution on test set for the three models. What is later confirmed by applying bootstrapping technique (paragraph 3.2.2), is that the majority of the errors is distributed towards the value zero, resulting in distributions with thin tails. This is a first good indicator that the average error is small in the vast majority of cases; this statement is then confirmed by calculating a confidence interval on trained models.

Cross-validation results for average daily volatility swing ±5% - NO VIX					
Model	MAE (mean)	MAE (variance)	R2	Single Fit Time	Cross-val Time
GBR	1.2777	0.1299	0.968	0.12	3.12
RFR	1.351	0.1369	0.970	1.55	29.3
MLPR	1.272	0.127	0.967	2.33	27.24
Cross-validation results for average daily volatility swing ±5% - VIX					
GBR	1.2643	0.1286	0.972	0.12	3.12
RFR	1.31	0.1538	0.977	1.55	29.3
MLPR	1.1415	0.1425	0.972	2.33	27.24

Table 4.2: 10-folds Cross-Validation results for the three regression models with ±5% daily volatility swing

Cross-validation results for daily average volatility swing ±1.5% - NO VIX					
Model	MAE (mean)	MAE (variance)	R2	Single Fit Time	Cross-val Time
GBR	0.5745	0.0375	0.9899	0.12	3.12
RFR	0.5981	0.0699	0.9902	1.55	29.3
MLPR	0.5660	0.0301	0.9899	2.33	27.24
Cross-validation results for daily average volatility swing ±1.5% - VIX					
GBR	0.5655	0.05643	0.9891	0.12	3.12
RFR	0.5553	0.05841	0.99060	1.55	29.3
MLPR	0.4978	0.05087	0.98935	2.33	27.24

Table 4.3: 10-folds Cross-Validation results for the three regression models with ±1.5% daily volatility swing

4.1.2 Effect of Volatility Swings

The first consideration can be done about the different scores achieved in the scenarios with different implied volatility swings: the trained models are able to forecast IV with greater accuracy when IV swings are smaller and in the usual range of the index (±1.5%). When we consider a wider interval of swings (±5%), the models perform worse because these cases are less frequent and constitute in fact an exception in the index fluctuations. This is also motivated by the R2 score in the two different scenarios: the explained variance is greater and close to 1 when considering a smaller range of IV swings, while in the ±5% scenario the models show a lower score.

MAE cross-validation results show an acceptable error in both scenarios, since a forecast value that is wrong by 1% or 0.5% is able to detect anomalies is the monitored pricing system. If such system gathered wrong market data or some of its parameters were wrong for any reason, their repercussions of implied volatility would be of great impact with far greater error ranges, with errors in the order of integer percentage points.
Another consideration has to be done about the scenario in which the trained models are deployed. Models are more accurate when implied volatility swings are included in a small range, implying that market conditions are normal. In a more volatile market, where IV swings are larger, these models loose prediction accuracy and may not perform as well as under stable conditions.

Models comparison After this consideration we look at the performances of the three different models. GBR and RFR are both tree-based models and they perform similarly, while MLRP tends to perform better than the others in all the

different setups. This could be due to the fact that the importance given to the features is too narrowed down on the IV feature for the tree-based models, while the ANN exploits differently the features. This adaptation capacity of MLRP can be seen when we introduce an additional feature, the VIX index; the other models do not exploit the provided extra information very well, showing little improvement, while the multi-layer perceptron improves its score by almost 8%. We define a gain measure to compare the models with and without the VIX index information:

$$G = 1 - \frac{MAE_{vix}}{MAE}$$

MLRP is the model that better exploits the new feature, looking at gains:

- GBR: +1.5%
- RFR: +7.1%
- MLPR: +12%

Bootstrapping Results of bootstrapping procedure, explained in paragraph 3.2.2, have been calculated for Gradient Boosting Regressor only, since it was computationally unfeasible to apply it to the two other models. Fitting 1000 models on random subsamples of the dataset, with replacement, allowed us to statistically validate GBR by computing a 95% confidence interval. The resulting interval has a mean of 0.00558 and a variance of 0.0172. Results are shown in figure 4.4.
The error distribution shows that the model commits, in average, an error of 0.5% when forecasting next day IV values, and by looking at the variance we can state that there is a 95% probability that the average errors fall the interval of values: $0.558\% \pm 0.17\%$.

4.1.3 Use Case

The results show a very good preliminary analysis for a future application of this framework in the financial sector. The thesis is focused only on European call options on S&P500 index, but in the future the same methodology could be applied to options on other indexes and be extended to work on different types of options. The price checking framework could be integrated in an automated corporate workflow to detect anomalies in option prices. Given enough data, the approach could be applied to *OTC* markets to provide an automated price checker at service of banks and corporations.
In comparison to other approaches, mostly based on autoregressive models such as ARIMA and GARCH, our methodology provides an extensible method, easily

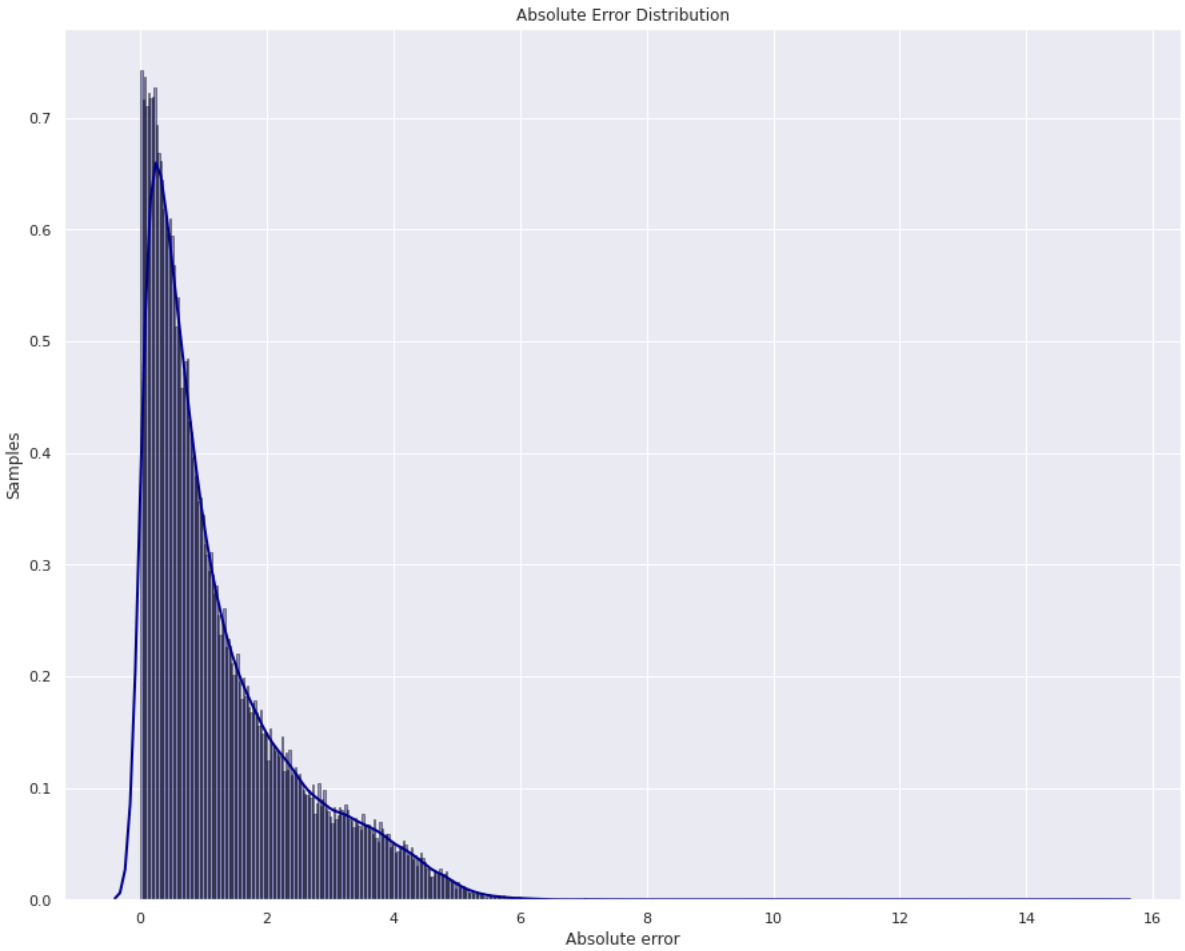

Figure 4.1: MAE Error Distribution for Gradient Boosting Regressor

applicable to different types of financial derivatives, completely automated and self-adaptive to different market scenarios. The novelty of the approach consists in reducing the complex problem of price-checking to the single problem of volatility forecasting. Tackling the problem with machine learning allows for more flexibility of application, without the necessity to create new models for different market scenarios but simply training ML algorithms on different data and evaluating them.

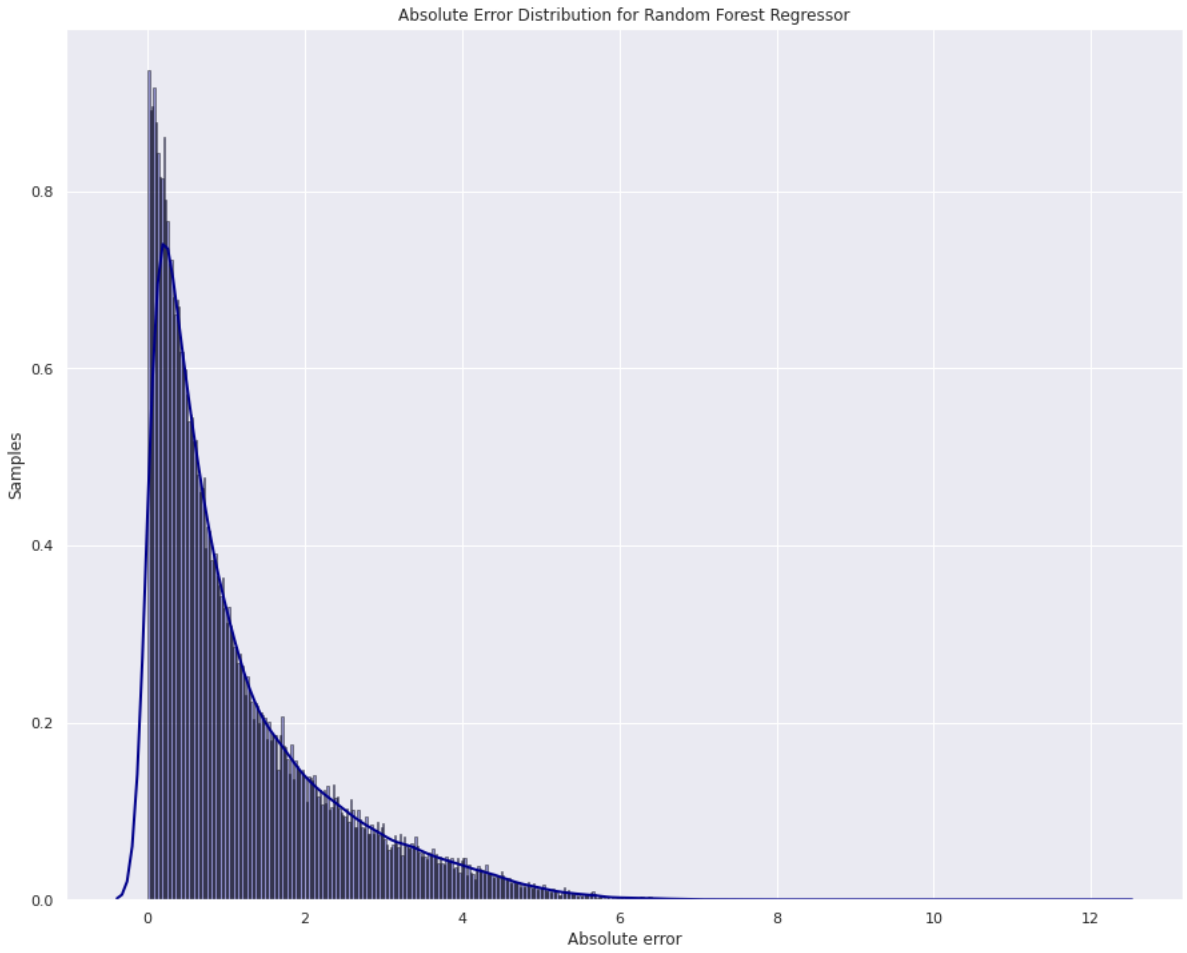

Figure 4.2: MAE Error Distribution for Random Forest

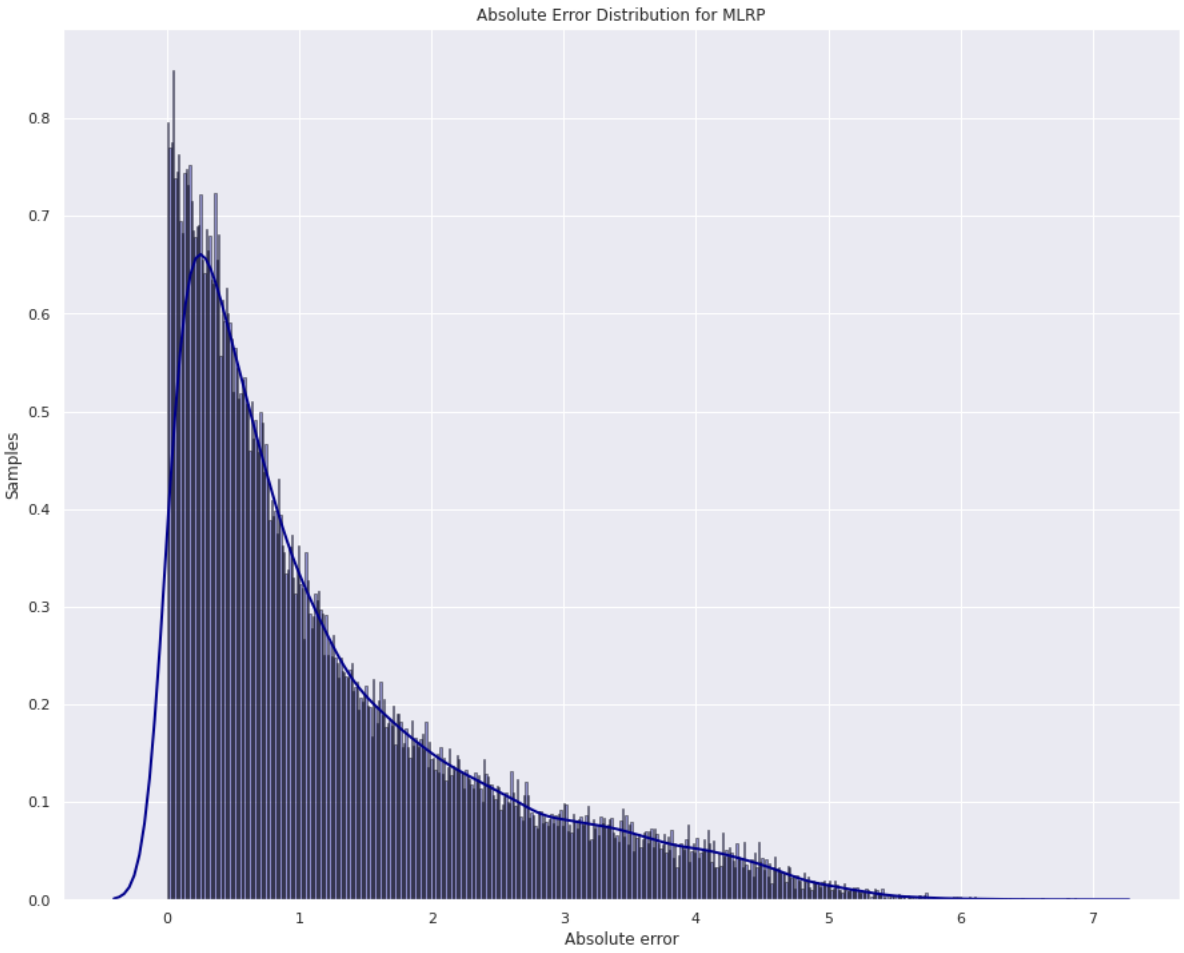

Figure 4.3: MAE Error Distribution for MLPR

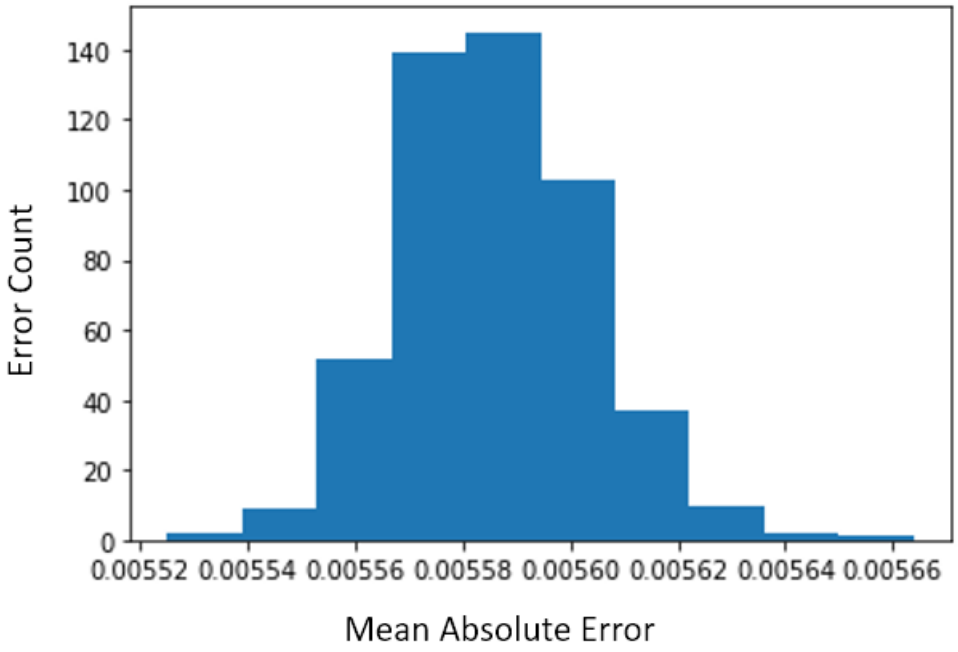

Figure 4.4: Distribution of absolute errors of GBR model with bootstrapping technique. 1000 iterations. Confidence interval: 0.00558 ±0.00172

Chapter 5

Conclusions and Future Work

The market of derivatives is in constant growth in terms of number of traded contracts and volume; it has become fundamental for banks and corporations to handle their operations around the world, in an always more connected global market. Derivatives always bring an associated risk to contractors, a risk that could have heavy financial consequences and can also be hidden, this is why a system that continuously monitors the effective value of these contracts is needed and fundamental.

The high number of traded derivatives generates a lot of data that can be used to apply machine learning algorithms to solve different modeling problems. What has always been solved with analytic theoretical models, can now be faced with a data-driven approach. Forecasting models are based on the analysis of historical data, where the quality of the data used to train the algorithms heavily influences the quality and reliability of the predicted values.

The thesis consisted in a preliminary analysis to verify whether the proposed approach, to solve the anomaly detection problem with machine learning, could in fact be applied in a real-life scenario and bring an added value to existing models and operational workflow.

We conducted an analysis on call options quoted on the Standard&Poor500 index, with historical data from 2010 to 2021.
Experimental results showed that:

- The application three different types of regressors could successfully learn how to predict next-day implied volatility value with an acceptable error.

- The models are more accurate when market conditions are normal, therefore implied volatility swings are included in a small range.
- Including extra information, like VIX index, can lead to improvements in the precision of the models.
- ANN model performed better than random forest regressor and gradient boosting regressor
- ANN model exploited the additional VIX index information better than the other models, producing the best results.

Considering these takeaways, we trained three models that are able to solve the task of forecasting the implied volatility in an accurate way and showed that the method could be integrated in a real use case.

This preliminary analysis could be extended and improved under many aspect:

- Train the models on different types of options
- Train the models on other market indexes like Dow Jones
- Try out different machine learning models such as Long Short Term Memory, which could incorporate temporal information of previous days
- Explore the applicability of the approach to the OTC market
- Integrate the framework in a real application
- Explore the possibility of including other features in the dataset that may bring additional useful information to improve IV forecast precision

Bibliography

[1] European Central Bank and Europäische Zentralbank. *ECB Guide to Internal Models*. European Central Bank, 2019. ISBN: 9789289937153. URL: https://books.google.it/books?id=ZLQQzgEACAAJ (cit. on p. 1).

[2] Najmi Samsudin and Azhar Mohamad. «Implied Volatility Forecasting in the Options Market: A Survey». In: *Sains Humanika* 8 (Mar. 2016), pp. 9–18. DOI: 10.11113/sh.v8n2.721 (cit. on pp. 4, 21).

[3] Ser-Huang Poon and Clive W.J. Granger. «Forecasting Volatility in Financial Markets: A Review». In: *Journal of Economic Literature* 41.2 (June 2003), pp. 478–539. DOI: 10.1257/002205103765762743. URL: https://www.aeaweb.org/articles?id=10.1257/002205103765762743 (cit. on pp. 5, 6).

[4] O.I. Bacha. *Financial Derivatives: Markets and Applications in Malaysia*. McGraw-Hill (Malaysia), 2012. ISBN: 9789675771408. URL: https://books.google.it/books?id=LQ6RngEACAAJ (cit. on p. 4).

[5] John Hull and Alan White. «Optimal delta hedging for options». In: *Journal of Banking Finance* 82 (2017), pp. 180–190. ISSN: 0378-4266. DOI: https://doi.org/10.1016/j.jbankfin.2017.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0378426617301085 (cit. on pp. 5, 26, 29).

[6] Torben Andersen and Tim Bollerslev. «Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts». In: *International Economic Review* 39.4 (1998), pp. 885–905. URL: https://EconPapers.repec.org/RePEc:ier:iecrev:v:39:y:1998:i:4:p:885-905 (cit. on p. 6).

[7] Robert Engle. «Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation». In: *Econometrica* 50.4 (1982), pp. 987–1007. URL: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:50:y:1982:i:4:p:987-1007 (cit. on p. 6).

[8] Andrew Yang Alexander Ke. «Option Pricing with Deep Learning». In: *CS230: Deep Learning* 8 (2019), pp. 1–8 (cit. on p. 6).

[9] Shengli Chen and Zili Zhang. «Forecasting implied volatility smile surface via deep learning and attention mechanism». In: *Available at SSRN 3508585* (2019) (cit. on p. 6).

[10] Jay Cao, Jacky Chen, and John Hull. «A neural network approach to understanding implied volatility movements». In: *Quantitative Finance* 20.9 (2020), pp. 1405–1413. DOI: 10.1080/14697688.2020.1750679. eprint: https://doi.org/10.1080/14697688.2020.1750679. URL: https://doi.org/10.1080/14697688.2020.1750679 (cit. on pp. 6, 26, 37, 44).

[11] Will Kenton. «Financial Instrument». In: (Mar. 2020). URL: https://www.investopedia.com/terms/f/financialinstrument.asp (cit. on p. 6).

[12] John C. Hull. *Options, futures, and other derivatives*. 6. ed., Pearson internat. ed. Upper Saddle River, NJ [u.a.]: Pearson Prentice Hall, 2006. URL: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+563580607&sourceid=fbw_bibsonomy (cit. on pp. 7, 9, 10, 23).

[13] James Chen. «Risk-Free Rate of Return». In: (Apr. 2021). URL: https://www.investopedia.com/terms/r/risk-freerate.asp (cit. on p. 11).

[14] Adam Hayes. «Guide to Dividend Investing». In: (Mar. 2021). URL: https://www.investopedia.com/terms/d/dividend.asp (cit. on p. 11).

[15] Fischer Black and Myron S Scholes. «The Pricing of Options and Corporate Liabilities». In: *Journal of Political Economy* 81.3 (May 1973), pp. 637–654. DOI: 10.1086/260062. URL: https://ideas.repec.org/a/ucp/jpolec/v81y1973i3p637-54.html (cit. on p. 12).

[16] Mervyn A. King and Sushil Wadhwani. «Transmission of Volatility between Stock Markets.» In: (Feb. 1990) (cit. on p. 18).

[17] Justin Kuepper. «CBOE Volatility Index (VIX) definition». In: (Mar. 2021). URL: https://www.investopedia.com/terms/v/vix.asp (cit. on p. 19).

[18] Adam Hayes. «Greeks». In: (Feb. 2020). URL: https://www.investopedia.com/terms/g/greeks.asp (cit. on p. 19).

[19] Nian et al. «Learning minimum variance discrete hedging directly from the market». In: (Mar. 2018) (cit. on p. 22).

[20] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: http://www.scipy.org/ (cit. on p. 28).

[21] Yassine Maaroufi. *MibianLib Options Pricing Open Source Library*. 2011. URL: https://github.com/yassinemaaroufi/MibianLib (cit. on p. 28).

[22] «S&P-500 Index». In: (). URL: https://www.bloomberg.com/quote/SPX:IND (cit. on p. 28).

[23] Will Kenton. «SP 500 Index – Standard Poor's 500 Index». In: (Mar. 2021). URL: https://www.investopedia.com/terms/s/sp500.asp (cit. on p. 28).

[24] W. McCulloch and Pitts W. «A logical calculus of ideas in nervous activity». In: (1943) (cit. on p. 29).

[25] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. «Multilayer feedforward networks are universal approximators». In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208 (cit. on p. 30).

[26] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on pp. 33, 35).

[27] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/ (cit. on pp. 36, 42).

[28] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 43).

[29] Charles R. Harris et al. «Array programming with NumPy». In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2 (cit. on p. 44).

[30] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134 (cit. on p. 44).

[31] Ed Easterling. «Volatility In Perspective». In: *Crestmont Research* (Jan. 2020), pp. 1–6 (cit. on p. 44).