STOCK MARKET TRENDS FORECASTING

FINAL YEAR PROJECT

Dioula DOUCOURE
Department of Mathematics
Télécom SudParis
Evry, France
dioula.doucoure@telecom-sudparis.eu

Hatem RABAA
Department of Mathematics
Télécom SudParis
Evry, France
hatem.rabaa@telecom-sudparis.eu

June 11, 2021

ABSTRACT

Stock markets are an essential component of the economy. Their prediction naturally arouses a fascination in the academic and financial world. Indeed, financial time series, due to their wide range application fields, have seen numerous studies being published for their prediction. Some of these studies aim to establish whether there is a strong and predictive link between macroeconomic indicators and stock market trends and thus predict market returns. Stock market prediction however remains a challenging task due to uncertain noise. To what extent can macroeconomic indicators be strong predictors of stock price ? Can they be used for stock trends modeling ? To answer these questions, we will focus on several time series forecasting models. We will on the one hand use statistical time series models, more specifically the most commonly used time series approaches for stock prediction : the Autoregressive Integrated Moving Average (ARIMA), the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) and the Vector Autoregressive (VAR) approach. On the other hand, we will be using two deep learning models : the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU) for our prediction task. In the final section of this paper, we look directly at companies to detect trends.

Keywords Stock Markets · Times Series · Macroeconomic · Deep Learning · GARCH · ARIMA · VAR · LSTM · GRU · Portfolio Management

1 INTRODUCTION

Financial time series modelling plays a pivotal role in today's world economic activities. This can be explained by stock market's ability to not only convert savings into profitable investments and but also reallocate funds in different sectors of the economy. Thus, this question has been widely studied over the past decades.

Louis Bachelier started, in his 1900 PhD thesis called "Théorie de la spéculation", by modelling stocks with a simple Brownian motion, which was firstly used to describe the irregular movement of particles suspended in a liquid that was observed by Robert Brown in 1828. He also introduced the concept of martingale, which nowadays plays a crucial role in mathematical finance, that will only be systematically studied by mathematicians in the 1940s by Paul Levy.

Louis Bachelier's work established the ground rules for financial mathematics and stock market modelling since it was extended by Paul Anthony Samuelson in the 1960s who developed the efficient markets hypothesis, which maintains that market prices fully reflect all available information.

During this time, theoretical developments in time series analysis started in the 1920s with Herman Wold who introduced AR-MA models, but it was necessary to wait for the Box-Jenkins approach to time series analysis in 1970 to be able to determine maximum likelihood estimation of the parameters and propose a full modeling procedure for individual series. This led to the arise of new classes of models from the econometricians side with the first generalization of multivariate AR-MA models, particularly VAR models, and the non-linear generalizations, mainly ARCH and GAR-CH models. These models emphasise the correlation between current and past values and have become very popular for financial time series analysis. Moreover, the non-linear models have proved useful for modeling macroeconomic time series since they highlight non-linear characteristics, notably volatility.

Indeed, volatility is the crux of the matter in financial fields since is one of the crucial non-linear characteristics of financial time series. Its study remains one of the core issues in financial research and represents the uncertainty of the future price that makes it unpredictable. [1]

These models constitute a first class of so called statistical time-series model which use previous values to predict the future. Despite the widespread use these models in the literature, they encounter limitations such as a need of stationary, an increase effect of estimation error. With the enhancement of computational performance and the advent of big data, the use of features, a second category of models emerges to overcome these limits and use features to break free from the naive idea that the future can be merely predicted by using the past.

Indeed, over the past years, with the progress of learning techniques and AI, machine learning techniques are being more and more used in stock market analysis in order to improve the prediction performances [2], [3]. Recurrent neural networks in general and LSTM in particular have provided interesting results in trends prediction. For instance, Long et al [4] have used bidirectional LSTM in assessing stock price trends and they have obtained great performances.

Finally, the link between stock markets and macroeconomic factors has been widely studied in the past years. This incentive can be explain by the fact that stocks market embody the financial health of a country, which is described by macroeconomic indicators. Our study aims at identifying which are the most relevant macroeconomic indicators that can be used to predict stocks market. The idea is to establish a comparative study using both statistical time-series model and machine learning techniques to highlight their limits, to fulfil them and points out their advantages.

2 METHODOLOGY

The data considered in our study are S&P 500 daily historical returns. This is a good representation of US equity market since S&P 500 is based on the market capitalizations of 500 large companies that are listed on the NYSE or NASDAQ. The daily historical data is used to compute the closing price which represents our target (or dependant variable). Along with the daily historical returns, we have a dozen macroeconomic variables that are used as predictors (independant variables)

Our first models (see Section 4) are regression based models. They allow us to examine the relationship between macroeconomic variables and S&P 500 closing price and their influence on the latter. We will discuss which variables and factors among the macroeconomic indicators matter most, which can be ignored and how do these variables influence S&P 500 closing price.

In Section 5, we will be using classification models. We will predict S&P 500 up and downtrends with a LSTM. In section 6, we will try a different approach to our modeling. Instead of classifying up and downtrends, we will look directly at the companies and from data such as the stock price, its volatility, its previous 1, 3 and 6 months performance, predict its next 6 months performance and try to make a portfolio management by determining each time the 10 best stocks on which we could invest.

3 DATA ANALYSIS AND PREPROCESSING

On a predictive modeling task, machine learning algorithms learn a mapping from input variables to a target variable. These algorithms cannot be fitted and evaluated on raw data; instead, we must transform the data to meet some requirements of individual machine learning algorithms and also put the data in a format that we can gain massive amount of information.

More than that, we must choose a representation for the data that best exposes the unknown underlying structure of the prediction problem to the learning algorithms in order to get the best performance given our available resources on a predictive modeling project. Thus pre-processing and data analysis play a very important role in every machine learning project. We will discuss in this section all the techniques and methods used to analyze, transform and put our data in the right format.

3.1 DATA ANALYSIS

Data analysis plays an important role in all modeling project. Understanding the data makes it easy to chose the most suitable model for the prediction task.

One of the first things we noticed when looking at the data was the difference in frequency between the different variables. Some macroeconomic indictors had daily values while others were weekly sampled. So we had to make some choices. Should we work only with daily or weekly variables ? Or both ? Should we resample the data by interpolation techniques ? These are we asked ourselves at the beginning.

Moreover, since we had about numerous macroeconomic variables, we tried to determine the most relevant ones with statistical tests. For example, we tested the correlations between these variables and the closing price. We used Spearman's correlation test. We also decided not to work with the data corresponding to the covid period and only business work days. Below are some plots of the macroeconomic variables.

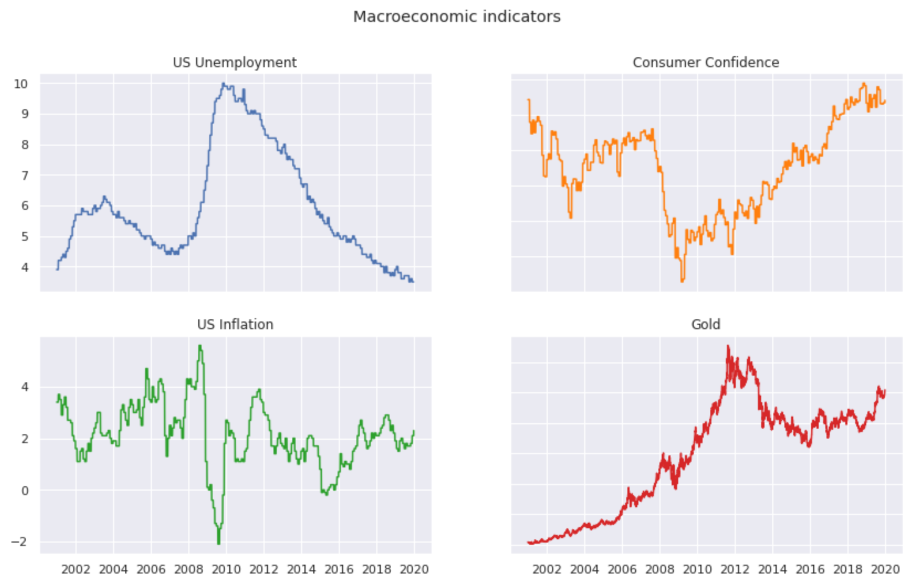


Figure 1: Macroeconomic variables plots

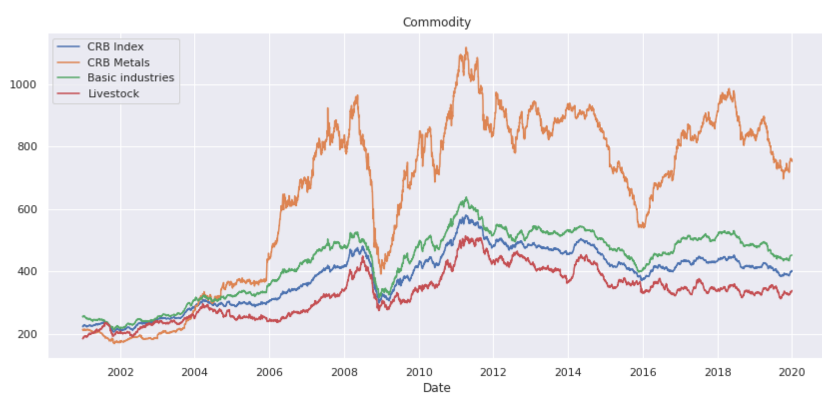


Figure 2: Commodities

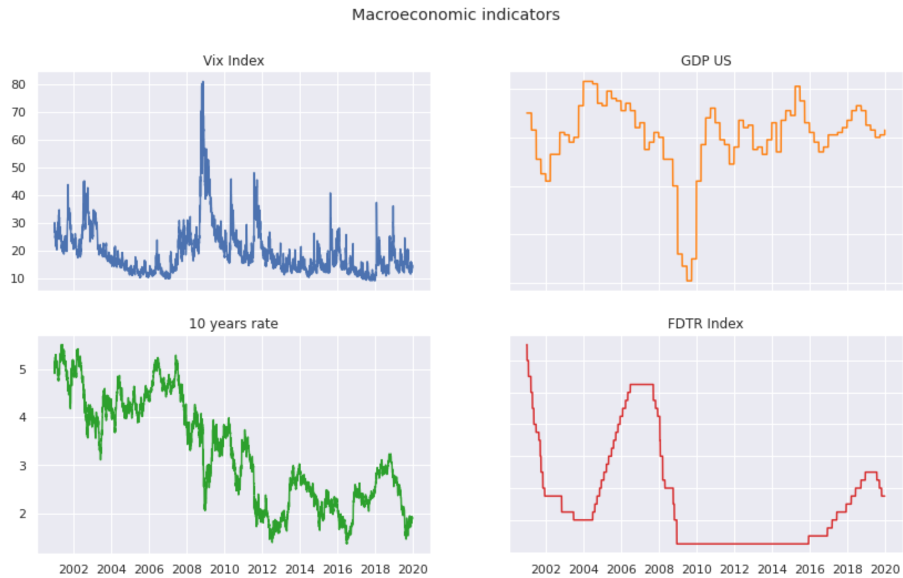


Figure 3: Macroeconomic variables plots

3.2 TIME SERIES SAMPLING

As mentioned in the previous section, we encountered a difficulty with the frequency our predictors. Indeed some were daily sampled as daily stocks returns and other were weekly sampled. We have therefore made a hypothesis to be able to work on as much variables as possible. We assumed that the missing values of the time series are constant until the date of publication of a new data item.

3.3 DENOISING

In order to overcome the difficulties of our models in dealing with the nonlinear characteristics and the noise of the stock data, we use *Wavelet Transform (WT)* denoising method. We help our models improve their generalization ability by filtering out the noise. WT eliminates noise in financial time series and fully retain the characteristics of the original signals [6]. It can be defined as time-frequency decomposition that decomposes time series data into a set of wavelets; both time and frequency domains even if the data is non-stationary.

The idea behind WT denoising is very simple. Let's denote $f(t)$ our observed signal. In our case, it is the stock data.

$$f(t) = x(t) + \epsilon(t)$$

Where $\epsilon(t)$ is a centered Gaussian white noise and $x(t)$ the *useful signal*. The purpose of WT is to filter out $\epsilon(t)$ as much as possible. A threshold is selected to separate the useful signal from the noise.

There are two types of Wavelet Transforms: Continuous and Discrete. The key difference between these two types is the Continuous Wavelet Transform (CWT) uses every possible wavelet over a range of scales and locations i.e. an infinite number of scales and locations. While the Discrete Wavelet Transform (DWT) uses a finite set of wavelets i.e. defined at a particular set of scales and locations [7], [8].

Below are the steps to wavelet decompose a signal (financial time series):

1. Selection of a wavelet basis function
2. Determination of the number of decomposition layers
3. Determination of the threshold value
4. Selection of the threshold function

We have used the most suitable wavelet functions for financial data denoising and have chosen the best among them. 3 decomposition layers were selected. To evaluate the effect of the WT, Qiu and al [9] have used its *signal-to-noise ration* (SNR) and the *root mean squared error* (RMSE). The higher the SNR and the smaller the RMSE, the better the denoising effect of the wavelet transform. The same evaluation techniques have been used in this paper. Below the formula for the SNR and the RMSE :

$$\mathrm{SNR} = 10 \log \left[\frac{\sum_{j=1}^{\mathrm{N}} \mathbf{x}_j^2}{\sum_{j=1}^{\mathrm{N}} (\mathrm{x_j} - \hat{\mathrm{x}}_\mathrm{j})^2} \right] \tag{1}$$

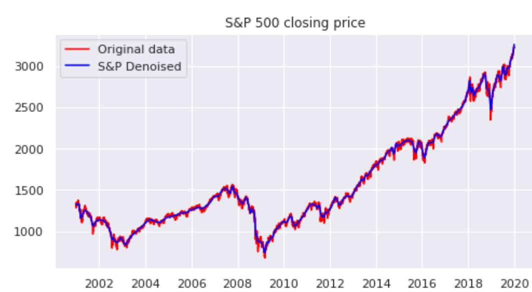
$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} (\hat{\mathrm{y}}_\mathrm{i} - \mathrm{y}_\mathrm{i})} \tag{2}$$

Just like Qiu and al [9], we found that *coif3* wavelet function was the most suited wavelet for the decomposition task with the highest SNR et the lowest RMSE as you can see in the following figure.

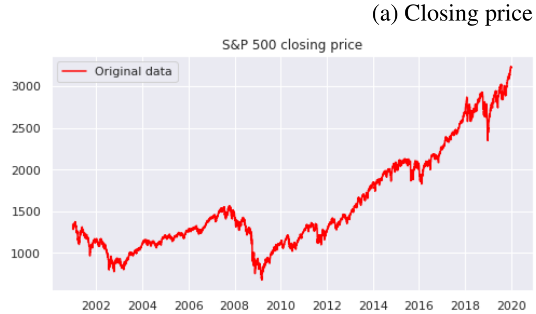
	haar	sym3	coif3	db3
wavelet	haar	sym3	coif3	db3
SNR	81.6796	83.3746	84.2824	83.3746
RMSE	28.4686	26.1596	24.9941	26.1596

Figure 4: Evaluation of wavelet denoising functions

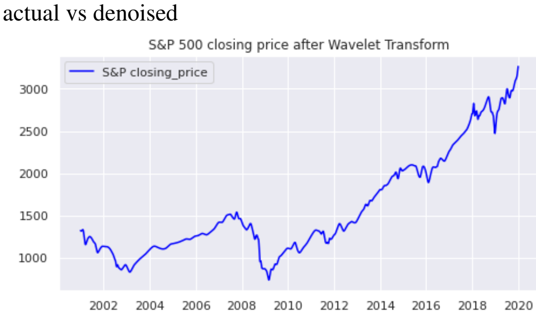
We therefore chose the *coif3* function to denoise the closing price. Fig2 compares the closing price curves with and without wavelet transform. As expected, the noise seems to be smaller after denoising.



(a) Closing price actual vs denoised



(b) Actual closing price



(c) Denoised closing price

Figure 5: Closing price with and without denoising

3.4 TRENDS MODELING

As stated in the introduction of this paper, our main goal is to predict S&P 500 trends with macroeconomic variables. Several trends modeling techniques have been used in our study.

3.4.1 Stock price mean and difference

Given a timestep T ranging from 1 to 15 days, trends are computed as follow:

- **Difference** : $Diff = C_t - C_{t-T}$ where C_t is the closing price at timestep t. If the $Diff$ value is positive then the trend is $+1$ (uptrend) otherwise it is -1 (downtrend)
- **Mean** : Compute the average closing price over T days (Av_T) and compare it to the average of the T previous days price (Av_{t-T}). If Av_T is greater than Av_{t-T} then the trend is $+1$ otherwise it is -1.

3.4.2 Technical indicators

A part from these two trends indicators, we have also used technical indicators to compute trends. Technical indicators are mathematical formulas based on the study of stock price or volume. These indicators are used in technical analysis to analyze the viability of a potential investment. Below are the indicators and their formulas [5]:

- **RSI** : The Relative Strength Index is a momentum oscillator that measures the speed and change of price movements. It oscillates between 0 to 100. If the RSI value surpasses 70 then the trend is a *downtrend*, and if the value goes below 30 then the trend is an *uptrend*. For values between 30 and 70, if the current value (time t) is larger than the prior value (time $t-1$) then the trend is an *uptrend*, otherwise it is a *downtrend*.

  $$\text{RSI} = 100 - \frac{100}{1 + \frac{AvgU_N}{AvgD_N}}$$

  Where $AvgU_N$ is the average of all up moves over the last N periods and $AvgD_N$ is the average of all down moves in the last N periods.

- **MACD :** The Moving Average Convergence Divergence is what we call a "trend follower" indicator. Indeed, it is used by traders to determine the direction of a trend since it indicates changes in the speed of price movement. It is computed by substracting two Exponential Moving Averages (EMA) : a 12 periods and 26 periods EMA.

  $$MACD_t = EMA(12)_t - EMA(26)_t$$

  If the current value (time t) of the computed MACD is greater than the previous value (time $t-1$), then we have an *uptrend*, otherwise it is a *downtrend*.

- **CCI :** The Commodity Channel Index measures the difference between the current stock price and the historical average price. This indicator defines the time of entry or exit for investors by computing special signals. If the CCI value surpasses 200 then the trend is *downtrend* and if the value goes below 200 then the trend is an *uptrend*. For values between 200 and 200, if the current value (time t) is larger than the prior value (time $t-1$) then the trend is an an *uptrend*, otherwise it is a *downtrend*. Below the formula to calculate CCI

  $$CCI = \frac{1}{0.015} \frac{p_t - SMA(p_t)}{MD(p_t)}$$

  Where $p_t = \frac{p_{high} + p_{low} + p_{close}}{3}$ is what we call the *typical price*,
  $SMA(p_t)$ is the simple moving average and $MD(p_t)$ is the mean absolute deviation.
  p_{close} is the closing price at time t, p_{high} and p_{low} are respectively the high price and the low price at time t.

- **STCK :** Stochastic oscillator is a technical indicator that compare a given closing price to a range of its prices over a certain period of time. It was first developped in 1950s and is a popular indicator for generating overbought and oversold signals. The formula for the stochastic indicator is :

  $$STCK = 100 \frac{C_t - L_N}{H_N - L_N}$$

  Where C_t is the current closing price,
  L_N and H_N are respectively the lowest and the highest prices traded of the N previous trading sessions (the

lowest low and the highest high prices in the last N days).

If the current value (time t) of the STCK indicator is greater than the previous value (time $t-1$), then we have an *uptrend*, otherwise it is a *downtrend*.

- **ADO :** The Accumulation distribution Oscillator indicator is employed to observe the flow of money into or out of stock. Investors ordinarily use ADO line to find buying or selling time of stock or confirm the strength of a movement. The formula for the ADO indicator is :

$$ADO = \frac{H_t - C_t}{H_t - L_t}$$

H_t and L_t are respectively the high and low price a time t.

Date	close	rsi	MACD	CCI	STCK%	ADO	ADO_trends	STCK%_trends	MACD_trends	rsi_trends	cci_trends
2001-01-03	1347.560059	100.000000	-11.375763	66.666667	59.091918	0.002734	0.0	1.0	1.0	0.0	0.0
2001-01-04	1333.339966	80.762337	-9.479367	78.716507	65.278406	0.800950	1.0	1.0	1.0	0.0	1.0
2001-01-05	1298.349976	53.492969	-10.676785	-29.735146	53.979125	0.914615	1.0	0.0	0.0	0.0	0.0
2001-01-08	1295.859985	52.143676	-11.691893	-86.857456	45.348682	0.112874	0.0	0.0	0.0	0.0	0.0
2001-01-09	1300.800049	54.590891	-11.959890	-32.288047	41.772636	0.658622	1.0	0.0	0.0	1.0	1.0
2001-01-10	1313.270020	60.132876	-11.038807	-20.805097	44.885375	0.018504	0.0	1.0	1.0	1.0	1.0
2001-01-11	1326.819946	65.115028	-9.110450	63.308543	52.933349	0.238985	1.0	1.0	1.0	1.0	1.0
2001-01-12	1318.550049	60.172512	-8.173869	48.680341	54.653290	0.678072	1.0	1.0	1.0	0.0	0.0
2001-01-16	1326.650024	63.124868	-6.682430	51.471103	59.370391	0.080112	0.0	1.0	1.0	1.0	1.0

Figure 6: Technical indicators and trends

3.5 METRICS TO ASSESS MODELS PERFORMANCE

We have used different metrics to assess the performance of our models. Below are some classification metrics we computed.

Classification metrics		
Metric	Formula	Interpretation
Specificity	$\frac{TN}{TN+FP}$	Coverage of actual negative sample
Precision	$\frac{TP}{TP+FP}$	Coverage of actual positive sample
Recall	$\frac{TP}{TP+FN}$	Describes how accurate the positive predictions are
F1 score	$\frac{2*precision*recall}{precision+recall}$	Combines recall and precision (weighted average of these two metrics)
False Positive Rate	$\frac{FP}{TN+FP}$	Proportion of individuals with a known negative condition for which the test result is positive
False Negative Rate	$\frac{FN}{TP+FN}$	Proportion of individuals with a known positive condition for which the test result is negative

Where :

- TP : True Positives - the model correctly predicts the *uptrends*
- FP : False Positives - Actual trend is an *uptrend* but the model predicts a *downtrend*
- TN : True Negatives - the model correctly predicts the *downtrends*
- FP : False Negatives - Actual is a *downtrend* but the model predicts an *uptrend*

In addition to the above metrics, we also computed *AUC-ROC curves*. This is also a measure of classification models' performances. The Receiver Operating Characteristics, noted ROC, is the plot of True Positive Rates versus False

Positive Rates by varying thresholds. The ROC curve is a probability curve and AUC represents the area under that curve. This area indicates how much our classification model is able to distinguish between our two classes. The higher its values are, the better the model is capable of correctly classifying up and downtrends.

Another interesting approach to assess the performance of our classification models is to create a **trading bot** and try to put ourselves in the shoes of an investor. The main goal of this bot is to compare the gains that an investment could acquire by basing its decision on the trends predicted by our models and the actual gains.

4 STATISTICAL MODELS FOR FINANCIAL TIME SERIES FORECASTING

4.1 ARIMA(p,d,q)

Auto-Regressive Integrated Moving Average (ARIMA) models are the generalization of Autoregressive Moving Average (ARMA) models. ARIMA models are based on the idea that the information in the past values of the time series can alone be used to predict the future values. ARIMA models are most prominent methods in financial forecasting. Indeed, these models have shown efficient capability to generate short-term forecasts [10]. In ARIMA model, the future value of a variable is a linear combination of past values and past errors, expressed as follows:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots + \phi_q \epsilon_{t-q} \tag{3}$$

where :

- Y_t is the actual value
- ϵ_t is the random error at time t
- β_i and ϕ_j are the coefficients
- p and q are integers that are often referred to as autoregressive and moving average, respectively.

An ARIMA model is characterized by 3 terms : p, d, q. The parameter d refers to the number of differencing required to make the time series stationary.

In our ARIMA modeling, we also added macroeconomic indicators are exogenous variables. We have followed the following steps [11] to build our predictive model : model identification, parameter estimation and diagnostic checking.

Arima models just like other statistical modeling methods assume or require the time series to be stationary to be effective. Indeed, when a time series is stationary, it can be easier to model. Time series are stationary if they do not have trend or seasonal effects. Summary statistics calculated on the time series are consistent over time, like the mean or the variance of the observations.

We used the **Augmented Dickey-Fuller** to test whether the time series is stationary. The null hypothesis of the Augmented Dickey-Fuller is that there is a unit root, with the alternative that there is no unit root, in other words, stationarity exists. Unit roots are a cause for non-stationarity. If the *p_value* is above a critical size, then we cannot reject that there is a unit root.

Once we have seen that the time series is not stationary, we can make them stationary by differencing - compute the differences between consecutive observations. This will determine the d term of the Arima model. We instead choose to use **Statsmodel auto-arima** function which directly determine the model best parameters p, d, and q based on criteria like AIC, BIC. We found that the best parameters are $p = 2$, $d = 1$ and $q = 2$.

ARIMA Model Results

Dep. Variable:	D.close	**No. Observations:**	3763
Model:	ARIMA(2, 1, 2)	**Log Likelihood**	-15453.058
Method:	css-mle	**S.D. of innovations**	14.697
Date:	Sun, 11 Apr 2021	**AIC**	30932.116
Time:	13:47:09	**BIC**	31013.145
Sample:	1	**HQIC**	30960.928

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	16.6597	4.439	3.753	0.000	7.960	25.359
FDTR Index	-0.0301	0.273	-0.110	0.912	-0.566	0.506
Vix Index	-0.3084	0.031	-9.888	0.000	-0.369	-0.247
Surprise économique	-0.6077	0.879	-0.691	0.489	-2.330	1.115
Inflation US	-0.5864	0.216	-2.716	0.007	-1.010	-0.163
Confiance des consommateurs US	-0.1058	0.025	-4.163	0.000	-0.156	-0.056
Taux 10 ans US	0.8440	0.371	2.274	0.023	0.117	1.571
Chômage US	-0.4809	0.318	-1.513	0.130	-1.104	0.142
ar.L1.D.close	0.1704	0.112	1.516	0.130	-0.050	0.391
ar.L2.D.close	-0.7612	0.078	-9.744	0.000	-0.914	-0.608
ma.L1.D.close	-0.2276	0.114	-1.996	0.046	-0.451	-0.004
ma.L2.D.close	0.7662	0.074	10.400	0.000	0.622	0.911

Roots

	Real	**Imaginary**	**Modulus**	**Frequency**
AR.1	0.1119	-1.1407j	1.1462	-0.2344
AR.2	0.1119	+1.1407j	1.1462	0.2344
MA.1	0.1485	-1.1327j	1.1424	-0.2292
MA.2	0.1485	+1.1327j	1.1424	0.2292

Figure 7: ARIMA model parameters

Before predicting trends, we validate our model by checking the residual distribution which has to be a normal distribution centered at 0.

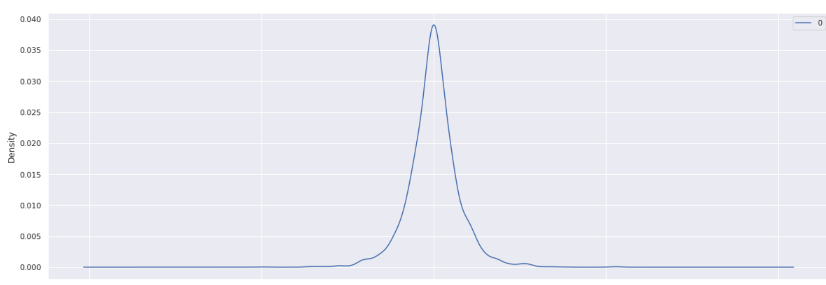


Figure 8: ARIMA Residuals distribution

4.2 Generalized AutoRegressive Conditional Heteroskedacity Model

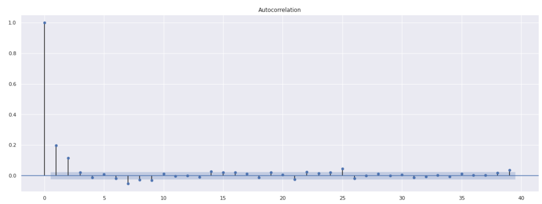
Financial time series are characterized by statistical properties known as stylized facts such as low autocorrelation, leptokurtic and assymetric distribution, long-range dependence and heteroskedasticity. Indeed, financial time series usually include floating and volatility models. Whereas the floating component is modelled with previous studied mean models such as ARIMA, the volatility if studied using ARCH/GARCH models.

4.2.1 Heteroskedacity

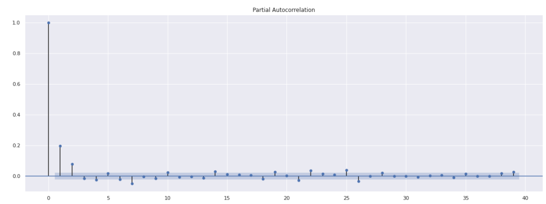
ARIMA-type models turn out to be inefficient with financial time series since they are appropriate when our process is a function of a series of unobserved shocks. Moreover, they are highly engrossed on the auto-covariance structure. The thing is, from a statistical point of view, financial returns are tightly knit to white noises.

To verify if a process is a white noise, we can check its autocorrelation and partial autocorrelation, among with its distribution.

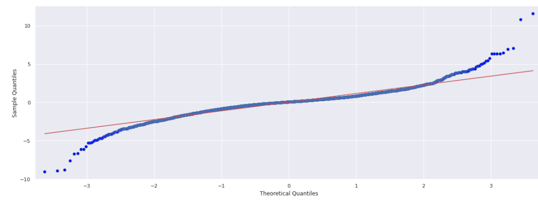
(a) SP500 Daily Return



(b) SP500 Daily Return Autocorrelation



(c) SP500 Daily Return Partial Autocorrelation



(d) SP500 Daily Return Q-Q Plot

Figure 9: SP500 daily return embodies white noises' properties

As we can see, SP500 daily return seems to be a random a centered about zero process. Moreover, we can notice that the autocoerraltion and partial autocorrelation highlight no statistically significant serial correlation. Finaly, the QQ Plots shows that our distribution is following a standard normal distribution. Thus, our daily return is definitely randomly distributed and seems to follow a Gaussian white noise.

Besides, financial time series are characterized by a non-constant conditional standard deviation. This phenomenon is known as conditional heteroskedasticity.

4.2.2 ARCH and GARCH models

ARCH, introduced by Engle in 1982, and GARCH models, developed by Bollerslev who generalized ARCH in 1986, are considered as the first statistical models that handle conditional heteroskedasticity. The general principle proposed by Engle (1982) is to assume that the variance depends on the information set available. Whereas Bollerslev defined the GARCH process with conditional variance dynamics.

4.2.3 Fitting GARCH

The idea with GARCH is that we can estimate the conditional volatility, which is a crucial component of financial time series. From the volatility, we deduce the trend of the financial time. Indeed, the higher the volatility is, the greater the fear index will be and people will start to sell, which would result on a decreasing trend of the financial time serie.

In our case, the volatility is embodied by the VIX Index which is computed by averaging the weighted prices of out-of-the-money puts and calls.

To perform our GARCH model, we start by making the SP500 Closing Price stationary. Several methods have been tested such as bijective transformation (differentiate, log function, square root function etc...). Once our closing price is stationary, we attempt to determine the best order (p,o,q) model. Based on the Akaike information criterion, we try every combination of $(p, o, q) \in \{0, ..., 10\} \times \{1, ..., 10\} \times \{0, ..., 10\}$. We find that the AIC of best model is 29813.48030 and order of best model is (0, 2, 2).

Before predicting trends, we validate our model by checking:

- The standardized residuals which should approach normal distribution

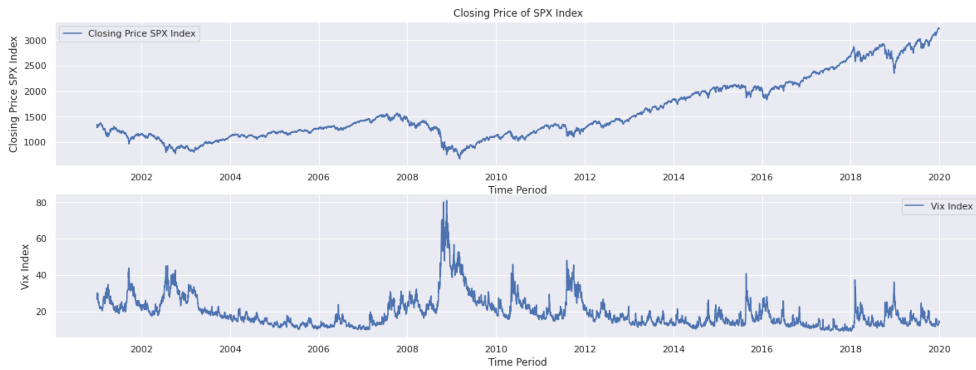
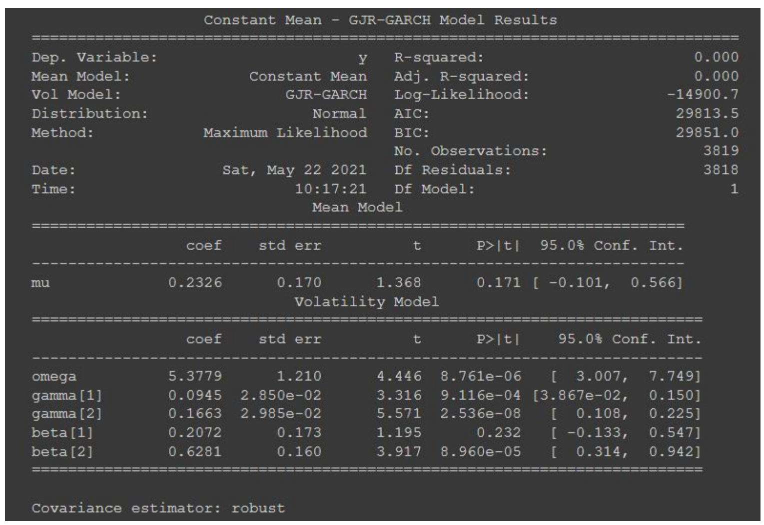


Figure 10: The link between SP500 Closing Price and VIX Index

- The standardized squared residuals which should not be auto-correlated



```
                     Constant Mean - GJR-GARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                  GJR-GARCH   Log-Likelihood:               -14900.7
Distribution:                  Normal   AIC:                           29813.5
Method:            Maximum Likelihood   BIC:                           29851.0
                                        No. Observations:                 3819
Date:                Sat, May 22 2021   Df Residuals:                     3818
Time:                        10:17:21   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.2326      0.170      1.368      0.171 [ -0.101,  0.566]
                               Volatility Model
==============================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
------------------------------------------------------------------------------
omega          5.3779      1.210      4.446  8.761e-06   [  3.007,  7.749]
gamma[1]       0.0945  2.850e-02      3.316  9.116e-04 [3.867e-02,  0.150]
gamma[2]       0.1663  2.985e-02      5.571  2.536e-08   [  0.108,  0.225]
beta[1]        0.2072      0.173      1.195      0.232   [ -0.133,  0.547]
beta[2]        0.6281      0.160      3.917  8.960e-05   [  0.314,  0.942]
==============================================================================

Covariance estimator: robust
```

Figure 11: GARCH Model results

Once our model is validated, we check the distribution of the standard residuals and compare it to the distribution of the SP500 return.

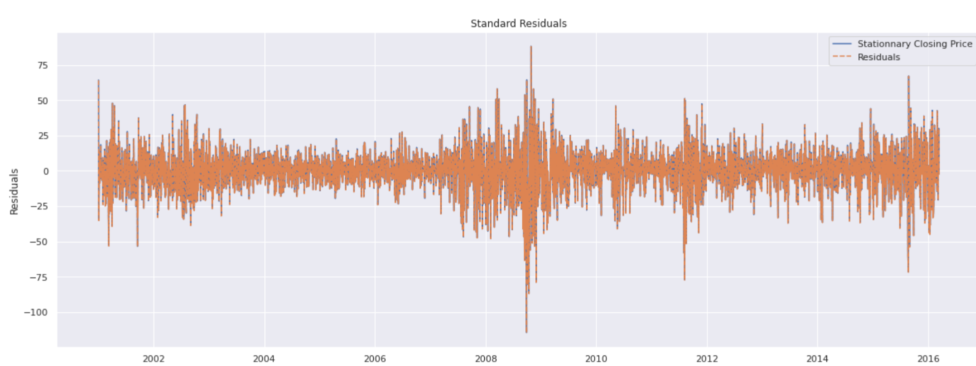


Figure 12: Standard Residuals of GARCH model

The residuals of the GARCH model are perfectly superimposed on the SP500 return. Thus, our standardized residuals appear to be randomly distributed and seem to follow a Gaussian white noise

Then, it is about computing the conditional volatility and compare it to the VIX Index to see if our model approximate well the actual conditional volatility of the closing price.

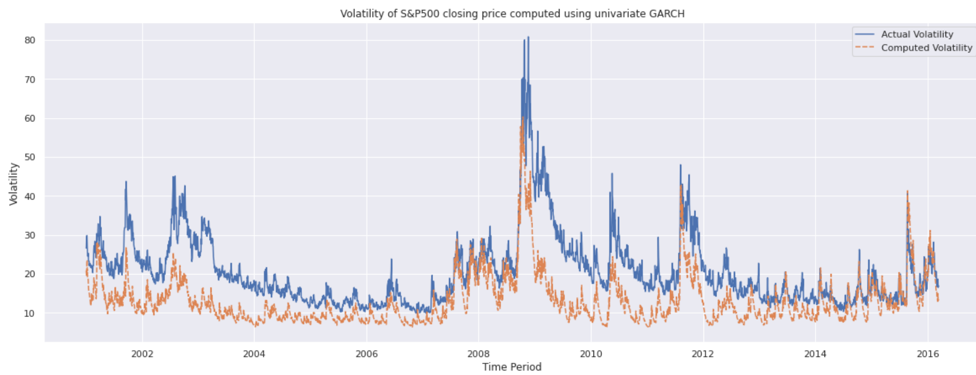


Figure 13: Conditional volatility of GARCH model and VIX Index

We can notice that the calculated volatility is very close to the real volatility. Moreover, we note the sensitivity of the GARCH model to strong variations which are amplified compared to reality. Overall, we note very similar patterns, despite the increased noise.

4.3 VAR

Vector autoregression (VAR) is a statistical model used to capture the relationship between multiple quantities as they change over time. In other words, VAR is a multivariate forecasting algorithm that can be used when two or more time series influence each other. In the VAR model, each variable is modeled as a linear combination of past values of itself and the past values of other variables in the system. The VAR model has proven to be especially useful for describing the dynamic behavior of economic and financial time series and for forecasting.

VAR model is different from other autoregressive models (AR, ARMA, ARIMA) since the latter are uni-directional, i.e only the macroeconomic variabes in our case influence the target (S&P 500 closing price) and not vice-versa. Whereas, Vector Auto Regression (VAR) is bi-directional. All the variables influence each other.

In order to test the relationship between macroeconomic indicators and the stock market price, we performed **Granger Causality** test before building the model. Note that VAR model requires all time series to be stationary. Among all the variables, only two were tested stationary with *ADF statistical test* : the VIX Index and the US Inflation rate. We made the remaining variables stationary by differencing as we did in the Arima model.

Granger's causality evaluates the influence between time series in a system and determines whether or not they are related. Granger's causality is based on the fact that X_t (macroeconomic variable) causes Y_t (closing price) if the prediction of Y_t conditional on its past values is improved by also taking into account X_t past values. This test helps us understand and know if the information we have about X_t and its past improves the knowledge of Y_t. The null hypothesis is that the coefficients of past values in the regression equation is zero. In other words, the past values of time series X_t do not cause the other series Y_t
We implemented the Granger's Causality test for all possible combinations of the time series and stored the p-value (see figure below).

Looking at the p_values in the above table, we can observe that all the time series in the system are interchangeably causing each other. We can keep these macroeconomic variables for our forecasting task. We splitted the dataset into a training and test set (80% vs 20%). The training set will be used to fit the model and the test set to assess its performance.

	FDTR Index	Vix Index	GDP US	Surprise économique	Inflation US	Confiance des consommateurs US	Taux 10 ans US	close	volume	Chômage US
FDTR Index	1.0000	0.0000	0.2644	0.0188	0.0000	0.0026	0.0990	0.0000	0.0052	0.0746
Vix Index	0.0002	1.0000	0.0001	0.0050	0.0002	0.0049	0.0006	0.0000	0.0000	0.0138
GDP US	0.0000	0.0000	1.0000	0.0001	0.0000	0.3946	0.0001	0.0856	0.0022	0.0000
Surprise économique	0.0185	0.0002	0.8357	1.0000	0.5204	0.8062	0.0002	0.0010	0.0020	0.0000
Inflation US	0.0000	0.0000	0.0141	0.0081	1.0000	0.0000	0.0001	0.0000	0.0003	0.5551
Confiance des consommateurs US	0.0000	0.0000	0.8647	0.0000	0.1591	1.0000	0.0001	0.0000	0.0000	0.1643
Taux 10 ans US	0.0350	0.0001	0.1505	0.1018	0.0005	0.0083	1.0000	0.0034	0.0004	0.0917
close	0.0647	0.0000	0.2976	0.2509	0.0094	0.0119	0.0849	1.0000	0.0104	0.0665
volume	0.0383	0.0000	0.0280	0.0483	0.1846	0.0017	0.0001	0.0000	1.0000	0.0000
Chômage US	0.0000	0.0000	0.0103	0.0007	0.0000	0.0356	0.0719	0.0009	0.0074	1.0000

Figure 14: Granger Causality test

VAR model is characterized by its order p. To conduct optimal order search, we used Information Criterion Akaike (AIC) as a model selection criterion. We selected VAR p order based on the best AIC score. The best order found is $p = 1$ as shown in the figure below.

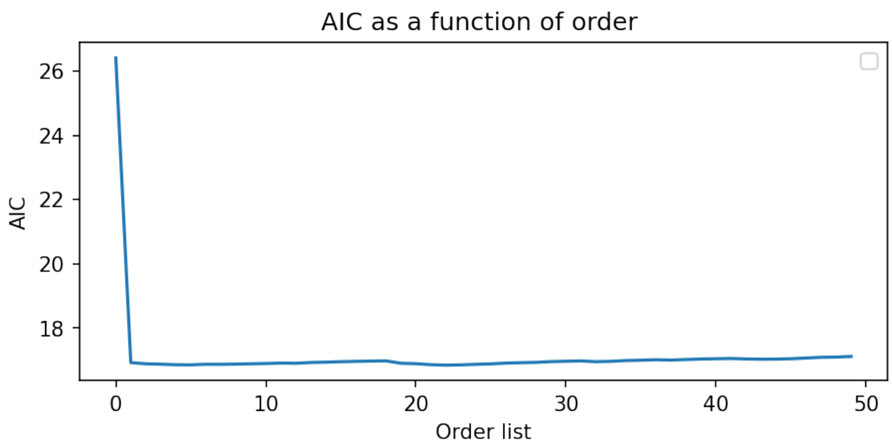


Figure 15: Granger Causality test

Once the best order found, we train the VAR model on the selected order.

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Fri, 26, Mar, 2021
Time:                     10:39:43
--------------------------------------------------------------------
No. of Equations:         9.00000    BIC:                    17.5448
Nobs:                     3758.00    HQIC:                   17.1025
Log likelihood:          -79254.2    FPE:                2.09646e+07
AIC:                      16.8583    Det(Omega_mle):     1.87903e+07
--------------------------------------------------------------------
```

Figure 16: VAR model summary

As in ARIMA model, before making any prediction, we have to validate the model by checking the residuals. There should not be any correlation left in the residuals, otherwise it would mean that there is some pattern in the time series that is still left to be explained by the model. In that case, the typical course of action is to either increase the order of the model or induce more predictors into the system or look for a different algorithm to model the time series. In order to check for correlation in the residuals which will ensure that the model is sufficiently able to explain the variances and patterns in the time series, we implemented **Durbin Watson Statistic**.

The Durbin–Watson statistic is a test statistic used to detect the presence of autocorrelation at lag 1 in the residuals (prediction errors) from a regression analysis. It reports a test statistic, with a value from 0 to 4, where:

- The closer it is to the value 2, then there is no significant serial correlation
- The closer to 0, there is a positive serial correlation
- The closer it is to 4 implies negative serial correlation

```
FDTR Index : 2.0
Vix Index : 2.0
Surprise économique : 2.0
Inflation US : 2.0
Confiance des consommateurs US : 2.0
Taux 10 ans US : 2.0
close  : 2.0
volume : 2.01
Chômage US : 2.0
```

Figure 17: VAR residuals correlation

4.4 STATISTICAL MODELS RESULTS

This section aims at summing up the main results obtained with our statistical models. Overall, after a linear regression performed with ARIMA, GARCH and VAR models, we proceeded to a trends classification from our regression. Thus, our models will be evaluated using both regression metrics and classification metrics since we compare the predicted values of Closing Price or VIX Index to the actual values and the trends deduced from the regression.

Regarding our regression, we perform different kind of predictions. Indeed, the key parameters was the forecasting window. Given that it is pretty difficult perform long-term predictions due to the error propagation, we decided to focus on short-term predictions, namely 3 days, 7 days or 10 days. Another idea was to perform successively 1 day prediction over the whole test set to avoid any propagation issue.

Once our prediction is done, we classify the trends of our regression given different indicators that have been introduced previously.

4.4.1 GARCH - Results, interpretation and performance analysis

In this section, we present our results for a 3-days prediction performed with GARCH. We noticed that greater prediction window led to an exponential increase of metrics that makes our model worser.

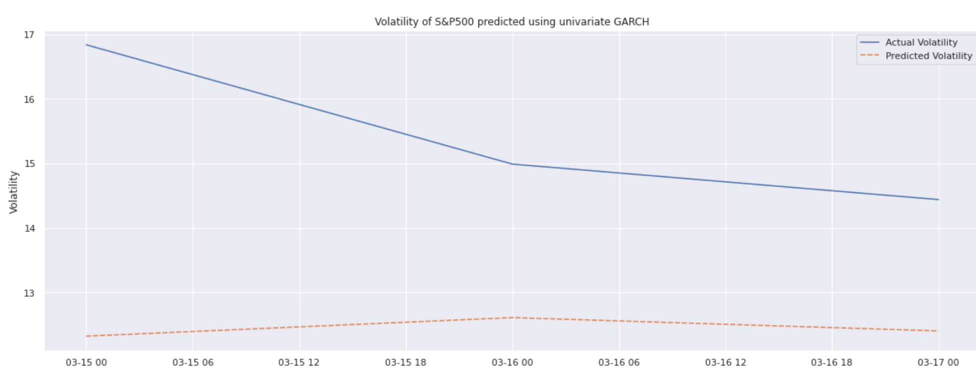


Figure 18: GARCH : Actual and predicted volatility on 3-days

MAE	MAPE	MEDAE	MSE	RMSE
2.977934	18.935064	2.379776	10.072738	3.173758

Globally, predicted values are far from actual ones. Our model is not even able to forecast the next day value accurately.

MAE	MAPE	MEDAE	MSE	RMSE
4.658431	29.081371	2.943814	47.145374	6.866249

We can notice that performing on the whole test set increases regression metrics. However, our method allowed us to avoid any exponential increase and have relevant predictions.

About classification, we will use difference and average indicators. Some metrics are used to evaluate our model performance quantitatively.

Metrics	Average-3	Difference-3
Accuracy	0.545455	0.526646
F1	0.52459	0.597333
Precision	0.519481	0.5
Recall	0.529801	0.741722

First of all, the average indicator and the difference one seem to be quite similar since no significant out-performance is made by none of them. Moreover, we notice that even with a small forecasting window, our model is similar to a thrown of parts, namely a probabilistic Bernoulli's experience.

To evaluate our classification, we will focus on the confusion matrix which shows how far our model is able to predict the right class of trend.

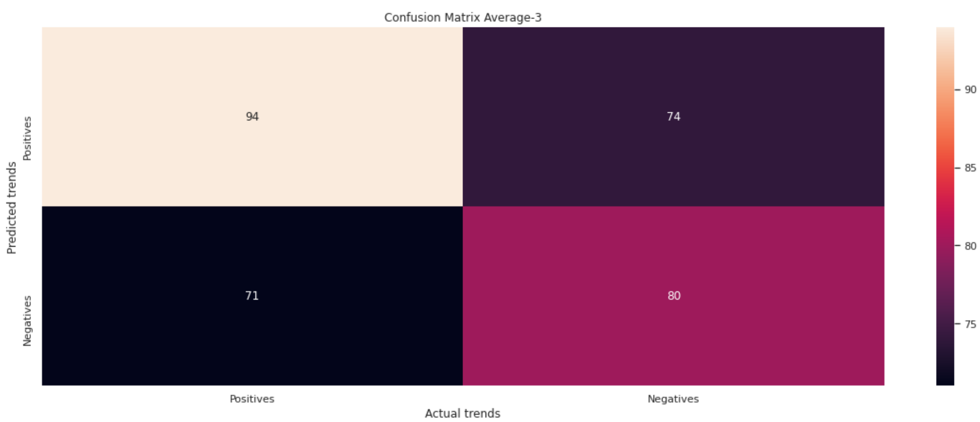


Figure 19: GARCH : Confusion matrix using average indicator for 3-days forecasting

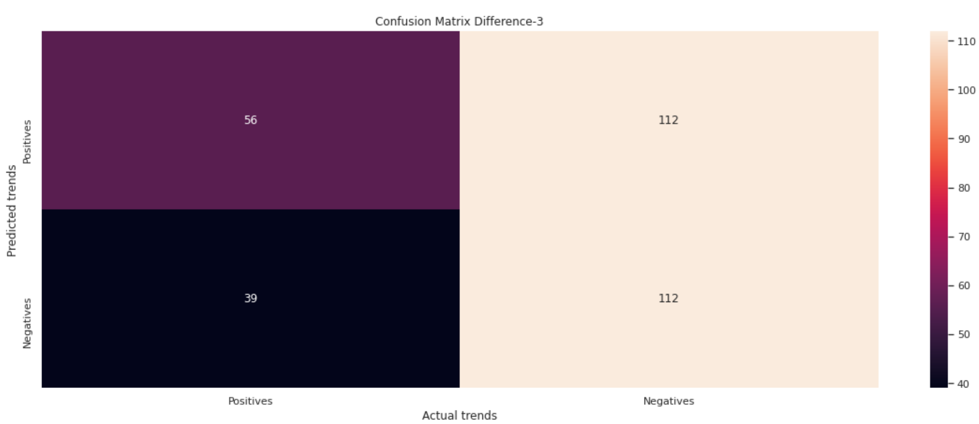


Figure 20: GARCH : Confusion matrix using difference indicator for 3-days forecasting

Confusion matrix are in the picture of regression and classification metrics. Our model associate randomly each trend since

To conclude on GARCH model, it is pretty clear that it does not manage to predict conditional volatility's values. Thus, trends classification is doomed to fail. As a result, our model is enable to forecast conditional volatility, even on short-term window.

4.4.2 ARIMA - VAR : Results, interpretation and performance analysis

In this section, we present our results for a 5-days prediction performed with ARIMA and VAR.

After performing the regression, we focus on the trends classification. We decided to perform directly successively 5-days prediction over the whole test set.

In order to have a significant test set that will allow any relevant interpretation, we decided to perform successively 3-days forecasting on the whole test set. We then focus on how our model is able to classify correctly trends.

Metrics	Arima Difference-5	VAR Difference-5
Accuracy	0.5558	0.597
F1	0.5434	0.561
Precision	0.5375	0.610
Recall	0.5434	0.605

We notice that ARIMA and GARCH are quite similar. However, VAR outperforms those models by better classification metrics.

5 STOCK TRENDS PREDICTION WITH LSTM

Instead of predicting stock price via a regression and use the predicted values to classify up and downtrends, in this section we focus on a classification task using our different trends modeling and a LSTM.

5.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks (RNN). RNN are neural nerworks that have internal memory. They are designed to recognize patterns in sequences of data, such as numerical times series data. These networks take as their input not just the current input example they see, but also what they have perceived previously in time. They are commonly used in multiple machine learning tasks : financial time series forecasting, Natural Language Processing (Machine Translation, Text Summarization ect.), Speech recognition (Apple Siri) etc.

The core concept of LSTM networks are the *cell state* with different *gates*. The cell state is the memory of the network. It transfers information all the way down the sequence chain. The reason why LSTMs are called networks with memory is that information from earlier time steps can make it's way to later time steps, reducing the effects of short-term memory. Information get's added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training. There are 3 gates in a LSTM network : the input gate, the forget gate and the output gate.

5.2 LSTM classification task results

For our classification task, we have used different LSTM architectures and all our trends modeling methods. At first, we only classified trends with the *difference* and *average* methods but we quickly realized that these trends modelings would not allow us to properly predict S&P500 up and downtrends. We then tried to add in our inputs technical indicators along with the macroeconomic variables. The results obtained were not quite different from the previous ones and this approach would deviate us a little from the initial task which was to use only macroeconomic indicators as inputs to our neural network in order to predict trends. We therefore opted for other trend models, i.e. those computed with technical indicators (see section 3.4.2), and used only macroeconomic variables as inputs. We also added in the inputs the denoised closing price.

The most suited LSTM architecture found for the classification is : a network with an input layer of 400 neurons and dropout of 0.4, 2 hidden layer of 150 and 300 neurons with dropout of 0.2 and 0.3 respectively. The second hidden layer has a $tanh$ activation function. The output layer contains 2 neurons with a softmax activation function. Below the results for the following four technical indicators : MACD, CCI, STCK and ADO.

Metrics			
Indicators	Downtrend accuracy (%)	Uptrend accuracy (%)	AUC score
STCK K‰	74.6	60.0	0.72
ADO	56.4	69.4	0.67
CCI	57.4	60.3	0.65
MACD	73.1	69.8	0.79

The Moving Average Convergence Divergence (MACD) and the Stochastic Oscillator (STCK) seem to outperform the other trends indicators in terms of accuracy per class (up and downtrend). AUC score of MACD is higher than STCK trend indicator auc. Based on the above table, we can say that Moving Average Convergence Divergence is the most accurate trend indicator among the four chosen indicators as we can observe in the figure below.

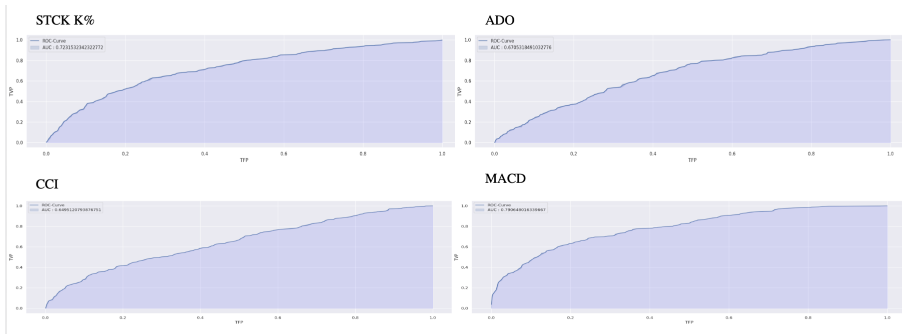


Figure 21: Indicators' ROC curves

5.3 Trading bot

The previous performance evaluation methods are not sufficient to conclude whether an investment with our model would be successful or how much an investment gains over time. Classification accuracy or error rate should not be the only assessment method in our case. Indeed, Not all mistakes are equal. Making investment mistakes on a high growth day is more penalizing than an error on a low variation day. In order to take this effect into account we decided to implement a trading bot. This allows us to put ourselves in a real investment situation.

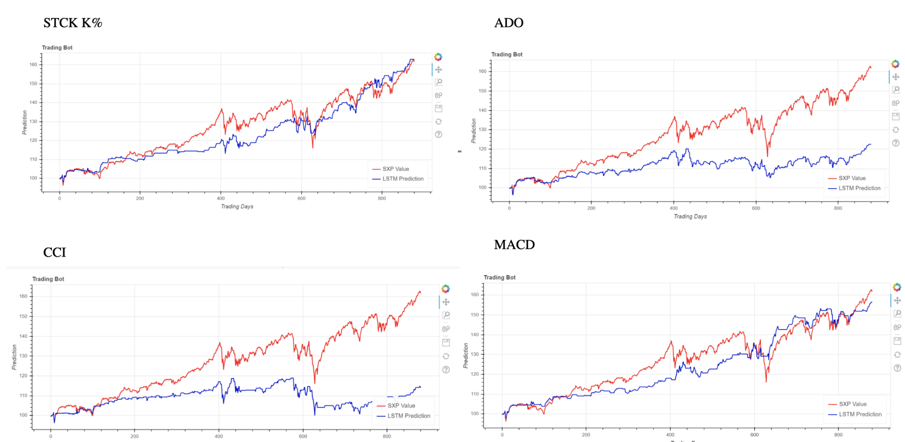


Figure 22: LSTM trading bot

How to interpret the tradind bot ?

Let's assume we have invested $base$ amount of money. If the bot trading curve (the blue curve which corresponds to the LSTM predictions) is above the red curve (S&P actual price), it means that our investment strategy generated by the model has allowed us to gain extra money. As we can see in the above figure, STCK and MACD are the only indicators that make an investor gains money on certain period. For the other two remaining indicator, the bot is always under the actual stock price. We now can conclude without any mistake that MACD is the best indicator among the four.

5.4 Hyper-parameters tuning

Since among the technical indicators used for trends modeling, MACD was the most accurate, we decided to use it for hyper-parameters tuning. We found that the hyper-parameters of the LSTM model that help maximize an investment profit are :

- Batchsize of 32
- 25 epochs
- Sparse Categorical Cross Entropy loss function
- Adam optimizer with 0.001 as learning rate

Below the metrics and trading bot of this model.

Classification	
Metrics	Value
Accuracy	0.678774
F1 score	0.684504
Precision	0.713953
Recall	0.657388

Figure 23: LSTM trading bot

6 PORTFOLIO MANAGEMENT

In this section, we looked directly at the stocks and tried to predict the best performing companies. To do so, we had several variables at our disposal. We had the prices of 291 stocks as well as their volatility, markups and markdowns. A markup is the difference between the market price of a security personally held by a broker-dealer and the price paid by a customer. A markdown in finance is the difference between the highest current bid price among dealers in the market for a security and the lower price that a dealer charges a customer. A bid price is a price for which somebody is willing to buy something, whether it be a security, asset, commodity, service, or contract.

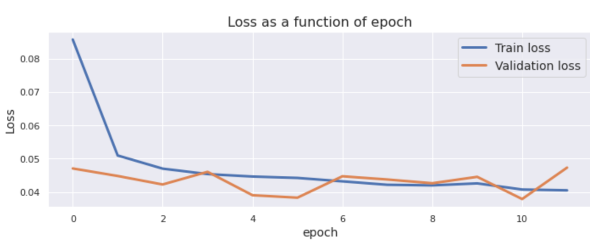
The main goal here is to predict the 6-month performance of stocks with the following inputs : volatility, markup and markdown and the previous 1, 3 and 6-month performances of the stocks. To compute these performances, we used the following formula:

$$Perf(t) = \frac{Current\ stock\ price}{Stock\ price\ (t)} - 1$$

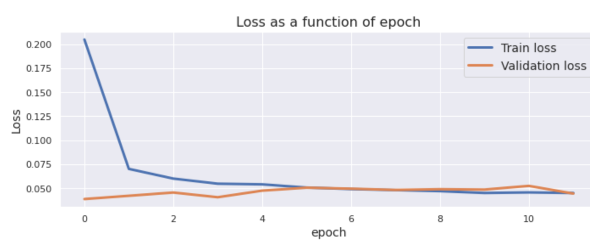
During the pre-processing step, we realized that around 80 stocks had 75% of null values which would result in error when computing the performances since we cannot divide by zero. We therefore decided to delete these stocks from our dataset and worked with the remaining 207 stocks. Two reccurent neural networks were used for the prediction task : a LSTM and GRU. Below the parameters used in the neural networks :

- Batchsize : 32
- Epochs : 12
- Input layer with 128 neurons and a dropout of 0.2
- 2 hidden layers with 100 neurons in each and dropout of $O.2$ and 0.1
- 1 output layer with a linear activation
- Mean squared error as loss function
- Adam optimizer with a learning rate $\rho = 0.001$

Below are the loss functions obtained after training the model. The GRU model converges better than the LSTM model.



(a) LSTM training vs validation loss



(b) GRU training vs validation loss

Figure 24: LSTM and GRU loss functions

	MAE	MEDAE	MSE	RMSE
LSTM	0.174590	0.136998	0.047296	0.217476
GRU	0.163357	0.117839	0.044314	0.210509

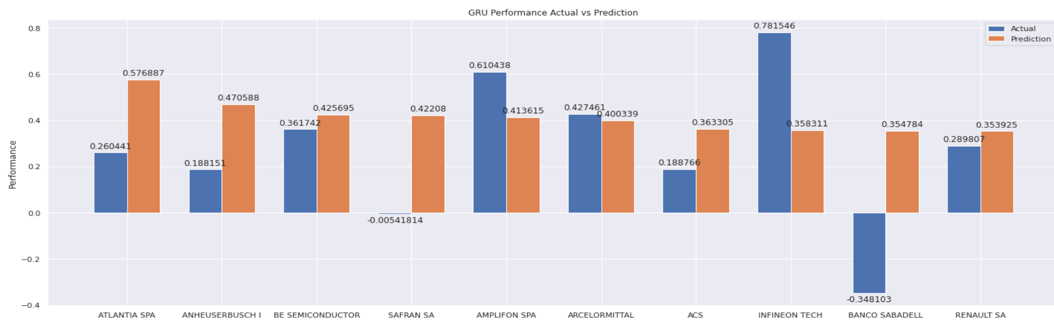
Figure 25: GRU vs LSTM regression metrics

We finally predicted the 207 stocks next six months performances after training the two deep learning models with 80% of the original dataset as the training set. Once we had made our six-month performance predictions for the LSTM and

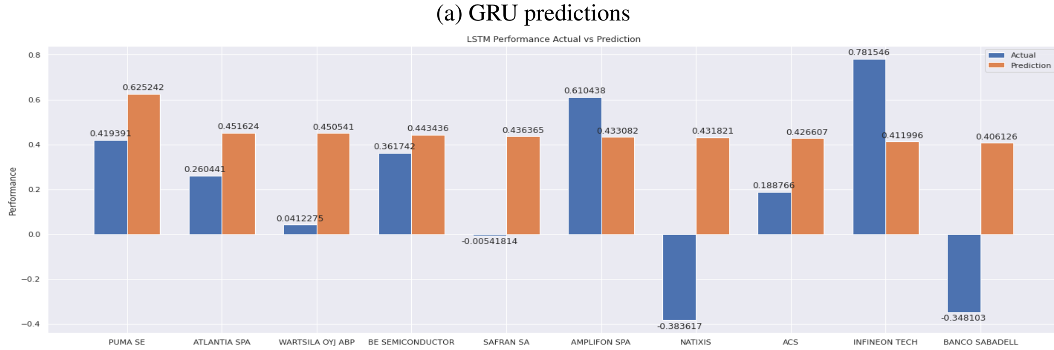
GRU model, the idea was to select the top ten stocks. Here is the approach we took.

Initially, we restricted ourselves to a smaller test set (a few months instead of 4 years) in order to be able to analyze all stocks. We have made an arbitrary first choice of market entry on the last day of each month of our test set, which ranges from March to October 2020, to get a first glimpse of the results. We can then generalize the approach to all the dates of our test set or optimize our market entry by considering for each model the best performing day of each month. Then, for the predictions of each model, we ranked in ascending order the six-month performance of the stocks for each day chosen to enter the market, to then select the ten best companies.

For each model, the idea is to compare the predicted performance of the top ten stocks with the actual performance of these same stocks. Below is a comparison of the results of the two models for the month of March.



(a) GRU predictions



(b) LSTM predictions

Figure 26: LSTM vs GRU on march

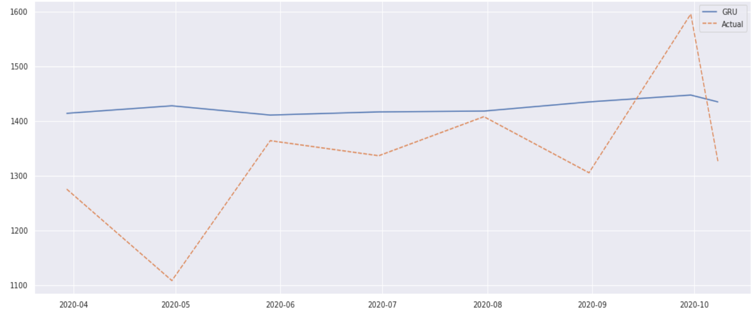
For the spanish company BANCO SABADEL, the GRU model predicts a negative six-month performance while the actual performance is positive. The remaining nine stocks have been predicted in the right direction even though for the some stocks the predicted value is too far from the actual one. As for the LSTM, two stocks were predicted in the wrong direction.

Once these results were established, we implemented our trading bot as follows: for a given investment denoted $invest$, we compute the portfolio value as follow :

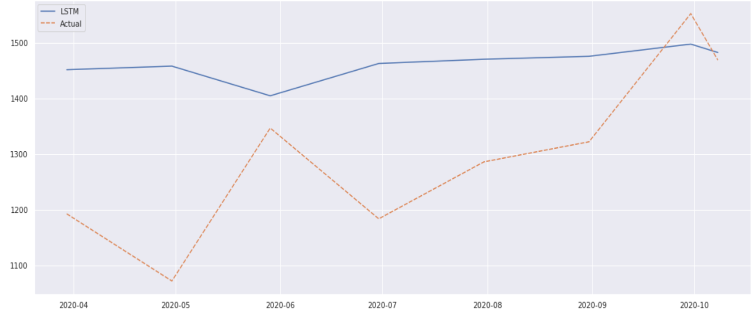
$$Value = (performance + number\ of\ stocks) * invest$$

This gives us for each chosen date, the return on our investment. We compare this predicted return to the actual return (considering in the same way the actual performance of these same stocks) in order to evaluate the relevance of our predictions. We obtain the following results:

The orange curve corresponds to the actual ROI of the ten top stocks once investment has been made and the blue curve is what an investor might earn if they take into account our model prediction. As seen previously, GRU performs better than the LSTM. GRU predictions are closer to the actual values than the LSTM.



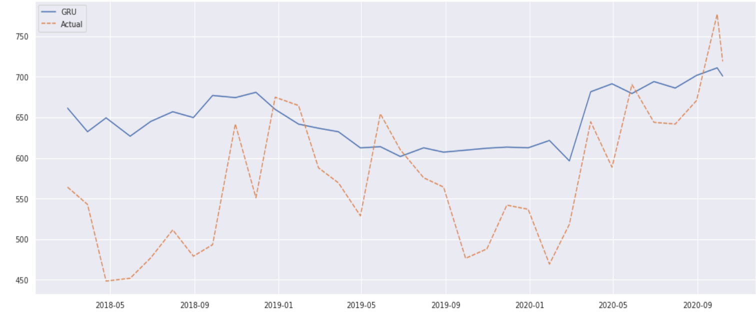
(a) GRU bot



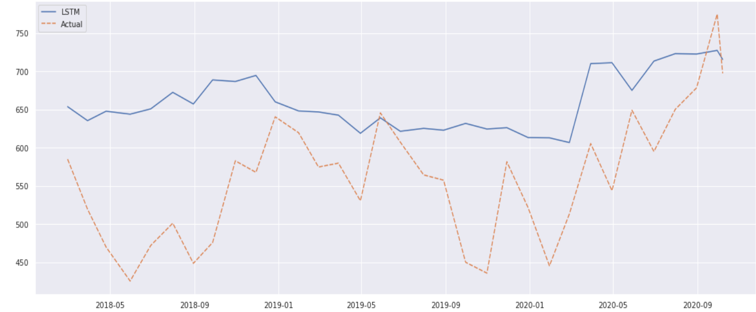
(b) LSTM bot

Figure 27: LSTM vs GRU trading bot

When we restrict ourselves to around 150 stocks, we have a larger base of comparison than the previous one and we obtain the following results :



(a) GRU bot



(b) LSTM Bot

Figure 28: LSTM vs GRU bot

Once again, GRU outperforms the LSTM model. However, actual value are quite different from prediction even though GRU predictions still are the closest to actual computed values of the portfolio.
We finally tried to look at country level. To predicts the top 10 stocks on which we could invest within the same country. Here is what we get for France.

Figure 29: France stocks : LSTM vs GRU

Predictions are much more interesting and accurate when we restrict ourself to a country level analysis.

7 CONCLUSION

In conclusion, this study allows us to examine several kinds of algorithms, from the basic linear models to sophisticated deep learning models to highlight the limits and fill gaps of each one. The statistical models were useful to points out how is to deal with financial time series by exploring their statistical properties and understand the major component of them. As we could expect, statistical models did not show any brilliant results. Trying to predict the future by studying the past values remains hardly realizable. However, VAR model shed light on the relevance of statistical multivariate model, that are more appropriate for our study. Better results were obtain with VAR, they remains yet insufficient. Exploring machine learning techniques was the opportunity to adopt another approach by studying time dependent neural networks, setting up another perspective by performing classification models and developing trading bot which is easier to understand from a client point of view. Moreover, we realized the limit of our preliminary statistical study about correlation between macroeconomic indicators and SP500. Undoubtedly, their is a clear link between them. However, the market is not only affected by macroeconomic indicators. In order to claim yearning for predicting the SP500, others variables or study-axis have to be used. Our portfolio management study sheds light on another approach which consist at looking at several stocks' performance using trading data. Overall, it is pretty difficult to precisely determine the most performing companies. However, we managed to highlight a sufficient number of companies in profit growth.

References

[1] J. B. Duarte Duarte, L. H. Talero Sarmiento, and K. J. Sierra Juárez, Evaluation of the effect of investor psychology on an artificial stock market through its degree of efficiency. Contaduría y Administración, vol. 62, no. 4, pp. 1361–1376, Oct. 2017.

[2] W. Long, Z. Lu, and L. Cui Deep learning-based feature engineer- ing for stock price movement prediction. Knowl.-Based Syst., vol. 164, pp. 163–173, Jan. 2019.

[3] H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, M. M. Selim, and K. Muhammad. A local and global event sentiment based efficient stock exchange forecasting using deep learning Int. J. Inf. Manage., vol. 50, pp. 432–451, Feb. 2020.

[4] J.Long, Z.Chen, W.He, T.Wu, and J.Ren An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in chinese stock exchange marke Appl. Soft Comput., vol. 91, Jun. 2020, Art. no. 106205.

[5] MOJTABA NABIPOUR, POOYAN NAYYERI, HAMED JABANI, SHAHAB S. AND AMIR MOSAVI, Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis IEEE ACCESS, 2020.

[6] Z.Y. Pang and L. Liu, Can Agricultural Product Price Fluctuation be stabilized by Future Market: Empirical Study Based on Discrete Wavelet Transform and GARCH model Financial research 11 (2013) 126-139.

[7] Z. Li and V. Tam, Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-8, doi: 10.1109/SSCI.2017.8280883.

[8] Towards DataScience, Wavelet Transform : An Introduction and example https://towardsdatascience.com/the-wavelet-transform-e9cfa85d7b34

[9] Jiayu Qiu, Bin Wang, Changjun Zhou Forecasting stock prices with long-short term memory neural network based on attention mechanism PLoS ONE 15(1): e0227222. https:// doi.org/10.1371/journal.pone.0227222

[10] A. Meyler, G. Kenny and T. Quinn, Forecasting Irish Inflation using ARIMA Models Central Bank of Ireland Research Department, Technical Paper, 3/RT/1998.

[11] B.G. Tabachnick and L.S. Fidell, Using multivariate statistics 4th ed., Person Education Company, USA 2001