uc3m | Universidad Carlos III de Madrid

Master in Big Data Analytics

2021-2022

Master Thesis

Machine Learning for Predicting Trends in Futures of Commodities

Antonio Carrera Maestro

Emilio Parrado Hernández

Madrid, June 28th 2022

SUMMARY

Financial time series prediction has always been a challenge in the industry and the academia. These series are influenced by many different factors, and change in a really fast way. Artificial intelligence has recently emerged as an alternative solution to econometric models for these problems. This project studies the application of different machine learning models to a case of interest, related to the prices of commodities.

The availability of open-source financial data allows easily obtaining price series of any commodity on the Internet. The objective of this project is to design a machine learning based classification scheme to identify different trend directions of the price of a commodity future contract. For that purpose, features must be extracted from the raw dataset and a label must be generated to identify the trend direction. The features that are under study are trading indicators and features related with the temporal dynamics of the series.

Six commodity futures series are compared on this project, belonging to distinct categories. Once the dataset is generated, four classifiers with different structures are trained to predict the generated labels. Several cross-validation methods are evaluated and the prediction periods are also varied to assess its impact on the final result. Multiple feature combinations are analyzed to describe the data from different points of view. The results show that the problem changes significantly for each series, and that although similarities can appear between them, it is difficult to find a global model to describe them jointly.

Keywords: Machine learning, commodities, futures, prediction, cross-validation, classification, trend.

DEDICATION

This project could not have been completed without the help of Emilio Parrado. Thanks for introducing me to the financial world and for all the tips that you have given me, both in technical and personal terms to face my future career.

Thanks to Marina, who has supported me through thick and thin and has experienced the development of this project with me. Thanks to my parents too, who have always backed me in my decisions and have given me everything.

Thanks to my personal friends and my classmates. These years have been wonderful because of you. Long live Consulate.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Context

The irruption of Artificial Intelligence has transformed the working procedures in several industrial sectors. The financial sector has experienced a great transformation associated to this breakthrough. Multiple parts of the sector are experiencing an incremental AI deployment across different industries, such as banking, asset management, trading or insurance [1]. These improvements constitute a key part of the digital transformation that the financial sector is undergoing.

Focusing on the trading sector, new possibilities have opened up. In the past, decisions were exclusively dependant on traders, which are the experts in the field, that took the decisions to buy or sell a certain product based on their expertise. Nowadays, traders are still operating, but artificial intelligence is breaking new ground in the field of research. The employment of these novel techniques allows for trading in an automatic way, either by themselves or complementing the decision of an expert. Hence, the research in this field is a relevant topic at the present time.

The purpose of research is generally looking for the most profitable approach, either to forecast the price of a financial instrument or to determine its future trends and maximize the income. The focus in this project differs from the business point of view and tries to approach the problem in an academic way. The goal is to define an interpretable machine learning scheme, where it is not necessary to outperform the current professional classifiers, but to extract insights about the behaviour of the series and interpret the results.

1.2. Objective

The main objective of this project is to design a classification scheme that is able to, starting from a raw time series containing the price of a commodity future contract, predict the future trend of the price. This approach differs from the classical time series forecasting framework, where the objective is to predict the exact price of a series and their respective confidence intervals. Therefore, the time series samples need to be summarized into a group of features. Then, a label needs to be generated to separate the samples in groups, according to the corresponding trends. This data will then be introduced to conventional machine learning classifiers to discover whether the generated dataset is useful to predict the future trends or not.

Nowadays, there is a great amount of information available on the Internet. Several web sites as *Yahoo! Finance* or *Tradingview* offer APIs and open-source tools to download the price series of almost any asset. This fact impacts directly the characteristics of the data, for example in terms of potential missing values or time granularity of the samples. On the other hand, the companies that are specialized in this field have additional information from private sources that allows them taking advantage over their competitors. Because of this, the results need to be put into context, as there are many underlying factors that are difficult to define without those pieces of information.

Financial time series prediction is a challenging issue, as markets involve a great number of people and lots of money. Hence, uncertainty is a factor that is always present in this kind of problems. The experts in the field are able to develop trading strategies to decide when the actions have to be executed. In exchange, a data scientist does not need this information to develop a classification strategy. Consequently, the objective is to explore alternative approaches to the traditional point of view, trying to extract value from data without considering its nature, but just focusing on its time series structure.

The goal of this project also differs from the classical approach since the aim is to be able to properly separate the classes instead of maximizing the final profit. For this purpose, a different strategy needs to be defined. Initially, there is not a defined reference to follow, so several approaches can be tested. A key part of this strategy is to define the time window for the prediction, because the problem can change significantly depending on if the prediction is one day ahead or one month ahead, as an example. Finally, the code needs to be modular in order to easily adapt to different experiments.

1.3. Structure of the report

This section includes a brief description of the contents of each of the following chapters, in order to ease the reading of the document.

Chapter 2 gives an overview of how the problem of financial time series prediction has evolved with time and how artificial intelligence is playing a key role on its development. Commodities and future contracts are also described in detail to put the problem in context. Finally, state-of-the-art models are also commented.

Chapter 3 describes the methodology that has been followed and the reasoning behind each decision. It starts from the raw data extraction and ends with a labeled dataset that is ready to be used. The feature extraction process and the trend detection are described here. The classification schemes and the cross-validation methods are also presented.

Chapter 4 includes the results for each of the classifiers, in terms of accuracy and standard deviation. Various alternatives are compared and some insights are extracted from each one of them. Finally, a global comparison is made and the best-performing models for each problem are shown.

Chapter 5 summarizes the achieved goals and compares them to the previously defined objectives. There is also a brief commentary on potential future works related to this topic.

2. STATE OF THE ART

2.1. Financial time series forecasting

Financial time series modeling and forecasting has been in the spotlight of researchers for a long time. This topic is still relevant nowadays, and there is much research both in theoretical and applied points of view [2]. These series are characterized by non-linearity and non-stationarity, which in conjunction with the random walk behavior and the influence of external factors make them one of the most challenging tasks when dealing with time series forecasting [3].

There are several ways to approach this problem. Econometric models rely on mathematical formulations to forecast the price of an asset, and are able to predict the series up to a certain point. Prediction can also be based on trading strategies, which are defined by experts based on technical criteria. Finally, the engineering approach suggests applying artificial intelligence and eventually including exogenous predictors to the forecasting paradigm [4]. Another way to categorize the approaches is by distinguishing between technical analysis and fundamental analysis [5]. The first one focuses on predicting the prices based on historical data and indicators, involving mathematical and statistical processes to obtain a technical point of view. The second one uses information based on the financial situation of a single instrument and is more human-based, as it can refer to previous experience and interpretations of specialists. Currently, research is more advanced on the technical analysis due to its easier application with machine learning. On the other hand, fundamental analysis strategies are still being developed, due to the difficulties to integrate the processing of unstructured data in the artificial intelligence environment [6].

Classical time series methods have been widely used for forecasting financial time series. Nowadays, several studies involving Machine Learning techniques have been published, outperforming classical models in some cases [7]. Specifically, Deep Learning methods have shown the best performance among Machine Learning algorithms. Financial instruments are influenced by many different factors aside from economic aspects, as commented before. Artificial intelligence has emerged as a potential solution for this new challenge [8].

Several sources compare the performance of the classical econometric models and the new machine learning approaches. In 2010, [9] showed a comparison between Neural Networks and traditional ARCH models to forecast the exchange rates of various currencies, where Neural Networks outperformed classical models by 10%-15%. Another example can be found at [10], where multiple journals are compared, both including econometric models and Machine Learning models. The result was that Machine Learning models helped improving the overall accuracy. The paper also mentions the possibility of combining both types of models into a hybrid model as a potential future line of research.

2.2. Commodity markets

The previous point covers all types of financial time series. Depending on the source, markets can be classified in different ways according to the actives that are exchanged or their structure. A five-group classification [11] can be defined as: stock market for large corporations, bond market for loans, commodities for natural resources and primary products, derivatives for products basing their value in underlying assets and forex for currencies.

The target of this project is the commodity market, which is focused on trading with raw materials or primary products. Commodities can be categorized under two groups [12]:

- Hard commodities. Natural resources to be extracted or mined. Examples of this category are gold, oil or natural gas.
- Soft commodities. Agricultural products or livestock. Examples of this category are sugar, cotton or coffee.

One of the objectives of the project is to characterize both hard and soft types separately, to assess potential differences between them. According to theory [13], the price of soft commodities is strongly dependent on external factors such as the weather or the environment, making them more volatile. In exchange, hard commodities have a more fixed nature, which makes them more predictable and likely to be set as wealth-preservation assets.

These aforementioned groups can be divided into more categories or following a different scheme. As an example, commodities can also be grouped into energy, agriculture and metal commodities. Nevertheless, the distinction in this case will be made for soft and hard commodities. Figure 2.1 summarizes the classification of commodities [14]:

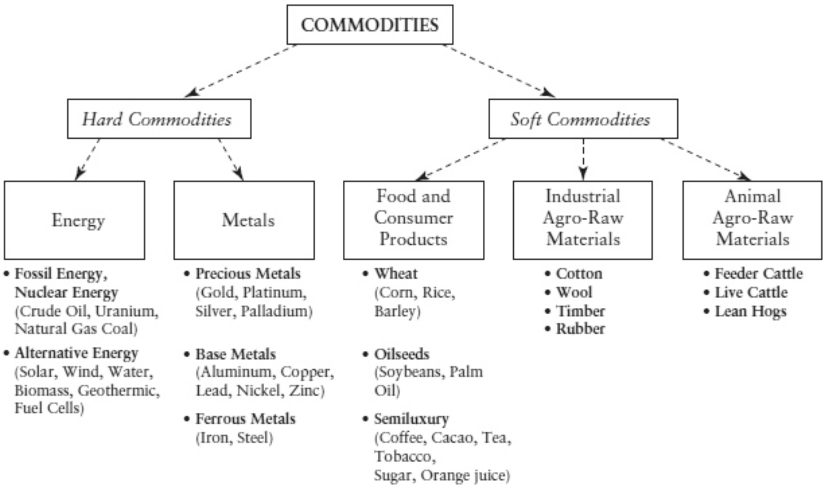

COMMODITIES
Hard Commodities
Soft Commodities
Energy
Metals
Food and Consumer Products
Industrial Agro-Raw Materials
Animal Agro-Raw Materials
• Fossil Energy, Nuclear Energy (Crude Oil, Uranium, Natural Gas Coal)
• Alternative Energy (Solar, Wind, Water, Biomass, Geothermic, Fuel Cells)
• Precious Metals (Gold, Platinum, Silver, Palladium)
• Base Metals (Aluminum, Copper, Lead, Nickel, Zinc)
• Ferrous Metals (Iron, Steel)
• Wheat (Corn, Rice, Barley)
• Oilseeds (Soybeans, Palm Oil)
• Semiluxury (Coffee, Cacao, Tea, Tobacco, Sugar, Orange juice)
• Cotton
• Wool
• Timber
• Rubber
• Feeder Cattle
• Live Cattle
• Lean Hogs

Fig. 2.1. Schematic of commodity types [14]

Commodities can be traded in different ways. The most known option is to buy or sell commodities immediately, technically known as spot markets. The main topic of this project is commodity futures market. The contracts from this market allow fixing a price for buying or selling an amount of a certain commodity at a future date that is previously agreed between the parties [15]. Until that time arrives, the price of the commodity can fluctuate significantly, so there is room for both large gains and losses. For this reason, the typical users of this market are big companies or institutions, which try to reduce risk from potential future price changes or speculate to profit from market movements.

Future contracts can be influenced by many different factors, such as market speculations, economic growth or unforeseen circumstances [16]. As the target is to set a price for a future date, it is important to consider the forecasting of the commodity prices apart from the factors mentioned above. Hence, the future prices modeling needs to account for all these drivers together, adding an extra difficulty to the problem.

The spot market and the futures market are then mutually dependent. The principal factors that relate them are the interest rate, the storage cost and the convenience yield. The convenience yield defines the status of the market, defining two states: *contango* when the future price is higher than the spot price and *backwardation* when it is the other way around. Based on these assumptions, [17] sets the futures prices as a benchmark for predicting spot commodity prices and compares it to a random walk benchmark, concluding that it is a valid baseline.

Once the commodity markets are properly defined, the next step is to look for a prediction approach related to the previously defined points. In this way, a feasible prediction scheme can be described to set a reference for further sections.

2.3. Trend detection and prediction

The main line of research related to commodity futures is linked to forecast the exact price of the series. As mentioned before, Deep Learning has shown a solid performance for this problem, so there is a great number of studies proposing solutions to this problem. The focus of this project is designing a classification scheme summarizing the information present on the trends and achieving a reasonable score. The number of projects related to this topic is lower than in the previous case, but there are still some approaches that are useful to set a reference.

The first step to proceed with the classification scheme is to define the output classes in a precise way. Regression schemes look for predicting an exact value, so in order to adapt to a classification problem, these values need to be grouped into classes. One of the most common approaches is to predict the future price direction of the price, grouping prices by 'up' and 'down'. More granularity can be introduced to the classification scheme by increasing the number of classes. There are some examples on the literature that define strategies to retrieve the trend from a price series.

Wu et al. describe at [18] a new method to carry out the trend labeling in a different way from traditional methods. Traditional methods usually focus on short-term variations, so that depending on the daily variation of a series, the signal can be very noisy. The authors focus in this case on detecting trends for longer periods of time, under the assumption that trends last for a relatively long time. For this purpose, a fluctuation parameter was defined to account for these small variations, acting as a threshold. The differences are easier to appreciate on Figure 2.2, where the black line represents the traditional method and the red line represents the improved method:

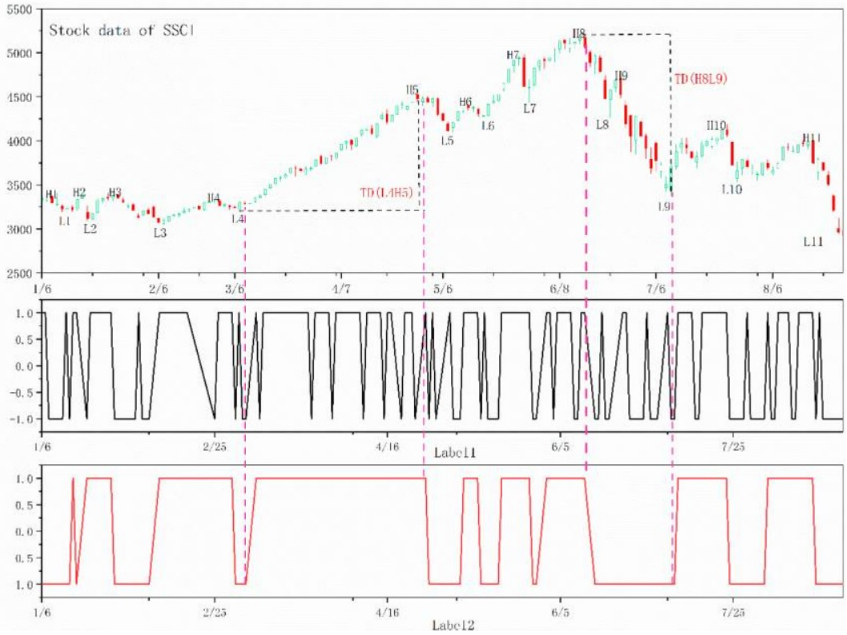


Fig. 2.2. Schemes for trend detection [18]

As it is observed above, the new method is able to successfully identify longer trends than the classical approach. Nevertheless, this previous method requires from a mathematical definition of the trend detection problem, that is designed based on technical criteria. There are alternative approaches, such as the one shown in [19]. In this case, the one-day-ahead trend is defined based on the values of the 25-day and 65-day moving averages and their comparison to the current prices. The moving averages have the advantage of smoothing the signal, so that daily peaks are eliminated and the medium to long trends are easier to spot. These two previously commented methods offer a preliminary approach for trend detection in short periods of time.

After defining the output classes, it is necessary to define the features that will be used as predictors. Following the classical approaches, technical indicators are the most commonly used variables, as many publications are based on technical analysis. An example of this scheme is shown in [20], where three types of indicators are used: trend-following indicators as moving averages to try capturing the trend dynamics, oscillators as RSI and Williams to guess future trend movements and volume as an indicator of demand and supply on the market.

As commented before, the commodity markets and in general all stock markets are influenced by many heterogeneous factors. Hence, some authors include exogenous predictors on their models, as it is done in [21]. This approach combines numerical and textual information to predict the future trend of a stock. Numerical features are based on the appliance of technical indicators to the raw data of the stock. On the other hand, textual features are generated through a sentiment analysis of economic-related texts collected from the Internet. These texts are pre-processed and then automatically labeled to obtain a discrete variable that can be used as a predictor for the model. This approach allows obtaining two perspectives from the market to carry out the classification: the technical perspective and the market sentiment perspective.

The classification scheme is the following step to define after obtaining the dataset. The literature proves that several different schemes can be useful for classifying trends, but usually they are oriented to a particular commodity or a concrete dataset structure. Nevertheless, there are some particular Machine Learning algorithms that have outperformed conventional techniques as logistic regression [22]. Among these models, three of them are specially interesting for this project:

- **Decision Trees**. Among all possible tree models, Random Forest is selected. Decision Trees are simple models that offer result interpretability, so they are useful for classification and for feature selection.
- **Support Vector Machines**. SVM algorithms are kernel-based and look for a solution maximizing the margin to separate groups over the N dimensions of the dataset.
- **Neural Networks**. The selected network structure is a Multi-Layer Perceptron. MLP Classifier can learn from the train set a non-linear function to classify the test set according to different solvers.

These algorithms are based on different fundamental concepts, so they can provide different solutions to the same problem. Considering that the prediction of financial time series is challenging, using a wide variety of methods may help finding the most suitable approach for each financial instrument individually. Following the procedures described on [22], the models can be combined as an ensemble to obtain joint predictions in different ways: averaging, weighted averaging and max voting. Hence, the combination may help solving problems that a single model may not be able to solve alone.

The final step is to define the validation strategy. Depending on the article, different methods are observed, specially k-fold cross-validation and walk-forward cross-validation. When defining the classification problem, it is necessary to account for the temporal dependency among samples, so that information leakage does not occur and the evaluation is carried out properly. This is also key for overfitting avoidance. Figure 2.3 shows an example of how walk-forward cross-validation can be implemented on a real scenario:

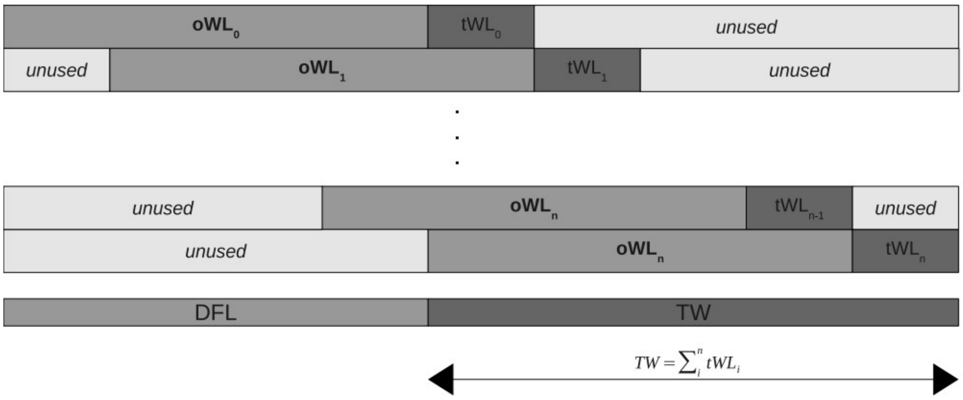


Fig. 2.3. Sliding-window walk forward cross-validation scheme [20]

The classification pipeline is now completely characterized. To assess performance, it is necessary to define an evaluation method to allow comparing results from different series under the same conditions. The most widely used metric in literature is accuracy [19], [22], and in some approaches a profit metric is also introduced to evaluate the earnings that the method would achieve [23].

3. DESIGN AND METHODOLOGY

The main point to be addressed in this section is the generalization capability of the problem. The classification scheme ranges then from a two-class classifier to a regression scheme with an infinite number of classes. Too simple classifiers could lead to trivial results and too complex classifiers could not be capable of classifying instances properly. Hence, there is a trade-off between usefulness and simplicity that needs to be evaluated. The best classifier will then be the most complex one giving reasonable results.

The problem definition is very flexible, as it can be tackled in many different ways. Several points have to be considered to define a strategy, and these points are the following:

1. **Dataset**. Open-source datasets are available in different trading platforms. The selected dataset should have at least a daily frequency and enough information about the price for each instance (e.g. Volume of exchange, Opening and Closing prices, etc.). All datasets for the different commodity futures should be extracted from the same source to have a common reference.

2. **Features** . The characteristics of the instances need to be properly extracted to add relevant information to the classifier. This process includes selecting the proper features from the original dataset. Some additional features are generated, such as technical indicators and temporal dynamics.

3. **Classification strategy**. The classification methods need to be properly selected to offer different alternatives for reaching the optimal result. Several techniques can be evaluated, including Decision Trees, Support Vector Machines or Neural Networks, among other options. Finally, ensembles combining the results from these methods can help refining the results.

All these points will be discussed with more detail on the following sub-sections of this chapter.

3.1. Dataset acquisition and structure

The chosen source for obtaining the dataset is *Yahoo! Finance* [1]. This web site contains a huge database of different financial markets, including a wide range of commodities. The integration with *Python* is very simple, as it provides with an API called *yfinance* that allows directly downloading the database and store it on a *pandas* dataframe. Hence, the chosen tickers are specified and the data is downloaded.

[1] https://finance.yahoo.com/commodities/

The data has a daily frequency, and depending on the commodity the date of the first register can change. The years between 2000 and 2008 contain certain periods with NA values, so 2008 has been chosen as the common starting point for all series. This way, all series of futures are analyzed under the same conditions. If needed, some imputation techniques could be applied to check if it is worthwhile increasing the size of the dataset at the expense of adding synthetic instances.

The dataset attributes are organized as follows. The index is the date corresponding to the market session, and the days in which the market is not open are not included. Hence, it is not necessary to look for weekends or holidays to eliminate or impute them, as the series is continuous. It must be remarked that the market closes on Friday and re-opens on Monday but the price does not change during the weekend, so this day hops do not affect the continuity of the series. The attributes of the dataset are the following:

- **Open & Close**. Opening and closing prices for the corresponding market session.
- **High & Low**. Highest and lowest values that the future price reaches over the course of a market session.
- **Adjusted Close**. This feature reflects the continuous value of the series considering possible variations due to actions taken by investors or providers.
- **Volume**. Number of future contracts that are traded along a session.

These attributes can be visualized together to get a first idea about the shape of the series. For that purpose, the *plotly* library can be used to create a candlestick chart where the prices and their variations along each day are shown, in conjunction with the volume. Figure 3.1. shows the candlestick chart for the coffee futures price:

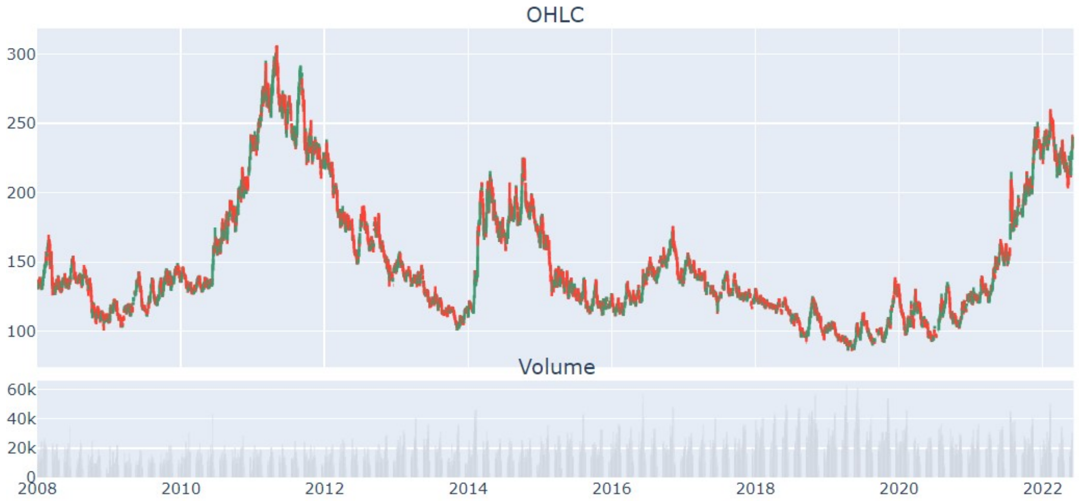


Fig. 3.1. Candlestick chart for Coffee price

The candlestick chart eases understanding the behavior of the series. In this particular case, the impact of some events can be spotted on the graph, such as the COVID-19 pandemic outbreak, that caused a continuous increase of the price from 2020 on. A similar but more significant pattern is observed in year 2010. This pattern can be appreciated generally in all the future contracts that will be included in this project, so it needs to be considered.

Initially, only the Adjusted Close is selected from the series, as it is the attribute containing the actual price of the future contract. The OHLC (Open High Low Close) attributes can be useful for daily predictions as they are directly related to the intra-day movements, but they may be confusing for longer periods of time, so they are initially discarded or used to obtain other indicators. The Volume is also initially discarded, in order to start with a uni-variate time series.

3.2. Feature extraction

The objective is to generate a dataset starting from the Adjusted Close. This series could be analyzed using a classic time series approach, but in this case this analysis needs to be adapted to a classification scheme. Hence, the temporal dynamics need to be captured from the series, so that they can be used as features for each sample. The target variable needs to be generated, too. As the goal is to determine the direction of the price along a period of time, a trend detector will be defined according to different strategies. These two procedures are described on the following sub-sections.

3.2.1. Predictors

As commented in the previous description, the dataset starts from the Adjusted Close price of the series. No additional attributes from the original source are considered in principle. This means that the employed features need to be synthetically generated. There are several ways to approach this problem, but only two of them will be considered: temporal dynamics and trading indicators.

Temporal dynamics refer to the procedure of extracting information from the series itself. This approach relates to a machine learning point of view, as no technical information is added, and the focus is set on the series as such. This approach could be considered as more generalist, as it can be applied to any kind of series following a similar structure to the ones considered on this project. The interpretability may also be easier, as the results have a temporal meaning and are easy to understand. Among all possibilities, two of them are employed: delayed percentage returns and moving averages.

The delayed percentage returns allow including information on a concrete sample about past time movements, so that the classifier can try to infer the future price direction based on the past trend directions. The advantage of these indicators is that they are very simple to generate, as only the percentage return for a period of time can be calculated and delayed for N time instants defined by the user. As an example, the 10-day returns could be obtained for the series, and delayed N time instants to capture the differences between day 0 and day 10, between day 10 and day 20, and so on. The trend can then be summarized along different time periods and with different granularities. These results can be even more summarized to simplify the problem, by assigning discrete values to the samples according to its magnitude.

Moving averages can be considered as a link between temporal dynamics and trading indicators, as they are frequently used in finance. They are similar to the delayed percentage returns, as they are able to condense the information of the current and past samples in a single value. Depending on the kind of moving average, all samples will be weighted equally (simple moving average, MA) or exponentially with respect to the last sample (exponential moving average, EMA). These functions can be combined to obtain different perspectives from the same series. Figure 3.2 shows an example for cotton with three different moving averages:

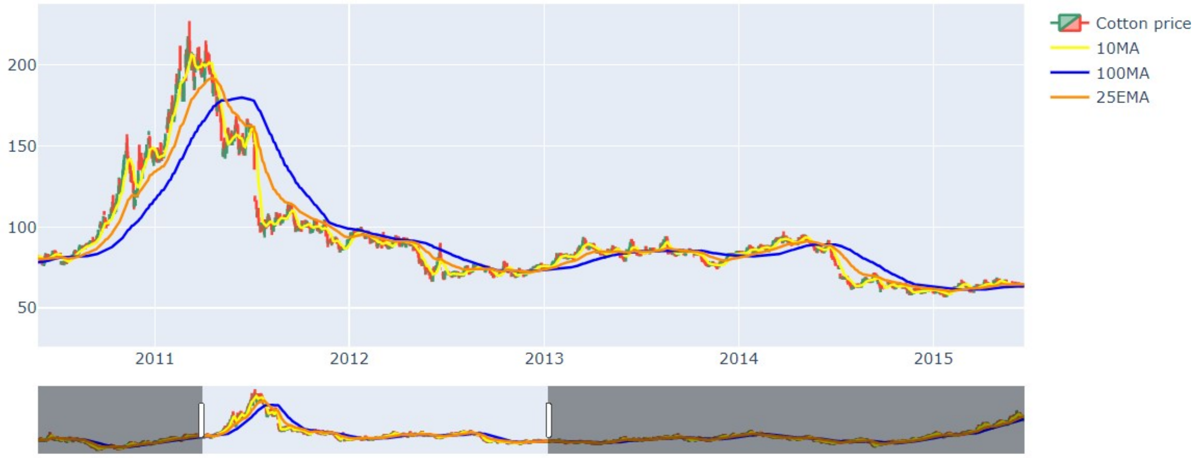


Fig. 3.2. Moving averages for Cotton price

The graph has been zoomed in to have a better understanding. Generally, moving averages can help simplifying the problem, as they smooth the series, eliminating the daily or weekly peaks. The shorter the period of average, the greater the similarity with the original series. It is appreciated on the graph how the 10-days MA almost follows the original signal while the 100-MA is a smoother series and has a delay due to the influence of the 100 previous samples. For the case of the EMA, the weighting is different and it can give alternative information.

The other group of features that can be included to capture temporal dynamics are the **trading indicators**. These indicators are mathematical functions generally used by experts, which can give relevant technical information about a financial instrument. The previously commented moving averages are a part of this group. Some other commonly used indicators can be calculated, such as [24]:

1. RSI (Relative Strength Index). RSI looks for defining momentum, which consists on identifying situations where a price may reach certain key points. It gives a per-

spective of when an instrument may be overbought or oversold. Its value oscillates between 0 and 100.

2. ADX (Average Directional Index). ADX indicates the strength of a trend, which eases identifying whether a trend is likely to continue or not. As for the RSI, its value oscillates between 0 and 100.

3. Bollinger Bands. This indicator defines the expected range of movement of the price, acting as a volatility indicator. In this case, the indicator is defined by the two lines indicating the expected upper and lower limits.

Indicators may be difficult to understand just by themselves. Traders use these calculations to generate specific signals when they reach a determined value [24]. RSI indicates oversell when it is lower than 20-30, while it indicates overbuy when it is greater than 70-80. ADX indicates that the trend is strong when it takes a value over 25. Bollinger Bands can indicate oversell or overbuy when the price is over or under the corresponding band. Then, these values can be simplified to ease their interpretation, discretizing their values and summarizing them. To ease the understanding of how this works, Figure 3.3 shows how these indicators would be with gold price:

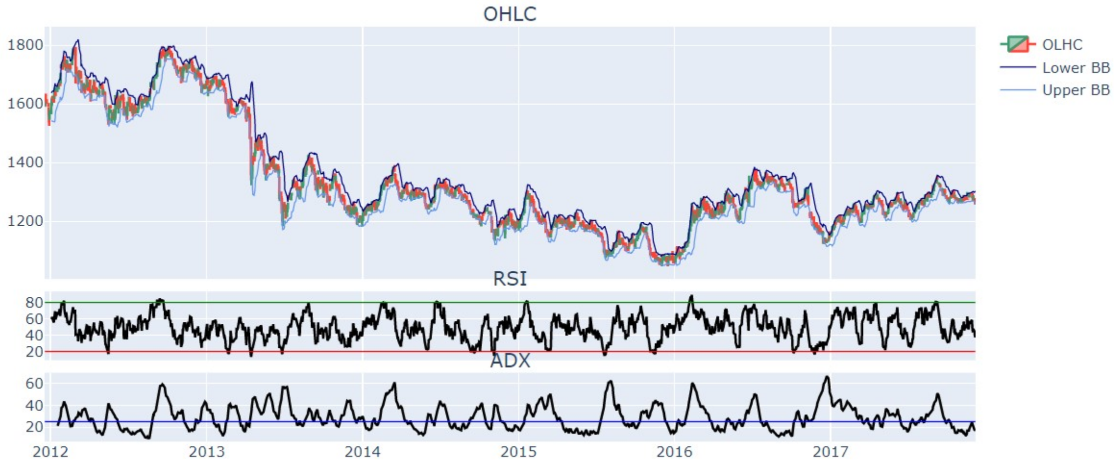


Fig. 3.3. Plot of trading indicators for Gold price

The previously defined limits have been plotted in conjunction with the indicators, all of them calculated for a 10-day window. The greater the window, the smoother the indicators will be. Then, the indicators could be calculated for different time windows to create some signals according to trading strategies. Finally, depending on the situation, different combinations between the temporal dynamics and the trading indicators can be assessed to achieve the best possible prediction of the target variable.

3.2.2. Target variable

Depending on the problem, the target variable can be either a continuous variable or a discrete variable. As the regression paradigm has been discarded, the target needs to be a discrete variable taking as many different values as the required granularity. The original dataset does not have a definition for the target, so it has to be manually generated. For that purpose, a trend detector needs to be implemented, so that it can automatically determine the trend present on the series according to a previously defined criteria.

The trend needs several parameters to be determined. The first one is the period of time between which the trend is detected (e.g. two weeks, one month, etc.). Then, some thresholds can be defined to define the labeling policy at the detector. Finally, a common metric for all samples should be defined to extract a single value for each sample and compare it against the thresholds for each class. In this way, the whole dataset can be labeled consistently.

The metric that will be used to compare all samples is the percentage return. This metric defines the difference in percentage between two samples, comparing their magnitudes. It can be either positive or negative, and therefore define whether the price is increasing or decreasing. The window plays a key role in this detection, as it can be very noisy for short periods of time or too similar to the original series for long periods of time. Figure 3.4 illustrates this difference for the coffee futures prices:

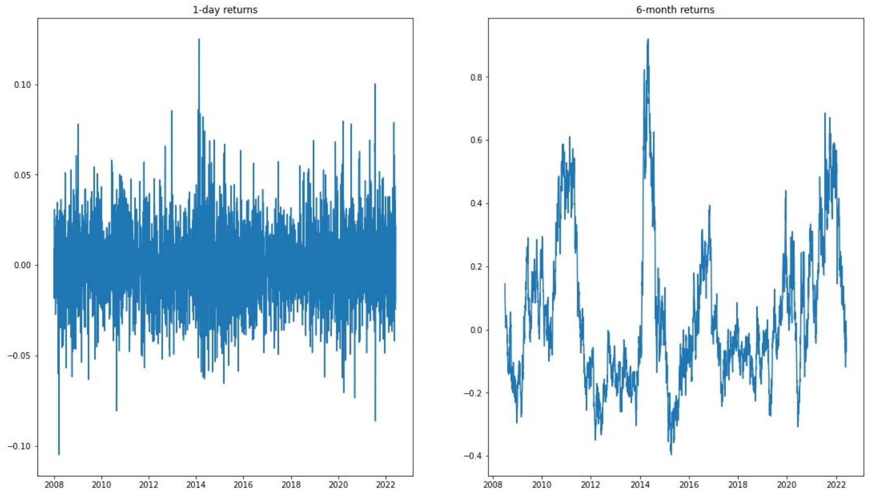


Fig. 3.4. Coffee returns with different time windows

The picture clearly shows that the wider the window, the less noisy the series is. Also, the range of variation is wider and longer trends can be identified. Considering this previous result, it is concluded that the window value has a direct impact on the final result and should be studied in detail. It must be remarked that due to the variability of the different series, the optimal window size for predicting the movement of the price can change depending on the considered commodity. Several window sizes can give different pieces of useful information.

The last parameter to be set is the threshold definition. This problem can be tackled in two different ways, depending on the perspective from which the problem will be approached. From the Machine Learning point of view, the optimal scenario has equiprobable classes to avoid the issues of dealing with imbalanced datasets. On the other hand, if a financial view is sought, the thresholds could be manually defined depending on the trading strategies. To have an initial picture of the situation, the monthly returns are calculated for all series, and the ranges of variation of the samples are included on the following table:

Commodity	Change between ±1%	Change between ±3%	Change between ±5%
Cotton	10.57%	32.37%	50.98%
Gold	18.39%	49.41%	72.03%
Heating oil	10.24%	28.21%	42.39%
Coffee	10.62%	29.53%	45.53%
Natural gas	6.30%	18.74%	32.84%
Sugar	10.16%	27.91%	43.41%

Table 3.1. PERCENTAGES OF VARIATION OF MONTHLY RETURNS DEPENDING ON THE COMMODITY

The results shown on Table 3.1 prove the aforementioned differences between the commodity futures series. As an example, for the case of cotton, ±3% would be a good threshold to obtain equiprobable classes in a three-class scheme, as usually the number of samples above and below the threshold are similarly distributed. For the other commodities, the threshold should be refined to obtain more balanced splits. The easiest way to determine the thresholds for equiprobable classes is to plot the histogram and select the corresponding quartiles according to the number of classes that is wanted. As an example, the histogram of returns is shown for the coffee and cotton series:

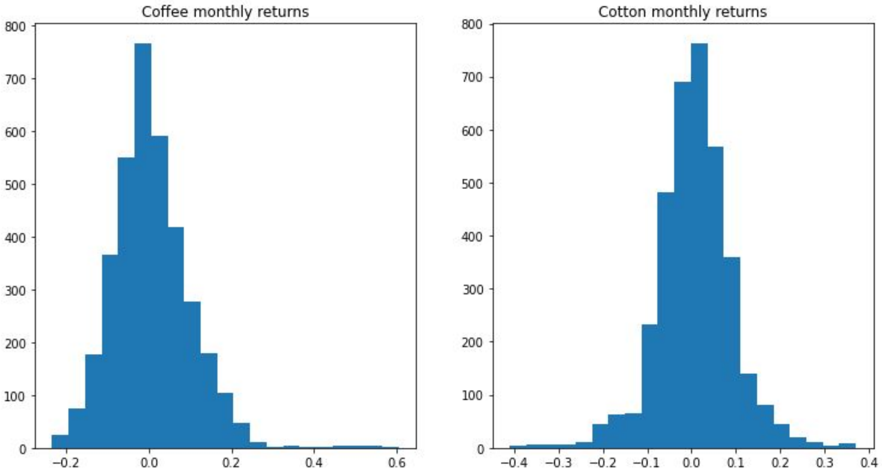


Fig. 3.5. Histogram of returns for Coffee and Cotton

As it has been commented above, the distribution of the returns is usually symmetric. This fact can be easily checked on the previous histograms. Hence, a balanced three-class problem can be defined, having the intermediate class on the center of the distribution and the increasing and decreasing classes on the tails.

The correct labeling of the dataset is a key task, as a wrong assignment of the classes would give a misleading result. To ensure that the labeling has been performed correctly, the fastest way to check it is to print the corresponding colors of the trend over the values of the commodity. Depending on the time window, the delay of the trend detector can be easier to detect. This result appears because the label of a sample corresponds to the value of the future price after a window of N days. A piece of code was developed to show this results. In this case, to keep up with the previously commented results of cotton, the following picture shows the labeling for a 10-day window and with a threshold of ±3% to split the data into three classes:

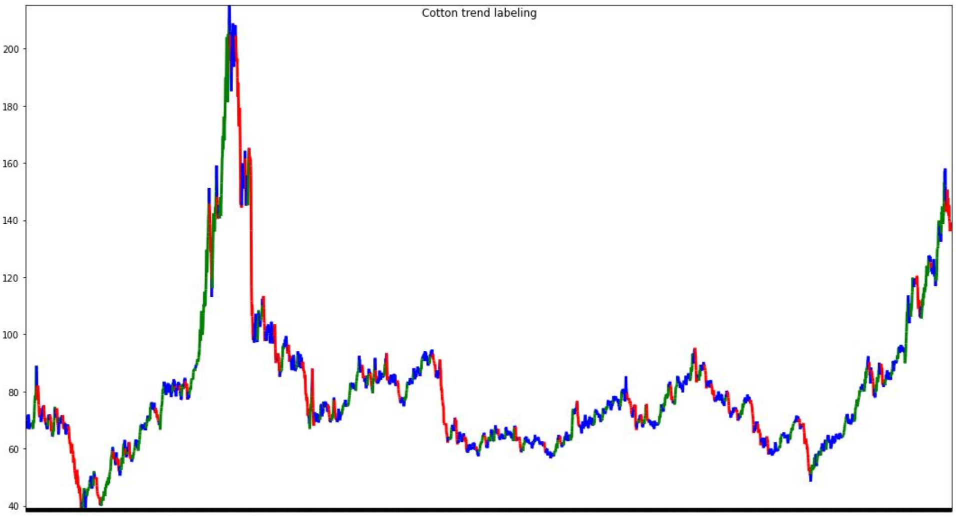


Fig. 3.6. Trend detection for Cotton

The green points are those samples where the price grew more than 3% in the 10-day period. The red points represent the opposite situation, and the blue points are those where the price stayed between ±3% after 10 days. The correct operation of the trend detector is now checked, so that the target variable is correctly generated, and ready to be used on further steps of the project.

3.3. Classification strategy

Different classification methods will be used to try to predict the future trends present on the data. The objective in this section is to present a wide variety of classifiers with different intrinsic characteristics, so that each one can give different insights about the final result. Depending on the result, they can be weighted and combined to obtain the optimal output as a combination of the outputs of all classifiers. The cross-validation strategy is also an important point, as the data come from a time series, so it can not be treated as a common classification dataset.

3.3.1. Classifiers and ensembles

The classification stage will be approached with three different classifiers. These three classifiers are Random Forest Classifier, Support Vector Classifier and Multi-Layer Perceptron Classifier. Each of them comes from a different family, with the objective of giving different perspectives about the same problem.

The series have a great variability and different characteristics, so it is expected not to obtain a global optimal classifier, but a particular classifier with a particular scheme for each series. Initially, all series will be tested under the same conditions to set a benchmark for the following improvements to be compared against. This benchmark will consist on classifying the series using as features only the time dynamics based on percentage returns. Then, the problem can be enriched with some buy/sell signals or trading indicators to see if the combination of features gives a better result.

The performance will be assessed through the accuracy and its standard deviation. The assessment process will also be subject to the confusion matrix given by each classifier. Up to this point, two strategies could be followed depending on the perspective: machine learning strategy or financial strategy. The characteristics of each one are the following:

- Machine Learning strategy. This strategy is more related to an academic point of view. Then, the objective is to be able to separate the classes properly. The confusion matrix is a useful tool to determine whether the classifier is distinguishing between classes or just focusing on achieving the best accuracy. Therefore, in cases of similar results for two classifiers, the one having a better balance between classes is preferred.
- Financial strategy. This strategy is related to business itself. The objective in this case is to maximize the incomes. Hence, it is not that important to distinguish between classes, but to achieve the highest possible number of correct responses. Then, it is reasonable to choose the majority class in cases of doubt.

The objective of this project is merely academic, so the first evaluation method will be used. The financial strategy can be left as a potential improvement for the future if the academic results show a good behavior of the models.

3.3.2. Cross-validation strategy

The cross-validation scheme is key to achieve a consistent result. The data has an important time-related component, so the train and test sets should be generated according to this characteristic. Then, data shuffling needs to be avoided to ensure that no information leakage is produced.

The K-fold strategy is not useful for this scheme, following the previous statements. There are two alternatives to perform the cross-validation. The first one is to roll forward and continuously increase the size of the train set, while keeping the test size constant. The second possible way consists on a sliding window strategy, where train and test set sizes keep constant as the window moves forward. The strategies are illustrated on Figure 3.7:

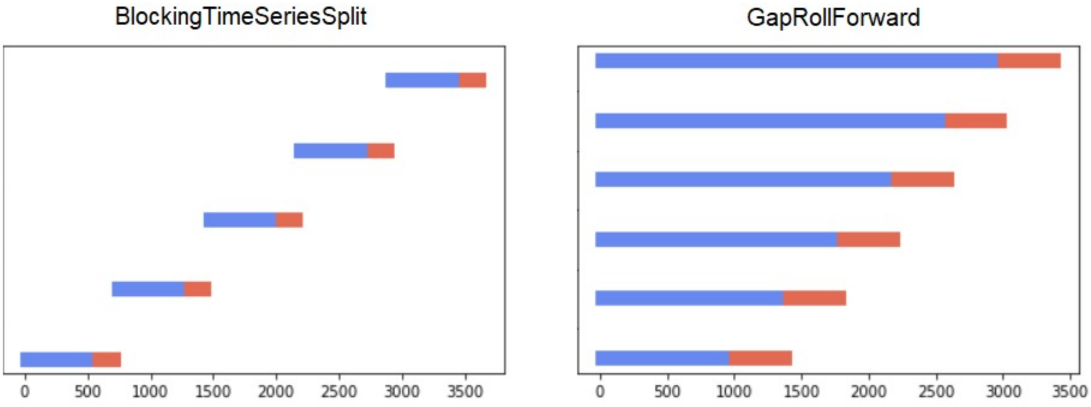


Fig. 3.7. Cross-validation schemes

Both strategies are equally feasible. They will be tested one against each other to see which one performs better. The main conclusion that can be extracted depending on which structure gives better results is if more distant samples are useful to predict current trends or not. Finally, it must be explicited that the *BlockingTimeSeriesSplit* [2] function has been obtained from a web site, and after checking that it works properly, it has been included in the project. Some modifications have been introduced to adapt it to the desired cross-validation structure of 80% train set and 20% test set, chosen by design.

3.4. Development and tools

This section gives an overview of the implementation process, describing the time sequence and the resources that have been used during the project development.

3.4.1. Development

Initially, an in-depth literature review was carried out, to become familiar with the topic and get a preliminary idea of the methods that are related to solve these problems. Although the focus has always been on machine learning techniques, it was necessary to acquire some knowledge about finance and commodities to understand how they work and their characteristics. During the development of the project, more literature was consulted, increasing the complexity as the project advanced to obtain further details about examples of implementations.

[2] https://hub.packtpub.com/cross-validation-strategies-for-time-series-forecasting-tutorial/

The code implementation has been the fundamental and the longest-lasting part of the project. The first step was to import and visualize the data on *Python*, to appreciate its structure and define the pre-processing strategy. Then, the first algorithms were applied to the data and the results were not good, as the performance was not able to improve a random guess. During this process, several approaches were tested, and although they did not work at first, they were useful for the following parts of the project. Two of these intermediate implementations were:

- Appliance of a Hidden Markov Model. The models were not working as expected, so a HMM was used to generate a series with low transition probability between states and the means of the states were initially separated by a long distance. This series simulated a series of returns, so that a long period of positive value could be interpreted as a long-lasting tend and vice versa. It worked well, so this was enough to check that the models were correctly implemented, but the conclusion was that maybe the problem was too complex. This issue was solved later.
- Appliance of classic forecasting scheme. This approach was used as a potential alternative to the classification scheme in case it did not work. Finally, it was discarded as a solution was found for the classification scheme.

Once the experiments were finished and after a new literature review, the final solution was designed. A preliminary experiment was made with the lagged returns in different periods of time, and it gave good results, so it was the chosen option. The code was then modularized and everything was summarized into two functions: one for generating the dataset according to the parameters introduced by the user and other for carrying out an automatic evaluation of the results. Then, the validation step was straightforward and the only task to do was to introduce the corresponding parameters, wait for the execution and finally analyze the results.

The graphic representation of the sequence is available on Appendix 1.

3.4.2. Tools

The development of the project has been entirely based on *Python*. The environment to develop it is *Jupyter Notebook*, as it eases the visualization of results in conjunction with the code. The *Rstudio* environment has also been employed during the development of the project for very concrete purposes as studying the correlations between series or perform time series forecasting, but these elements have not been finally included on the project.

In order to obtain the dataset, the *yfinance* library has been used, as commented before. All commodities are downloaded at the same time, so the output dataset comes in a multi-column dataframe format. To carry out pre-processing and transform the dataset into the desired format, *numpy* and *pandas* have been used. Some complementary libraries as *pandas_ta* have been useful to simplify the calculations of trading indicators. Once the data was ready to be used, *sklearn* library has been used for machine learning and cross-validation, unless for the particular functions described above. An extension of *sklearn* called *skforecast* was tried for carrying out classical time series forecasting, as described on the previous sub-section. The libraries that have been used to make the plots are *matplotlib* for simpler plots and *plotly* for more complex plots.

4. EVALUATION AND RESULTS

The experimental evaluation is presented in an incremental way. Initially, all series are classified under the same conditions, to set a common reference. Then, some potential improvements are introduced to see how this affects the results and if it offers some additional interpretability. The prices of commodity futures and financial instruments are in general difficult to predict as they are subject to many different factors, so the results need to be contextualized.

4.1. Benchmark

A first baseline result is set with four classifiers used with a basic hyper-parameter tuning. This scheme is the same for all commodities to test them under the same conditions. No feature selection is performed at this stage and the time horizon is also varied to assess the performance in different time windows. The employed cross-validation methods are those commented on section 3.3.2. The target has balanced classes, to ease the operation of the Machine Learning algorithms. Finally, the number of classes is equal to two and three, as it has been proved that the results are more inaccurate as the number of classes grow.

The classes are balanced, so accuracy is used as the evaluation metric. The standard deviation derivated from the cross-validation is also considered. Additionaly, the feature importances derived from Random Forest are retrieved to see the most influential variables over the final result. It must be remarked that only temporal-dynamics features are included in this first approach.

The accuracies obtained with **blocking cross-validation** are shown in Table 4.1 and Table 4.2. Standard deviations are later commented to ease reading.

	21-day forecasting				**10-day forecasting**			
Commodity	**RF**	**SVM**	**NN**	**Voting**	**RF**	**SVM**	**NN**	**Voting**
Coffee	0.66	0.625	0.645	0.62	0.56	0.515	0.45	0.585
Cotton	0.485	0.42	0.45	0.52	0.545	0.535	0.64	0.505
Sugar	0.47	0.49	0.48	0.48	0.445	0.465	0.485	0.475
Gold	0.505	0.495	0.62	0.51	0.415	0.49	0.405	0.475
Heating oil	0.51	0.60	0.485	0.555	0.55	0.655	0.615	0.655
Natural gas	0.38	0.365	0.32	0.36	0.475	0.535	0.42	0.4

Table 4.1. ACCURACIES FOR ALL CLASSIFICATION SCHEMES IN 2-CLASS CLASSIFICATION AND BLOCKING CROSS-VALIDATION

	21-day forecasting				10-day forecasting			
Commodity	**RF**	**SVM**	**NN**	**Voting**	**RF**	**SVM**	**NN**	**Voting**
Coffee	0.525	0.485	0.34	0.495	0.43	0.365	0.425	0.43
Cotton	0.32	0.195	0.37	0.295	0.365	0.395	0.365	0.355
Sugar	0.2	0.265	0.37	0.245	0.275	0.32	0.405	0.305
Gold	0.285	0.34	0.34	0.345	0.33	0.355	0.305	0.365
Heating oil	0.215	0.32	0.405	0.2	0.38	0.39	0.39	0.35
Natural gas	0.195	0.205	0.365	0.225	0.305	0.24	0.315	0.29

Table 4.2. ACURACIES FOR ALL CLASSIFICATION SCHEMES IN 3-CLASS CLASSIFICATION AND BLOCKING CROSS-VALIDATION

The test set has a size of 250 samples, approximately corresponding to the number of market days in a year. In terms of performance, the execution is computationally expensive, as the splits are automatically generated according to the number of samples in the test set. Hence, the train set grows significantly if the test set is small. Nevertheless, positive results are obtained for some of the series.

The most relevant series for this sub-section is coffee. Knowing that the classes are equiprobable, the thresholds to be surpassed are 50% accuracy on the two-class problem and 33% on the three-class problem to beat a random-guessing classifier. For coffee series, both thresholds are beaten, and in general Random Forest seems to be the best classifier for the series. As commented before, feature importances can be retrieved from this classifier. Figure 4.1. shows these importances for the three-class classifier predicting 21 days ahead:

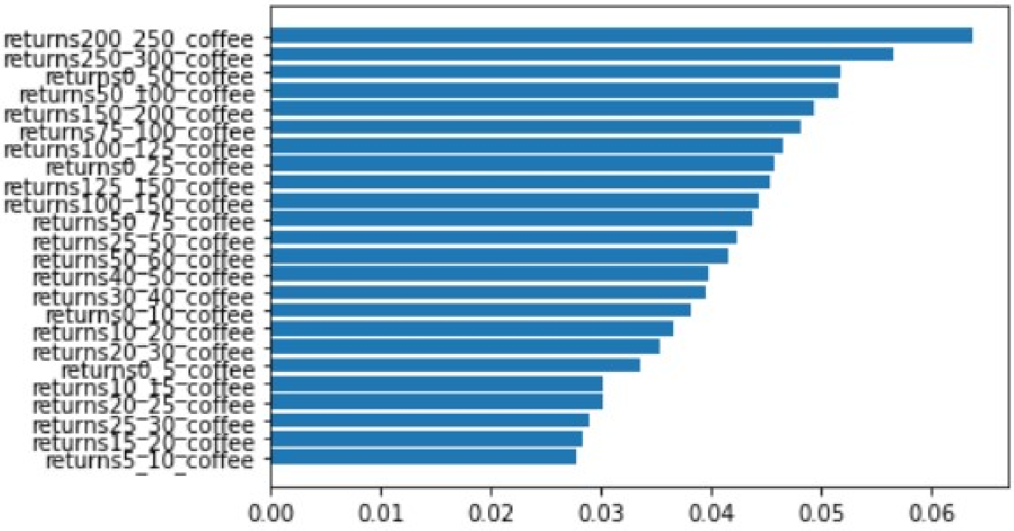


Fig. 4.1. Feature importances for three-class Random Forest with blocking cross-validation

The most relevant returns are generally the 50-day returns. It is appreciated how the variables with highest importance almost cover the range between day 0 and 300 days before with steps of 50 days. This result states that wider return periods are useful to predict coffee trend in comparison to shorter return periods, which occupy the last positions in terms of feature importance for Random Forest. It must be remarked that the two-class classifier for 21-day forecasting with 0.66 not only overcomes the threshold, but achieves the aforementioned condition of being able to separate classes. Hence, its confusion matrix has greater numbers on the diagonal than in the non-diagonal positions.

Another relevant result to be commented is the one relating to heating oil. In this case, the classifiers work well for the two-class problem, particularly the SVM classifier. The opposite happens for the three-class problem, where only one result of the NN classifier is successful. Then, SVM can be configured in further sections to improve performance for heating oil series.

Voting classifier does not give good results, as all classifiers take part on the voting process independently of their positive or negative results. Hence, it depends on the coincidences between classifiers. Nevertheless, it is useful in some cases to balance the predictions and separate classes properly, which may not give the best result but it is positively valued, as classes are better separated.

The last point to be commented is the good performance of the Multi-Layer Perceptron when predicting the three-class labels. In many cases, specially for 21-days predictions, it is the only classifier capable of beating the benchmark. By observing the confusion matrices of the different classifiers, it is appreciated that the most difficult class to identify is the middle class ('stay'), and in most of the cases NN is the classifier that assigns most samples to this class. Then, this classifier is stated as a potential solution to be enhanced in further sections. Click here to see an example of a confusion matrix as the one described above.

The cross-validation scheme is now changed to compare both methods. The results derived from **sliding window cross-validation** are shown in Table 4.3 and Table 4.4:

	21-day forecasting				**10-day forecasting**			
Commodity	**RF**	**SVM**	**NN**	**Voting**	**RF**	**SVM**	**NN**	**Voting**
Coffee	0.561	0.579	0.647	0.583	0.581	0.558	0.603	0.621
Cotton	0.699	0.669	0.706	0.699	0.692	0.594	0.571	0.669
Sugar	0.496	0.496	0.503	0.496	0.474	0.473	0.518	0.474
Gold	0.511	0.593	0.511	0.571	0.639	0.556	0.564	0.601
Heating oil	0.533	0.549	0.330	0.473	0.443	0.699	0.345	0.368
Natural gas	0.781	0.691	0.676	0.736	0.639	0.691	0.609	0.646

Table 4.3. ACCURACIES FOR ALL CLASSIFICATION SCHEMES IN 2-CLASS CLASSIFICATION AND SLIDING WINDOW CROSS-VALIDATION

	21-day forecasting				10-day forecasting			
Commodity	**RF**	**SVM**	**NN**	**Voting**	**RF**	**SVM**	**NN**	**Voting**
Coffee	0.375	0.278	0.338	0.323	0.345	0.248	0.353	0.308
Cotton	0.556	0.526	0.616	0.578	0.330	0.308	0.308	0.323
Sugar	0.225	0.278	0.300	0.263	0.255	0.255	0.300	0.285
Gold	0.390	0.436	0.428	0.428	0.496	0.345	0.390	0.338
Heating oil	0.263	0.225	0.323	0.278	0.293	0.180	0.218	0.180
Natural gas	0.631	0.631	0.609	0.669	0.390	0.578	0.406	0.443

Table 4.4. ACCURACIES FOR ALL CLASSIFICATION SCHEMES IN3 -CLASS CLASSIFICATION AND SLIDING WINDOW CROSS-VALIDATION

The train-test proportionality that has been selected is 80%-20% for a total of five splits, as it was shown in Figure 3.7. The size of the test set changes then to approximately 140 samples, but the train set is also reduced. In terms of computational cost, this approach is much cheaper. All blocks have now the same number of samples, which is always smaller than in the train set of the previous section. The results are generally better in terms of accuracy in comparison to the previous case.

Natural gas achieves the best results in 2-class classification, reaching a maximum accuracy of 71.8% for 21-day prediction. As this result is achieved with a Random Forest classifier, feature importances can be studied and compared to the previous case. Figure 4.2 shows the result:

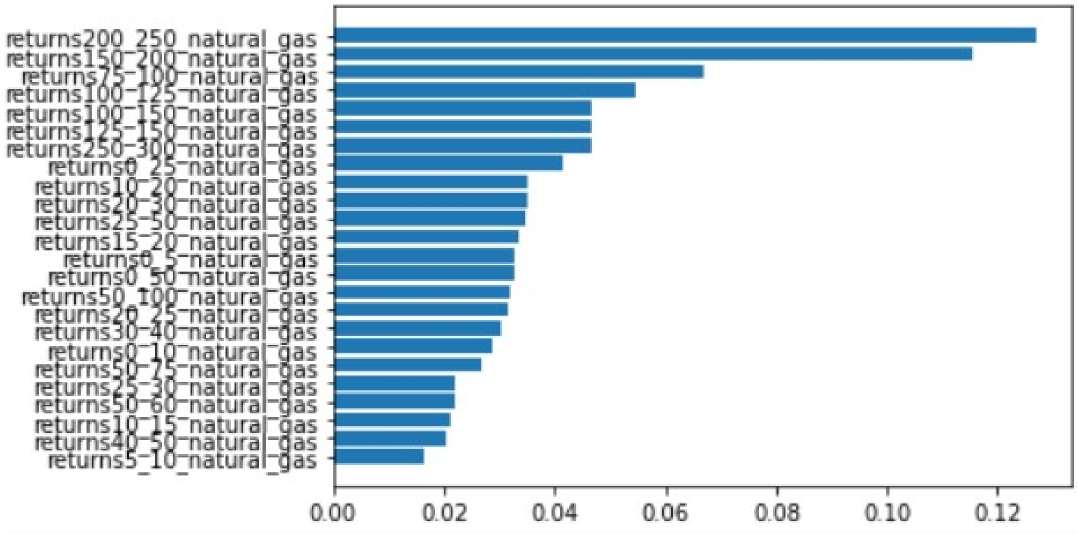


Fig. 4.2. Feature importances for two-class Random Forest with sliding-window cross-validation

The shape of the graph changes significantly with respect to Figure 4.1. Now, shifted 50-day returns are still the most useful variable, but clearly standing out from the rest. Nevertheless, the conclusion is similar to the one from the previous sub-section. It seems that trends for natural gas are long-lasting, and samples delayed half a year or more can be useful to determine whether the trend state will hold or not in comparison with short-term measurements. The main drawback of this classifier is appreciated on the confusion matrix. As commented before, separating classes is important from the machine learning point of view. For this particular scenario, there are more samples of 'up' than 'down' in the test set. The positive result is that all 'up' samples are correctly classified but 'down' samples only achieve 29% class prediction accuracy.

Natural gas is still the series with highest accuracy in the three-class problem. These results may lead to think that the best way to approach its classification is using a sliding window. Hence, it can be concluded the far past may not be really useful to predict the current trend, as eliminating it gives better results than when it is kept. The aforementioned problem of only focusing in one class is still present in the three-class scenario. The 'up' class is again properly detected, unlike for the case of 'down' and 'stay' classes.

Generally speaking, the two-class problem improves clearly with respect to the previous cross-validation method, having a result that outperforms the random-guessing classifier in all cases. This means that, in general, classifiers are not confusing classes. The problem that is detected in this case is that classifiers tend to predict a single class in some cases, but they hit the correct majority class on the test set. Voting classifier is still affected by the bad results of some of the other classifiers.

The three-class problem also improves the results for some of the commodities. In general, it performs much better for 21-day prediction than for 10-day prediction. This result may be due to the wider variation of the residuals as the window increases, which allows separating classes in an easier way. The problem with 10-day prediction may be that the boundary between classes is narrower than in the previous cases, and that makes it difficult to separate the samples into three groups.

Distinguishing between hard and soft commodities, some intuitions can be extracted. Broadly speaking, the Neural Network classifier achieves a better performance for soft commodities, while Support Vector Classifier does the same for hard commodities. Random Forest classifier does not stand out from the rest, achieving a similar score to the rest of classifiers when the results are tighter. Finally, voting classifier is not useful to refine the score in most of the cases, but it offers some good results in terms of class separation, which is observable through the confusion matrices.

The last item to comment are the standard deviations of accuracies, that have not been included on the tables to ease the document reading. Blocking cross-validation is a heavier method as it considers more samples at each training stage, so in most of the cases, the standard deviation of accuracy is around 5% and 10%. On the other hand, sliding-window cross-validation has more variability, as its values are around 10% and 20%, and eventually reaching higher values. This is due to the lower number of samples in each evaluation in comparison with the previous case. The final results are commented in deeper detail in section 4.3.

4.2. Further Experiments

The next step to be taken consists on adding new variables and assess their impact on the final result. Technical indicators are now introduced as features to the algorithm in conjunction with the temporal dynamics. The technical indicators that are included are RSI and ADX, together with the moving averages. The implicit feature selection present on Random Forest allows identifying the importance that is assigned to these new variables. Through comparing the accuracy with the baseline, it can be checked if they add useful information.

The first section of this block has also been useful to select the most adequate cross-validation method. The sliding window method is selected, as it has a lower computational cost and a better performance on the previous section. In this way, more experiments can be conducted and the complexity of the classifiers can also be incremented to deal with a higher number of variables at the expense of increasing slightly the computational cost. The remaining characteristics are left as they were in the previous section.

Table 4.5 shows the results for this approach on the two-class scheme:

	21-day forecasting				10-day forecasting			
Commodity	RF	SVM	NN	Voting	RF	SVM	NN	Voting
Coffee	0.609	0.496	0.578	0.571	0.518	0.458	0.518	0.496
Cotton	0.691	0.639	0.646	0.699	0.691	0.571	0.451	0.729
Sugar	0.601	0.563	0.496	0.563	0.548	0.473	0.451	0.473
Gold	0.458	0.496	0.511	0.518	0.541	0.481	0.601	0.526
Heating oil	0.398	0.255	0.248	0.248	0.268	0.305	0.373	0.298
Natural gas	0.631	0.736	0.616	0.736	0.611	0.694	0.664	0.686

Table 4.5. ACCURACIES FOR ALL CLASSIFICATION SCHEMES IN 2-CLASS CLASSIFICATION AND SLIDING WINDOW CROSS-VALIDATION FOR IMPROVED SCHEME

The results in comparison with the baseline are worse, generally speaking. Heating oil is the case where the results experience the most significant performance drop, as the classifier is worse than a random-guess classifier for all cases. This result leads to think that the classes are being confused against the other. The Multi-Layer Perceptron classifier, which previously had a great performance for soft commodities, is now less accurate for these series. It is then thought that feature selection prior to its appliance may be needed to enhance performance. In contrast, the voting classifier improves its average performance. Hence, classifiers seem not to be able to predict the samples properly on their own, but the majority vote is useful to face this issue.

Keeping with the comparison against the baseline, the same pattern is identified on both cases in terms of class separation. The classifiers usually focus on predicting a single class for all samples, leading to great results if the majority class is identified on the test set. It is observed on the confusion matrices that for some of the cases, all predictions belong to the same class. An example of this situation can be appreciated here. This issue is again solved through the voting classifier, but it is not achieved for all cases, and may imply an accuracy loss. The complexity has been increased in terms of number of variables and the structure of the classifiers, but classifiers still choose one class as their final answer in some cases. Hence, another option to be tested is to increase the size of the batches on the cross-validation, to check if having a great number of training samples helps improving generalization.

Table 4.6 shows the results for this approach on the three-class scheme:

	21-day forecasting				10-day forecasting			
Commodity	RF	SVM	NN	Voting	RF	SVM	NN	Voting
Coffee	0.368	0.330	0.323	0.330	0.353	0.398	0.270	0.375
Cotton	0.609	0.586	0.684	0.691	0.315	0.375	0.436	0.375
Sugar	0.390	0.315	0.360	0.398	0.285	0.255	0.240	0.270
Gold	0.428	0.443	0.375	0.428	0.406	0.315	0.421	0.451
Heating oil	0.263	0.165	0.157	0.142	0.231	0.231	0.216	0.231
Natural gas	0.624	0.721	0.676	0.766	0.410	0.574	0.507	0.567

Table 4.6. ACCURACIES FOR ALL CLASSIFICATION SCHEMES IN 3-CLASS CLASSIFICATION AND SLIDING WINDOW CROSS-VALIDATION FOR IMPROVED SCHEME

Comparing the results with the baseline, it is appreciated that the new accuracies change slightly, but there is no clear improvement. In some cases as natural gas, the performance is improved from 66.9% to achieve a 76.6% accuracy, which is a remarkable result for a three-class problem of this kind. The rest of the accuracies tend to grow too, but the growth is not significant considering that the dataset is now more complex. Then, a trade-off between accuracy and dataset complexity needs to be assessed, as it may not be worth-wile to add multiple variables just to achieve a performance increase around 1%-2%.

The execution time is now higher than in the previous case due to the increased complexity of the models, that was introduced to cope with the increase in the number of variables. The feature importance plots from Random Forest change the shape with respect to the previous section, as there are new variables that appear at the top. Hence, the problem is not that the new variables do not add information but rather that the selected group achieves a result that is similar to the previous case. Figure 4.3 on the following page shows feature importances for natural gas series.

The graph is similar to Figure 4.2, as the most important returns are still the long-term returns for a fifty-day period. It is appreciated how the moving averages with different time granularities are also important, but still the most important feature is the one encompassing a longer period of time. ADX seems to be more influential than RSI, but

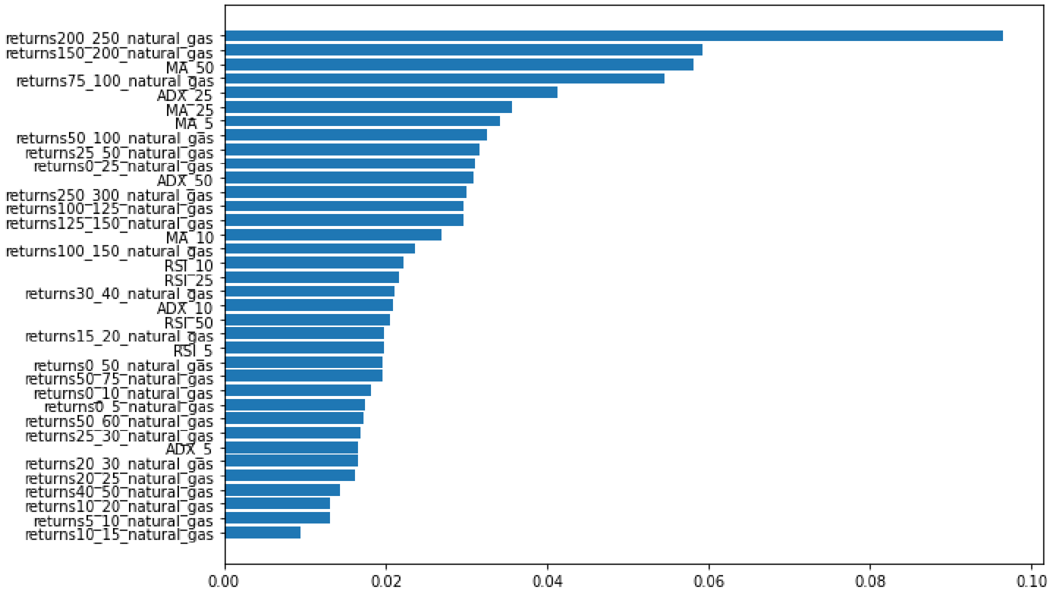


Fig. 4.3. Feature importances for three-class Random Forest with sliding-window cross-validation for improved scheme

neither of them have a great score unless for the case of 25-day ADX.

The main conclusion that can be extracted from this experiment is that technical indicators are more difficult to integrate in our classification scheme. These indicators are usually included by traders on their strategies according to technical criteria, instead of being assigned with arbitrary time windows. On the other hand, moving averages are easier to integrate with temporal dynamics, as their interpretation is more straightforward. It is appreciated how RSI and ADX are almost not considered by the algorithm while moving averages are given a similar importance to the long-term returns. This may be because the classifier is able to reconstruct the trend both with the values in percentage, corresponding to the returns, and the actual value prices, corresponding to the moving averages.

Despite the addition of new features to the dataset, the series that obtain the best results are the same as in the previous case. The 21-day forecasting is still more precise than the 10-day forecasting, too. It is concluded then that series as sugar or heating oil may need to be treated in an alternative way to try improving the results.

Lastly, the standard deviations of accuracies are briefly commented. In this case, due to the sliding window cross-validation usage, the standard deviations are around 10% and 25% in most of the cases, as in the previous section. This variation can give a false perspective of results, so it needs to be assessed when choosing the right model. A potential way to reduce the standard deviation would be to use three folds instead of five, as more samples are assigned to each data batch.

4.3. Final results

The results can be interpreted in many different ways. The previous sections show the tables containing all results for all cases, allowing for extracting certain conclusions about each method separately. The purpose of this final section is to summarize the results of all methods into a single table, to compare the performance of the different schemes. Both the two-class problem and the three-class problem are studied separately, and the best-performing scheme is included for each commodity and prediction period. Then, the machine learning algorithms, the cross-validation method and the dataset structure are compared. As this is the final section, which includes the results of both previous sections, the best accuracy metrics are presented with the standard deviations. This factor adds one more factor to be considered.

4.3.1. Two-class problem

Table 5.1 summarizes all the experiments for the two-class problem, including the best result for each particular problem:

Commodity	Period	Best accuracy	Algorithm	Cross-validation	Dataset
Coffee	21 days	0.660 ±0.068	RF	Blocked	Benchmark
Coffee	10 days	0.621 ±0.159	Voting	Sliding window	Benchmark
Cotton	21 days	0.706 ±0.237	MLP	Sliding window	Benchmark
Cotton	10 days	0.729 ±0.206	Voting	Sliding window	Combined
Sugar	21 days	0.601 ±0.176	RF	Sliding window	Combined
Sugar	10 days	0.548 ±0.308	RF	Sliding window	Combined
Gold	21 days	0.620 ±0.096	MLP	Blocked	Benchmark
Gold	10 days	0.639 ±0.171	RF	Sliding window	Benchmark
Heating oil	21 days	0.600 ±0.117	SVC	Blocked	Benchmark
Heating oil	10 days	0.699 ±0.256	SVC	Sliding window	Benchmark
Natural gas	21 days	0.781 ±0.174	RF	Sliding window	Benchmark
Natural gas	10 days	0.694 ±0.279	SVC	Sliding window	Combined

Table 4.7. BEST ACCURACY SCORE FOR EACH SERIES ON THE TWO-CLASS PROBLEM

Following with the initial statements, it is appreciated that in all scenarios the 50% accuracy has ben overcome at least once. Hence, as the classes have been defined as equiprobable, this result means that the method is adding extra value to the the trend classification problem.

Several aspects can be commented about Table 5.1:

- Period of forecasting. In chapter 4, it could be appreciated how performance was generally better for the 21-day problem, but in this table is appreciated that this does not happen when considering only the best results. In half of the cases, the performance is better for the 10-day problem.

- Cross-validation strategy. The sliding-window cross-validation strategy gives better results in comparison with blocked cross-validation. Nevertheless, the pattern of problems where blocked cross-validation is effective is easy to distinguish: 21-day problems. Then, it is concluded that for some series as coffee, gold and heating oil it may be worth-while not to eliminate past samples. In exchange, for cotton, sugar and natural gas it seems to be better not to consider far past samples to predict the future trend. For the 10-day problem, sliding window works better in all cases.

- Machine learning algorithms. Random Forest is the algorithm achieving the highest number of best performances, but there is not a clear algorithm outperforming the rest. There are some algorithms that fit well with particular series, as Random Forest to predict sugar price and Support Vector Classifier to predict the price of heating oil. Voting classifier is useful for enhancing global performance in particular cases. Nevertheless, the positive aspect of this classifier is that it helps balancing classes, as stated in previous chapters.

- Dataset features. Temporal dynamics alone are able to obtain good results, beating the dataset that combines them with technical indicators. Combined dataset is only useful for predicting sugar in both problems, but then in most of the problems it offers a worse performance than temporal dynamics.

The criteria that has been used to decide which is the best option is to select the model with highest mean accuracy. Nevertheless, the standard deviation needs to be accounted too, as it is a key factor to determine how precise is that accuracy measurement. As an example, for 21-day coffee prediction, a 66% ± 6.8% accuracy is achieved, so it can be guaranteed that in almost all cases it will be above 50%. In exchange, for 10-day heating oil it is observed that that it can vary within a range of 25.6%. Hence, this result is more uncertain even though the mean accuracy is higher. These results need to be considered when using a model to determine its reliability.

The initial criteria to determine the maximum number of classes to be used was based on the trade-off between simplicity and accuracy. For the two-class problem, all classifiers have an acceptable performance and the problem can become more complicated by adding more classes. Hence, the three-class problem is the step to be tested, to see if algorithms are able to deal with a more complex scenario.

4.3.2. Three-class problem

Table 5.2 shows the best-performing methods for each of the problems:

Commodity	Period	Best acuracy	Algorithm	Cross-validation	Dataset
Coffee	21 days	0.525 ±0.074	RF	Blocked	Benchmark
Coffee	10 days	0.430 ±0.061	RF/Voting	Blocked	Benchmark
Cotton	21 days	0.691 ±0.112	Voting	Sliding window	Combined
Cotton	10 days	0.436 ±0.206	MLP	Sliding window	Combined
Sugar	21 days	0.398 ±0.217	Voting	Sliding window	Combined
Sugar	10 days	0.405 ±0.068	MLP	Blocked	Benchmark
Gold	21 days	0.443 ±0.161	SVC	Sliding window	Combined
Gold	10 days	0.496 ±0.130	RF	Sliding window	Benchmark
Heating oil	21 days	0.405 ±0.086	MLP	Blocked	Benchmark
Heating oil	10 days	0.390 ±0.077	MLP	Blocked	Benchmark
Natural gas	21 days	0.766 ±0.236	Voting	Sliding window	Combined
Natural gas	10 days	0.578 ±0.126	SVC	Sliding window	Benchmark

Table 4.8. BEST ACCURACY SCORE FOR EACH SERIES ON THE THREE-CLASS PROBLEM

As in the two-class problem, there is a solution for every problem overcoming the 33% expected success rate from a random-guessing classifier. The structure of the table changes with respect to the two-class problem in the following aspects:

- Period of forecasting. The 21-day period generally obtains better results in this case, but still it depends on the series that is under consideration.

- Cross-validation strategy. The blocking cross-validation is now more important, as it is associated to better results. Coffee and heating oil, where the blocking strategy already worked in the two-class problem, both still have blocking as the best method. This result may be obtained because the problem is now more complex, as the distinction is made among more classes, and it is useful to keep information from more distant past. Nevertheless, sliding window is the chosen option for a higher number of methods.

- Machine learning algorithms. The voting classifier is more important in this approach. This may be associated to the increasing complexity, where classifiers are not able to separate the three classes, but majority vote is able to select the best options. It is remarkable that none of the series has the same algorithm as the best-performing for both prediction periods.

- Dataset features. The combined dataset is more important in this case. Following in the line of the previous sub-sections, the problem is now more complex, so adding new attributes can help refining predictions.

Focusing on the standard deviation of the mean accuracy, several aspects can be commented. The case of 21-days natural gas has a standard deviation of 23.6%, which is really big, but as the mean accuracy is 76.6%, the result is still positive as it is clearly above the 33% random-guess benchmark. Generally, the uncertainty is higher when using the combined dataset, both for the two-class and the three-class problems. In case of doubt, it may be less risky to choose the temporal dynamics to control the potential deviations. Therefore, a potential improvement for future works could be related to focus on reducing the standard deviation of the models.

As in the two-class problem, all classifiers overcome random guessing and add new insights about the trend prediction, so it is concluded that the result of the three-class problem is also satisfactory.

5. CONCLUSIONS

5.1. Results

The outcome of the project has been satisfactory. The initial goal has been achieved, as the classification scheme has been successfully implemented. An alternative method to the common procedures found on literature has been tested, being able to outperform the random-guessing classifiers. These random-guessing classifiers have been defined as the new benchmark instead of the classic benchmarks, due to the change from a regression problem to a classification problem. The functions that have been defined for each task, as trend labeling or automated algorithm appliance, work well for all cases and can be easily modified to reshape the problem if necessary.

The process of development has also been useful for me personally to learn many different concepts about finances and machine learning. Despite the fact that initially results were not as good as expected, the first approaches were useful to understand how the series behave. In this way, by knowing which approach works and which does not, new information has been extracted for further implementations. The addition of technical indicators has also been useful both for giving a new point of view and for deepening a little bit my knowledge in finance.

Although the explored models have been able to improve random guesses, these time series change in a fast way. As commented before, financial time series generally have non-linear and non-stationary characteristics, and are influenced by many different factors that can not be always identified. Hence, the algorithms need to be re-trained periodically to analyze how performance evolves with time and if the model requires for new approaches. These results should then only be considered for the present time of June 2022, as they will surely change as time passes.

The project has established a basis for a system that can be improved in different ways. This approach had an academic focus, as the main goal was to separate the classes, which has been achieved for some of the cases. Research papers are more focused on technical and fundamental analysis for financial time series, so there are some alternative possibilities to be explored relating to time dynamics. The percentage returns can be handled in many different ways, and so happens with the simple moving averages and exponential moving averages. Consequently, these features relating to time dynamics could also be useful for approaching the problem from a business point of view and some strategies could be created based on them.

The biggest difficulty that has been faced is to handle the great amount of information that is available, and condense it into a single implementation. Facing an open problem has been a challenge, but it has been a learning experience for the future. All the effort has paid off, and it has been a rewarding experience.

5.2. Future work

A solid basis has been established, but there is still room for improvement. The problem is very flexible, and it has been proved that the addition of exogenous predictors can be useful in some cases. Certain research papers cited on Chapter 2 mention the possibility of adding information to the features from various sources as the news. Many sources of additional information can be tested, but two of them are described below:

- Inter-relationships among series. The prices of commodities are influenced by multiple factors. In order to produce a certain agricultural commodity, certain energetic resources are needed, so the price of energetic commodities has a direct impact on their price. A high price on oil can have an action-response effect on agricultural commodity prices, and hence on the food market [25]. Therefore, it may be useful to monitor oil price to improve future predictions of other commodities. Some other relationships can also be studied.
- Commodity spot prices forecasting. As mentioned in [17], the future prices can be a good predictor for the spot prices. Consequently, a precise forecasting of commodity spot prices can give an idea of how future prices will fluctuate. Futures market is directly influenced by the sentiment of the investors, as the prices may rise or fall depending on the predictions of the spot price. A proper combination of these factors may lead to an improvement on the predictions.

Artificial intelligence is constantly evolving, so new technologies will be developed on the future that may allow to improve the results. The series will still change and may follow new patterns, so this challenge will remain a major focus of researchers for a long time to come. Machine learning and deep learning have shown a new potential way to proceed, but there is still a world of possibilities to explore.

BIBLIOGRAPHY

[1] Anonymous, “Oecd business and finance outlook 2021,” *OECD Business and Finance Outlook*, 2021. DOI: 10.1787/ba682899-en.

[2] V. Derbentsev, A. Matviychuk, N. Datsenko, V. Bezkorovainyi, and A. Azaryan, “Machine learning approaches for financial time series forecasting,” 2020. DOI: 10.31812/123456789/4478.

[3] B. Alhnaity and M. Abbod, “A new hybrid financial time series prediction model,” *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103 873, 2020. DOI: 10.1016/j.engappai.2020.103873.

[4] C. Bousono-Calzon, J. Bustarviejo-Munoz, P. Aceituno-Aceituno, and J. J. Escudero-Garzas, “On the economic significance of stock market prediction and the no free lunch theorem,” *IEEE Access*, vol. 7, pp. 75 177–75 188, 2019. DOI: 10.1109/access.2019.2921092.

[5] G. V. Attigeri, M. P. M. M, R. M. Pai, and A. Nayak, “Stock market prediction: A big data approach,” *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015. DOI: 10.1109/tencon.2015.7373006.

[6] I. K. Nti, A. F. Adekoya, and B. A. Weyori, “A systematic review of fundamental and technical analysis of stock market predictions,” *Artificial Intelligence Review*, vol. 53, no. 4, pp. 3007–3057, 2019. DOI: 10.1007/s10462-019-09754-z.

[7] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning : A systematic literature review: 2005–2019,” *Applied Soft Computing*, vol. 90, p. 106 181, 2020. DOI: 10.1016/j.asoc.2020.106181.

[8] A. Bahrammirzaee, “A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems,” *Neural Computing and Applications*, vol. 19, no. 8, pp. 1165–1195, 2010. DOI: 10.1007/s00521-010-0362-z.

[9] D. A.K. and B. V.K., “Financial time series forecasting: Comparison of neural networks and arch models,” *EuroJournals*, no. 49, 2010.

[10] M. E. Pérez-Pons, J. Parra-Dominguez, S. Omatu, E. Herrera-Viedma, and J. M. Corchado, “Machine learning and traditional econometric models: A systematic mapping study,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 12, no. 2, pp. 79–100, 2021. DOI: 10.2478/jaiscr-2022-0006.

[11] K. Amadeo, *An introduction to the financial markets*, Jan. 2022. [Online]. Available: https://www.thebalance.com/an-introduction-to-the-financial-markets-3306233.

[12] A. Hayes, *Commodity market*, Feb. 2022. [Online]. Available: https://www.investopedia.com/terms/c/commodity-market.asp.

[13] Anonymous, *Hard vs soft commodities*, Sep. 2021. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/hard-vs-soft-commodities/.

[14] F. J. Fabozzi, F. Roland, and D. G. Kaiser, "30," in *The Handbook of Commodity Investing*. Wiley, 2008, pp. 679–711.

[15] C. F. T. Commision, *Basics of futures trading*. [Online]. Available: https://www.cftc.gov/LearnAndProtect/AdvisoriesAndArticles/FuturesMarketBasics/index.htm.

[16] E. Antwi, E. N. Gyamfi, K. A. Kyei, R. Gill, and A. M. Adam, "Modeling and forecasting commodity futures prices: Decomposition approach," *IEEE Access*, vol. 10, pp. 27 484–27 503, 2022. DOI: 10.1109/access.2022.3152694.

[17] M. Kwas and M. Rubaszek, "Forecasting commodity prices: Looking for a benchmark," *Forecasting*, vol. 3, no. 2, pp. 447–459, 2021. DOI: 10.3390/forecast3020027.

[18] D. Wu, X. Wang, J. Su, B. Tang, and S. Wu, "A labeling method for financial time series prediction based on trends," *Entropy*, vol. 22, no. 10, p. 1162, 2020. DOI: 10.3390/e22101162.

[19] V. Vo, J. Luo, and B. Vo, "Time series trend analysis based on k-means and support vector machine," *Computing and Informatics*, vol. 35, pp. 111–127, 2016.

[20] P. Ładyżyński, K. Żbikowski, and P. Grzegorzewski, "Stock trading with random forests, trend detection tests and force index volume indicators," *Artificial Intelligence and Soft Computing*, pp. 441–452, 2013. DOI: 10.1007/978-3-642-38610-7_41.

[21] K. Prachyachuwong and P. Vateekul, "Stock trend prediction using deep learning approach on technical indicator and industrial specific information," *Information*, vol. 12, no. 6, p. 250, 2021. DOI: 10.3390/info12060250.

[22] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock-market prediction," *Journal of Big Data*, vol. 7, no. 1, 2020. DOI: 10.1186/s40537-020-00299-5.

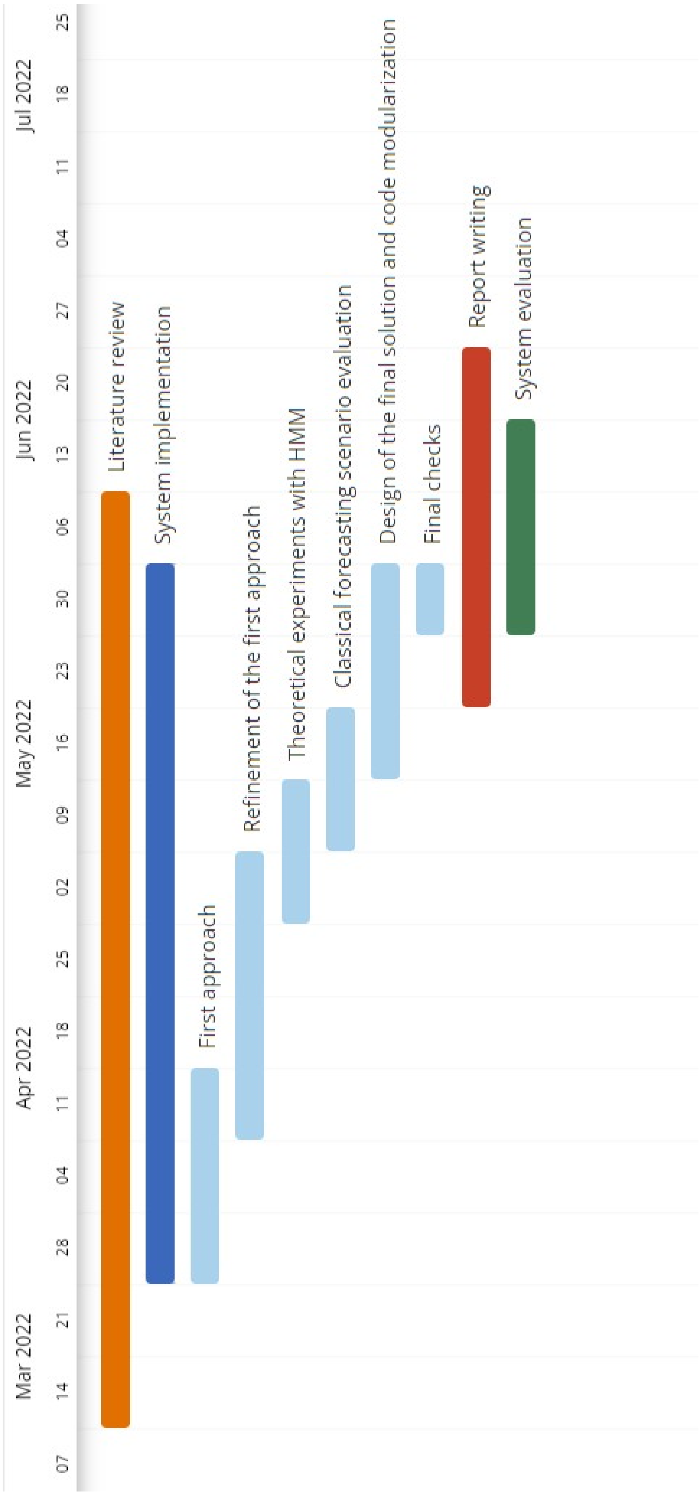
[23] B. Areej Abdullah and F. Mohamed Waleed, "Forex trend classification using machine learning techniques," *Proceedings of the 11th WSEAS international conference on Applied computer science*, vol. 1, pp. 41–47, Oct. 2011.

[24] Anonymous, *10 trading indicators every trader should know*. [Online]. Available: https://www.ig.com/en/trading-strategies/10-trading-indicators-every-trader-should-know-190604#relativestrengthindex.

[25] A. Kapusuzoglu and M. Karacaer Ulusoy, “The interactions between agricultural commodity and oil prices: An empirical analysis,” *Agricultural Economics (Zemědělská ekonomika)*, vol. 61, no. No. 9, pp. 410–421, 2016. DOI: 10.17221/231/2014-agricecon.

APPENDIX I: GANTT CHART

The following diagram shows the timeline describing the stages of the project.



Click here to go back to Section 3.4.1.

APPENDIX II: CONFUSION MATRICES

The commented confusion matrices are included in this appendix. Some examples will be included for each section, to illustrate the facts commented on the report.

Three-class blocking cross-validation

The matrices are shown for the case of coffee and a 10-day ahead prediction. The commented problem is easy to appreciate, as Random Forest an SVC tend to focus on separating 'up' and 'down' classes, while NN seems to be capable to identify the middle class, but still is not able to separate it from the rest. The positive aspects of Voting Classifier are appreciated here too, as it enhances the prediction of 'up' and 'down' classes. The class 'stay' is more difficult to predict as the two first classifiers have a low number of predictions of this class, and the voting classifier uses hard decision.

Random Forest	SVC	Neural Network	Voting
$\begin{bmatrix}19 & 3 & 17\\37 & 5 & 19\\34 & 4 & 62\end{bmatrix}$	$\begin{bmatrix}26 & 8 & 5\\33 & 8 & 20\\52 & 9 & 39\end{bmatrix}$	$\begin{bmatrix}13 & 13 & 13\\17 & 20 & 24\\21 & 27 & 52\end{bmatrix}$	$\begin{bmatrix}24 & 4 & 11\\41 & 3 & 17\\37 & 4 & 59\end{bmatrix}$

Click here to go back to section 4.1.

Two-class sliding-window cross-validation

The matrices are shown for the case of cotton with a 21-day ahead prediction. The aforementioned problem is easy to appreciate, as classifiers focus mostly on predicting the majority class. Voting classifier is able to balance the classes in a positive way, but the 'down' class still has a low accuracy, as SVC and NN almost do not consider it. This is a common situation for the sliding-window cross-validation scheme, but as far as the classifiers identify the majority class, the overall accuracy is not affected.

Random Forest	SVC	Neural Network	Voting
$\begin{bmatrix}11 & 37\\4 & 81\end{bmatrix}$	$\begin{bmatrix}3 & 45\\3 & 82\end{bmatrix}$	$\begin{bmatrix}7 & 42\\1 & 84\end{bmatrix}$	$\begin{bmatrix}10 & 38\\2 & 83\end{bmatrix}$

Click here to go back to section 4.2.