Research Project (PRe)

Field of study : Quantitative Finance
School year : 2022-2023

Meta labeling for portfoilio optimisation

Author : Benjamin GERINO (Promotion 2024)

Host organisation : University of Toronto
Department : RiskLab
Address : 40 St. George St,Suite 7256,Toronto On M5S2E4

Supervisors : Pr. Luis Seco (UofT), Pr. Fransesco RUSSO (ENSTA)

This document is not confidential.

Privacy Statement

This document is not confidential. It can therefore be communicated externally in paper format but also distributed in electronic format.

Acknowledgement

First, I want to thank Professor Seco for having me during this internship and for proposing me this very interesting subject for the internship. I also want to thank all the RiskLab department staff for being very welcoming and for handling all the administrative work, which allowed me to work there. Moreover, I am grateful to them for providing me with the opportunity to attend several conferences during this internship. I also want to extend my special thanks to Professor Hamid for his invaluable assistance in this research subject and for offering wise advice that greatly contributed to my learning and progress during this internship.

Finally, I want to express my gratitude to ENSTA Paris for enabling me to undertake this exceptionally rewarding internship and to Porfessor Russo for being my referring teacher.

Abstract

Meta-labeling is an innovative technique that combines machine learning and financial expertise to improve the performance of algorithmic trading systems. This paper explores the concept of meta-labeling, its applications in finance and algorithmic trading, and its potential to enhance the accuracy and efficiency of trading models. We discuss the primary benefits of meta-labeling, including trade signal filtering, risk management, model validation, execution optimization, and adaptive trading strategies. By incorporating a secondary predictive layer, meta-labeling provides traders and portfolio managers with a competitive edge in today's fast-paced, data-driven financial markets. Our analysis demonstrates the significant potential of meta-labeling in revolutionizing the landscape of algorithmic trading and finance.

Keywords

Meta-labeling, Algorithmic Trading, Risk Management, Portfolio Optimization, Adaptive Trading Strategies.

Table des matières

I Introduction

I.1 Context

This paper presents my work during my internship at the University of Toronto, supervised by Pr. Luis Seco (director of the risklab). The objective of this internship was to implement the meta-labeling method for portfolio optimisation. This project was inspired by the book of Marcos Lopez de Prado called Advances in Financial Machine Learning. This book presents all major issues in financial machine learning, and solutions are offered to carry these issues. This project at the risklab department was new, so I had to start from scratch and do all the research to understand the method. A chapter of this book is dedicated to the labeling of data. This is where Meta-Labeling is explained. In this chapter, meta-labeling is loosely explained, giving the main idea of its functioning and the benefits of this technique. Nevertheless, I needed more explanation to understand the method. But, this topic is very little discussed in the research area, so I found almost no other papers on this subject than the book. This made my task way more difficult and made me spend a lot of time on doing research online to understand what meta-labeling was about. The main explanations I found were made by the group Hudson&Thames on youtube or the internet.

I.2 Organisation

Even though I had to do all the research and the work on my own (since the project was new), I was helped by Pr. Hamid Harian (assistant professor at Sharif University of Technology). Therefore, he was there to help me understand the concept, clarify some points and to give some feedbacks on my algorithms.

I.3 Structure of the report

My report contains three main parts about my research internship project :

- The first one , Portfolio Optimisation, explains the concept of portfolio optimization, portfolio management, and the main indicators in portfolio
- The second part, Existing strategies, presents different strategies that are already developped and known. Some of them are used to be compared with the meta-labeling method.
- The last part, Meta Labeling, explaines the methodology of this technique, the different steps and shows the results.

II Portfolio Optimization

Portfolio optimization is the process of selecting the best portfolio (asset distribution), out of the set of all portfolios being considered, according to some objective. The objective typically maximizes factors such as expected return, and minimizes costs like financial risk

II.1 Portfolio Management

Portfolio optimization is used in portfolio management. Portfolio management consists in investing a certain amount of money in different assets. The objective is to maxime the risk-return-trade-off. This means to give the best allocation in those assets to get the best return for a given risk, usely given by the investor's goal.
We can seperate portfolio management in two categories :

- Discretionnary portfolio management : portfolio managers make investment decisions that do not follow a particular theory or rationale. They rely on their judments or intuitions.

- Quantitative portfolio managers : uses mathematical models and statistical techniques to make investment decisions. It relies on analysing large amounts of data to identify patterns, trends, and opportunities. This is the type of investments strategies we will focus on.

II.2 Markowitz Theory

Markowitz founded the portfolio optimization theory. It assumes that an investor wants to maximize a portfolio's expected return contingent on any given amount of risk.

There were several assumptions originally made by Markowitz. The main ones are the following :

1. the risk of the portfolio is based on its volatility (and covariance) of returns.
2. analysis is based on a single-period model of investment.
3. an investor is rational, averse to risk and prefers to increase consumption. Therefore, the utility function is concave and increasing.
4. an investor either minimizes their risk for a given return or maximizes their portfolio return for a given level of risk.

Many trading strategies are based on Markowitz theory.

II.3 How to evaluate a strategy

To evaluate a strategy, there are several features of the results to take into account. These are used to compare strategies.

II.3.1 Return

A portfolio return is a reference to how much an investment portfolio gains or loses in a given period of time.
Therefore, the portfolio return between time t and t+1 is $R_{t+1} = \frac{V_{t+1}-V_t}{V_t}$ where V_t represents the value of our portfolio at time t.
Let's say that our portfolio is made of n assets. We note $r_{t+1,i}$ the return of the asset i between

t and t+1 : $r_{t+1,i} = \frac{p_{t+1,i}}{p_{t,i}} - 1$.
We also note $w_{t,i}$ the weight of the asset i in the portfolio at time t. We can therefore write the portfolio return as :
$R_{t+1} = \sum_{i=0}^{n} w_i r_{t+1,i}$

II.3.2 Risk

Portfolio risk is used to describe the potential loss of value or decline in the performance of an investment portfolio due to various factors, including market volatility, credit defaults, interest rate changes, and currency fluctuations.
Within the portfolio, there are different investment stocks that the investor holds, and all of them individually have different risk assessments.
However, when they all combine in a portfolio, the risk is diversified and then different for a portfolio of the existing different stocks.
Portfolio Risk is measured by calculating the standard deviation of the portfolio. In this regard, standard deviation alone cannot calculate the portfolio risk. There is a need to ensure that all the different standard deviations are accounted for with their weights and the existing covariance and correlation between the existing assets. In this regard, covariance can be defined as the extent to which stocks move in the same direction. Usely assets from a same sector, for example automobile, are correlated.

II.3.3 indicators to mesure risk

Here are the different indicators :

Standard deviation : Standard deviation is a measure of the dispersion of returns around the mean of a portfolio. It is a widely used measure of portfolio risk and represents the volatility of the portfolio.

Bêta : Beta measures the sensitivity of an investment's returns to changes in the overall market. A beta of 1 indicates that the investment's returns move in line with the market, while a beta greater than 1 indicates that the investment is more volatile than the market.

Value at risk : VaR is a statistical measure of the maximum potential loss that a portfolio may experience over a given time period with a certain level of confidence.
For example, "We have a portfolio VaR of 250,000 USD over the next month at 95% confidence." means that, with 95% confidence, we can say that the portfolio's loss will not exceed 250,000 USD in a month.

Sharp ratio : Sharpe ratio is a risk-adjusted measure of portfolio performance. It measures a portfolio's excess return over the risk-free rate relative to its volatility.
It can be calculated with this formula : $SR_p = \frac{\mu_p}{\sigma_p}$ where SR_p is the sharp ratio of the portfolio, μ_p is the return of the portfolio and σ_p is the volatility of the portfolio.

III Existing Strategies

III.1 Naive diversification

Naive diversification is a portfolio construction strategy that consists of investing in a large of assets without considering their risk or return. One of the naive diversification strategy is to build an equally-distributed portfolio : all the assets have the same weight. For a portfolio that has N assets, each weight would be $w_i = \frac{1}{N}$
Naive diversification is based on the idea that by investing in a large number of assets, we can potentially reduce the risk of the portfolio and achieve more stable returns over time.

The naive strategy is known to surprisingly beat other strategies. The idea is that it follows the market and that we can not beat the martket

III.2 Markowitz Theory : Critical Line Algorithm / Efficient Frontier

The Efficient Frontier is a hyperbola representing portfolios with all the different combinations of assets that result into efficient portfolios.

Under constraints, one of the simplest ways to calculate the efficient frontier is by using Markowitz's Critical Line Algorithm (CLA).

There are several trading strategies that are based on the efficient frontier (cf annexe1)

But, the issues with these methods are the instability, concentration and underperfomormance. The following method should answer those issues.

III.3 Hierarchical Risk Parity : Allocation Method

This is a method that gives weights for assets in a portfolio. The objective of this method is to give weights to assets regarding their correlation.

HRP applies modern mathematics (graph theory and machine learning techniques) to build a diversified portfolio based on the information contained in the covariance matrix. However, unlike quadratic optimizers, HRP does not requires the invertibility of the covariance matrix.

This strategy is explained in the book of Marcos Lopez de Prado, and here is how it works, and the results. The algorithm consists in clustering assets, based on the correlation matrix, and to give weights to each cluster.

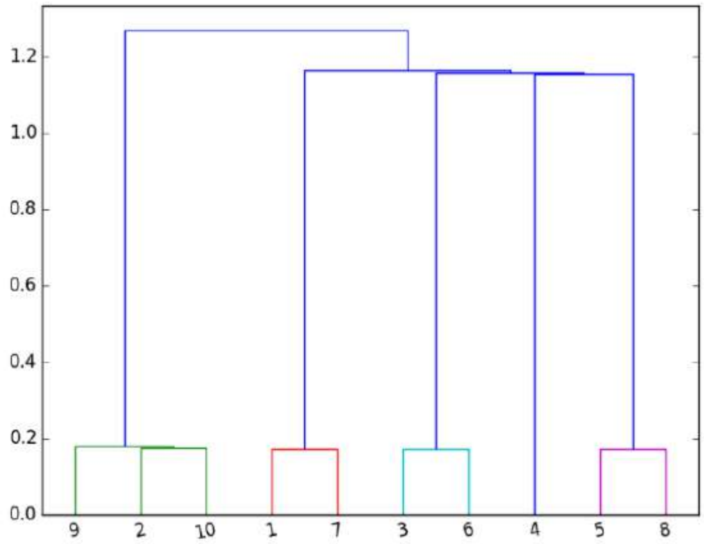


FIGURE 1 – Dendogram of cluster formation

This dendogram gather assets that are the most correlated and create a hierarchy of clusters (it clusters the cluster and so on). For example, here the algorithm identifies that assets 9,2 and 10 forms a cluster, then 1 and 7 together, 3 and 6, abd 5 and 8. In this example the asset number 4 is the only asset for which the clustering algortihm found no similarity.

Then the weight distribution is done as follow :

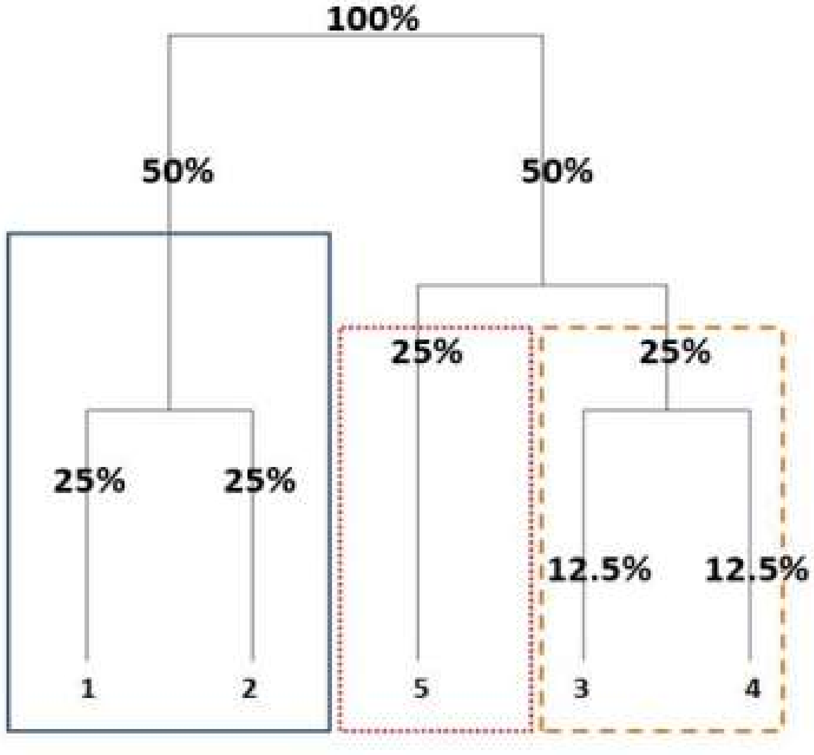
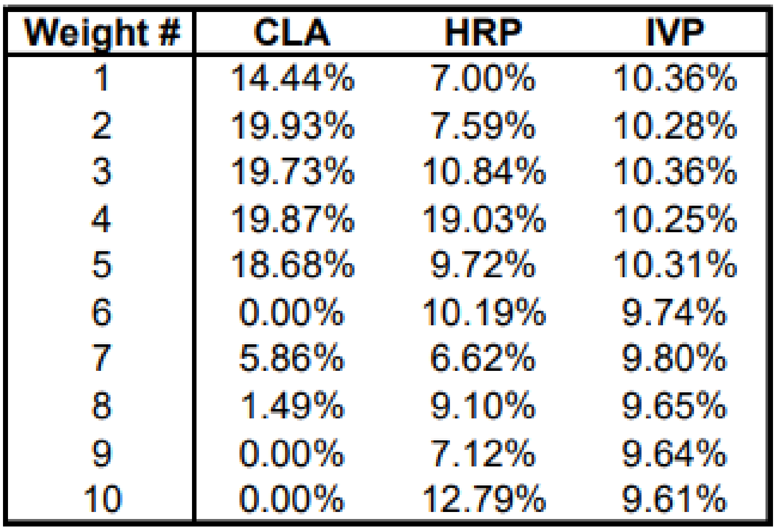


FIGURE 2 – Dendogram of cluster formation, weight distribution



Weight #	CLA	HRP	IVP
1	14.44%	7.00%	10.36%
2	19.93%	7.59%	10.28%
3	19.73%	10.84%	10.36%
4	19.87%	19.03%	10.25%
5	18.68%	9.72%	10.31%
6	0.00%	10.19%	9.74%
7	5.86%	6.62%	9.80%
8	1.49%	9.10%	9.65%
9	0.00%	7.12%	9.64%
10	0.00%	12.79%	9.61%

Figure 3 – comparison of methods

On this table, we can see the weight of each asset for three allocation methods : Capital Line Allocation ; Hierarchical Risk Parity ; Inverse Volatility portfolio. We can notice that CLA concentrates weights on a few investments (so more dependant to shocks), IVP evenly spreads weights through all invesments (ignoring the correlation) and HRP find a compromise between diversifying across investments and diversifying across cluster (making it more resilient against both shocks)

HRP strategy will be the one used to compare meta labeling with.

IV Meta Labeling

OBJECTIVES : In this section, we

1. Develop a detailed framework for the Constrained Meta-Labeling approach.
2. Describe how the constraints are incorporated into the primary model, and how the secondary model (meta-labeling) dynamically adjusts to these constraints while filtering and refining trading signals.
3. Propose relevant algorithms, techniques, and metrics for implementing Constrained Meta-Labeling in a practical setting, considering various constraint types, such as asset exposure limits, sector diversification, risk tolerance, and regulatory requirements.
4. Test our algorithm and compare it with the existing methods such as naive and HRP methods.

IV.1 Meta Labeling Strategy

First,a meta labeling strategy should, given some stocks in a portfolio, give some weights for those stocks in the portfolio. Therefore, this is a quantitative portfolio management strategy.

The difference between a regular quantitative portfolio management strategy and the meta labeling strategy is that meta labeling strategy uses a secondary machine learning model.

The role of this secondary model is to evaluate the primary model. Indeed, the primary model has to give a prediction (long/short) and the secondary model is trained to determine if the prediction of the primary model is likely to be true or not.

Here is below a diagram to give an overlook of the meta labeling strategy principles. The next paragraphs will explained in detail the different step of the method.

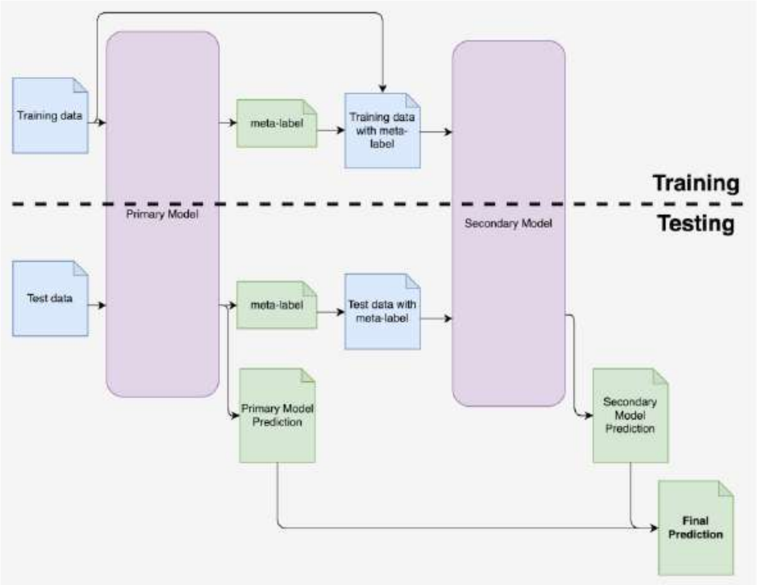
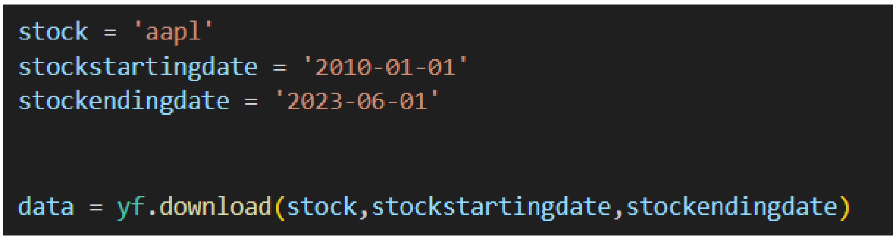


Figure 4 – Meta labeling

IV.2 Financial Data Structure

The first step in this project is to gather the data we need to train our model. To train our model we need to :

1. Choose what will be the source of data. I will use the library yfinance from yahoo. To get the data, we need to specify the stock, the starting date and the ending date :

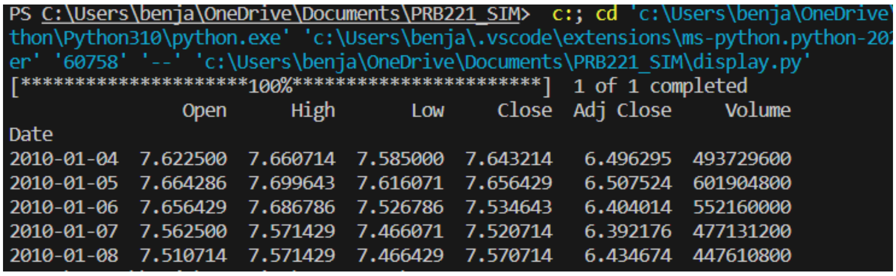


```
stock = 'aapl'
stockstartingdate = '2010-01-01'
stockendingdate = '2023-06-01'


data = yf.download(stock,stockstartingdate,stockendingdate)
```

FIGURE 5 – yfinance data

This allows us to get a dataframe like this one :

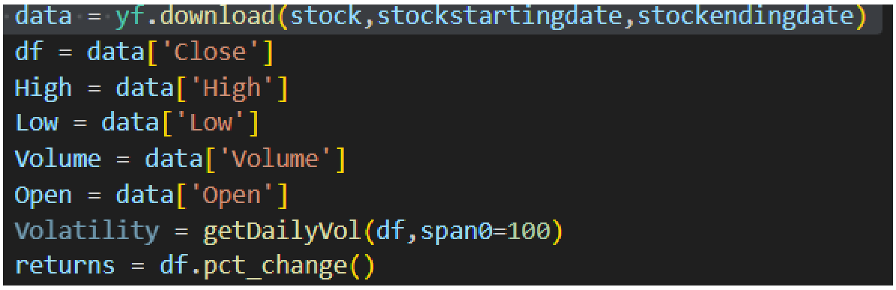


```
PS C:\Users\benja\OneDrive\Documents\PRB221_SIM>  c:; cd 'c:\Users\benja\OneDrive
thon\Python310\python.exe' 'c:\Users\benja\.vscode\extensions\ms-python.python-202
er' '60758' '--' 'c:\Users\benja\OneDrive\Documents\PRB221_SIM\display.py'
[*********************100%***********************]  1 of 1 completed
                Open      High       Low     Close  Adj Close     Volume
Date
2010-01-04  7.622500  7.660714  7.585000  7.643214   6.496295  493729600
2010-01-05  7.664286  7.699643  7.616071  7.656429   6.507524  601904800
2010-01-06  7.656429  7.686786  7.526786  7.534643   6.404014  552160000
2010-01-07  7.562500  7.571429  7.466071  7.520714   6.392176  477131200
2010-01-08  7.510714  7.571429  7.466429  7.570714   6.434674  447610800
```

FIGURE 6 – yfinance data frame

We can see the features it gives us, such as Open, High, Low, Close or Volume, but it can also give others like dividends.

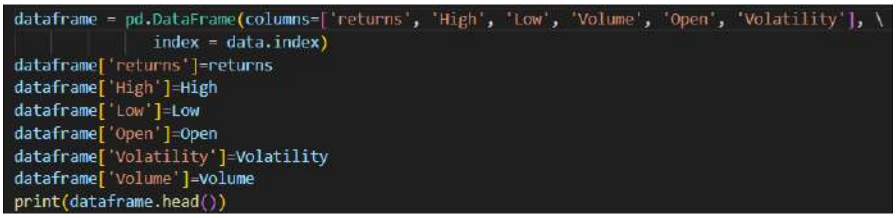
2. Choose which financial data to use. In our case, I will use the following features : Close, Open, High, Low and Volume. I will also derive the daily volatility and the returns from this data. To get this specific data, I will do as follow :



```
data = yf.download(stock,stockstartingdate,stockendingdate)
df = data['Close']
High = data['High']
Low = data['Low']
Volume = data['Volume']
Open = data['Open']
Volatility = getDailyVol(df,span0=100)
returns = df.pct_change()
```

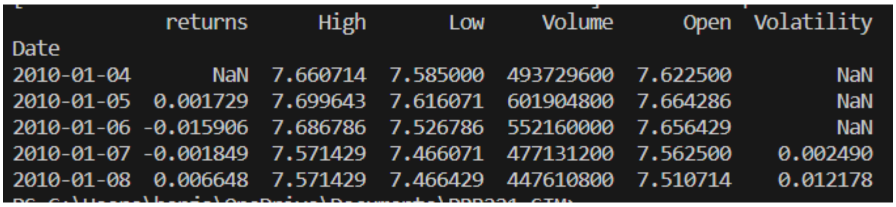
FIGURE 7 – get features

3. Create a data frame that can be used by our model. The library Pandas is used to create a data frame for machine learning models



```
dataframe = pd.DataFrame(columns=['returns', 'High', 'Low', 'Volume', 'Open', 'Volatility'], \
                    index = data.index)
dataframe['returns']=returns
dataframe['High']=High
dataframe['Low']=Low
dataframe['Open']=Open
dataframe['Volatility']=Volatility
dataframe['Volume']=Volume
print(dataframe.head())
```

FIGURE 8 – get pandas dataframe



```
             returns      High       Low     Volume      Open  Volatility
Date
2010-01-04       NaN  7.660714  7.585000  493729600  7.622500         NaN
2010-01-05  0.001729  7.699643  7.616071  601904800  7.664286         NaN
2010-01-06 -0.015906  7.686786  7.526786  552160000  7.656429         NaN
2010-01-07 -0.001849  7.571429  7.466071  477131200  7.562500    0.002490
2010-01-08  0.006648  7.571429  7.466429  447610800  7.510714    0.012178
```

FIGURE 9 – pandas dataframe

IV.3 Constraints and objectives

In theory,

We have to define the portfolio optimization objectives, such as maximizing risk-adjusted return, minimizing portfolio risk, or maximizing diversification. We also have to Identify and specify the relevant constraints, including asset exposure limits, sector diversification, risk tolerance, risk budget, and regulatory requirements.

Nevertheless, due to a lack of time, I was not able to add those constraints to my model.

IV.4 Primary model

We discussed how to produce a matrix X of financial features with yfinance and the Pandas library. This data can be used by a primary trading model that has to generates primary trading signals (long or short) on each stocks in our portfolio, based on the underlying data. This primary trading model can be any trading strategies/model, it can be :

1. Statistical arbitrage model based on the spread between two assets.
2. Machine learning model such as an SVM or Neural Network.
3. Fundamental value or events based strategy where the portfolio manager generates the signal.
4. Rules based, technical trading strategy such as moving average crossovers.

In our case, we will focus on basic technical trading strategy.

IV.4.1 Moving agerage crossover

The crossing moving average strategy (CMA) is a simple technical analysis tool that smooths out price data by creating a constantly updated average price. The average is taken over a specific period of time, like 10 days, 20 minutes, 30 weeks, or any time period the trader chooses. When the shortest average price surpass the longest average price, it means that there is an uptrend and vice versa.

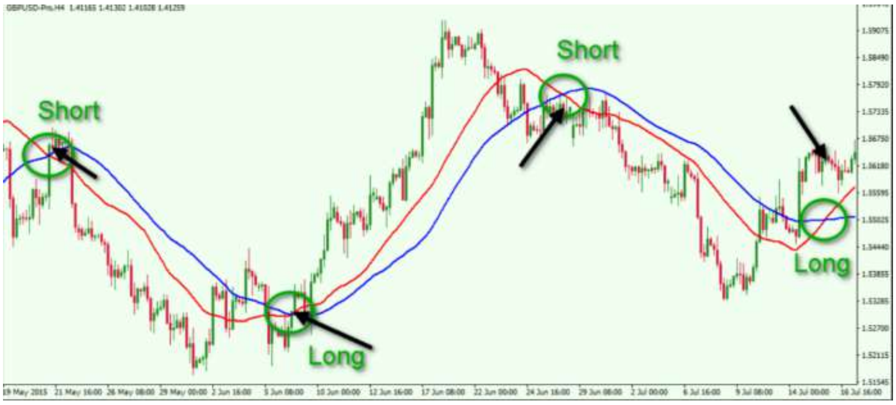
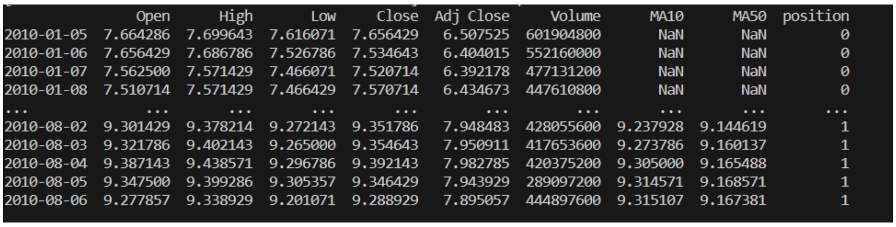


Figure 10 – CMA strategy

In this picture, we can see the blue curve that represents the average stock price over the past 50 days and the red curve represents the average stock price over the past 10 days.

Whenever the blue line overtakes the red line, this means that there is a downtrend, and whenever the red line overpass the blue line, there is an uptrend.

For each day, we can determine a position (long or short) associated with a label (1 or 0). This postion can be added to our pandas data frame, as shown below.



```
                Open      High       Low     Close  Adj Close     Volume       MA10       MA50  position
2010-01-05  7.664286  7.699643  7.616071  7.656429   6.507525  601904800        NaN        NaN         0
2010-01-06  7.656429  7.686786  7.526786  7.534643   6.404015  552160000        NaN        NaN         0
2010-01-07  7.562500  7.571429  7.466071  7.520714   6.392178  477131200        NaN        NaN         0
2010-01-08  7.510714  7.571429  7.466429  7.570714   6.434673  447610800        NaN        NaN         0
...              ...       ...       ...       ...        ...        ...        ...        ...       ...
2010-08-02  9.301429  9.378214  9.272143  9.351786   7.948483  428055600   9.237928   9.144619         1
2010-08-03  9.321786  9.402143  9.265000  9.354643   7.950911  417653600   9.273786   9.160137         1
2010-08-04  9.387143  9.438571  9.296786  9.392143   7.982785  420375200   9.305000   9.165488         1
2010-08-05  9.347500  9.399286  9.305357  9.346429   7.943929  289097200   9.314571   9.168571         1
2010-08-06  9.277857  9.338929  9.201071  9.288929   7.895057  444897600   9.315107   9.167381         1
```

FIGURE 11 – CMA data

IV.4.2 SVM : Support Vector Machines

The primary model could be a machine learning model such as support vector machines or neural networks. This case will not be studied .

IV.4.3 Labels generated

Our Primary model generates labels (1 for long and 0 for short). Therefore, the labels indicate the side of the trade and will be used for the side of our final bet (after the secondary model).

But now, we need to create a dataset to train our secondary model. For that, we have to add a column of labels telling when our primary model was right and when he was not. Those labels will be called meta labels. A meta label 1 means that our primary model was right on the trade, and a meta label 0 means our primary model was wrong on the trade.

The question is now, how to generate those meta labels ? We need for that a method telling when have a trade was positive or not, in order to compare with our primary model prediction.

IV.5 Meta-labeling : Triple barrier method

The labeling method usely used is called the Fixed-Time Horizon Method. This method can be described as follows.

Consider a series of a stock prices $(X_i)_{i=1..T}$, drawn from some bars index t= 1,...,T. Let's consider an intervalle of a length h and an arbitrary threshold τ.

Then each observation X_i is assigned a label $y_i \in \{-1, 0, 1\}$

$$y_i = \begin{cases} -1 & \text{if } r_{t_{i,0},t_{i,0}+h} < -\tau \\ 0 & \text{if} |r_{t_{i,0},t_{i,0}+h}| \leq \tau \\ 1 & \text{else.} \end{cases}$$

Where $r_{t_{i,0},t_{i,0}+h} = \frac{P_{t_{i,0}+h}}{P_{t_{i,0}}} - 1$ is the price return over a bar horizon h, $P_{t_{i,0}}$ being the price at time t_i

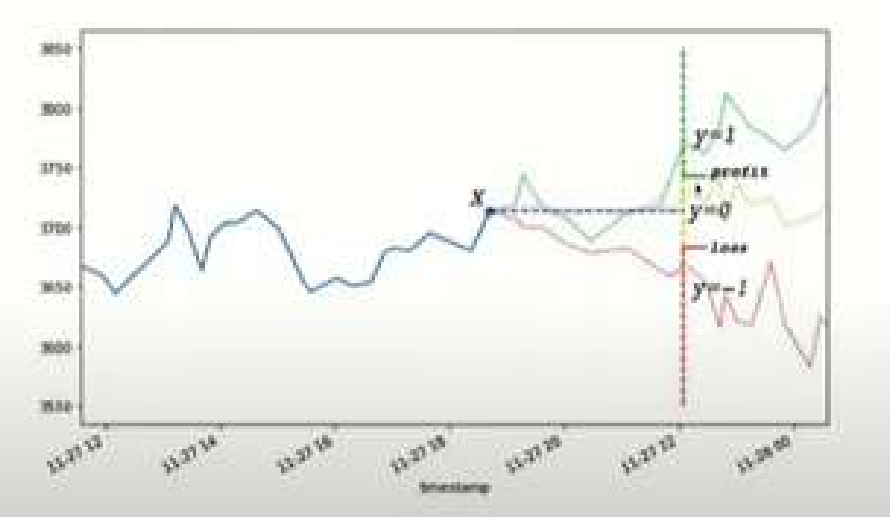


FIGURE 12 – fixed time horizon

But, this method has 2 main flaws :

First, the profit taking/stop loss threshold τ is fixed while the volatility is not

Second, this method does not take into account the path of the price during the intervalle length.

A solution proposed by De Prado is called the Triple Barrier Method. This method sets 3 barriers :

1. Upper horizontal barrier (= profit taking barrier). If touched first we label 1
2. Lower horizontal barrier (= stop loss barrier). If touched first, we label -1
3. Vertical horizontal barrier (= end of the horizon). If touched first we label the sign of the return over the term

Here is an illustration of how triple barrier method works on the stock apple, we can see that the upper barrier is touched first, so we label it 1.

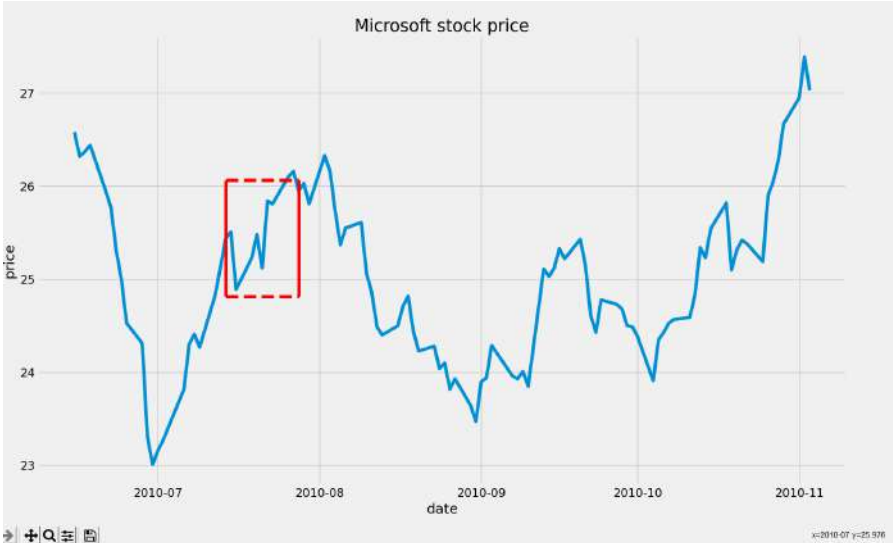


FIGURE 13 – triple barrier method

The labels generated by this method will be compared with the prediction of the primary model, if they were the same, the meta label will be 1 (= prediction was correct), otherwise the meta label will 0 (= the prediction was incorrect)

IV.6 Secondary model (meta-labeling Model)

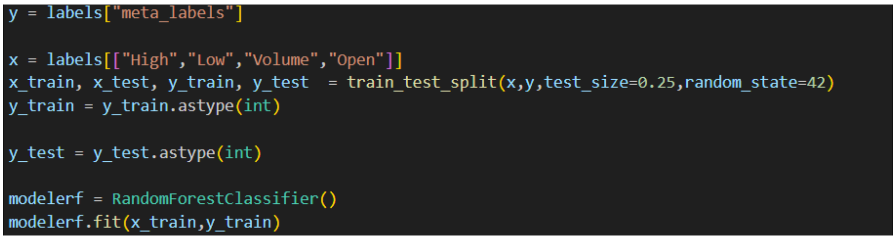
This part is where meta-labeling method takes place. The objective is to develop a secondary model which is responsible for filtering, denoising and refining the trading signals given by the primary model.

The secondary model will be a supervised machine learning algorithm that has to determine when the primary model is correct or not. Therefore, as a training dataset, we give the financial features and a meta label (0 or 1) that says when the primary model prediction was correct or not.

The question is : Which machine learning model should be used as the secondary model ?
First, we need a classifier because we want 0 or 1 as an output (as explained previously).
Then we need a model with good caracteristic for our problem.

The machine learning model suggested by De Prado is the Random Forest. This model is better than the decision trees models that are known to be prone to overfitting, which increases the variance of the forecast. Random forest models are a solution to this issue. More, random forest evaluates feature importance, meaning that it ranks the importance of each features.

I will use the library sklearn to build the random forest model. The data used to train our model will be the features : High, Low, Volume, Open. Those features will be associated with the meta label. Below is the python code showing how to train my model



```
y = labels["meta_labels"]

x = labels[["High","Low","Volume","Open"]]
x_train, x_test, y_train, y_test  = train_test_split(x,y,test_size=0.25,random_state=42)
y_train = y_train.astype(int)

y_test = y_test.astype(int)

modelerf = RandomForestClassifier()
modelerf.fit(x_train,y_train)
```

FIGURE 14 – random forest

IV.6.1 Filtering signals

We apply the secondary model (meta-labeling) to the signals generated by the primary model. This process helps in distinguishing between false and true trading signals by comparing them with the learned patterns of signals. Those false positive trades will be removed of our trades, and only the long position trades from our primary model that are validated by our secondary model will be considere.

This should improve some features of the confusion matrix, such as accuracy and F1-score.

IV.6.2 Bet Sizing

As we just see, our Meta Labeling model can filter out the false positive trade, meaning that it increases the accuracy of our model. But it can also help us to size our bet. Indeed, our primary model gives us the side (long or short) but not the size. It is extremely important to be able to size our bets, otherwise we could have a losing strategy even if we are right with our side prediction.For that, we can use the probability of misclassification of our secondary model to derive the bet size. In other words, we want to size our position according to how confident the model is about the prediction. How to do so ?

Let us denote p_x the probability that label x takes place. We would like to use this predicted probability to derive the bet

— For two possible outcomes, $x \in \{-1, 1\}$, we test the null hypothesis $H_0 : p[x = 1] = \frac{1}{2}$.

— We compute the test statistic $z = \frac{p[x=1]-\frac{1}{2}}{\sqrt{p[x=1](1-p[x=1])}} = \frac{2p[x=1]-1}{2\sqrt{p[x=1](1-p[x=1])}} \approx Z$ with $z \in \{-\infty, \infty\}$ and where Z represents the standard normal distribution.

— We derive the bet size as $m = 2Z[z] - 1$, where $m \in [-1, 1]$ and $Z[.]$ is the cumulative distribution function (CDF) of the standard Normal distribution.

At the end of the day, giving weights of my stocks in my portfolio will do as follow :

1. We have a primary model that tells us to go long or short. In my case, I only consider long positions. So I will only consider the trading signals telling me to go long.

2. The secondary model gives us a prediction on wether the primary model is right or not. I will keep the trades that are confirmed by the secondary model. This filters out the

false positives trades.

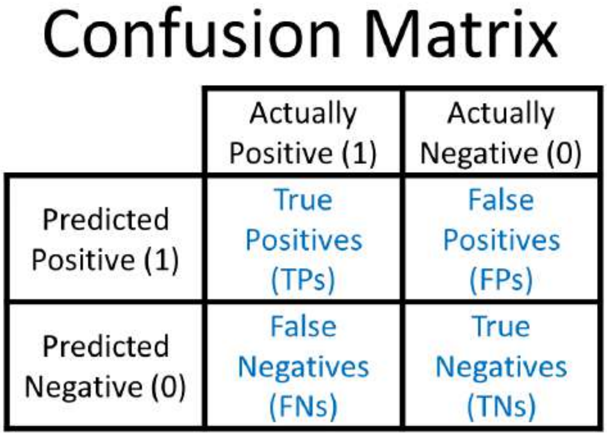
3. The predicted long trades confirmed by the secondary model are assigned a weight, given by the bet sizing method presented. This will give me a list of weights for the assets.

4. Normalize the list so that the sum of the weights equals 1.

5. Repeat the process each day and calculate the returns of the portfolio

IV.7 Metrics for evaluating Constrained Meta-Labeling performance

We have to define appropriate performance metrics to evaluate the effectiveness of the Constrained Meta-Labeling approach. One of the objective of the meta labeling technique is to increase the F1-score. The confusion matrix gives us an overlook of the efficiency of our model by showing the True/False Positives and the True/False Negatives, as presented below



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

FIGURE 15 – Confusion Matrix

This matrix is given by the library sklearn. The different indicators of this matrix are :

precision $= \frac{TP}{TP+FP}$

recall $= \frac{TP}{TP+FN}$

accuracy $= \frac{TP+TN}{total}$

F1 score $= 2 * \frac{precision*recall}{precision+recall}$

The use of those mesures is very specific to the topic of research. For example, a machine learning algorithm used to detect a cancer should have a very high recall since we want to detect all the true positives and avoid the false negatives. In our case, we want the primary model to have a high recall (in order to get a maximun of trading opportunities) and the secondary model to have a higher precision, by filtering out the false positive trades.

The confusion matrix will help us to compare the strategy with meta-labeling and without.

The indicators used to evaluate the portfolio strategy will be :

1. Return
2. Risk
3. Sharpe Ratio

There are also some indicators that will not be used here but that helps to evaluate the portfolio strategy :

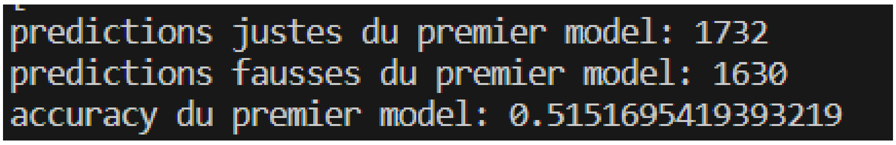
1. Information Ratio (IR) : measures the risk-adjusted return of the optimized portfolio relative to a benchmark.
2. Turnover Rate : reflects the level of trading activity and can be used to gauge the impact of transaction costs on the portfolio's performance.
3. Diversification Ratio : measures the extent to which the portfolio is diversified across different assets or sectors.
4. Compliance Ratio : quantifies the extent to which the portfolio satisfies the imposed constraints and regulatory requirements.

IV.8 Implementation in a practical setting

Now it is time to implement our strategy in python. We will consider a portfolio of ten stocks : ["AAPL", "MSFT", "GOOGL", "AMZN", "TSLA", "JPM", "V", "JNJ", "PG", "WMT"]

IV.8.1 Primary Model

First, let's look at primary model strategy (the crossing moving average strategy). The image below shows how many prediction were right and how many false.



```
predictions justes du premier model: 1732
predictions fausses du premier model: 1630
accuracy du premier model: 0.5151695419393219
```

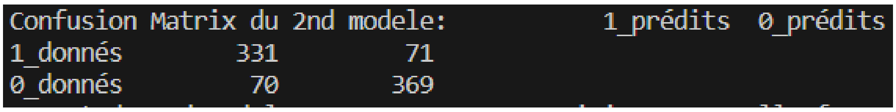
Figure 16 – 1st Model predictions

We can notice that the performance of the primary model is not very good (which is understandable for a basic technical trading model).

The accuracy is of 0.515 which is barely better than random trades (accuracy $= \frac{\text{True predictions}}{\text{Total predictions}}$)

IV.8.2 Secondary Model

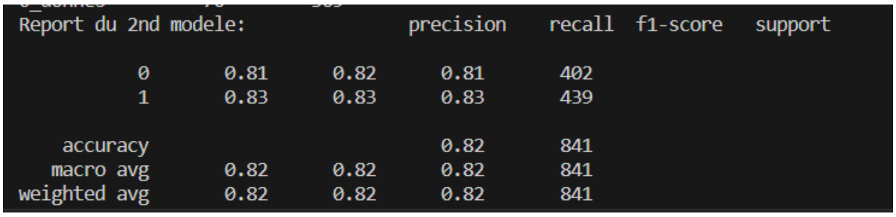
We train our secondary model with the features High, Low, Open, Volume and the meta labels generated by the primary model and the triple barrier method. The confusion matrix is presented below :



```
Confusion Matrix du 2nd modele:          1_prédits  0_prédits
1_donnés        331        71
0_donnés         70       369
```

Figure 17 – Confusion Matrix 2nd Model

We can see that our model has a quite good ability to predict when our primary model is correct or not. This report (that can be gotten by the sklearn library) confirms it :



```
Report du 2nd modele:             precision    recall  f1-score   support

           0       0.81      0.82      0.81       402
           1       0.83      0.83      0.83       439

    accuracy                           0.82       841
   macro avg       0.82      0.82      0.82       841
weighted avg       0.82      0.82      0.82       841
```

Figure 18 – features 2nd Model

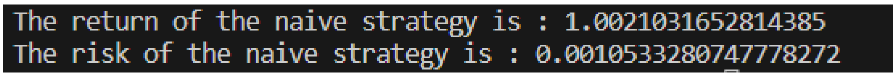
The accuracy of the secondary model is of 0.82. This means that 82% of the time, the model is able to predict if the primary model is correct or not. As a result, this allows us to correct the trading signals. The report also gives us some information on the precision and the F1-score.

IV.8.3 Backtesting our strategy

We can now compare our meta-labeling strategy with the existing strategies (Naive and HRP). Since our models take a lot of time to run, and have some issues with testing strategies on a long term, I will present my strategy backested on 2 intervalls.

The first one is on a short term intervall : 10 days (01/06/2023 to 10/06/2023).

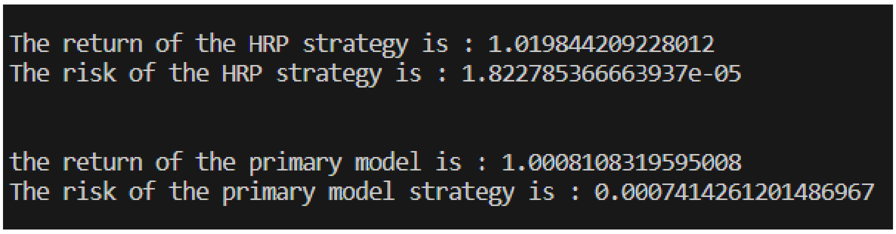
For each strategy, I coded a function that gives the cumulative returns and the risk of the strategy are given (risk being the standard deviation of the returns).



```
The return of the naive strategy is : 1.0021031652814385
The risk of the naive strategy is : 0.0010533280747778272
```

Figure 19 – Result Naive strategy 10 days

The Naive strategy seems not very efficient on this intervall since the return is low and the risk quite high



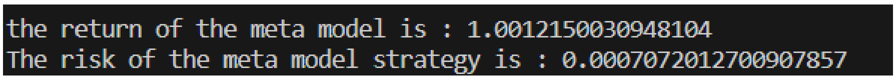
```
The return of the HRP strategy is : 1.019844209228012
The risk of the HRP strategy is : 1.822785366663937e-05


the return of the primary model is : 1.0008108319595008
The risk of the primary model strategy is : 0.0007414261201486967
```

Figure 20 – Result HRP and 1st Model strategies 10 days

The HRP strategy has good returns on 10 days only and the risk is very low, so the strategy has been very efficient on this intervall.

Regarding the primary model strategy, both the return and the risk are low.

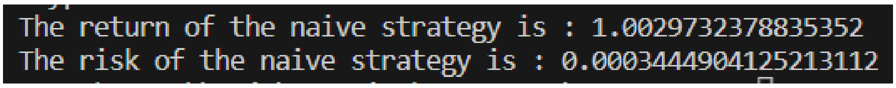


```
the return of the meta model is : 1.0012150030948104
The risk of the meta model strategy is : 0.0007072012700907857
```

Figure 21 – Result meta model strategy 10 days

For the meta model strategy, we can see a higher return than the primary model with a low risk. On this intervall, the meta model has been efficient, even though it is less efficient than the HRP method (but more than the Naive strategy)

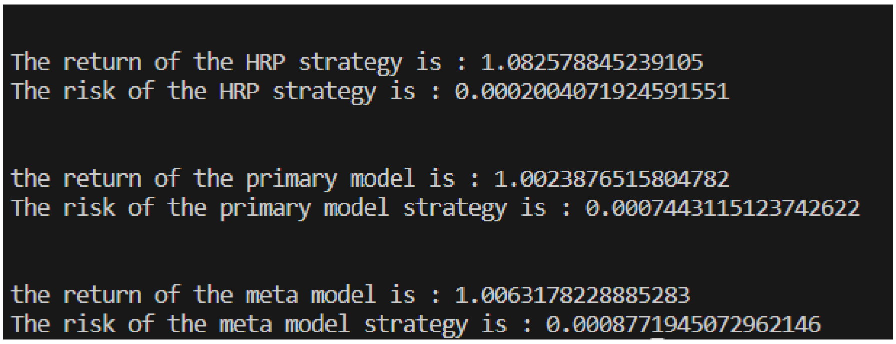
The second one is on a longer term intervall : 1 month (01/06/2023 to 01/07/2023). For a longer intervall, the strategy is less sensitive to short term fluctuations.



```
The return of the naive strategy is : 1.00297323788835352
The risk of the naive strategy is : 0.0003444904125213112
```

Figure 22 – Result Naive strategy 1M

We can see a higher return and a lower risk than the same strategy on 10 days. However the return looks still weak.



```
The return of the HRP strategy is : 1.082578845239105
The risk of the HRP strategy is : 0.0002004071924591551


the return of the primary model is : 1.0023876515804782
The risk of the primary model strategy is : 0.0007443115123742622


the return of the meta model is : 1.0063178228885283
The risk of the meta model strategy is : 0.0008771945072962146
```

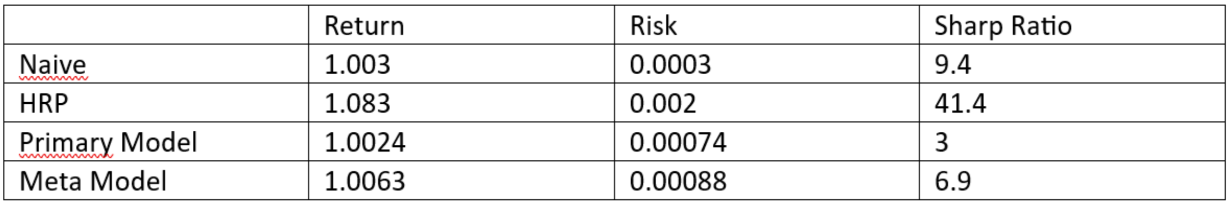
Figure 23 – Result HRP + 1st Model + Meta Model strategies 1 M

The HRP strategy has a very good renturn (8.2% in a month) for a low risk (0.002%). During this period, this strategy was extremely efficient.

However, the crossing moving average model keeps a low return (0.24%).

The meta model improves once again this primary model by increasing the returns (to 0.64%) and maintaining the risk around (0.008%)

Here is a sum up of the strategies results on the 1-month intervall :



	Return	Risk	Sharp Ratio
Naive	1.003	0.0003	9.4
HRP	1.083	0.002	41.4
Primary Model	1.0024	0.00074	3
Meta Model	1.0063	0.00088	6.9

Figure 24 – Sum Up 1 M

Commentary on the results :

- The HRP method seems to be the most efficient one, it has a way better return than the other strategies.
- The naive strategy has very low risk, compared to the others. This can be explained by the fact that with this strategy we invest in all stocks, all equally distributed. As a result, it is less sensitive to the fluctation of each stock, so a lower risk. Nevertheless, the return is weak (no optimisation, no good trade detected).
- The primary model is not efficient : it has a very low return. Furthermore, it has the lowest sharp ratio.
- The Meta model signifficently improves the primary model. However, it seems to be less efficient than a complex quantitative strategy like HRP.

Even if the meta-model adds some efficiency to the model, I think that we could do way better with more work on it, and the next paragraph discusses what could be done.

IV.9 What could be added

There are some points that were not discussed or implemented in my research project but that would have been very interesting to or important to study.

1. The constraints and objectives as discussed in the paragraph 4.3
2. I should have take into account the cost of trades because in this case, strategies such as the naive strategies would much more interesting (no trades)
3. We could also try different transaction rates (others than 1 day) to see if the strategy would be more efficient on a longer transaction rate (such as 5 days, a week or two weeks).
4. Try a different primary model, a machine learning model for example. I think it would be much more interesting and efficient to use a primary model such as neural networks.

Indeed, the secondary model has to learn when the primary model is true or not, meaning that the secondary model has to find a pattern on the primary model.

5. We could also use the HRP model to bet size our strategy. If we detect long trades with the primary model, then we eliminate the false positive trades with the secondary model, we could finally size our long position with the HRP method.
6. One of the most important and biggest topic discussed in the book of Marcos Lopez De Prado that I barely worked on, is the financial data analysis. There is a lot of work to do on how to gather and select the important data, which data to be collected, etc... For example, should we collect data every N-days or should we have an event based strategy (and use a cusum filter for example). Since our machine learning models depend on this data, it is very important to work on improving the quality of this data.

Conclusion

To conclude, the project I worked on was very interesting and exciting. Portfolio managment is an important topic in finance (and not only quantitative finance). Working on using recent machine learning techniques to develop a portfolio strategy was a rewarding opportunity for me. Indeed it allowed me to :

1. Learn and understand financial concept that were completly new for me.
2. Use my mathematics knowledge to understand the markowitz theory and other financial mathematics concept.
3. Discover existing portfolio strategies and execute them on python
4. Develop my computer science skills (mostly in python). Learn how to use machine learning models, how to prepare the data for the model, how to evaluate the model and how to get the different caracteristics of the model. I also learn to manipulate data with the library Pandas.
5. I understood the different issues with quantitive strategies for portfolio management, such as backtesting, overfitting,etc...

I would say that, even though I was not able to work on all topics I wanted to, I had a very wide understanding of quantitative portfolio management techniques and issues. My research internship was not very theoretical (so I did not develop precise theoretical skills), but it was very practical, so it allowed to developp a lot of valuable skills that I will surely reuse later on my career.

As a result, I am very happy to have worked on this project, because of all I learned with that.

Moreover, living in Toronto was a wonderful experience. I met a lot of nice people there, discovered a new culture, new landscape and mostly, I could practice and improved my english level !

Références

[7(2021)] Markowitz model. *QuantPedia*, 2021.

[A. Singh(2022)] J. Joubert A. Singh. Does-meta-labeling-add-to-signal-efficacys. *Hudson-Thames*, 2022.

[Chan()] Ernie Chan. Meta-labeling : a key machine learning tool. *Epchan*.

[De Prado(2018)] Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

[Gui(2020)] Ke Gui. Data labeling ; the triple barrier method. *Towards Data Science*, 2020.

[Hsia(2022)] Michael Hsia. Momentum trading : Use machine learning to boost your day trading skill - meta-labeling. *Michael's blog*, 2022.

[Raffinot(2018)] Thomas Raffinot. The hierarchical equal risk contribution portfolio. 2018.

A Annexes 1

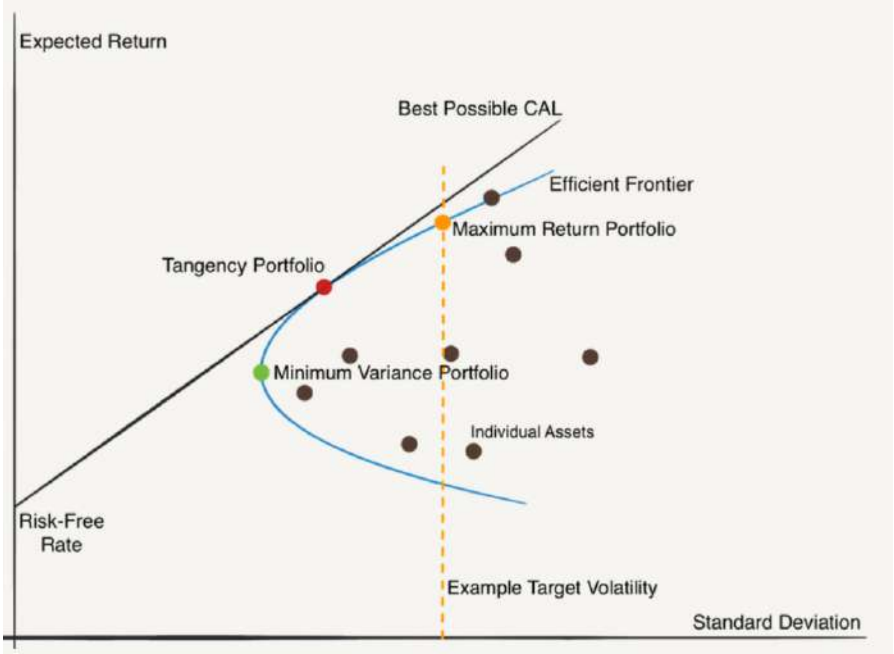


Figure 25 – Critical Line Allocation

Table des figures