Machine Learning Engineer Nanodegree

Capstone Proposal

Gregorio Polito
May 08th, 2020

Proposal

Domain Background

The current project proposal aims to solve one of the most interesting problems in finance: Stock Price prediction. Such problem is assuming a central role in finance for both trading and risk-management purposes in order to ensure that no future crises would be unpredictable. The use of the Machine Learning is providing a way to success. Several hedge funds and banks are using Machine Learning for stock price prediction and are generating higher earnings. The use of Machine Learning helps finding patterns in the dataset which may help in a more accurate prediction of the future prices. In addition, such predictions are commonly applied for the portfolio optimization (whose aim is to find the best way to combine stocks in order to maximize a metric of interest, e.g. utility function, log-term return, Sharpe Ratio, etc.).

Sometimes only large corporates and hedge funds can account for Machine Learning algorithms for investment purposes, but everyone can be interested in good stock prices predictors in order to assure the profitability of the own investments: such solutions could help people in finding the best strategy.

Problem Statement

Taking into account a portfolio of Stocks, defined through the couples (S_i, w_i) where at each stock S_i is assigned a weight w_i such that the sum of the weights is 1 (each weight can be either positive or zero), the problem is to find the best weights combination in order to assure higher annualized Sharpe Ratio[1] over a given period. In particular, the problem is divided into the following components

1- Stock prices forecasting: for each stock in perimeter the Adjusted Closing Stock price is predicted on a future date T
2- Taking into account the predictions computed in 1., the portfolio composition (i.e. the stock weights w_i) is optimized in order to maximize the annualized Sharpe Ratio.

[1] For the definition of the Sharpe Ratio, please refer to the link https://en.wikipedia.org/wiki/Sharpe_ratio

Datasets and Inputs

The dataset used for the prediction component consists in the historical closing prices over the training time window for the main European Stock Indices. The whole dataset is retrieved from Yahoo! Finance APIs (i.e. *yfinance* python module). The dataset is made of the daily stock values such as Open, High, Low, Close, Volume and Adjusted Close and only the last value is used for the calibration and prediction.

Regarding the portfolio composition optimization, the input is represented by

1- Future time T (after the last datapoint used in the training part)
2- The current portfolio composition, whose stocks are a subset of the whole Yahoo! Finance dataset

Solution Statement

The Stock prices prediction uses the Amazon SageMaker DeepAR which is a supervised learning algorithm based on recurrent neural networks (RNN).

On the other hand, the portfolio optimization is performed computing the optimal weights composition in order to maximize the annualized Sharpe Ratio on a given future date. Such metrics takes as input the Expected Value and the Standard Deviation of the daily Excess Return computed at portfolio level. Considering the simulated Stock prices S_i^T in T, the current spot price S_i observed in t_0 and the portfolio weights w_i, the Sharpe Ratio is computed according the following steps (for sake of simplicity, the risk-free rate is assumed 0%)

1. Daily Excess Return of the predicted portfolio on a single predicted path

$$\frac{1}{T-t_0}\cdot\sum_i w_i\cdot\frac{S_i^T}{S_i}$$

2. Expected Value $\mathbb{E}$ of the predicted Excess Return (computed over the whole set of simulated values)

$$\mathbb{E}=E_{Sim}\left[\frac{1}{T-t_0}\cdot\sum_i w_i\cdot\frac{S_i^T}{S_i}\right]=\sum_i w_i\cdot E_{Sim}\left[\frac{1}{T-t_0}\cdot\frac{S_i^T}{S_i}\right]$$

3. Standard Deviation σ of the predicted Excess Return (computed over the whole set of simulated values)

$$\sigma=\sqrt{Variance_{Sim}\left[\sum_i\frac{S_i^T}{S_i}\cdot\frac{1}{T-t_0}\cdot w_i\right]}$$

4. Annualized Sharpe Ratio (considering a year made of 252 business days)

$$\frac{\mathbb{E}}{\sigma}$$

the optimal portfolio composition is obtained maximizing the Sharpe Ratio over the different weights combination.

Benchmark Model

The benchmarking activity is performed for the prediction part of the problem: once the RNN is trained, the prediction performance, on a given stock and on a given future date, is assessed in terms of the actual stock price on the given future date. If the actual value is contained in the predicted quantiles, the performance is assumed good.

Evaluation Metrics

For the evaluation of the prediction component, the daily prediction is performed and the predicted quantiles are compared with the actual stock value observed during the prediction horizon. If the actual stock price systematically exceeds the predicted quantiles, the performance is assumed bad, otherwise the performance is good. For this kind of problems, it is difficult to assess the goodness of the prediction performance in a rigorous quantitative way: in case of crises, the actual prices may reach levels which have never be observed in the history.

The performance of the optimization is assessed in terms of the output of the optimization function which highlights if the optimization algorithm succeeds.

Project Design

According to the problem, the project will be divided into 2 steps: the first one relies to the stock prediction and the second on the portfolio optimization.

Regarding the stock prediction, the first step consists in the definition of the training set: the historical time window consists in the year 2019, starting from 1st January 2019 up to 31st December 2019, and the perimeter of stocks is made of the main European Stock Indices. The prediction process is implemented using the RNN (in particular Amazon SageMaker DeepAR library). Once the network is trained, the prediction of the indices will be performed starting from 1st January 2020 and, potentially, up to today.

Due to the common geographic area, the expected behavior is that the indices will be highly correlated and consequently the DeepAR algorithm could perform well in the prediction process.

Concerning the portfolio optimization, the portfolio will be defined as a list of indices and the aim of the optimization is to assign a weight to each stock in order to maximize the Sharpe Ratio.

Let's start with the definition of the Sharpe Ratio: given the list of weights, it is computed as the ratio between the Portfolio Expected Value and Standard Deviation (apart from the annualization scaling factor). Such statistics (both Expected Value and Standard Deviation) are computed on the predicted daily excess return (as defined in the Solution Statement). Using matrix notation, with w the vector of weights of N dimension, the Expected Value $\mathbb{E}$ is defined as

$$\mathbb{E} = w^T \cdot \begin{bmatrix} E_{Sim}\left[\frac{1}{T-t_0} \cdot \frac{S_1^T}{S_1}\right] \\ \vdots \\ E_{Sim}\left[\frac{1}{T-t_0} \cdot \frac{S_N^T}{S_N}\right] \end{bmatrix}$$

and the Standard σ as

$$\sigma^2 = w^T \cdot varcov_{Sim}\left(\frac{1}{T-t_0} \cdot \frac{S_1^T}{S_1}, \dots, \frac{1}{T-t_0} \cdot \frac{S_N^T}{S_N}\right) \cdot w$$

Hence considering a portfolio with unit weights with Expected Value $\mathbb{E}_1$ and the variance-covariance matrix between the whole set of Indices COV the statistics of the general case (with different weights) are defined as follows

$$\mathbb{E} = w^T \cdot \mathbb{E}_1 \quad and \quad \sigma = \sqrt{w^T \cdot COV \cdot w}$$

defining

$$\mathbb{E}_1 = \begin{bmatrix} E_{Sim}\left[\frac{1}{T-t_0} \cdot \frac{S_1^T}{S_1}\right] \\ \vdots \\ E_{Sim}\left[\frac{1}{T-t_0} \cdot \frac{S_N^T}{S_N}\right] \end{bmatrix} \quad and \quad COV = varcov_{Sim}\left(\frac{1}{T-t_0} \cdot \frac{S_1^T}{S_1}, \dots, \frac{1}{T-t_0} \cdot \frac{S_N^T}{S_N}\right)$$

In order the find the maximum Sharpe Ratio according the different portfolio compositions, the Sharpe Ratio is seen as a function of the different weights and the algorithm is summarized as

```
def SharpeRatio(w, EV, COV)
        EV_portfolio = w*EV
        STD_portfolio = sqrt(w.T * COV * w)
        SharpeRatio = sqrt(252)*EV_portfolio/STD_portfolio
```

The optimization is defined as

```
w_star = argmax(ShapeRatioMC)
subject to
        w>=0
        sum(w)==1
```

where the weights are bounded to be nonnegative and to have unit sum.
The result of the optimization function represents the weights to be assigned to each Index in order to maximize the Sharpe Ratio. The maximization is performed using the *scipy.optimize* module, in particular minimizing the function $-1 \cdot SharpeRatio$ (which is equivalent to maximizing the SharpeRatio).