Forecasting the direction of High-Frequency Returns: An ensemble-trees application

A. A. P. Santos* and R. S. Wagner†

July, 2018

Abstract

We propose the construction of an intra-daily algorithmic trading strategy based on machine learning models, specifically Tree-Based Methods as Gradient Boosting, Random Forests and BART. The models are constructed to capture the dynamic of market microstructure variables and model their impact into the sign of the future returns. We also verify the impact in the statistical and financial performance of time aggregation and minimum returns requirements for class identification. In the end we verify the financial significance of our models through a trading simulation.

Introduction

Forecast the financial markets is of huge interest for different agents. Besides the variety of different interests this is not an easy task, expectation of asset returns is of hard predictability, but in spite of that some success has been achieved along the years including the forecasts of returns signs. Recent works made by Dixon, Klabjan e Bang (2016), Tsantekidis et al. (2017) and Fletcher e Shawe-Taylor (2013) already verified that the sign of high-frequency returns can be predicted by modeling the trading dynamic.

We propose to verify the ability of an algorithms family that as far as we know hasn't yet been fully tested, Ensemble-trees are a hot topic inside the Machine Learning (M.L.) community specially due to their recently excellent performance in M.L. competitions (CHEN; GUESTRIN, 2016). Breiman et al. (1984) were the responsible to popularize the decision trees, and since then a lot of tree ensemble methods has been presented in the literature. We'll investigate the forecast ability of three of them: Boosting (FREUND; SCHAPIRE, 1997), Random Forests (BREIMAN, 2001) and BART (CHIPMAN et al., 2010).

We'll use this models to forecast the signal of asset returns based on 27 Market Microstructure variables, and asses models quality with the major Brazilian asset, the Petrobras Company (PETR4.SA), recently in media due to political corruption scandals. The series starts in Dec/2015 and goes to Jun/2018, part of the troubled period.

1 Theoretical Reference

1.1 Sign of Returns

Due to recent results to be presented in section 1.2, we'll investigate if exists predictability of returns sign in intra-daily horizons. In high frequency horizons, observations without any change on prices are quite common, this kind of situation defines a 3 class prediction problem, returns R_t can be positive, negative or null. Null return can be intuitively understood as $R_t = 0$, but we'll define it in a different way, due to the noise on high frequency signs we'll test different minimums of price oscillation to classify a return as different from null, in this work null return will be defined as $|R_{t+1}| < m$ where m is a real constant, this was originally proposed by Tsantekidis et al. (2017).

Should be clear by now that we have on our hands a three classification problem, this kind

* Economics Department, Federal University of Santa Catarina.
† Graduate program in Economics from Federal University of Santa Catarina.
Contact: rafaelsilvawagner@gmail.com

Full Article Available at <https://bit.ly/2LuIeWd>

of problem is usually described by a multinomial density function like (1):

$$P(X_1 = n_1, \ldots, X_r = n_r) = \frac{n!}{n_1! \ldots n_r!} p_1^{n_1} \ldots p_r^{n_r}, \quad (1)$$

for the rest of this paper we'll assume that our signals follows (1).

1.2 Related Work

Machine learning uses generic algorithms of high versatility without requiring specific hypothesis about the data generating process (DGP), this versatility makes them highly adaptable to different datasets. Recently several works achieved good performance using machine learning to forecast high-frequency returns signs, our interest is inside a share of this literature that combines those algorithms to Market Microstructure variables.

Inside the Deep Learning paradigm we have the works of Dixon (2017) and Tsantekidis et al. (2017), that modeled the volumes into the top 10 levels of the Limit Order Book (LOB). The evaluated architectures goes from Traditional Multilayer Perceptrons passing through Recurrent, Long-Memory and Convolution Neural Networks, they also compared these architectures to Support Vector Machines, a technique that was previously analyzed by Fletcher e Shawe-Taylor (2013) and Kercheval e Zhang (2015). On these two last works the variety of variables, and consequently, richness of modeled dynamic was bigger, they verified that the most important variables for forecasting were inside the trading dynamics like last price and number of traded stocks. Han et al. (2015) extended this rich dynamic to the classification trees world, and achieved even better forecasting capacity.

The lack of one predominant technique to model the relationship between the variety of Microstructure estimators and returns signals turns this an open point in literature. In our work we'll increase one more test to this literature, Ensemble trees have recently returned to popularity due to the good performance of the XGBoost system in M.L. competitions (CHEN; GUESTRIN, 2016). Despite this awesome performance other ensemble tree methods are quite famous, as stated before we'll recover the work of Han et al. (2015) and evaluate the random forest model, in addition to that inspired by the recently great performance of Bayesian Additive Trees we will also add MPBART (KINDO; WANG; PEÑA, 2016) to our analysis.

2 Methodology

2.1 Data-Set

To construct our dataset we'll use the stock high frequency database from the Brazilian Stock Exchange B3.SA; we are in debt with Perlin e Ramos (2016) for the construction of a system that makes the manipulation of this dataset so simple. Our database starts in 01/12/2015 and goes to 12/06/2018. With this dataset we constructed the following descriptive variables: Time; Last Price; Last Return; Sign of last Return; Realized Volatility; Number of trades; Quantity of Trades; Volume; Weighted Price by Volume; New, Updated and Canceled Orders; Offers Maximum, Minimum and Weighted Price.

To realize the frequency analysis we constructed 60 databases with frequencies ranging between 1 and 60 minutes, increasing the frequency in each successive databases by one minute. To add the time series structure we'll work with variables lagged two periods and use the time until the trading sections open as covariate. After we find an optimal frequency we'll apply an approach proposed by Casals, Jerez e Sotoca (2009) and combine variables aggregated in other frequencies.

2.2 Competitor Models

As usual we want to construct a function regressing the labels y_i into our variables $x_i = (x_{1,i}, x_{2,i}, ..., x_{m,i})$ with $i = 1, \ldots, N$, and we want to use some model $\hat{f}(x)$ to make this function. we will test four different models to construct this function. We'll make forecasts for the returns signs using three Tree-Ensemble methods, Random Forests, Boosting and BART, additionally we'll estimate the Logistic regression. The interest for the tree tree algorithms is due to the low volume of works evaluating their performance, with exception of Random Forests that has already been tested by Han et al. (2015).

2.2.1 Logistic Regression

The interest in logistic regression is due to it being one of the most traditional linear regression model for classification, usually when we see a non-linear classification method we ask if the improvement of forecasting quality is due to capacity of generating non-linearities in the regressors space, Logit can give us some intuition for that. We can initially define the logits of each reference class as:

$$log\frac{\pi_{ij}}{\pi_{iJ}} = log\frac{\pi_{ij}}{1-\sum_{j=1}^{J-1}\pi_{ij}} = \sum_{k=0}^{k} x_{ik}\beta_{kj}, \quad (2)$$

where we define the log-chances of one class j in relation to some chosen basis class J, in an specific i scenario. Solving for π_{ij}, and standardizing the probabilities to sum 1 and be positive we get to:

$$\pi_{ij} = \frac{exp(\sum_{k=0}^{K} x_{ik}\beta kj)}{1+(exp\sum_{k=0}^{J-1} x_{ik}\beta kj)}, \quad j \leq J, \quad (3)$$

$$\pi_{iJ} = \frac{1}{1+exp(\sum_{j=0}^{J-1} x_{ik}\beta kj)}. \quad (4)$$

The equation (3) describe the probability of each of the j classes in the i scenario and (4) describes the probability of the basis class. To find the weights β_i we make a gradient search seeking to maximize the likelihood obtained through the joint density.

2.2.2 Regression Trees

The next three models are all created expanding the concept of regression trees (BREIMAN et al., 1984). Regression Trees divide the regressors space into hyper-rectangles and assign a real constant w_j to each region so that given a vector $x \in \mathbb{R}_j \to f(x) = w_j$ we can present the trees as $T(x;\Theta) = \sum_{j=1}^{J} w_j I(x \in \mathbb{R}_J)$, where the parameter $\Theta = \{R_j, w_j\}_1^J$, contains the R_j hyper-rectangles and the w_j constants used to forecast in each of the J regions, usually J is treated as an hyper-parameter previously chosen.

2.2.2.1 Boosting

Boosting (SCHAPIRE, 1991) is a modeling philosophy that combines different weak-learners, seeking to achieve one combined strong-learner. We can represent our tree combination as:

$$\hat{y}_i = \phi(\boldsymbol{x}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \ \ f_k \in \mathcal{F}, \quad (5)$$

where $\mathcal{F} = \{f(x) = w_{q(x)}\}$ $(q : \mathbb{R}^m \Rightarrow T, w \in \mathbb{R}^t)$ is the Classification and Regression Tree (CART) space, defining our final model (5) as a sum of K trees.

Analytically this problem is not well-defined and we need a numeric optimization algorithm to solve this problem, usually, as a sum of trees, which in each step t we estimate one additional tree conditionally on the previous ones:

$$\begin{aligned} \hat{y}_i^{(0)} &= 0, \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i), \\ &\dots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i), \end{aligned} \quad (6)$$

in each iteration the estimated function $f_t(x_i)$ will be the one that minimizes the following regularized objective function:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad (7)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda|||w||^2, \quad (8)$$

The penalization element (8) helps to smooth the learned functions preventing the over-fitting, when we try to minimize (7) with more trees T or leafs w; γ and λ are hyper parameters that control the learning smoothness, and l is a convex loss function that measures the forecasts quality, in our classification case usually the mlogloss function. There's a number of different implementations of this sequential estimation, in this work we'll use the famous system of Chen e Guestrin (2016)

2.2.3 Random Forests

Another famous ensemble-tree method is Random Forests (BREIMAN, 2001) which is inspired in Bagging. Bagging is a forecast combination methodology that aggregates B models and uses their means as the punctual forecast as in (9), we'll select a sample of size $b < N$ from our dataset B times and fit B models $\hat{f}^{*B}(x)$ with them.

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{g=1}^{B} \hat{f}^{*g}(x) \quad (9)$$

Random Forests adds a new procedure to Bagging trying to decrease the model correlation in the growing tree process. When constructing the classification trees we'll in every split change the underlying DGP by changing the split candidate-variables. In every split we'll select randomly $r < k$ covariates that are eligible for a split and restrict the candidate split variables to

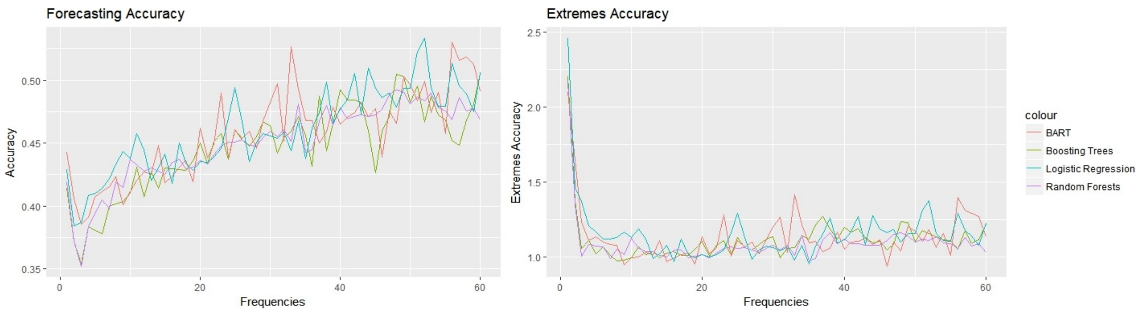

Figure 1 – Forecasting Accuracy - Frequency Analysis

Horizontal axis display the evolution from 1 to 60 minute frequency. Vertical axis indicate the Forecasting Accuracy. Left Graphic presents the Accuracy of all forecasts. Right Graphic presents only the forecasting accuracy of the extreme classes, positive and negative.

this subset, this randomization procedure tries to decrease the correlation between the B models. After we grow our B trees $\{T(x;\Theta_g)\}_1^B$ the Random Forests classifier $\hat{T}_{rf}$ is found rearranging the trees output to a vector with $K-1$ zeros and a one inside the cell of the K^{th} predicted class, after that we sum all trees and use the k^{th} biggest cell to make the forecast $\hat{T}_{rf}(x)$:

$$\hat{f}^B_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b) \tag{10}$$

$$\hat{T}_{rf}(x) = argmax_k \hat{f}_{rf}(x) \tag{11}$$

Exists many implementations of this algorithm and in this work we'll use the implementation from Candel et al. (2016).

2.2.4 Bayesian Trees and MPBART

BART (CHIPMAN et al., 2010) and his multinomial version MPBART (KINDO; WANG; PEÑA, 2016) can be seen as a Bayesian version of boosting. We've already told that a tree structure is compound of two parts, the tree splits also known as the tree-structure that define the hyper-rectangles R_j and the vector of parameters in the terminal leafs w_j. Once this two parameters are known the tree is unique and fully specified. So in the usual Bayesian spirit we need to make a prior for our Bayesian-Tree parameters. We'll use the same decomposed prior $p(w,R) = p(w|R)p(R)$ as in Chipman, George e McCulloch (1998) implicit stating that the prior for the tree structure is independent of the leafs.

We can view our classifier as surging from a response latent vector that now we'll call by z_i with dimensions $K-1$, when all elements of this vector became negative we'll forecast class K, and other else if at least one of them is positive we forecast the correspondent class as in (12):

$$y_i(z_i) = \begin{cases} \text{j} & \text{if } \max(z_i) = z_{ij} > 0, \\ \text{K} & \text{if } \max(z_i) < 0. \end{cases} \tag{12}$$

This vector z_i is a function of the regressors variables plus a random noise:

$$z_i = G(X_i;\Theta) + \epsilon, \quad \text{for } i = 1,\ldots,n, \tag{13}$$

$$G(X_i;\Theta) = (G_1(X_i;\Theta_1),...,G_{K-1}(X_i;\Theta_{k-1})) \tag{14}$$

$$\epsilon_i = (\epsilon_{i,1},\ldots,\epsilon_{i,K-1})' \sim N(0,\Sigma), \tag{15}$$

where $\Theta = \{T, M\}$ is the collection of tree structures. Equation (14) is specified as a vector that in each cell have a sum of j classification trees with respective structures $R_{k,j}$ and leafs parameters $w_{k,j} = w_{k,j,1},\ldots,w_{k,j,l}$ for all the $l = 1,\ldots,b$ trees terminal nodes:

$$G_k(X_i;T,M) = \sum_{j=1}^{n_T} g(X_i, R_{kj}, w_{kj}) \tag{16}$$

Where g are the individual regression tree. The parameter estimation as usual is made through a Metropolis Within Gibbs algorithm. For further details look into the work of Kindo, Wang e Peña (2016).

3 Simulation Results

3.1 Minimum Threshold and Optimum Frequency

3.1.1 Frequency Analysis

The sample that we'll use in this first part is composed by one-hundred days going

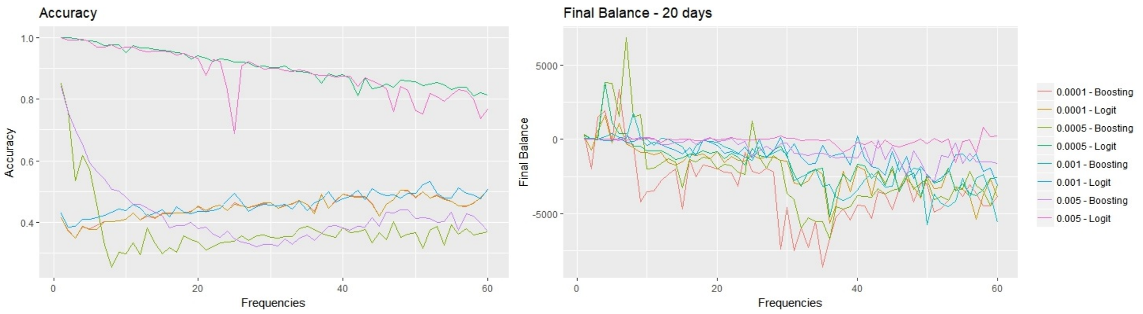

Figure 3 – Forecasting Accuracy - Threshold Analysis

Horizontal axis display the evolution from 1 do 60 minute frequency. Left Graphic presents the Accuracy of all forecasts. Right Graphic presents the Final Balance of the financial simulation

from 26/05/2017 to 17/10/2017, the first 80 days were used to train and the last 20 to test. Were constructed 60 databases with different temporal aggregations, in each successive base we enlarge the sampling frequency by one minute, the most disaggregate basis is of 1 minute and the most aggregate is of 60 minutes, in this first part we'll use the minimum threshold $0,01\%$ for classify a return as different from null. We'll evaluate the forecast quality of all 4 models: Boosting Trees, Random Forests, BART and Logistic regression in all 60 different frequencies.

As we can see in the left graphic into figure 1 as we aggregate more our data-set the models became more accurate. This behavior is natural, as we came from the lower to the biggest frequencies we slowly get out from a 3 classes to a 2 classes problem, in the 60 minutes database is really rare that the price don't change, but in the 1 minute base this is relatively common. Perhaps a more interesting statistics would be verify the number of times that we predict correctly into one of the extreme classes, positive or negative, right graphic at figure 1 presents this statistics. We can see that the accuracy have the highest values in the smallest frequencies reaching a minimum just a little before the 20 minutes, after that the curves slowly raises as we aggregate more our dataset.

When we look to the financial performance we don't really get a clear picture. Figure 5 shows the ending balance of a short financial simulation in the last 20 days. The vertical axis presents the budget in Brazilian Real (R$), we make this simulation with only one stock for getting intuition, the mid price during this 20 days was close to R$15,50.

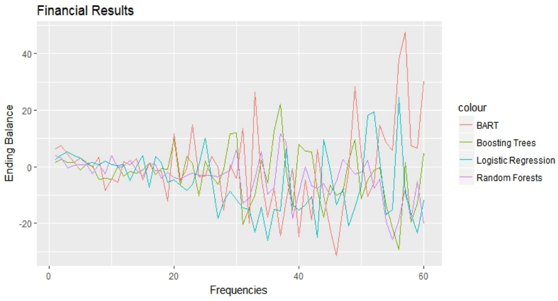

Figure 5 – Financial Simulation

Final Balance of the four models. Horizontal axis presents the frequencies evolution and vertical axis presents the final balance of each model in each time aggregation.

Figure 5 shows to us that the variance of financial results increases dramatically as we start to aggregate more our dataset, starting in the lowest frequencies with strictly positive results going to the biggest frequencies with a variety of final results, both positives and negatives.

3.1.2 Minimum Threshold

Following this first analysis we selected only two models, Boosting and the Logistic Regression, to analyze the impact of a minimum threshold to classify a return as different from null. This selection is simply due to this two models be the faster, allowing us to evaluate a bigger space of different combinations. We classified the returns as Positives, Negatives or Null requiring 0,01%, 0,05%, 0,1%, 0,5% of price oscillation. Right graphic at Figure 3 presents the accuracy of the forecasts for all 60 analyzed frequencies.

The left graphic at figure 3 tells us that the highest levels of accuracy occur in the small-

Table 1 – Forecasting Accuracy - 500 days

Forecasting Accuracy of all three models with respective confidence intervals.

Algorithm	Boosting Trees	Random Forests	B.A.R.T.	Logit
Accuracy	0.4113	0.4135	0.4124	0.3988
C.I. Superior	0.4142	0.4164	0.4152	0.4016
C.I. Inferior	0.4085	0.4107	0.4095	0.396

est aggregations together with the biggest thresholds, as long we aggregate more our dataset less accurate became the forecasts for the biggest threshold. We believe that this behavior is due to in the small windows doesn't occur many situation where the prices oscillates so much. An interesting behavior also occurs with the second greatest threshold 0,1%, he starts the forecasts with a great accuracy but suddenly fall, only starting to recover the forecasting quality after the 20 minutes aggregation. The other two thresholds presents a constantly and improving forecasting behavior as we aggregate more our dataset.

To refine the analysis we again make a short financial simulation with only 20 days to evaluate the forecasting quality, for making the simulation a little different we changed the number of shares for the minimum lot required to trade at B3.SA, 100 shares. The final balance of all models with respective threshold is presented in the right graphic at figure 3, displaying that the unique strategies that would finalize the simulation with positive final balance would be the ones with smallest frequencies, specially going from 4 to 8 minutes. All others frequencies would result in negative balances.

3.2 Financial Simulation

In the last section 3.1 we verified some interesting behaviors, first of all seem to have a great increase in forecasting accuracy as we move for the greatest forecasting windows, but we never see this greater accuracy translate into profits, indeed we verified that only at the smallest frequencies the models achieve positive final balances. We also don't see a great benefit in set a higher threshold for class identification, despite the greatest forecasting performance of the highest threshold in the smallest frequencies.

Now let's verify the forecasting quality five minutes ahead requiring only 0.01% as minimum return for classifying it as different from null, different from the previous analysis we'll expand our test-set. In this simulation our

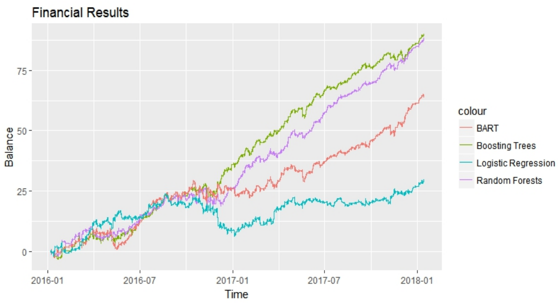

Figure 7 – Financial Simulation

Balance evolution of the trading simulation for all 4 tested models. Horizontal axis presents the time evolution and vertical axis presents the balance.

data-base will start at 2015/12/01 and end at 2018/06/12. We'll make a rolling window simulation with the training set having the fixed number of 100 days and test set having the fixed number of 10 days. How we verified that greatest forecasting horizons seems to have good statistical properties, we'll try to insert their quality into the 5 minutes forecasts, together with the variables sampled in intervals of 5 minutes we will also add variables aggregated in 20 and 60 minutes. This will let us add the notion of acceleration to the model and hopefully achieve better statistical results.

We can see at table 1 that the overall performance of all tree based methods are really similar, with the confidence intervals overlapping. The only model that appears to be significant worst than the others is the logistic regression, in addition to that if we think of ourselves in a 3 category regression we achieved a performance that is superior to the 33,33% baseline. A good way to differentiate all models is their financial result, in figure 7 we present the wallet evolution of a trader that trades only one stock in every entry, again the vertical axis is in Brazilian Real (R$), and the horizontal axis presents the time evolution.

We can verify by figure 7 that all models generates a positive financial result with the

Table 2 – Forecasting Accuracy - 500 days

In the first line we have the average financial return of all four models. In the second line we have this numbers standardized by the returns financial standard deviation.

Algorithm	Boosting Trees	Random Forests	B.A.R.T.	Logit
Avg. Fin. Ret.	0.00084643	0.00080354	0.00056521	0.00025881
Std. Fin. Ret.	0.0314323	0.0301379	0.0212663	0.00971084

Boosting Trees model achieving the highest final balance. Worth call the attention that PETR4.SA was being traded by an average of R$12,79 in our sample period so a final profit of R$70 is a really exciting result. In a similar spirit of the sharp index we divided the average results of this models by their standard deviation, the results are at table 2.

Table 2 shows that the highest average return provided by the boosting trees model is also benefited by a lower risk return proportion. But shaw we call your attention for the negligible average financial return, we would need an average cost of less than 0,001% to make this strategy profitable.

4 Conclusions

We reviewed three state of art ensemble-trees methodologies in this work: Random Forests, Boosting and BART, together with the logistic regression, and looked for their performance when forecasting high-frequency financial signals. We also investigated the impact of the series temporal aggregation and a threshold for classify a return as different from null.

First we verified that as more we aggregate our dataset usually the models start to present better statistical properties, but on the other hand only the smallest frequencies have a positive financial result. In addition we verified that the the forecasting quality gets better and worst in a similar way by all models, indicating that all findings are characteristics of the true DGP.

Secondly we verified that require higher thresholds to classify a return as different from null can generate better statistical properties, but this came with a cost, the lack of predictions in extreme classes. In addition to that in the smallest thresholds usually we improve the forecasting performance as we aggregate more our data, and in the highest thresholds the behavior is the extreme opposite.

By end we verified that all analyzed algorithms achieve a positive forecasting performance in our 500 days simulation, ending the period with profits. But we have to call the attention for the Boosting Trees implementation of Chen e Guestrin (2016), that achieved the best risk return relationship. Despite of that, the average returns of all models would probably be corrupted by the transaction costs, a future problem that will have to be analyzed.

In conclusion, we verified that smallest frequencies and smallest thresholds provide good trading strategies independently of the used model, with special attention for Boosting Trees, but none of the models achieve sufficient profits to compensate the transaction costs. Future work directions include analyze the size of detected movements, insert a risk management strategy like stop rules and analyze variables importance in the typical regression tree way.

Bibliography

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado 2 vezes nas páginas 1 and 3.

BREIMAN, L. et al. *Classification and regression trees.* [S.l.]: CRC press, 1984. Citado 2 vezes nas páginas 1 and 3.

CANDEL, A. et al. Deep learning with h2o. *H2O. ai Inc*, 2016. Citado na página 4.

CASALS, J.; JEREZ, M.; SOTOCA, S. Modelling and forecasting time series sampled at different frequencies. *Journal of Forecasting*, Wiley Online Library, v. 28, n. 4, p. 316–342, 2009. Citado na página 2.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. p. 785–794, 2016. Citado 4 vezes nas páginas 1, 2, 3, and 7.

CHIPMAN, H. A.; GEORGE, E. I.; MCCULLOCH, R. E. Bayesian cart model search. *Journal of the American Statistical Association*, Taylor & Francis Group, v. 93, n. 443, p. 935–948, 1998. Citado na página 4.

CHIPMAN, H. A. et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, Institute of Mathematical Statistics, v. 4, n. 1, p. 266–298, 2010. Citado 2 vezes nas páginas 1 and 4.

DIXON, M. Sequence classification of the limit order book using recurrent neural networks. *Journal of Computational Science*, Elsevier, 2017. Citado na página 2.

DIXON, M. F.; KLABJAN, D.; BANG, J. H. Classification-based financial markets prediction using deep neural networks. 2016. Citado na página 1.

FLETCHER, T.; SHAWE-TAYLOR, J. Multiple kernel learning with fisher kernels for high frequency currency prediction. *Computational Economics*, Springer, v. 42, n. 2, p. 217–240, 2013. Citado 2 vezes nas páginas 1 and 2.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, Elsevier, v. 55, n. 1, p. 119–139, 1997. Citado na página 1.

HAN, J. et al. Machine learning techniques for price change forecast using the limit order book data. *Machine learning*, 2015. Citado na página 2.

KERCHEVAL, A. N.; ZHANG, Y. Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, Taylor & Francis, v. 15, n. 8, p. 1315–1329, 2015. Citado na página 2.

KINDO, B. P.; WANG, H.; PEÑA, E. A. Multinomial probit bayesian additive regression trees. *Stat*, Wiley Online Library, v. 5, n. 1, p. 119–131, 2016. Citado 2 vezes nas páginas 2 and 4.

PERLIN, M.; RAMOS, H. Gethfdata: Ar package for downloading and aggregating high frequency trading data from bovespa. 2016. Citado na página 2.

SCHAPIRE, R. E. *The design and analysis of efficient learning algorithms*. [S.l.], 1991. Citado na página 3.

TSANTEKIDIS, A. et al. Forecasting stock prices from the limit order book using convolutional neural networks. In: IEEE. *Business Informatics (CBI), 2017 IEEE 19th Conference on*. [S.l.], 2017. v. 1, p. 7–12. Citado 2 vezes nas páginas 1 and 2.