

Sibyl - Sistema de recomendações para criptomoedas

Recommended system for cryptocurrencies trade

Teodoro B. Calvo¹, André R. A. Grégio, Dr.²

¹Aluno do programa de Especialização em Data Science Big Data *

² Departamento de Informática, Universidade Federal do Paraná[†]

Uma abordagem de aprendizado de máquina aplicada ao mercado financeiro, mais especificamente, ao mercado de criptomoedas. Neste contexto, buscou-se a construção de um produto viável mínimo (MVP, de *Minimum Viable Product*) para sugestão de investimento. Foram percorridas todas etapas do ciclo analítico: coleta e organização de dados, criação de variáveis explicativas, ajuste de modelos preditivos e por fim, disponibilização da informação predita. Fazendo uso de metodologia ágil, as etapas foram elencadas em cada *sprint* e cumpridas, o que possibilitou rápido desenvolvimento e utilização. Tal atitude trouxe ganhos de conhecimento e retornos financeiros em curto prazo. O desfecho deste trabalho é um protótipo que pode ser usado por operadores em suas negociações, ainda que haja espaço para melhorias incrementais.

Palavras-chave: bitcoin, xgboost, fronteira eficiente

A machine learning approach applied to the financial market, more specifically, to the crypto-coins market. In this context, we sought to develop a minimum viable product (MVP) for investment suggestion. All stages of the analytical cycle were covered: collection and organization of data, creation of explanatory variables, adjustment of predictive models and, finally, availability of predicted information. Using agile methodology, the steps were listed in each sprint and fulfilled, which made possible rapid development and use. Such an attitude has brought gains in knowledge and financial returns quickly. The outcome of this work is a prototype that can be used by traders in their negotiations, although there is incremental improvements possibilities.

Keywords: bitcoin, xgboost, efficient frontier

1. Introdução

Em dezembro de 2017 a cotação do Bitcoin ultrapassou o valor de US\$19.000,00, representando um crescimento anual maior que 2.400%. Porém, no mês seguinte sua cotação estava próxima a US\$11.800,00 (40% de baixa). Logo, se mostra um ativo financeiro volátil e de difícil predição (seja de aumento ou queda de seu valor). Seria possível que especialistas, usuários e entusiastas do mercado de criptomoedas pudessem prever esta súbita valorização? Ou ainda, sua perda de valor nos meses seguintes? A Figura 1 apresenta esta alta repentina e posteriormente uma queda significativa na cotação do Bitcoin.

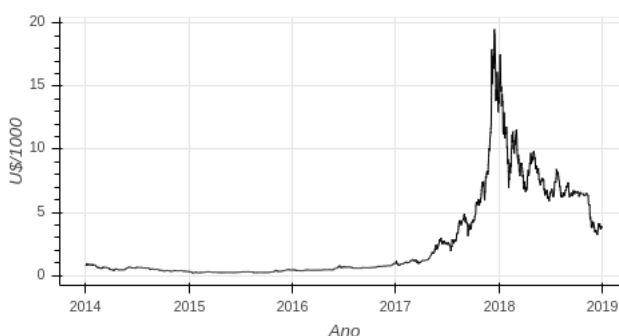


Figura 1: Cotação histórica do Bitcoin entre janeiro de 2014 e dezembro de 2018.

Embora tenha ganho muita atenção no mercado financeiro, o Bitcoin não é o único produto resultante do advento Blockchain. Surgiram inúmeras outras moedas (criptomoedas) para serem comercializadas [1]. Cada uma possui seu propósito, valor e nicho de mercado. Alguns exemplos são: Litecoin, Ethereum e Monero. Ainda com suas particularidades, essas *altcoins*

*teo.bcalvo@gmail.com

[†]gregio@inf.ufpr.br

possuem certa associação com o Bitcoin, uma vez que suas cotações são correlatas, isto é, quando o Bitcoin ganha valor, as *altcoins* tendem a valorizar, no caso de desvalorização, o mesmo ocorre.

Há, então, a possibilidade de não apenas realizar o câmbio entre dólar (ou qualquer outra moeda regulada) e Bitcoin, mas também negociar diversas outras criptomoedas existentes até então. Desta forma, entende-se que é possível gerar lucro a partir de transações neste cenário, de maneira similar ao que acontece tradicionalmente no mercado de câmbio.

A grande dificuldade é prever qual criptomoeda irá valorizar no próximo dia, semana ou mês. É o mesmo problema que acionistas de bolsas de valores e casas de câmbio enfrentam ao realizar seus investimentos. Para resolvê-lo, pode-se fazer uso de técnicas estatísticas e de aprendizado de máquina para auxiliar nos palpites e tomada de decisão.

No presente trabalho, é apresentado um sistema de recomendação de investimento para criptomoedas. Nele, sugere-se quais criptomoedas o investidor deve priorizar em seu portfólio diariamente (*day trade*). Isto é, todos os dias o sistema recomenda uma realocação dos recursos financeiros entre as criptomoedas, a fim de maximizar o retorno.

Nas seções subsequentes apresenta-se os passos realizados durante os estudos para alcançar o objetivo definido. No primeiro momento têm-se a coleta de dados, que possibilita a criação dos indicadores de performance como variáveis explicativas a partir dos dados históricos. Posteriormente tem-se a definição do modelo e seu treinamento, por fim, testes do modelo campeão na série histórica e apresentação dos resultados obtidos.

2. Coleta e armazenamento dos dados

Os dados são obtidos a partir do site CoinMarketCap [2], onde cotações diárias de inúmeras moedas no mercado de criptomoedas são disponibilizadas. Diferentemente da bolsa de valores, as negociações de criptomoedas não cessam ao longo do dia, assim, por convenção, o marco de fechamento e abertura do mercado se dá à zero hora do fuso de UTC. As informações de cotação referentes à abertura e fechamento ficam disponíveis no site momentos após este marco.

O desenvolvimento de um *Web Scraping* [3] foi necessário para obter as informações do site. Por meio de um script *Python*, uma requisição HTTP é realizada, onde é retornado pelo site uma página HTML. Nesta

página há as cotações históricas de determinado período da criptomoeda selecionada. O mesmo script é responsável por identificar todas as *tags* da página, encontrando dados relevantes para serem armazenados. A seguir, as informações são estruturadas e armazenadas em um banco de dados relacional (SQLite)[4]. O que facilita as ações de consultas e inserções corretas de dados, garantindo ainda a tipagem correta dos campos. A figura 2 apresenta o esquema prático de funcionamento da coleta e armazenamento de dados.

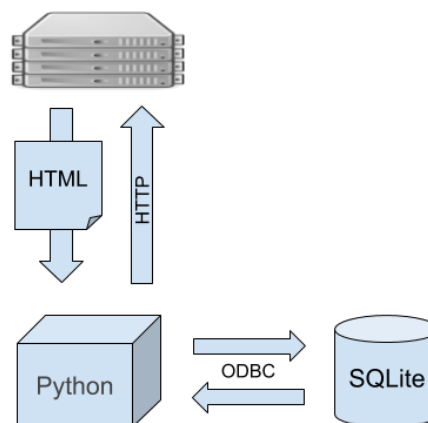


Figura 2: Esquema funcional do processo de coleta de dados diário a partir do script *Python* se comunicando com o servidor da CoinMarketCap, armazenando os dados capturados no banco de dados SQLite.

Diariamente o script criado consulta a última data de cotação que existem em sua base de dados local. Em seguida, é realizada uma varredura no site CoinMarketCap a partir da data encontrada até a presente data. O processo se repete de forma automática para todas as criptomoedas configuradas pelo usuário que deseja realizar a coleta.

3. Criação de variáveis

Com os dados devidamente armazenados, da-se início à etapa de criação de variáveis que auxiliarão a explicar o evento de interesse. No presente trabalho o evento de interesse a ser modelado é o sinal de alta ou queda na cotação de criptomoedas, isto é, um classificador criado para descrever em termos probabilísticos se a cotação de determinada moeda em $d + 1$ será de valorização ou não.

3.1. Indicadores de Performance do Ativo

Entende-se que para a definição de tais variáveis, o conhecimento do tema em questão traria conceitos

importantes, resultando em um melhor ajuste de modelo no futuro, por consequência a tomada de decisão se torna mais segura. Assim, para realizar a tarefa de criação das variáveis explicativas, ou características, buscou-se auxílio de ferramentas econômicas. Tais ferramentas consistem em indicadores que são obtidos à partir da própria série histórica de cotações do ativo financeiro [5].

Dentre os indicadores de performance existentes, o presente trabalho faz uso dos seguintes:

- Retorno;
- Log-Retorno;
- Média móvel;
- Volatilidade;
- Ganhos médios;
- Perdas médias;
- Índice de força relativa.

Ainda que haja inúmeros outros indicadores elegíveis à serem estudados, anseia-se pela obtenção de um produto mínimo viável no menor prazo possível, aplicando conceitos de metodologia ágil [6] neste contexto. Logo, até o momento, limita-se aos indicadores elencados anteriormente. Os demais existentes na literatura servirão de incrementais propostos em trabalhos futuros.

3.1.1. Retorno

O *Retorno* consiste na razão entre o período atual e seu antecessor na série histórica de determinado ativo [7]. No caso da cotação diária, cada dia representa um período, sendo o índice de retorno representado por:

$$R_d = \frac{P_d}{P_{d-1}} \quad (1)$$

Onde P_d é o período (ponto da série) atual e P_{d-1} é o período precedente. Note ainda que para $R_d > 1$, significa valorização em relação aos dois períodos avaliados.

3.1.2. Log-retorno

Ao aplicar o logaritmo neperiano no retorno, obtemos o log-retorno:

$$r_d = \ln\left(\frac{P_d}{P_{d-1}}\right) \quad (2)$$

O conceito é análogo ao retorno, resultando apenas em uma transformação que altera sua escala. É possível que o log-retorno traga melhores ajustes em termos de modelagem do que o próprio retorno. Assim, todos os indicadores apresentados a seguir são calculados a partir do log-retorno.

3.1.3. Média móvel

A média móvel utiliza os $(k - 1)$ pontos anteriores (períodos) e o ponto atual para ser calculada [7]. No presente trabalho a sigla $MA(k)$ representa o indicador de médias móveis, em que o argumento k corresponde a quantidade de pontos que se deseja utilizar em seu cálculo. Então, para calcular a média móvel correspondente ao período d , com uma quantidade de k períodos, têm-se:

$$MA_d(k) = \frac{1}{k} \sum_{i=0}^{k-1} r_{d-i} \quad (3)$$

3.1.4. Volatilidade

O racional para obter a volatilidade é o mesmo que para a média móvel. Embora, esta seja calculada a partir do desvio padrão dos períodos correspondentes. Assim, para obter a volatilidade do período d em n períodos, têm-se:

$$Vol_d(k) = \frac{1}{k-1} \sum_{i=0}^{k-1} [r_{d-i} - MA_d(k)]^2 \quad (4)$$

Note que a volatilidade faz uso da média móvel em seu cálculo. Logo, ambas devem ser calculados sobre a mesma quantidade de períodos, isto é, os mesmos dados históricos da séries são usados para os dois indicadores, dado o período d e $(k - 1)$ períodos anteriores.

3.1.5. Ganho médio

O ganho médio corresponde à média de ganhos nos últimos k períodos da série a partir do d -ésimo período. Ou seja, para $r_d > 0$, tem-se ganho no d -ésimo período, caso contrário o período representa perda. Basta então somar todos os ganhos e dividi-los pela quantidade de períodos observados:

$$G_d(k) = \frac{1}{k} \sum_{i=0}^{k-1} V_{d-i} \quad (5)$$

onde

$$V_{d-i} = \begin{cases} r_{d-i}, & \text{caso } r_{d-i} > 0 \\ 0, & \text{caso contrário} \end{cases}$$

É importante reforçar a ideia de que os valores de $V_{d-i} = 0$ são computados no cálculo da média, impactando assim o valor resultante.

3.1.6. Perda média

A forma de calcular a perda média é similar ao ganho médio. A diferença consiste na estrutura da condição criada:

$$L_d(k) = \frac{1}{k} \sum_{i=0}^{k-1} V_{d-i} \quad (6)$$

onde

$$V_{d-i} = \begin{cases} r_{d-i}, & \text{caso } r_{d-i} < 1 \\ 0, & \text{caso contrário} \end{cases}$$

3.1.7. Índice de força relativa

Os indicadores de ganho médio e perda média trazem consigo informações individuais. O índice de força relativa combina ambos, sintetizando a informação em um único indicador [8]:

$$IFR_d(k) = 100 - \left(\frac{100}{1 + \frac{G_d(k)}{L_d(k)}} \right) \quad (7)$$

A interpretação para este indicador se dá pela intensidade de alta ou queda da série no últimos k dias. O que pode fornecer em um único dado a informação tanto de ganho e perda média. Dito isso, apenas o indicador IFR é considerado para servir de variável explicativa, uma vez que há nele a síntese de informação dos indicadores elencados anteriormente.

3.2. Variáveis de datas

Além das variáveis explicativas criadas a partir da própria série histórica (por meio de indicadores econômicos), pode-se gerar variáveis que dizem a respeito de quando o evento ocorreu. Isto é, o dia da semana pode ajudar a explicar o evento de interesse, bem como qual a semana do ano, por exemplo. Essas informações contribuem explicando possíveis sazonalidades que a série possui, tornando o modelo melhor ajustado aos dados.

De forma, as variáveis a seguir também foram geradas na tentativa de explicar o evento de interesse:

- Dia do ano, valor entre 1 e 366;
- Dia do mês, valor entre 1 e 31;

- Dia da semana, valor entre 1 e 7;
- Semana do ano, valor entre 1 e 53.

Caso haja alguma tendência na série a ser capturada pelo modelo preditivo, é necessário a criação de uma variável de índice para os períodos observados. Isto é, para cada dado da série é atribuído um índice, onde o dado mais antigo recebe o valor 1 (um) e o mais recente n , sendo este o tamanho (dimensão) da série.

Na seção seguinte, é apresentado a estratégia da construção do modelo proposto. Isto é, como combinar todas as variáveis em busca de prever a alta ou baixa das criptomoedas.

4. Equação Generalizada

Ainda antes de propor um classificador, é necessário definir qual a variável que se deseja prever. Como dito em outras seções, deseja-se modelar o sinal de 'alta' ou 'baixa' de criptomoedas. Com isso, uma vez que se tem o log-retorno, é possível criar uma variável binária a partir do mesmo:

$$y_d = \begin{cases} 1, & \text{caso } r_d > 0 \\ 0, & \text{caso contrário} \end{cases}$$

Assim, os valores de r_d maiores que zero, correspondem ao sinal de 'alta', consequentemente, valores abaixo de zero correspondem à 'baixa'.

Entretanto, sob a ótica do investidor, é necessário ter algum indicador financeiro que balize de forma mais adequada a real significância da 'alta'. Ou seja, ainda que haja valorização de determinada criptomoeda, essa valorização supera outros investimentos do mercado? Para isso, é selecionado um critério arbitrário para definir se de fato a alta apresentada pela mesma é significativa ou não. Isto é, a alta de determinada criptomoeda só será definida como 'alta' caso a mesma for maior que determinado *cohort*, caso contrário o valor investido teria um rendimento maior em um instrumento financeiro de menor risco, por exemplo.

Para o sistema desenvolvido foi utilizada a taxa Selic como referência, ainda que seja flexível para o usuário definir seu próprio critério de *cohort*. A taxa Selic está fixada para o ano de 2019 à 6,5% [9], logo, a variável resposta para o modelo é definida por:

$$y_d = \begin{cases} 1, & \text{caso } r_d > \ln\left(\frac{0.065}{365}\right) \\ 0, & \text{caso contrário} \end{cases} \quad (8)$$

Ou seja, só é considerada alta caso a cotação da criptomoeda no dia d for 0,017% maior em relação ao dia $d - 1$, caso contrário será considerada como baixa.

4.1. Parcela auto regressiva

O primeiro vetor de características a ser criado é referente aos dados históricos que antecedem a previsão. Assim, ao fazer uso do log-retorno que já ocorreu no passado em determinado período, pode-se ter certa acuracidade ao utiliza-lo para prever o futuro, permitindo então escrever y_d em função destes dados:

$$\hat{y}_d = f(r_{d-1}, \dots, r_{d-k})$$

Para finalidade de síntese de informação, este vetor de características é representando por:

$$\mathbf{r} = r_{d-1}, \dots, r_{d-k}$$

Obtendo assim uma função do vetor de características \mathbf{r} que busca modelar y_d :

$$\hat{y}_d = f(\mathbf{r}) \quad (9)$$

Embora tal vetor de característica traga consigo informação para explicar a ocorrência de y_d , dificilmente será suficiente para obter uma boa predição. Logo, a criação novos vetores de características se mostra necessária.

4.2. Parcela dos indicadores de performance

Faz-se uso então das variáveis relacionadas à indicadores de performance de ativos, inserindo mais informações na equação 9. Tais variáveis contribuem no modelo com a informação do dia que precede a previsão. Ou seja, para prever y_d os indicadores devem ser calculados para $d - 1$, uma vez que o valor que deve ser predito ainda não foi observado. Assim, considera-se: MA_{d-1} , Vol_{d-1} e IFR_{d-1} .

A adoção do valor para k em cada um dos indicadores é feita de forma arbitrária, mesmo com análises técnicas há muita subjetividade em sua escolha. Isto é, escolher entre $MA_{d-1}(2)$ ou $MA_{d-1}(10)$? Com o intuito de obter um modelo preliminar que explique minimamente bem o evento de interesse, foi criada uma lista de valores para cada indicador, obtendo uma lista de características para cada indicador.

No presente trabalho foram utilizados valores de k entre 2 e 40, havendo a compreensão de que valores tão antigos não trariam tanta contribuição para o modelo. Ou seja, no caso das médias móveis, serão calculadas 40 médias móveis, a primeira, utilizando 2 pontos históricos: r_{d-1} e r_{d-2} . Logo, último cálculo das média móveis faz uso de 40 pontos históricos, isto é, r_{d-1}, \dots, r_{d-40} . Para os outros indicadores o mesmo ocorre.

Por fim, têm-se três vetores de características, um para cada indicador:

$$\mathbf{u} = [MA_{d-1}(2), \dots, MA_{d-1}(40)]$$

$$\mathbf{v} = [Vol_{d-1}(2), \dots, Vol_{d-1}(40)]$$

$$\mathbf{w} = [IFR_{d-1}(2), \dots, IFR_{d-1}(40)]$$

Ao combinar tais vetores com a equação 9, a mesma pode ser reescrita de seguinte forma:

$$\hat{y}_d = f(\mathbf{r}, \mathbf{u}, \mathbf{v}, \mathbf{w})$$

4.3. Parcela de datas

Buscando capturar possíveis sazonalidades apresentadas na série, faz-se uso de variáveis referentes à data em que a observação ocorreu. Tais variáveis são incorporadas no modelo por meio do vetor \mathbf{t} apresentado a seguir:

$$\mathbf{t} = \begin{bmatrix} \text{dia do ano} \\ \text{dia do mês} \\ \text{dia da semana} \\ \text{semana do ano} \end{bmatrix}^t$$

E ainda, o índice d que carrega consigo a informação de qual é a posição da informação a ser predita na série histórica. Tem-se então:

$$\hat{y}_d = f(\mathbf{r}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{t}, d)$$

4.4. Parcela de marcação de moedas

Dado que se deseja realizar a previsão para um conjunto de m moedas, é necessário que a base de treinamento do modelo tenha a marcação de qual moeda aquele dado diz respeito, uma vez que o treinamento será realizado considerando os dados históricos de todas moedas. Têm-se então o último vetor de características:

$$\mathbf{c} = [c_1, \dots, c_m]$$

onde a moeda correspondente à observação é representada por 1 e as demais como 0. Então $\sum_{i=1}^m c_i = 1$,

pois neste vetor só pode haver uma única moeda a ser representada por observação. Por fim, pode-se escrever y_d em função de todas as variáveis criadas a partir de:

$$\hat{y}_d = f(\mathbf{r}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{t}, \mathbf{d}, \mathbf{c}) \quad (10)$$

5. Ajuste e performance do modelo

Nesta seção apresenta-se a estratégia adotada para ajustar um algoritmo de aprendizado de máquina aos dados, bem como os resultados obtidos a partir do modelo campeão. Dentre a gama existente de algoritmos elegíveis, no presente trabalho fez-se uso do XGBoost [10]. Tal modelo tem se destacado em inúmeras premiações no site de competições de aprendizado de máquina Kaggle [11]. Havendo ainda, relatos de performances satisfatórias ao aplicá-lo para prever o comportamento da cotação do bitcoin usando as redes sociais [12], comparando-o com outros modelos da literatura.

5.1. Grid search

Em busca de obter o melhor ajuste de modelo possível com os dados coletados, utilizou-se o método de *grid search* (busca de grade) para otimização dos hiperparâmetros do XGBoost. Tal método realiza o ajuste do modelo para todas possíveis combinações dos hiperparâmetros que o usuário define, logo, acaba sendo extremamente exaustivo.

Considere ainda o uso de validação cruzada [13] ($k = 5$) em cada uma das iterações do método. A área sobre a curva ROC (AUC, *area under curve*) [13] foi adotada como medida de sucesso para escolha dos hiperparâmetros. Ou seja, para cada possível combinação dos hiperparâmetros foram treinados 5 modelos sobre diferentes partições da base de dados, onde se calcula a AUC na partição que não participou do treinamento (base de teste). A média desta medida serve de comparação com a próxima iteração do *grid search*. Assim, o melhor conjunto de hiperparâmetros será aquele que possui o maior valor médio de AUC nas bases de teste. O método realizou sua busca pelos melhores valores dos seguintes hiperparâmetros:

- Profundidade máxima;
- Taxa de aprendizado;
- Proporção de reamostragem;
- Quantidade de iterações;

Os dados de treinamento se limitaram entre as datas 28 de abril de 2013 e 09 de fevereiro de 2019, correspondendo à aproximadamente 20.000 linhas, sendo referentes à 19 criptomoedas distintas e seus históricos. Tais moedas foram consideradas a partir do ranking apresentado no site CoinMarketCap. Após a otimização realizada, o valor médio mais alto encontrado para AUC na base de teste foi igual 0,7032 com os seguintes valores de hiperparâmetros:

- Profundidade máxima = 16;
- Taxa de aprendizado = 0,001;
- Proporção de reamostragem = 0,1;
- Quantidade de iterações = 2.500;

Para a quantidade de iterações, entende-se que há possibilidades de melhoria, isto é, aumentar a quantidade de iterações pode trazer resultados melhores de ajuste. Porém, o esforço computacional necessário a ser desempenhado é muito alto, o que impacta no tempo de treinamento do modelo. É necessário então realizar um balanceamento entre tempo de treinamento e performance, buscando um equilíbrio entre ambos.

A figura 3 apresenta a evolução do desempenho do modelo ajustado para ambas bases (treino e teste) conforme o número de iterações aumenta. Onde o ganho de performance acaba estabilizando, sendo necessário um esforço muito maior para incrementá-lo.

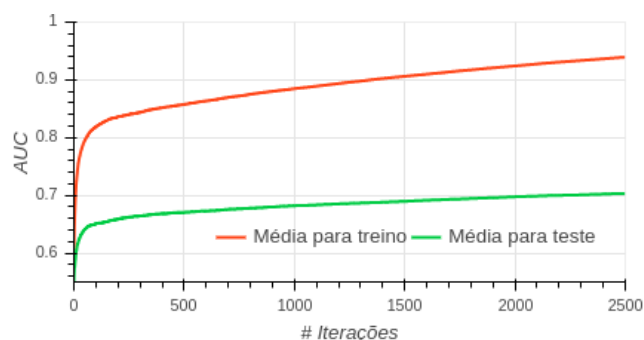


Figura 3: Relação entre desempenho do modelo pela área sob a curva ROC e a quantidade de iterações para treinamento.

6. Fronteira eficiente - Markowitz

As seções anteriores demonstraram a estratégia de ajuste do modelo preditivo bem como suas métricas de performance. Assim, o último passo necessário para cumprir o objetivo de sugerir a distribuição ideal de recurso entre os investimentos (moedas), consiste em análise de carteira.

Faz-se uso então da técnica de fronteira eficiente, formalizada matematicamente por Harry Markowitz. Em que a mesma consiste em minimizar os riscos com base na variância dos ativos em sua séries histórica, podendo ser expressa de forma matemática:

$$\hat{w} = \arg \min_w w^t \Sigma w \quad (11)$$

onde Σ é a matriz de covariância das moedas e w é um vetor de pesos com tamanho n -moedas. Considerando ainda as restrições:

$$\sum_{i=1}^n w_i = 1, \\ 0 \leq w_i \leq 1, i = 1, \dots, n$$

Em português, isso significa encontrar o vetor \hat{w} que minimiza a somas das variâncias de todas moedas. Logo, ao encontrar \hat{w} se tem a proporção ideal para distribuição do investimento entre as moedas. No presente trabalho o modelo preditivo é utilizado para filtrar as moedas com baixa probabilidade de alta. Assim, para o cálculo da fronteira eficiente são consideradas apenas as moedas que tem chance considerável de alta (segundo o modelo preditivo), realizando a sugestão de quanto deve ser investido diariamente em cada uma das moedas elencadas.

7. Disponibilidade e uso

Como protótipo, foi desenvolvido um *bot* no aplicativo *Telegram* para viabilizar o consumo das informações geradas. Onde, diariamente as etapas elencadas anteriormente são executadas. Para integração, foi utilizado a biblioteca *Telegram Bot API* da linguagem de programação *Python*, facilitando a implementação.

Basta que o usuário envie uma mensagem de texto para receber o *ouput* do modelo. As informações são referentes às chances de alta de cada moeda e a proporção de investimento sugerida para cada uma.

8. Considerações finais e próximos passos

Encerra-se este trabalho com um produto mínimo viável, cumprindo todo o ciclo analítico onde está inserido, bem como na disponibilização da informação gerada. Logo, o objetivo de desenvolver um sistema de recomendação de criptomoedas foi atingido, ainda que haja oportunidade de melhorias.

Embora a performance do modelo preditivo seja aceitável, entende-se que são necessários mais esfor-

ços para criação e seleção de variáveis para torná-lo mais acurado. Buscar outros algoritmos de classificação pode trazer resultados mais interessantes, possibilitando desenvolvimento em trabalhos futuros.

Agradecimentos

Agradeço ao meu orientador, pois permitiu que eu tivesse a autonomia necessária para desenvolvimento do trabalho, bem como na escrita deste texto, obrigado Dr. André Grégio. Minha querida esposa, Natália, obrigado pelo ânimo e sorriso ao encarar os desafios que constantemente trago para nós.

Referências

- [1] Crypto Tech Academy, *Cryptocurrency: A Complete Beginners Guide to Cryptocurrencies: Cryptocurrency Mining and Cryptocurrency Trading*, (Dibbly Publishing, 2018)
- [2] CoinMarketCap. Disponível em <www.coinmarketcap.com> Acesso em: 18 de março de 2019
- [3] Mitchell, R., *Web Scraping with Python*, (O'Reilly Media, Inc., 2015)
- [4] Nield, T., *Getting Started with SQL*, (O'Reilly Media, Inc., 2016)
- [5] Lemos, E., *Análise técnica dos mercados financeiros: um guia completo e definitivo dos métodos de negociação de ativos*, (Saraiva Educação, São Paulo, 2015)
- [6] Sutherland, J. e Sutherland, J., J., *SCRUM: A arte de fazer o dobro do trabalho na metade do tempo*, (Sextante, 2019)
- [7] Morettin, P. A. e Toloi, C. M. C., *Análise de Séries Temporais*, (Blucher, São Paulo, 2006)
- [8] Wilder, J.W., *New Concepts in Technical Trading Systems*, (Trend Research, 1978)
- [9] Valor Econômico, *Sob liderança de Campos Neto, Copom abre primeiro dia de reunião*, Disponível em <https://www.valor.com.br/financas/6168143/sob-lideranca-de-campos-neto-copom-abre-primeiro-dia-de-reuniao> Acesso em 19 de março de 2019
- [10] Extreme Gradient Boosting, *XGBoost Documentation*, Disponível em <https://xgboost.readthedocs.io> Acesso em 06 de abril de 2019
- [11] Kaggle, Disponível em <https://www.kaggle.com> Acesso em 06 de abril de 2019
- [12] Gillett, J., *Predicting Bitcoin: a robust model for predicting Bitcoin price directions based on network influencers*, (UOIT, 2016)
- [13] Faceli, K., et al., *Inteligência artificial: Uma abordagem de aprendizado de máquina*, (LTC, Rio de Janeiro, 2015)