



## Trabalho Final - Módulo I

Ultimamente, o mercado de criptomoedas tem atraído diversos investidores ao redor do mundo. Neste contexto, você deverá desenvolver um modelo de previsão do preço de fechamento de alguma criptomoeda<sup>1</sup>, utilizando uma rede neural multicamadas (MLP - “*Multi Layer Perceptron*”), ou algum outro modelo abaixo, caso sintá-se confortável.

[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html)

<https://scikit-learn.org/stable/modules/svm.html#regression>

<https://scikit-learn.org/stable/modules/sgd.html#regression>

<https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-regression>

<https://scikit-learn.org/stable/modules/tree.html#regression>

<https://scikit-learn.org/stable/modules/ensemble.html#voting-regressor>

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

[https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

Para desenvolver este modelo você deverá:

1. Utilizar o dataset disponível em: <https://www.cryptodatadownload.com>, como por exemplo: <https://www.cryptodatadownload.com/data/poloniex/>
2. Crie módulos Python reutilizáveis (exemplo: `data_load.py`, `features.py`, `models.py`, etc.) e inclua ***docstrings*** e ***type hints*** em todas as funções. Adicione tratamento de erros com o módulo ***logging***.
3. Desenvolver as seguintes análises estatísticas nos dados de 10 criptomoedas:
  - a. Obter medidas resumo e medidas de dispersão
  - b. Construir boxplot e/ou histograma do preço de fechamento

---

<sup>1</sup> BTC, ETH, LTC, XRP, BCH, XMR, DASH, ETC, BAT, ZRX, EOS, LSK, REP. Comportamento de cada criptomoeda pode ser observado em: <https://coinmarketcap.com/>



- c. Analisar a variabilidade entre as criptomoedas com base nas medidas de dispersão
  - d. Construir gráfico de linha com o preço de fechamento destacando a média, mediana e moda ao longo do tempo
4. Crie um *script* configurável (main.py) que possa ser executado em linha de comando (CLI) usando **argparse**. Exemplo de parâmetros: --crypto, --model, --kfolds, etc.
5. Adicione uma pasta **tests/** contendo ao menos três casos de teste automatizados para serem executados usando **pytest**. Execute os testes automatizados gerando relatórios de cobertura (**pytest-cov**).
6. Pesquisar e escolher variáveis (*features*) de entrada para a sua rede. Você pode usar dados externos (Ex: Relação da moeda com dólar, com real, indicadores macro-econômicos, etc.), ou dados da própria série (média dos últimos 7 dias, desvio padrão dos últimos 7 dias, correlação entre as moedas, etc.). Ou seja, você deverá encontrar as melhores *features* que irá considerar para o seu modelo.
7. Aplicar em seu treinamento a estratégia de validação K-fold cross validation.
8. Sempre que possível, utilize operações vetorizadas (np.where, np.cumprod, np.roll, etc) em vez de laços explícitos para acelerar cálculos estatísticos.
9. Computar o lucro obtido com seu modelo, caso tenha investido U\$ 1,000.00 no primeiro dia de operação, refazendo investimentos de todo o saldo acumulado diariamente, caso a previsão do valor de fechamento do próximo dia seja superior ao do dia atual. Comparar seu modelo MLP com modelo de regressão (linear, e polinomial - graus 2 até 10):
  - a. Diagrama de dispersão (com todos modelos)
  - b. Definir os coeficientes de correlação dos regressores
  - c. Determinar a equação que melhor representa os regressores
  - d. Cálculo do erro padrão
  - e. Cálculo do erro padrão **entre** o MLP e o melhor regressor



- f. Plotar um gráfico mostrando a evolução do lucro obtido em cada modelo
10. Crie uma função em Python que, com nível de significância de 5%, construa um teste de hipótese analisando se o retorno esperado médio será superior ou igual à  $x\%$  (a ser definido pelo usuário) baseado na amostra que você utilizou. Realize esta análise para todas as criptomoedas do dataset.
11. Realize análises de variância (ANOVA) para comparar os retornos médios diários das criptomoedas.
- Aplique ANOVA para verificar se o retorno médio diário difere entre as criptomoedas analisadas. Caso o resultado seja significativo, realize um teste post hoc para identificar quais moedas diferem entre si.
  - Agrupe as criptomoedas com base em alguma característica comum (ex: volatilidade, volume médio negociado, ou retorno médio) e aplique ANOVA para verificar se o retorno médio diário difere entre os grupos formados. Caso o resultado seja significativo, realize um teste post hoc.
12. Todos os gráficos devem ser gerados com **matplotlib** ou **seaborn**, salvos em **figures/** com resolução mínima de 150 dpi. Use *subplots* quando houver comparações entre modelos.

### **Boas práticas de código**

- Utilize ferramentas de formatação de código e lint (black, ruff, pylint flake8, etc.)
- Inclua **requirements.txt** e **README.md** com instruções de execução passo a passo.

### **Orientações gerais**

- Grupos de no **máximo** 3 alunos
- Entrega até o dia 10/Jul/2025
- O código deverá ser disponibilizado e compartilhado no Google Colab ([noronha@ifg.edu.br](mailto:noronha@ifg.edu.br), [otavio.xavier@ifg.edu.br](mailto:otavio.xavier@ifg.edu.br) e [eder.brito@ifg.edu.br](mailto:eder.brito@ifg.edu.br))