

---

## Inteligência Computacional

---

### **Docente**

Carlos Pereira  
Inês Domingues

### **Alunos**

Paulo Henrique Figueira Pestana de Gouveia - a2020121705  
Nuno Alexandre Almeida Santos - a2019110035

October , 2022

# Índice

Tema .....	3
Dataset .....	3
Objetivo .....	4
Sequência .....	4
Métricas .....	4
Network's .....	5
Análise de Resultados .....	7
Conclusões .....	8

## Tema

O tema escolhido foi Sustainable Economy, dentro deste tema iremos trabalhar sobre a Bitcoin.

A Bitcoin foi criada para ser usada para mandar dinheiro pela internet. Esta moeda digital tem o intuito de fornecer uma maneira alternativa de pagamento sem o controlo central ou de alguma supervisão de um banco ou governo tal como ocorre nas moedas tradicionais. Para muito gente é visto como um futuro método de pagamento.

Depende de programas “peer-to-peer” e de cryptography. A cryptography por detrás da bitcoin é baseado no algoritmo SHA-256 designado pela “US US National Security Agency. Muito seguro, pois existem mais “private keys” do que existem átomos no universo, o que para todos os efeitos torna impossível enfraquecê-lo.

## Dataset

O Dataset escolhido foi [Bitcoin Price USD](#), neste conjunto de dados os dados são gerados no intervalo de 1 minuto por uma API (Binance API\*) entre 1 de janeiro de 2021 a 12 de Maio de 2021.

Inclui várias colunas que mostram a mudança real no preço da Bitcoin também mostra o preço Open, High, Low, Close da Bitcoin em minutos específicos. O Open Time e o Close Time no conjunto de dados estão em Unix Timestamp.

### 📌 Features:

- #1 – Preço de abertura num minuto específico (Open Price of particular minute);
- #2 – Preço alto num minuto específico (High Price of particular minute);
- #3 – Preço baixo num minuto específico (Low Price of particular minute);
- #4 – Fechar Preço num minuto específico (Close Price of particular minute);
- #5 – Volume total num minuto específico (Total volume of particular minute);
- #6 – Volume de ativos de cotação (Quote asset volume);
- #7 – Número de negócios para determinado minuto (Number of trades for particular minute);
- #8 – Volume de ativos base de compra do tomador (Taker buy base asset volume);
- #9 – Volume de ativos de cotação de compra do tomador (Taker buy quote asset volume).

🔗 Exemplos: 188318

\*A API da Binance é um método que permite conectar-se aos servidores da Binance via Python ou várias outras linguagens de programação. Mais especificamente, a Binance possui uma API RESTful que usa solicitações HTTP para enviar e receber dados. Além disso, há também um WebSocket disponível que permite o streaming de dados, como cotações de preços e atualizações de contas.

## Objetivo

Deparamo-nos com um problema de Regressão em que o nosso objetivo é treinar uma rede neuronal capaz de estimar o valor da Bitcoin ao minuto.

Sabemos que não iremos conseguir estimar o valor exato da Bitcoin pois existem fatores externos que não conseguimos controlar nem prever (Ex.: Elon Musk e outros influenciadores), no entanto o nosso objetivo é ficar o mais próximo possível.

Visto que a nossa rede será treinada com exemplos de 1/01/2021 a 12/05/2021, após a rede treinada iremos comparar os nossos valores obtidos com os valores reais nos dias seguintes ao último dia em que foi treinada a nossa rede.

## Sequência

Dataset  
Network  
Main  
Prediction

## Métricas

As métricas utilizadas para avaliar o nosso modelo foram:

1. MAE (Erro quadrado médio)
2. MSE (Erro absoluto médio)
3. RMSE (Raíz quadrada do erro médio)
4. Rsquared (Coeficiente de determinação)

## Network's

Foram criadas 4 redes, cada uma com uma função de treino diferente e foi escolhida a melhor rede para efetuar várias alterações nos parâmetros tais como, função de ativação, número de neurónios e número de camadas.

### 1. Rede MSE:

- ☐ Nome - network\_MSE\_2Lay\_256\_Tanh\_8\_94e\_05\_.tar
- ☐ Função de Treino - MSE
- ☐ Função de Ativação - Tanh
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 8,94E-05

### 2. Rede RMSE:

- ☐ Nome - network\_RMSE\_2Lay\_256\_Tanh\_0\_00228\_.tar
- ☐ Função de Treino - RMSE
- ☐ Função de Ativação - Tanh
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,0028

### 3. Rede MAE:

- ☐ Nome - network\_MAE\_2Lay\_256\_Tanh\_0\_00173\_.tar
- ☐ Função de Treino - MAE
- ☐ Função de Ativação - Tanh
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,00173

#### 4. Rede RSQUARED:

- ☐ Nome - network\_R2\_2Lay\_256\_Tanh\_0\_062.tar
- ☐ Função de Treino - RSquared
- ☐ Função de Ativação - Tanh
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,062

A melhor rede obtida na primeira fase, alterando apenas as funções de treino foi a rede network MSE 2Lay 256 Tanh 8 94e 05.tar (Rede MSE) cujo a função de treino foi a MSE (Erro quadrado médio).

Foram criadas novas 5 redes cujo a função de treino é a função MSE pois foi a função que obtivemos melhor resultados, nestas novas 5 redes irão variar vários parâmetros tais como a função de ativação, número de neurónios e número de camadas.

#### 1. Rede MSE (Softmax para função de ativação):

- ☐ Nome - network MSE\_2Lay\_256\_Softmax\_0\_065.tar
- ☐ Função de Treino - MSE
- ☐ Função de Ativação - Softmax
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,065

#### 2. Rede MSE(RReLU para função de ativação):

- ☐ Nome - network MSE\_2Lay\_256\_RReLU\_0\_0019.tar
- ☐ Função de Treino - MSE
- ☐ Função de Ativação - RReLU
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,0019

#### 3. Rede MSE(Sigmoid para função de ativação):

- ☐ Nome - network\_MAE\_2Lay\_256\_Sigmoid\_0\_00173.tar
- ☐ Função de Treino - MSE
- ☐ Função de Ativação - Sigmoid
- ☐ N° de neuronios - 256
- ☐ N° de camadas - 2
- ☐ Learning rate - 3e-4
- ☐ Erro de teste - 0,00173

#### 4. Rede MSE(com uma camada):

- Nome - network\_MSE\_1Lay\_256\_Tanh\_9\_04E\_05\_.tar
- Função de Treino - MSE
- Função de Ativação - Tanh
- Nº de neuronios - 256
- Nº de camadas - 1
- Learning rate - 3e-4
- Erro de teste - 9,04E-05

#### 5. Rede MSE(com 50 neuronios):

- Nome - network\_MSE\_2Lay\_50\_Tanh\_0\_062\_.tar
- Função de Treino - MSE
- Função de Ativação - Tanh
- Nº de neuronios - 50
- Nº de camadas - 2
- Learning rate - 3e-4
- Erro de teste - 0,062

## Análise de Resultados

### *Diferença nas funções de ativação(métricas de avaliação)*

MSE é uma função diferenciável que facilita a execução de operações matemáticas em comparação com uma função não diferenciável como MAE.

Portanto, em muitos modelos, o RMSE é usado como métrica padrão para calcular a Função de Perda, apesar de ser mais difícil de interpretar do que o MAE.

O menor valor de MAE, MSE e RMSE implica em maior precisão de um modelo de regressão. No entanto, um valor mais alto de RSquared é considerado desejável.

Para comparar a precisão entre diferentes modelos de regressão linear, RMSE é uma escolha melhor do que R Squared.

### *Diferença nas funções de ativação*

Como podemos visualizar nos resultados acima, o função Tanh continua a ter a melhor performance, isto tudo tem haver da maneira como foram normalizados os dados.

Os nossos dados estão normalizados entre -1 e 1, sendo que a Softmax e a Sigmoid limitam o output entre 0 e 1, o que não ajuda para a performance. Para RReLU aplica a função de unidade de revestimento retificado com vazamento aleatório, elemento a elemento, podendo obter melhor resultados se o seu

lower e upperbound foram bem definidos.

### ***Diferença nas configurações da rede***

Sendo um modelo linear, precisamos apenas de uma camada para rede, sendo que o aumento de camadas não afeta a performance da rede que é o que acontece nos nossos testes.

Enquanto que na diminuição da quantidade de neurónios podemos ver que a nossa performance da rede piora, pois que quantos mais neurónios para a nossa rede melhor ela será ao custo de um aumento do tempo de treino.

### **Conclusões**

Podemos concluir que para este caso a melhor métrica de avaliação é a MSE com a função de ativação adequada para a maneira como foram normalizados os dados.

Ao testar com valores esta rede dá valores favoráveis, que infelizmente não podem ser levados para uso de ganhos financeiros no mercado da criptomoeda devido ao facto que a nossa rede não tem dados para prever fatores externos que possam causar por exemplo um crash no mercado.

Combinamos o interesse ao tópico das criptomoedas hoje em dia, para aprofundar a nossa compreensão no tópico de criar redes neuronais para previsão de valores de um modelo de regressão.