

Inteligência Computacional

**Docente**

Carlos Pereira

Inês Domingues

**Alunos**

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

October 13, 2022

Índice

[Tema 3](#_Toc119278949)

[Dataset 3](#_Toc119278950)

[Objetivo 4](#_Toc119278951)

[Sequência 4](#_Toc119278952)

[Métricas 4](#_Toc119278953)

[Network´s 5](#_Toc119278954)

[Análise de Resultados 6](#_Toc119278955)

[Conclusões 6](#_Toc119278956)

# Tema

O tema escolhido foi Sustainalbe 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](https://www.kaggle.com/datasets/aakashverma8900/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

# 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\_Linear\_8\_94e\_05\_.tar
* Função de Treino - MSE
* Função de Ativação - Linear
* Nº de neurónios - 256
* Nº de camadas - 2
* Learning rate - 3e-4
* Erro de teste - 8,94E-05

1. **Rede RMSE:**

* Nome - network\_RMSE\_2Lay\_256\_Linear\_0\_00228\_.tar
* Função de Treino - RMSE
* Função de Ativação - Linear
* Nº de neurónios - 256
* Nº de camadas - 2
* Learning rate - 3e-4
* Erro de teste - 0,0028

1. **Rede MAE:**

* Nome - network\_MAE\_2Lay\_256\_Linear\_0\_00173\_.tar
* Função de Treino - MAE
* Função de Ativação - Linear
* Nº de neurónios - 256
* Nº de camadas - 2
* Learning rate - 3e-4
* Erro de teste - 0,00173

1. **Rede RSQUARED:**

* Nome - network\_R2\_2Lay\_256\_Linear\_0\_062\_.tar
* Função de Treino - RSquared
* Função de Ativação - Linear
* Nº de neurónios - 256
* Nº de camadas - 2
* Learning rate - 3e-4
* Erro de teste - 0,062

# Análise de Resultados

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

Foram criadas 3 redes cujo a função de treino é a função MSE pois foi a função que obtivemos melhor resultados, nestas novas 3 redes irão variar vários parâmetros tais como a função de ativação, número de neurónios e número de camadas.

# Conclusões