

# Redes Neurais e Deep Learning aplicado no escopo econômico de colecionáveis digitais

Vinícius da Silva Dias  
Bacharelado em ciência e tecnologia  
UNIFESP, Campus ICT  
São José dos campos, Brasil  
[dias.vinicius@unifesp.br](mailto:dias.vinicius@unifesp.br)

Mateus da Silva Romani  
Bacharelado em ciência e tecnologia  
UNIFESP, Campus ICT  
São José dos campos, Brasil  
[mateus.romani@unifesp.br](mailto:mateus.romani@unifesp.br)

## I. INTRODUÇÃO E MOTIVAÇÃO

Atualmente a indústria de jogos eletrônicos competitivos vem crescendo de maneira exponencial, e como esperado, o mesmo acontece com seus mercados externos e internos, desde venda de colecionáveis e cosméticos à campeonatos profissionais que movimentam um mercado milionário anualmente. Com isso, ao analisarmos jogos e seus mercados internos foi possível notar um grande movimento financeiro ao redor de colecionáveis digitais ou Skins como são comumente chamados pela comunidade, do jogo Counter Strike: Global Offensive, um FPS (First Person Shooter) competitivo online desenvolvido pela empresa Valve Enterprise. O setor econômico do CS:GO se encontra em constante crescimento possuindo itens avaliados em mais de meio milhão de reais, tendo assim um ótimo potencial lucrativo.

Contudo, há uma lacuna no aproveitamento da grande gama de dados gerada pelo mercado em constante expansão. Sendo assim, temos em mão um grande potencial de desenvolvimento de projetos utilizando inteligência artificial para organização e compreensão dos dados.

Ao implementar tecnologias avançadas, como redes neurais, é possível explorar e aproveitar esses dados a fim de fornecer benefícios tanto para os usuários quanto para o mercado de skins por si só. Portanto, uma motivação para esse projeto seria desenvolver um sistema de predição de valores das skins com base em suas características, e potencializar a efetividade do modelo através de deep learning.

Com uma IA treinada em deep learning, seria possível analisar um conjunto de dados com características das skins, como raridade, popularidade e desgaste, e utilizá-las para prevermos valores do mercado referentes aos itens e suas características. Isso portanto permitiria aos usuários tomarem decisões mais informadas tanto ao comprar, vender ou trocar suas skins, maximizando assim os lucros e minimizando os riscos.

Além disso, como poderemos ver ao decorrer do relatório, o sistema de predição de valores poderia fornecer insights valiosos em relação às tendências de mercado, flutuações de preços e comportamento dos consumidores. Isso seria especialmente útil para investidores e negociantes do mercado, tais como colecionadores e lojas de revenda, permitindo-lhes tomar decisões estratégicas com base em dados confiáveis e atualizados.

Em resumo, um projeto de redes neurais com deep learning para predição de skins de CS:GO pode aproveitar o vasto potencial de dados gerados por esse mercado, fornecendo previsões precisas que beneficiam tanto os usuários quanto o próprio mercado, a fim de auxiliar na tomada de decisões mais precisas e lucrativas.

## II. CONCEITOS FUNDAMENTAIS

Outros modelos de IA foram levados em consideração, como Regressão Linear Múltipla (MLR) e Redes Neurais Artificiais. A Regressão Linear foi o ponto inicial de pesquisa por se tratar de um modelo de fácil interpretação e menor complexidade de implementação quando comparado aos modelos posteriores. Entretanto, após pesquisar mais a fundo descobriu-se que existem modelos que lidam melhor com predição de preços, como a Rede Neural Artificial (ou ANN). (Lim et al., 2016) concluíram que modelos ANN tiveram bom desempenho na predição de preços de casas em comparação à Análise de Regressão Múltipla (MRA). Ainda, (De Faria et al., 2009) fez uso de ANN para predizer a direção do principal indexador do mercado financeiro brasileiro.

Para a tarefa alvo escolhida dois modelos foram levados em consideração: MLP (Multi Layer Perceptron) e Regressão Linear. A MLP foi feita com três camadas (uma de entrada, uma camada escondida e a camada final com a saída da rede). A cama inicial recebe os 7 atributos escolhidos e fornece um total de 64 entradas para a camada escondida que, por sua vez, fornece 128 entradas para a última camada. A regressão linear foi feita utilizando os 7 atributos de entrada e fornecendo 1 saída apenas (valor predito).

A avaliação dos dois modelos escolhidos foi feita através de: Mean Absolute Error (Erro Absoluto Médio, MAE), RMSE (Raiz Quadrada do Erro Médio, RMSE) e o Coeficiente de Determinação. O MAE é uma medida muito utilizada em rotinas de regressão linear pois mede a média diferença entre os valores preditos e os valores reais. Vale notar que a medida do MAE é a mesma medida da variável de análise e que quanto menor o valor encontrado, melhor é o modelo. O RMSE, por sua vez, é uma métrica que mede não só a distância entre as predições e os valores reais como também a direção e a magnitude dos erros.

Por último tem-se o Coeficiente de Determinação (chamado também de "R quadrado"), que é responsável por representar a proporção da variância entre as variáveis dependentes e independentes de um dado problema (neste caso, a relação entre os atributos dos itens e seu preço). Um R-quadrado perto de 1 indica que o modelo está bem ajustado, enquanto um R-quadrado próximo de zero indica erros na implementação. Entretanto, a análise do R-quadrado deve ser feita com muito cuidado pois valores extremamente próximos de 1 podem indicar overfitting, ou seja, o modelo tem bom desempenho na etapa de treinamento porém não foi generalizado o suficiente para ter bom desempenho no conjunto de teste.

O teste de diferença utilizando foi o de Cross Validation (Validação Cruzada), pois é uma poderosa técnica utilizada para avaliar a performance de um modelo de aprendizagem de máquina. A técnica consiste em dividir o conjunto de dados em múltiplas sub-partes (chamadas de "folds"). Então, o

modelo é treinado com uma porção (chamada de "treinamento") e depois avaliada no conjunto restante (chamado de "teste"), sendo este processo realizado repetidas vezes, utilizando diferentes porções do conjunto de dados. O método de Cross Validation é uma boa técnica a ser usada pois ajuda o pesquisador a analisar a capacidade de generalização de um modelo, além do impacto da variabilidade dos dados no resultado final.

Após realizarmos testes com Cross Validation e MAE foi destacado pelo professor orientador que a abordagem via MAPE seria mais precisa uma vez que envolveria porcentagens da média, destacando novos horizontes de possível melhoria e adequação dos dados e do modelo.

### III. TRABALHOS RELACIONADOS

Outros modelos de IA foram levados em consideração, como Regressão Linear Múltipla (MLR) e Redes Neurais Artificiais. A Regressão Linear foi o ponto inicial de pesquisa por se tratar de um modelo de fácil interpretação e menor complexidade de implementação quando comparado aos modelos posteriores. Entretanto, após pesquisar mais a fundo descobriu-se que existem modelos que lidam melhor com predição de preços, como a Rede Neural Artificial (ou ANN). (Lim et al., 2016) concluíram que modelos ANN tiveram bom desempenho na predição de preços de casas em comparação à Análise de Regressão Múltipla (MRA). Ainda, (De Faria et al., 2009) fez uso de ANN para prever a direção do principal indexador do mercado financeiro brasileiro.

A escolha inicial havia sido o modelo LSTM pois envolvia prever o preço ao longo do tempo, este modelo é o que melhor pode descrever a dinâmica dos dados, como demonstrado por (Sunny et al., 2020). Ainda, (Pramod et al., 2020) realizaram predição de preços utilizando LSTM e os resultados foram expressivos, mostrando o quão impactante pode ser o uso deste modelo para a predição de preços futuros.

Contudo, a escolha final foi o modelo MLP uma vez que o valor de tempo era algo intangível no tempo do projeto, sendo assim uma possível melhoria ou abordagem posterior.

Portanto tomamos como decisão não utilizar o tempo como característica para predição do item, dessa forma, não poderíamos usar LSTM. Em contrapartida com um Multi Layer Perceptron é possível realizarmos a predição do preço em relação as características dos dados no presente e obter um resultado conclusivo.

### IV. OBJETIVO

O objetivo do projeto se trata de utilizarmos inteligência artificial para ajudarmos o mercado de skins de CS:GO a ser mais lucrativo, sendo assim o projeto ideal seria uma Rede Neural de que consiga prever os preços de diversas skins e identificar padrões de mercado, utilizando de deep learning para maior eficácia e confiabilidade dos dados. Podendo desta forma, auxiliar na tomada de decisão acerca de compra, venda ou troca dos itens de maneira lucrativa não só para o usuário como para o mercado como um todo.

### V. METODOLOGIA EXPERIMENTAL

O projeto será desenvolvido através da linguagem de programação python com o auxílio das bibliotecas selenium, csv, unidecode, numpy, pandas, scikit-learn e pytorch, com isso a primeira etapa será a coleta de dados através do programa desenvolvido para realizar o web scraping do

principal mercado web de skins para levantamento de dados. Com os dados coletados, o próximo passo será a construção da rede neural responsável pela predição dos preços e alimentá-la utilizando a metodologia de treinamento de redes neurais conhecida como deep learning.

Com a rede neural alimentada, precisamos analisar os resultados obtidos e refinar as etapas anteriores até que a rede possua confiabilidade e eficácia significativa. Com uma rede neural eficaz poderíamos entender melhor os dados e suas características, encontrando padrões e posteriormente podendo prever mais do que o preço de forma que transformaremos os padrões em conhecimento.

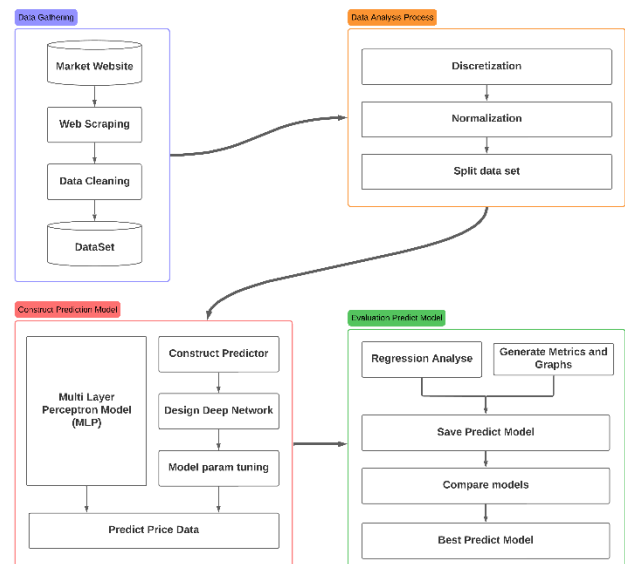


Figure 1 - Pipeline experimental

#### A. BASE DE DADOS

Para o trabalho de previsão de preços foi utilizado uma implementação em Python que realizou o web scrapping ("raspagem de dados") de um site no qual jogadores vendem suas skins. As bibliotecas utilizadas foram: Selenium (entrar em contato com o site e adquirir as informações desejadas), unidecode (importante para retirar acentos e símbolos indesejados dos dados), JSON e CSV (manipulação de arquivos) e a os (manipulação de caminhos dentro do computador para ler e escrever os arquivos).

O processo de web scrapping envolveu entrar no site e fazer com que o script pegasse todas as informações disponíveis de cada item na página e prosseguindo para a página seguinte. Este laço repete-se até que o número determinado previamente pelo usuário seja alcançado. Antes de limpar os dados um passo importante foi a normalização deles, visto que alguns caracteres no site são especiais (como ★ ou ™).

#### B. LIMPEZA DOS DADOS

A limpeza dos dados adquiridos inicia-se abrindo o arquivo gerado anteriormente e deletando uma das colunas, chamada de "tournament tags" (distintivos de campeonatos, que não serão utilizados no treinamento). Um importante passo nesta etapa de limpeza foi o processo de flattening ("achatamento") pois os dados adquiridos no site estavam em formato JSON (JavaScript Object Notation), uma notação muito semelhante à estrutura de dicionário da linguagem

Python. Ocorreu que diversas chaves possuíam dicionários dentro de si, causando um aninhamento de dicionários e, portanto, foi necessário achatar estes dicionários de maneira que, ao final do processo, os dicionários possuísem apenas um nível.

Uma vez que os dicionários estavam devidamente achatados, o próximo passo foi o percorrido de todos eles, substituindo os dados categóricos “rarity” e “quality” por uma graduação numérica (de 0 à 12 para “rarity” e 0 ou 1 para “quality”) para ajudar na hora do treinamento. Além disso, foi feita também uma substituição nos valores de “stickers”, que inicialmente eram uma lista contendo as descrições de possíveis stickers atrelados a cada item. No lugar das informações foi colocado apenas o total de stickers para um dado item, visto que maiores detalhes não seriam úteis para o treinamento.

Finalmente, um segundo arquivo chamado de “cleaned\_data.csv” foi criado. Neste arquivo apenas as colunas desejadas (“stickers”, “paintwear”, “name”, “steam\_price”, “quality”, “rarity”, “type”, “price”) foram inseridas, de modo a descartar todas as outras informações desnecessárias para o desenvolvimento do trabalho.

### C. PRÉ-PROCESSAMENTO DOS DADOS

Com os dados devidamente formatados, deu-se início à etapa de pré-processamento, no qual foi verificada a existência de valores inválidos (“NaN”) e a exclusão de itens que continham estes valores. Para esta base de dados foi preciso realizar a técnica de codificação, que é transformar dados categóricos em dados discretos para que seja possível utilizá-los no treinamento da rede.

Inicialmente a codificação utilizada foi a LabelEncoder, encontrada na biblioteca Scikit-learn sendo escolhida pois transforma dados categóricos em inteiros numéricos.

Contudo com análise posterior ao código destacamos um possível problema relacionado ao LabelEncoder uma vez que os dados transformados para inteiros numéricos possuíam pesos diferentes de acordo com seu valor, sendo assim destacavam uma possível relação de crescimento de valor com crescimento numérico, o que não acontece pois os dados nunca deixaram de ser qualitativos, apenas estavam sendo representados por números.

Com isso, afim de resolvermos essa vantagem conferida pela tipagem hierárquica dos dados, abordamos uma distribuição diferente utilizando OneHotEncoder encontrado também na biblioteca Scikit-learn, que converte os tipos únicos em colunas na tabela e adiciona valores binários onde, 1 e 0 representam respectivamente o pertencimento e o não pertencimento do grupo.

Uma vez que os dados foram devidamente convertidos para números, o próximo passo seguido foi a normalização dos dados, que consiste na uniformização das diferentes escalas que podem existir entre os dados, como é o caso do preço (que varia em escala da moeda) e as qualidades (que foram substituídas por um inteiro entre 0 e 12).

A normalização e discretização dos dados são procedimentos importantes pois melhoram a performance de algoritmos de aprendizagem de máquina.

## VI. ANÁLISE DOS RESULTADOS DO EXPERIMENTO

Para que possamos analisar os resultados obtidos em ambas as experimentações utilizando os modelos MLP e Regressão Linear, é necessário a geração de gráficos e tabelas que classifiquem a eficiência e confiança do modelo.

Onde para a experimentação MLP obtivemos:

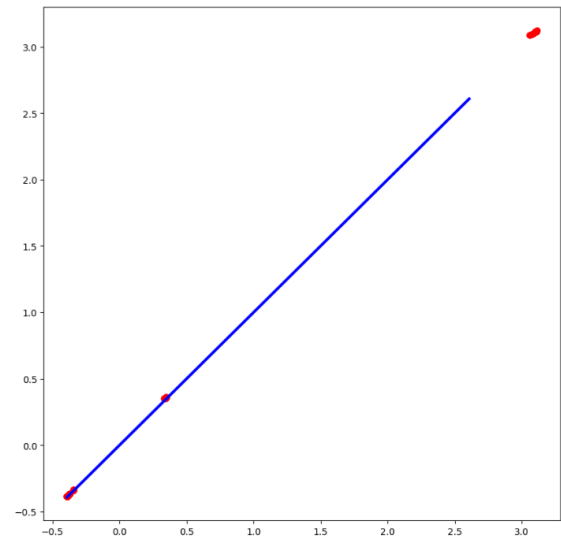


Figure 2 - Retas de regressão MLP (LabelEncoder).

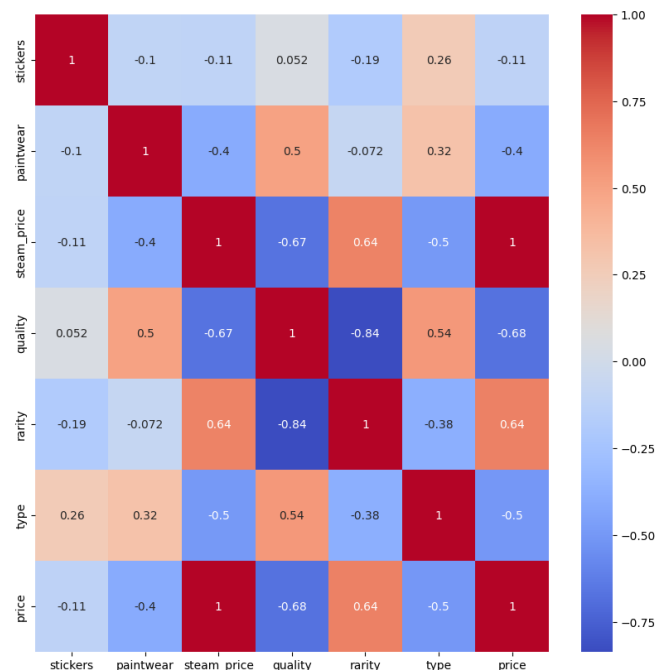


Figure 3 - Matriz de correlação MLP (LabelEncoder).

```
price predicted: [-0.34343379  3.10310945 -0.3903483 -0.38094366 -0.3903483 -0.3903483 ]
real price: tensor([[ -0.3435],
                    [ 3.1034],
                    [-0.3897],
                    [-0.3809],
                    [-0.3913],
                    [-0.3916]])
Mean Absolute Error: 0.00042285993709089044
Mean Squared Error: 3.391265313988737e-07
Root Mean Squared Error: 0.0005823457146737441
Coefficient of Determination: 1.00
```

Figure 4 - Resultados e medidas de avaliação – MLP (LabelEncoder).

Ao analisarmos os resultados obtidos na experimentação do modelo MLP é possível destacar que o modelo foi extremamente satisfatório no quesito ajuste, porém esse ajuste muito próximo pode indicar alguma relação de erro nos dados ou até mesmo erro na implementação ou treinamento do modelo.

Para Regressão Linear os resultados foram diferentes, com aproximações mais comuns no desenvolvimento de redes neurais, indicando talvez um bom uso do modelo, como destacado abaixo pelas figuras.

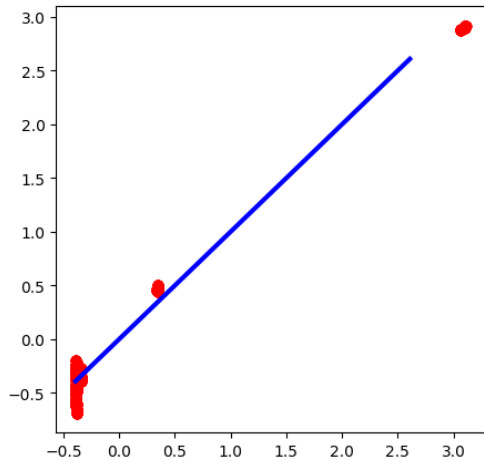


Figure 5 - Reta de regressão - Regressão Linear

```
price predicted: [[-0.4329197 ]
[-0.44440684]
[-0.36938086]
[-0.38064936]
[-0.207735 ]
[-0.4116724 ]]
real price: [[-0.39134318]
[-0.385569 ]
[-0.37504175]
[-0.37504175]
[-0.3427621 ]
[-0.38974905]]
Mean Absolute Error: 0.12172058
Mean Squared Error: 0.030308163
Root Mean Squared Error: 0.1740924
Coefficient of Determination: 0.97
```

Figure 6 - Resultados e medidas de avaliação - Regressão

Concluindo, o modelo MLP é mais atrativo no geral, porém apresenta características que possivelmente indicam algum possível erro, sendo assim é necessário a realização de mais testes para validação das dúvidas, até que se prove algum tipo de erro, MLP é o modelo mais favorável e que obteve os melhores resultados na predição dos preços, assim descartando o uso de regressão linear que mesmo contendo dados saudáveis não é tão preciso quanto o MLP.

Ao realizarmos mais experimentações com MLP foi possível notar que o erro denotado acima estava possivelmente sendo causado por conta do LabelEncoder citado anteriormente, portanto ao realizarmos testes com OneHotEncoder foram obtidos resultados muito mais homogêneos e conclusivos, destacando assim, a superioridade do uso do modelo de Multi Layer Perceptron.

	stickers	paintwear		name	steam_price	quality	rarity	type	price
0	0	0.738805	StatTrak™ MP5-SD   Necro Jr. (Battle-Scarred)		0.28	1	8	SMGs	1.50
1	0	0.727373	StatTrak™ MP5-SD   Necro Jr. (Battle-Scarred)		0.28	1	8	SMGs	1.53
2	0	0.560139	StatTrak™ MP5-SD   Necro Jr. (Battle-Scarred)		0.28	1	8	SMGs	1.55
3	0	0.598293	StatTrak™ MP5-SD   Necro Jr. (Battle-Scarred)		0.28	1	8	SMGs	1.57
4	0	0.787770	StatTrak™ MP5-SD   Necro Jr. (Battle-Scarred)		0.28	1	8	SMGs	1.70

Figure 7 – Dados (LabelEncoder).

	stickers	paintwear	steam_price	quality	rarity	price	Knives	Machineguns	Pistols	SMGs	Shotguns	Sniper Rifles
0	0	0.738805	0.28	1	8	1.50	0	0	0	1	0	0
1	0	0.727373	0.28	1	8	1.53	0	0	0	1	0	0
2	0	0.560139	0.28	1	8	1.55	0	0	0	1	0	0
3	0	0.598293	0.28	1	8	1.57	0	0	0	1	0	0
4	0	0.787770	0.28	1	8	1.70	0	0	0	1	0	0

Figure 8 - Dados (OneHotEncoder).

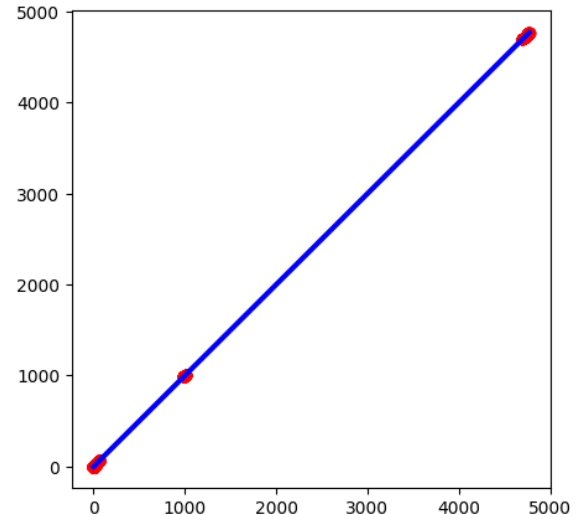


Figure 9 - Reta de regressão MLP (OneHotEncoder).

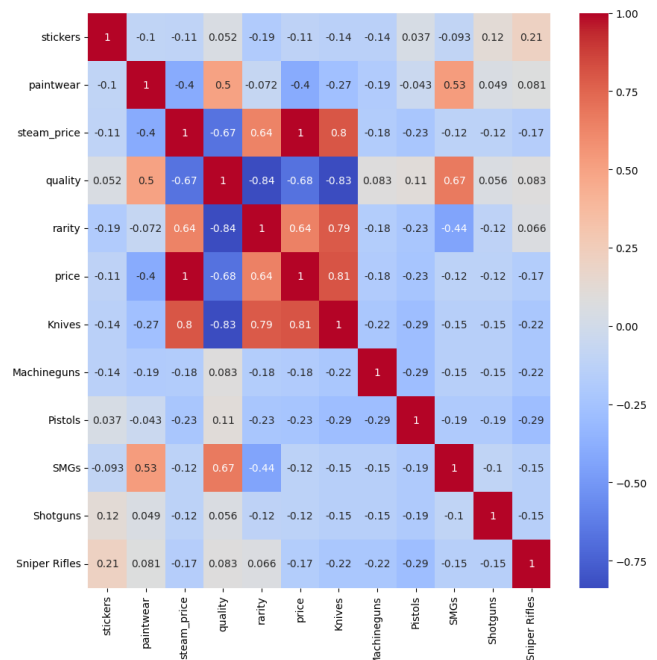


Figure 10 - Matriz de correlação – MLP (OneHotEncoder).



	Predicted price	Real price
0	-0.385428	-0.385400
1	-0.380942	-0.380919
2	0.347834	0.347834
3	-0.385428	-0.385554
4	-0.390346	-0.391424
	Metric	Value
0	Mean Absolute Error	4.337749e-04
1	Mean Absolute Percentage Error	1.174658e+02
2	Mean Squared Error	3.419462e-07
3	Root Mean Squared Error	5.847617e-04
4	Coefficient of Determination	9.999996e-01

Figure 11 – Resultados Finais MLP (OneHotEncoder).

	Predicted	Real
0	3.028837	3.102328
1	-0.182727	-0.344261
2	3.028837	3.102328
3	-0.353567	-0.385143
4	-0.331355	-0.375042
	Metric	Value
0	Mean Absolute Error	0.077400
1	Mean Absolute Percentage Error	19.376840
2	Mean Squared Error	0.009774
3	Root Mean Squared Error	0.098864
4	Coefficient of Determination	0.990331

Figure 12 - Resultados Finais Regressão Linear

## VII. CONCLUSÃO

Como nosso resultado final obtivemos uma rede neural no modelo MLP que prediz preços de skins baseando-se em suas características. Esse foi o modelo escolhido pois dentre os testados foi o que apresentou melhor valor entre os critérios escolhidos, além de apresentar maior escalabilidade e semelhança com a realidade.

No geral o projeto está bem usual, podendo auxiliar na atualização constante e avaliação de itens para o mercado, a ideia é torna-lo mais intuitivo e preciso com o tempo, uma vez que da forma que se encontra hoje necessita de uma inteligência por trás para que seja possível minimizar riscos e maximizar lucros.

Em nossos testes finais destacamos que ainda existem muitas etapas à serem otimizadas para que o projeto esteja em seu ápice, sendo assim várias ideias vieram à tona, sendo algumas delas:

- Aumento da base de dados e otimização da automação responsável pelo WebScraping;
- Utilização de modelos de otimização ao invés de predição, podendo incluir modelos genéticos como BRKGA;
- Implementação do tempo como variável para predições futuras utilizando modelo LSTM;
- Integração de uma API para melhor estruturação dos resultados e inclusão de operações mais complexas, como buscas e predições específicas.

Destacando assim diversas vertentes que podemos seguir para que o projeto fique cada vez mais e mais refinado de acordo com o tempo.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Newell, A., and Simon, H. A. 1976. Computer science as empirical inquiry: symbols and search. Communications of the ACM 19(3), pp.113–126.
- [2] Lim, T., Wan et al. 2016. Housing Price Prediction Using Neural Networks, 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).
- [3] Anuradha, G et al. 2019. House Price Prediction Using Regression Techniques: A Comparative Study, Department of CSE, VRSEC, Vijayawada.
- [4] Zhao, Bo. 2017. Web Scraping, Springer International Publishing AG, Disponível em: [https://www.researchgate.net/publication/317177787\\_Web\\_Scraping/](https://www.researchgate.net/publication/317177787_Web_Scraping/), Acesso em: 06 de jun. de 2023.
- [5] Yu, Pengfei, and Yan, Xuesong. 2020. Stock price prediction based on deep neural networks, Springer International Publishing AG, Disponível em: <https://link.springer.com/article/10.1007/s00521-019-04212-x/>, Acesso em: 06 de jun. de 2023.
- [6] İcan, Özgür, and Çelik, B. Taha. 2017. Stock Market Prediction Performance of Neural Networks: A Literature Review, Canadian Center of Science and Education, Disponível em: <https://www.ccsenet.org/journal/index.php/ijef/article/view/70421/>, Acesso em: 06 de jun. de 2023.
- [7] De Faria, E. L., Albuquerque M. P., Gonzalez, J. L., Cavalcante, J. T. P., & Albuquerque, M. P. 2009. Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods, Expert Systems with Applications, Disponível em: <https://doi.org/10.1016/j.eswa.2009.04.032/>, Acesso em: 06 de jun. de 2023.
- [8] Pramod B. S., Mallikarjuna Shastri P. M. 2020. Stock Price Prediction Using LSTM, CSE, REVA University, Bengaluru, Disponível em: [https://www.researchgate.net/publication/348390803\\_Stock\\_Price\\_Prediction\\_Using\\_LSTM/](https://www.researchgate.net/publication/348390803_Stock_Price_Prediction_Using_LSTM/), Acesso em: 06 de jun. de 2023.
- [9] Pawar, K., Jalem, S. R. and Tiwari, V. 2019. Stock Market Price Prediction Using LSTM RNN, Springer Nature Singapore Pte Ltd. 2019, Disponível em: [https://link.springer.com/chapter/10.1007/978-981-13-2285-3\\_58/](https://link.springer.com/chapter/10.1007/978-981-13-2285-3_58/), Acesso em: 06 de jun. de 2023.
- [10] Gonçalves, J.F., Resende, M.G.C. Biased random-key genetic algorithms for combinatorial optimization. J Heuristics 17, 487–525 (2011). Disponível em: <https://doi.org/10.1007/s10732-010-9143-1>.
- [11] Valve Enterprise. Counter-Strike:Global Offensive. Steam, 2012, Disponível em: [https://store.steampowered.com/app/730/CounterStrike\\_Global\\_Offensive/](https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/), Acesso em: 06 de jun. de 2023.
- [12] Gugelmin, Felipe. Skin de Counter-Strike: Global Offensive é vendida por US\$ 400 mil. Adrenaline, 2023 Disponível em: <https://www.adrenaline.com.br/games/skin-de-counter-strike-global-offensive-e-vendida-por-us-400-mil/>, Acesso em: 06 de jun. de 2023.