CENTRO DE PESQUISA E DESENVOLVIMENTO TECNOLÓGICO EM INFORMÁTICA
CEPEDI

Residência em Software – ResTIC36

Trilha 3 – Ciência de Dados

Relatório Técnico: Implementação e Análise do Algoritmo de Regressão Linear

Vinícius de Oliveira Souza

Sanley Pires Ferreira

Data de Entrega: 17 de Novembro de 2024

Resumo

Este relatório descreve a implementação e análise de um algoritmo de Regressão Linear para prever a pontuação de influência (*influence_score*) de influenciadores do Instagram. A metodologia envolveu a análise exploratória de dados, pré-processamento, treinamento do modelo e avaliação de desempenho usando métricas como R^2, RMSE e MAE. Os resultados indicam que o modelo, mesmo com otimizações, apresenta um desempenho limitado para este conjunto de dados, sugerindo a necessidade de explorar modelos não lineares ou baseados em árvores para capturar melhor as relações entre as variáveis.

1. Introdução

1.1. Contextualização do Problema

O objetivo deste projeto é prever a pontuação de influência (*influence_score*) de influenciadores do Instagram com base em diversas características, como número de seguidores, média de curtidas e taxa de engajamento. A compreensão dos fatores que impulsionam a influência online é crucial para marcas e profissionais de marketing, permitindo otimizar estratégias de marketing de influência.

1.2. Justificativa para o Uso da Regressão Linear

A Regressão Linear é um algoritmo amplamente utilizado para modelagem preditiva devido à sua simplicidade, interpretabilidade e eficiência computacional. É uma escolha inicial adequada para explorar a relação entre as variáveis e estabelecer uma linha de base para comparação com outros modelos mais complexos.

1.3. Descrição do Conjunto de Dados

O conjunto de dados utilizado foi obtido do Kaggle e contém informações sobre os principais influenciadores do Instagram, incluindo:

- ***rank***: Rank do influenciador com base na quantidade de seguidores.
- ***channel_info***: Nome de usuário do influenciador no Instagram.
- ***influence_score***: Pontuação de influência dos usuários, calculada com base em menções, importância e popularidade.
- ***posts***: Número de postagens feitas até o momento.
- ***followers***: Quantidade de seguidores do usuário.
- ***avg_likes***: Média de curtidas nas postagens do influenciador (curtidas totais/postagens totais).
- ***60_day_eng_rate***: Taxa de engajamento dos últimos 60 dias do influenciador, como fração do total de engajamentos feitos até agora.
- ***new_post_avg_like***: Média de curtidas em novas postagens.
- ***total_likes***: Total de curtidas que o usuário recebeu em suas postagens (em bilhões).
- ***country***: País ou região de origem do usuário.

2. Metodologia

2.1. Análise Exploratória

A análise exploratória inicial envolveu a inspeção dos dados, cálculo da matriz de correlação e visualização da distribuição das variáveis. Isso permitiu identificar padrões, relações entre variáveis e potenciais outliers. Para isso, foi necessário fazer um pré-processamento, a fim de extrair melhores resultados na visualização das correlações e distribuições das variáveis.

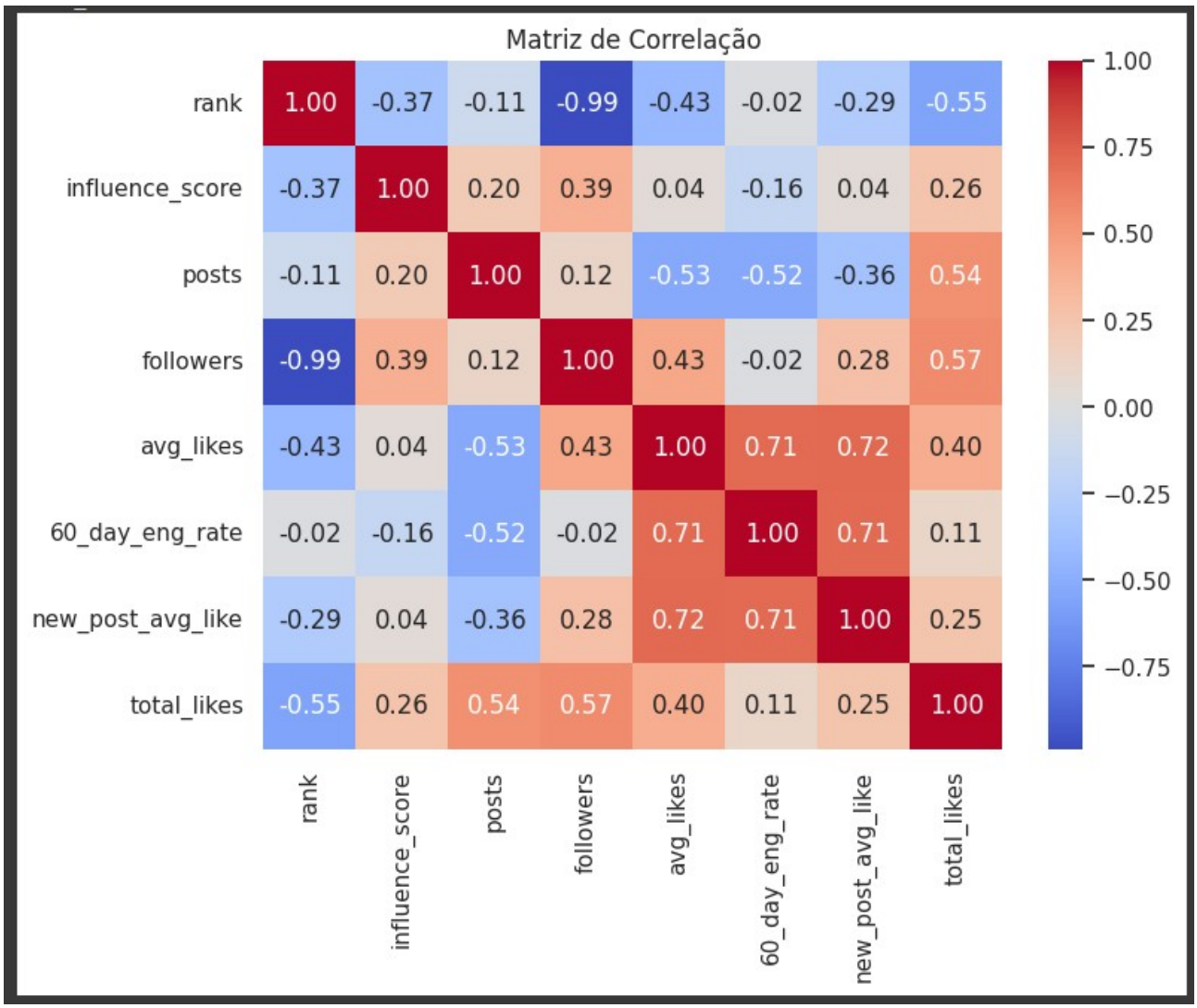
2.2. Pré-processamento dos Dados

O pré-processamento incluiu:

•Conversão de valores com sufixos ('k', 'm', 'b') para valores reais, ou seja, as colunas com sufixos de escala foram transformados para seus respectivos valores reais: milhares, milhões e bilhões, respectivamente.

•Remoção de linhas com valores NaN

•Tratamento de outliers usando uma transformação chamada Box-cox, utilizada para tentar **estabilizar a variância** e **melhorar a normalidade** dos dados, tornando-os mais adequados para a aplicação de modelos de regressão linear. A transformação Box-Cox aplica uma função matemática aos dados que depende de um parâmetro lambda (λ). O valor de lambda é estimado a partir dos dados para encontrar a melhor transformação. Essa transformação teve melhor desempenho, nesse caso, do que a logarítmica e a da raiz quadrada.



2.3. Implementação do Algoritmo

Foi utilizado o algoritmo de Regressão Linear da biblioteca scikit-learn. O conjunto de dados foi dividido em conjuntos de treinamento (80%) e teste (20%).

2.4. Validação e Ajuste de Hiperparâmetros

•**Normalização dos Dados:** Os dados foram normalizados usando MinMaxScaler, visando redimensionar as features para um intervalo específico, geralmente entre 0 e 1. Isso é crucial para algoritmos como a Regressão Linear e o SGDRegressor, que são sensíveis à escala das features. Dessa forma, evita que features com valores maiores dominem o processo de treinamento.

•**Algoritmo de Otimização:** Foi utilizado o SGDRegressor (Stochastic Gradient Descent Regressor) para encontrar os melhores parâmetros do modelo de Regressão Linear. Ele usa o método do gradiente descendente estocástico para minimizar a função de custo. Foi testado com diferentes taxas de aprendizado e tipos de regularização para otimizar o modelo e os que apresntaram melhores resultados foram:

Parâmetros:

- ***loss="squared_error"***: Define a função de custo como o erro quadrático médio (MSE), que mede a diferença entre os valores reais e previstos.
- ***alpha=0.01***: Taxa de aprendizado. Controla o tamanho do passo durante a atualização dos parâmetros do modelo. Valores menores levam a uma convergência mais lenta, mas podem evitar que o algoritmo fique preso em mínimos locais.
- ***penalty="elasticnet"***: Tipo de regularização. O Elastic Net combina as regularizações L1 (Lasso) e L2 (Ridge), penalizando os coeficientes do modelo para evitar overfitting.
    - L1 incentiva a esparsidade (coeficientes próximos de zero), selecionando features relevantes.
    - L2 distribui a importância entre as features, evitando que qualquer feature tenha um peso muito grande.
- ***max_iter=2000***: Número máximo de iterações do algoritmo. Define quantas vezes o algoritmo irá atualizar os parâmetros antes de parar.
- ***tol=1e-3***: Tolerância para parada. O algoritmo para se a mudança na função de custo entre duas iterações for menor que a tolerância.
- ***random_state=42***: Define a semente para o gerador de números aleatórios, garantindo a reprodutibilidade dos resultados.

•**Validação Cruzada:** A validação cruzada com 4 folds foi utilizada para avaliar o desempenho do modelo e evitar overfitting. Embora essa técnica forneça uma estimativa mais precisa do desempenho do modelo em dados não vistos, foi observado uma redução no $R^{2.}$

3. Resultados e Discussões

3.1. Métricas de Avaliação

As seguintes métricas foram utilizadas para avaliar o desempenho do modelo e esses foram os melhores resultado encontrados após as otimizações:

•**R^2:** 0.17

•**RMSE:** 10.24

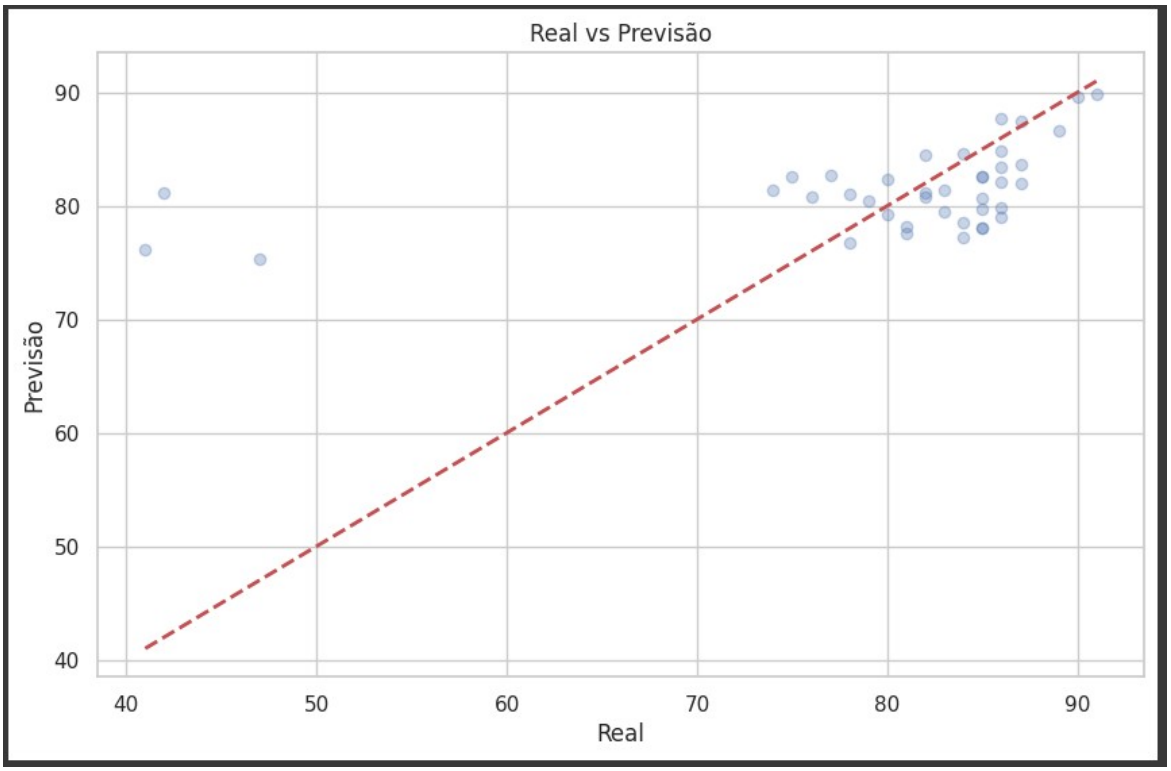
•**MAE:** 5.74

Esses valores indicam que o modelo de regressão múltipla, mesmo com as otimizações ainda apresenta um desempenho limitado para este conjunto de dados.

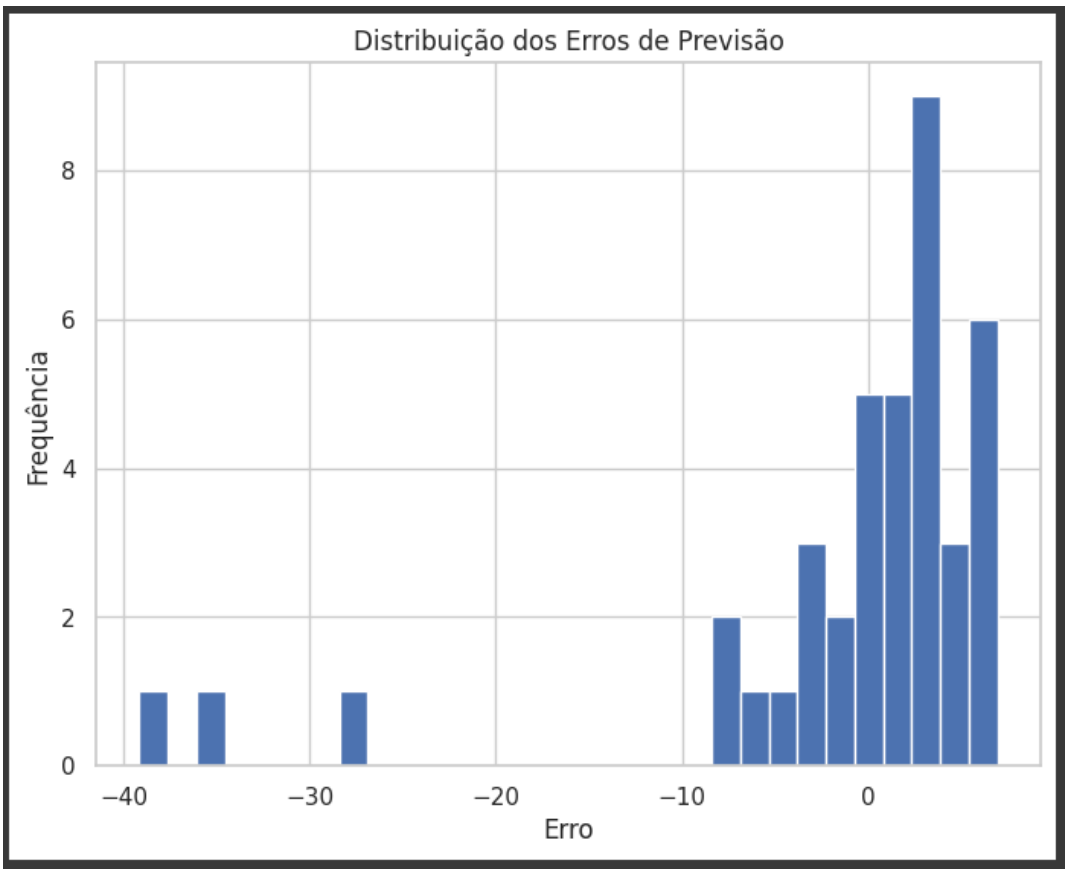
- O R^2 representa a proporção da variância na variável dependente que é explicada pelas variáveis independentes do modelo. Um R^2 de 0.17 indica que o modelo está explicando apenas 17% da variabilidade em influence_score, o que é considerado baixo. Isso significa que grande parte da variação na pontuação de influência não está sendo capturada pelo modelo.
- O RMSE e o MAE medem a magnitude dos erros de previsão do modelo. Valores altos indicam que o modelo está cometendo erros consideráveis na previsão da pontuação de influência. Nesse caso, um RMSE de 10.24 e um MAE de 5.74 sugerem que, em média, as previsões do modelo estão erradas em cerca de 10.24 e 5.74 pontos, respectivamente.

3.2. Visualizações do Desempenho do Modelo

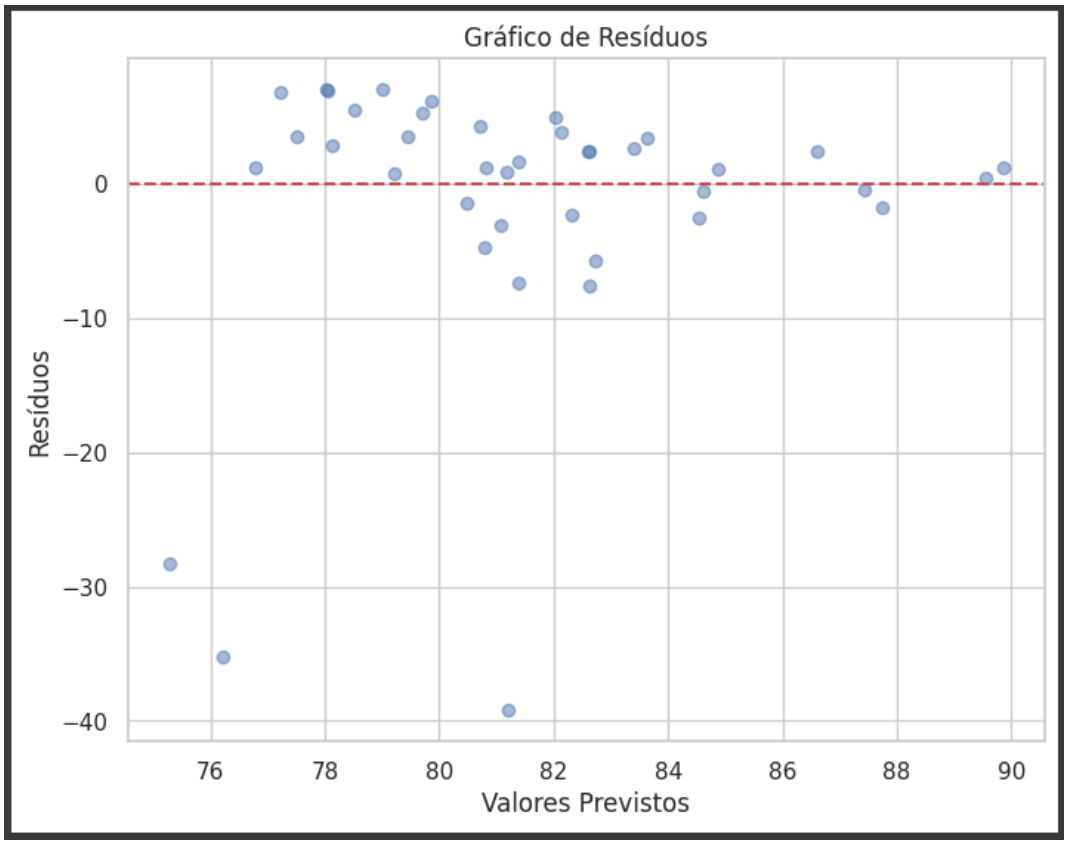
•Gráfico de dispersão entre valores reais e previstos



•Histograma dos erros de previsão



•Gráfico de resíduos



3.3 Discussões

A conclusão que chegamos é que não há uma relação linear clara entre a variável dependente e as variáveis independentes. Dessa forma, o modelo está encontrando dificuldades em se ajustar aos dados. Talvez experimentar outros modelos não lineares ou modelos basedos em árvores podem capturar melhor as relações.

Outro ponto observado é que mesmo como o tratamento de outliers, sem removê-los para não comprometer o dataset, ao traçar os boxplots percebe-se que ainda há pontos que ficam fora das margens. Isso se dá pelo fato de os dados terem uma grande diferença nas escalas e, embora essa diferença seja representativa, já que retratam a realidade dos dados, o modelo entende como outliers impactando na regressão linear.

Ademais, a distribuição de erros x previsões apresenta uma dispersão considerável dos pontos em torno de zero, indicando que o modelo não está capturando bem a relação entre as variáveis e que as previsões estão sujeitas a erros significativos.

Por fim, contradizendo os valores das métricas encontradas, os resíduos aleatoriamente dispersos em torno da linha horizontal em y=0 no gráfico é um bom sinal, indicando que o modelo até que está se ajustando aos dados. Entretanto, há pontos que se distanciam muito dos outros resíduos, o que, ainda, indicam presença de outliers.

4. Conclusões e Trabalhos Futuros

Os resultados indicam que o modelo de Regressão Linear apresenta um desempenho limitado para este conjunto de dados, explicando apenas 17% da variabilidade na pontuação de influência. Isso sugere que a relação entre as variáveis pode ser mais complexa do que uma relação linear. A exploração de modelos não lineares ou baseados em árvores pode ser necessária para melhorar a precisão das previsões. Além disso, realizar a engenharia de features para criar novas variáveis relevantes e coletar mais dados para aumentar o tamanho do conjunto de dados podem ser executados para melhorar o desempenho do modelo e enriquecer ainda mais o projeto.