CENTRO DE PESQUISA E DESENVOLVIMENTO TECNOLÓGICO EM INFORMÁTICA E ELETROELETRÔNICA DE ILHÉUS

Leonardo dos Santos Vaz
Ivanildo Gomes da Silva

IMPLEMENTAÇÃO E ANÁLISE DO ALGORITMO DE REGRESSÃO LINEAR SOBRE DADOS DOS PRINCIPAIS INFLUENCIADORES DO INSTAGRAM

Relatório Técnico: Atividade avaliativa referente a unidade 9 da Trilha 3 do curso de Ciência de Dados da residência TIC 36.

Ilhéus

17/11/2024

1. RESUMO

O presente relatório técnico descreve o desenvolvimento de um modelo preditivo utilizando o algoritmo de Regressão Linear para inferir a taxa de engajamento dos principais influenciadores do Instagram e suas variáveis independentes. O projeto abrange etapas fundamentais, incluindo a análise exploratória dos dados para identificação de padrões e correlações relevantes, a implementação do modelo preditivo, a otimização dos parâmetros e a validação do desempenho do modelo. O objetivo deste projeto é desenvolver um modelo preditivo usando o algoritmo de Regressão Linear para resolver um problema de inferência sobre taxa de engajamento dos principais influenciadores do Instagram. Todo o processo foi documentado neste relatório técnico em formato PDF, que foi incluído no repositório GitHub juntamente com o código-fonte do projeto e arquivo Readme, garantindo acessibilidade e reprodutibilidade.

2. INTRODUÇÃO

Com o desenvolvimento das Tecnologias Digitais da Comunicação e Informação (TDIC) e posterior popularização das redes sociais, as mesmas tornaram-se uma das principais plataformas de interação social, marketing e geração de influência ocupando lugares, que outras ocasiões eram preenchidas pelo Radio, a TV, a indústria fonográfica e uma series de outras tecnologias e industrias culturais que foram aos poucos, e nos últimos anos de maneira acelerada, fagocitadas pela internet. Enquanto há alguns anos, a internet apresentava uma maior diversidade de espaços alternativos de interação social, hoje navegar na internet torna-se cada vez mais sinônimo de entrar em alguma rede social hegemônica.

Dentre essas diversas plataformas, o Instagram destacou-se como um dos principais meios para conectar pessoas, marcas e influenciadores, criando um ecossistema único para a troca de informações, compartilhamento de conteúdo e engajamento em larga escala. Em geral, redes sociais produzem uma imenso volume de dados, o que a torna, um prolífico celeiro para a realização de testes de algoritmos de aprendizagem de máquina. O arquivo utilizado neste estudo contém informações sobre os 200 principais influenciadores do Instagram, organizados com base em sua classificação, determinada pelo número de seguidores, são 10 atributos principais os quais devemos encontrar suas respectivas correlações.

Utilizaremos para esta análise, a **regressão linear**, uma vez que, muitos parâmetros como número de seguidores e numero de curtidas apresentam comportamento linear em suas correlações, também esperamos avaliar, possíveis correlações que não podem ser estabelecidas por meio do método escolhido (A Regressão Linear), uma vez que, o cientista de dados deve compreender os problemas apresentados pelos dados a partir de perspectivas mais amplas e apresentar soluções diferente as partir de experiências com resultados insatisfatórios. Deste modo, elaboramos as seguintes **questões de pesquisa**: Existem relações lineares nos dados apresentados? Em que nível essas relações existem? A partir dessa questão de pesquisa elaboramos os seguintes objetivos específicos:

- Análise exploratória.
- Implementação da Regressão Linear utilizando a biblioteca Scikit-Learn do Python.
- Teste de diferentes configurações do algoritmo para ajuste e otimização.

Essa abordagem permite o esgotamento das possibilidades desta técnica de análise para a determinação das relações entre os dados, sejam essas correlações lineares, ou não. Em caso da não observação de relações lineares, outras técnicas devem ser sugeridas no tópico: conclusão e trabalhos.

3. METODOLOGIA

Neste tópico, descreveremos de maneira detalhada os procedimentos utilizados para a realização deste trabalho. Em linhas gerais, foi utilizado um algoritmo de regressão linear implementado por meio da linguagem Python. Os dados foram organizados em um arquivo .csv (planilha). A regressão linear é técnica de análise de dados utilizada para entender a relação entre algumas variáveis, evidente não esperamos que todas as variáveis que se relacionam tenha relações lineares, vamos investigar se isso ocorre, e quando ocorre neste conjunto de dados. O principal objetivo nesta técnica é prever o valor de uma variável (chamada de variável dependente) com base no valor de outra (chamada de variável independente). Foi utilizada Scikit-Learn, uma biblioteca de código aberto em Python amplamente utilizada para aprendizado de máquina.

3.1. Análise Exploratória

A análise exploratória de dados é uma etapa fundamental para compreender as principais características de um determinado conjunto de dados. Seu objetivo é identificar padrões, tendências, anomalias e relações entre as diferentes variáveis por meio de métodos gráficos e estatísticos. Esta etapa será dividida em 5 tarefas:

- Organização dos dados
- Avaliação do percentual de linhas com valores nulos
- Limpeza e tratamento de dados
- Estatística descritiva
- Produção e Análise de Gráficos (visualizações)

Dados desorganizados levam a análises incorretas, equivocadas, ou na melhor das hipóteses pode fazer o cientista de dados perder tempo precioso que poderia se utilizado de maneira mais produtiva, deste modo o primeiro passo de nosso trabalho foi a **organização dos dados**.

TABELA 1

	rank	channel_info	influence_score	posts	followers	avg_likes	60_day_eng_rate	new_post_avg_like	total_likes	country
0	1	cristiano	92	3.3k	475.8m	8.7m	1.39%	6.5m	29.0b	Spain
1	2	kyliejenner	91	6.9k	366.2m	8.3m	1.62%	5.9m	57.4b	United States
2	3	leomessi	90	0.89k	357.3m	6.8m	1.24%	4.4m	6.0b	NaN
3	4	selenagomez	93	1.8k	342.7m	6.2m	0.97%	3.3m	11.5b	United States
4	5	therock	91	6.8k	334.1m	1.9m	0.20%	665.3k	12.5b	United States
...	...	...	...	...	...	...	...	...	...	...
195	196	iambeckyg	71	2.3k	33.2m	623.8k	1.40%	464.7k	1.4b	United States
196	197	nancyajram	81	3.8k	33.2m	390.4k	0.64%	208.0k	1.5b	France
197	198	luansantana	79	0.77k	33.2m	193.3k	0.26%	82.6k	149.2m	Brazil
198	199	nickjonas	78	2.3k	33.0m	719.6k	1.42%	467.7k	1.7b	United States
199	200	raisa6690	80	4.2k	32.8m	232.2k	0.30%	97.4k	969.1m	Indonesia

Tabela 1: Prévia do conjunto de dados apresentados no desafio.

FIGURA 1

Utilizamos o código a seguir para a organização dos dados, observamos que a coluna country tem 31% de **valores nulos**, as demais colunas não possuem valores nulos.

```python
import pandas as pd
from IPython.display import display

# Caminho para o arquivo no Google Drive
caminho_arquivo = '/content/drive/My Drive/projetounidade9/tabela_influenciadores.csv'

# Leia o arquivo CSV especificando o separador correto (;)
df = pd.read_csv(caminho_arquivo, sep=';', encoding='latin-1')

# Exibir 200 linhas do CSV
display(df.head(200))

# Salve o arquivo com o separador corrigido (,)
caminho_corrigido = '/content/drive/My Drive/projetounidade9/tabela_influenciadores_corrigido.csv'
df.to_csv(caminho_corrigido, index=False, sep=',')

print(f"Arquivo corrigido salvo em: {caminho_corrigido}")

# Contar valores vazios por coluna
valores_vazios = df.isnull().sum()

# Calcular o percentual de valores vazios por coluna
percentual_vazios = (valores_vazios / len(df)) * 100

# Exibir o total de linhas com valores vazios
total_linhas_vazias = df.isnull().any(axis=1).sum()

# Exibir os resultados
print("Contagem de valores vazios por coluna:")
print(valores_vazios)

print("\nPercentual de valores vazios por coluna (%):")
print(percentual_vazios)

print(f"\nTotal de linhas com valores vazios: {total_linhas_vazias} ({(total_linhas_vazias / len(df)) * 100:.2f}%)")
```

Figura 1: Código utilizado para organização.

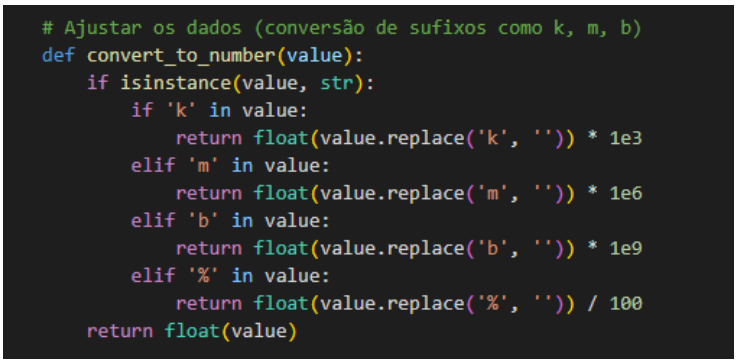
FIGURA 2

```
Arquivo corrigido salvo em: /content/drive/My Drive/projetounidade9/tabela_influenciadores_corrigido.csv
Contagem de valores vazios por coluna:
rank                0
channel_info        0
influence_score     0
posts               0
followers           0
avg_likes           0
60_day_eng_rate     0
new_post_avg_like   0
total_likes         0
country            62
dtype: int64

Percentual de valores vazios por coluna (%):
rank                0.0
channel_info        0.0
influence_score     0.0
posts               0.0
followers           0.0
avg_likes           0.0
60_day_eng_rate     0.0
new_post_avg_like   0.0
total_likes         0.0
country            31.0
dtype: float64

Total de linhas com valores vazios: 62 (31.00%)
```

Figura 2: percentual de linhas com valores nulos por coluna.

Nosso conjunto de dados apresentou valores numéricos formatados com sufixos para representar grandezas de forma compacta. Esses sufixos, como k, m, ou %, indicam múltiplos ou porcentagens e precisam ser convertidos para valores numéricos reais antes de serem usados em cálculos, esta é a etapa de **limpeza e tratamento dos dados**.

FIGURA 3



```python
# Ajustar os dados (conversão de sufixos como k, m, b)
def convert_to_number(value):
    if isinstance(value, str):
        if 'k' in value:
            return float(value.replace('k', '')) * 1e3
        elif 'm' in value:
            return float(value.replace('m', '')) * 1e6
        elif 'b' in value:
            return float(value.replace('b', '')) * 1e9
        elif '%' in value:
            return float(value.replace('%', '')) / 100
    return float(value)
```

Figura 3: parte do código utilizado para conversão de sulfixos.

O próximo passo foi realizar alguns cálculos estatísticos para apresentam um **panorama estatístico descritivo** avaliou que a análise do Heatmap era a opção mais adequada para decidir os caminhos para as implementações dos algoritmos de aprendizado de máquina. O gráfico de correlação (Heatmap) é representação gráfica de dados onde os valores são indicados por cores. Ele é usado para facilitar a visualização de padrões, relações ou tendências nos dados.

FIGURA 4

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from google.colab import drive

# Montar o Google Drive
drive.mount('/content/drive')

# Caminho do arquivo no Google Drive
caminho_arquivo = '/content/drive/My Drive/projetounidade9/tabela_influenciadores.csv'

# Carregar os dados
data = pd.read_csv(caminho_arquivo, sep=';', encoding='latin-1')

# Ajustar os dados (conversão de sufixos como k, m, b)
def convert_to_number(value):
    if isinstance(value, str):
        if 'k' in value:
            return float(value.replace('k', '')) * 1e3
        elif 'm' in value:
            return float(value.replace('m', '')) * 1e6
        elif 'b' in value:
            return float(value.replace('b', '')) * 1e9
        elif '%' in value:
            return float(value.replace('%', '')) / 100
    return float(value)

columns_to_convert = ['posts', 'followers', 'avg_likes', '60_day_eng_rate', 'new_post_avg_like', 'total_likes']
for col in columns_to_convert:
    if col in data.columns:
        data[col] = data[col].apply(convert_to_number)

# Gerar uma matriz de correlação, apenas para colunas numéricas
# O parâmetro numeric_only=True garante que apenas colunas numéricas sejam usadas
correlation_matrix = data.corr(numeric_only=True)

# Criar um heatmap usando Seaborn
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Heatmap de Correlação")
plt.show()
```

Figura 4: Código utilizado para a geração do Heatmap

Figura 5

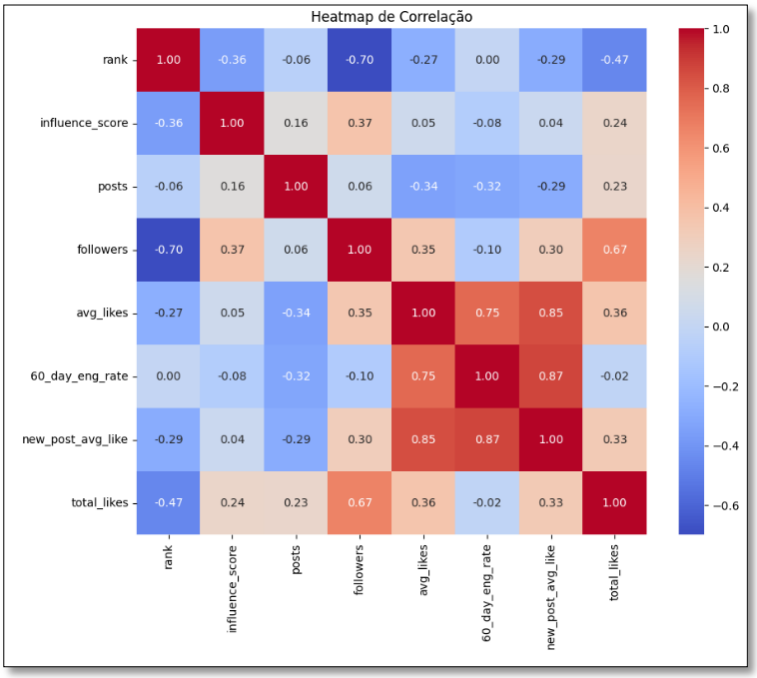


Figura 5: heatmap produzido por meio das bibliotecas: Seaborn e Matplotlib.

Além do Heatmap (para a escolha das melhores candidatas para as variáveis dependentes), foram utilizadas técnicas de normalização e optimização do algoritmo:

Mínimos Quadrados Ordinários (OLS): Método utilizado para calcular os coeficientes da regressão linear, minimizando a soma dos erros quadráticos.

Transformação Logarítmica: Aplicada em variáveis como **avg_likes** e **followers** para reduzir variâncias e lidar com distribuições assimétricas. Melhorou as relações lineares e estabilizou o modelo.

Remoção de Outliers: Utilizamos o método do Intervalo Interquartil (IQR) para identificar e remover outliers, reduzindo o impacto de valores extremos no modelo.

Normalização com Min-Max Scaling: Normalizou as variáveis para o intervalo de 0 a 1. Garantiu que as escalas das variáveis fossem consistentes, melhorando o desempenho do algoritmo.

Os detalhes da implementação dessas técnicas, serão melhor apresentados nos resultados e discussões dos tópicos posteriores. Ela foram utilizadas a medida que o algoritmo implementado não apresentou resultados satisfatórios, esses resultados foram avaliados de acordo com as seguintes métricas:

MAE (Erro Absoluto Médio): Deve ser o mais próximo possível de **0**.
MSE (Erro Quadrático Médio): Esperamos resultados próximos de **0**, embora sua interpretação dependa do problema.
R^2 (Coeficiente de Determinação): O mais próximo de **1**. No entanto, em alguns casos, um R^2 moderado (70–90%) já pode ser aceitável, dependendo da complexidade do problema e da qualidade dos dados. Este parâmetro foi mais utilizado, por apresentar uma visão geral do desempenho do algoritmo.

4. RESULTADOS

Neste tópico apresentaremos os resultados finais das implementações dos algoritmos desenvolvidos ao longo da execução deste projeto, os detalhes de sua execução serão apresentados nos tópicos finais.

Com base no heatmap construído na fase de análise exploratória, identificamos as variáveis mais correlacionadas com a variável dependente **(60_day_eng_rate)**. Neste caso, as variáveis **avg_likes** e **new_post_avg_like** apresentam alta correlação com 60_day_eng_rate **(valores próximos de 1)**, deste modo, consideramos que são boas candidatas a variáveis independentes (features). Por outro lado, variáveis com correlação muito baixa **(próximas a 0)** podem ser descartadas.

Figura 6

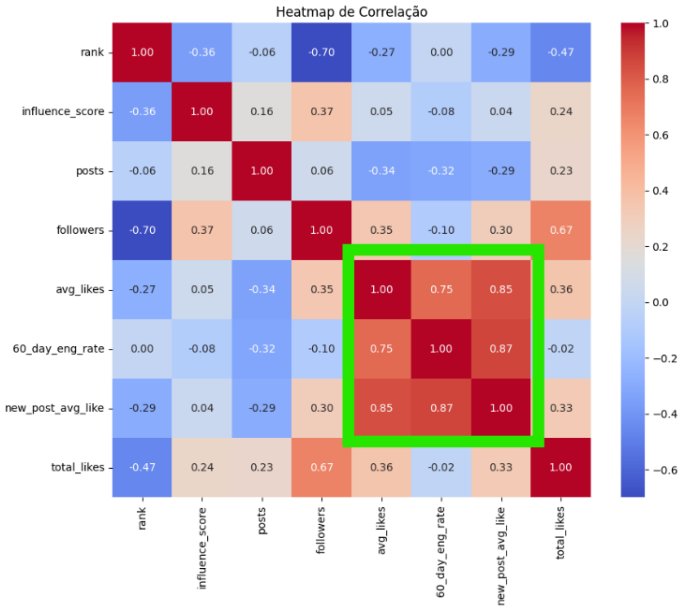


Figura 6: heatmap destacando a região que apresenta as correlações citadas.

Algumas informações sobre as variáveis de maior correlação de acordo com o heatmap:

TABELA 2

Métrica	Foco	Interpretação
avg_likes	Média geral de curtidas em todos os posts analisados	Reflete o desempenho histórico.
new_post_avg_like	Média de curtidas apenas nos posts recentes	Indica o engajamento atual.

Tabela 2: informações sobre as variáveis escolhidas com base no heatmap.

Foram feitas 4 implementações do código em python, todos foram avaliados de acordo com os parâmetros de avaliação determinadas na metodologia, a partir dos resultados, foram feitas modificações no código a fim de melhorar os resultados, buscando principalmente, atingir **R^2** entre 70 e 90%. Apresentaremos aqui, o resultado final da 4° implementação do algoritmo, trazendo nas discussões os resultados parciais das 3 primeiras implementações.

Figura 7

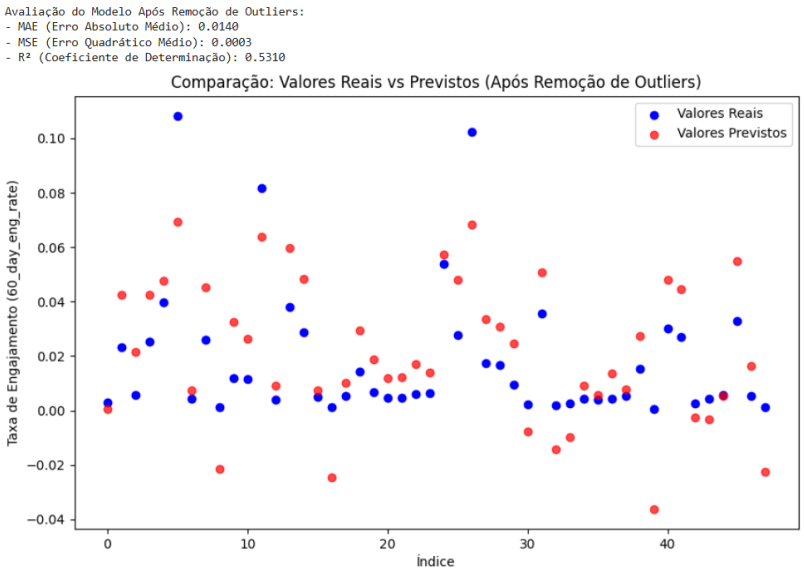


Figura 7: Melhor resultado obtido após as 4 implementações.

Em relação as métricas utilizadas na avalição do modelo, os seguintes dados foram apresentados:

Erro Absoluto Médio (MAE): 0.0140

Pequeno, indicando que o modelo faz previsões próximas dos valores reais.

Erro Quadrático Médio (MSE): 0.0003

Pequeno, sugerindo que os erros são bem distribuídos.

Coeficiente de Determinação (R^2): 0.5310

O modelo explica 53,10% da variação na taxa de engajamento.

O coeficiente de determinação atingiu, em nova melhor implementação justifica 53,10% da taxa de variação do engajamento nos próximo 60 dias. Esse valor está muito longe da meta de 70 a 90% para bons modelos de regressão linear. Utilizamos todos os processos e técnicas apresentados na metodologia, o que sugere que há relações não lineares entre os dados nesse conjunto. Avaliaremos de maneira mais detalhada no tópico de discussões.

5. DISCUSSÃO

Na primeira implementação, realizamos o treinamento de um modelo de regressão linear múltipla utilizando as variáveis **avg_likes**, **new_post_avg_like**, e **60_day_eng_rate** sem realizar transformações ou ajustes nos dados. Embora o modelo tenha produzido resultados funcionais, certas limitações foram identificadas:

Figura 8

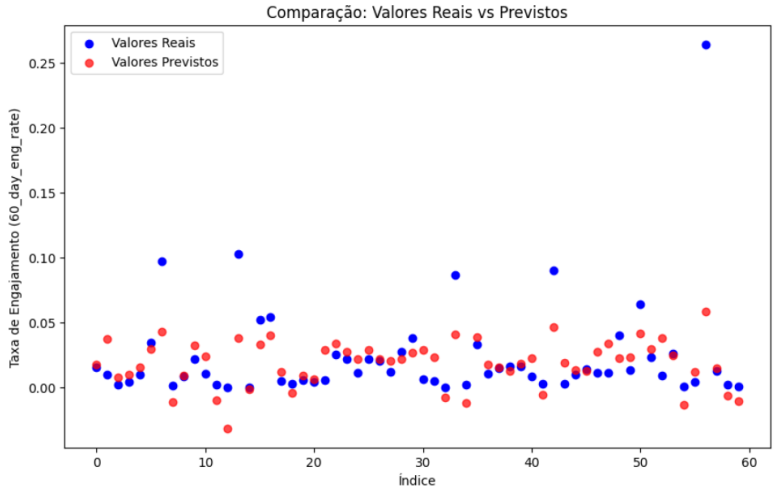


Figura 8: Resultados da primeira implementação (representação gráfica)

O **R² baixo**: O coeficiente de determinação foi de **33,57%** sugerindo que, embora apresente alguma linearidade, o modelo não estava capturando bem a relação entre as variáveis. **A Presença de Outliers** a principio, foi ignorada, uma vez que tratava-se de resultados reais, o que se mostrou um erro, pois percebeu-se que os dados apresentavam valores extremos que poderiam influenciar negativamente o desempenho do modelo. **Escalas de variáveis inconsistentes** foram identificadas, em nosso caso, algumas variáveis apresentavam escalas muito diferentes:

- **followers** (número de seguidores):
    - Valores na casa de milhões (ex.: 1.2M, 500K).

- **avg_likes** (média de curtidas):
    - Valores em centenas de milhares (ex.: 50K, 200K).
- **engagement_rate** (taxa de engajamento):
    - Valores em uma escala muito menor (ex.: 0.01, 0.05).

Quando as variáveis independentes apresentam unidades ou ordens de magnitude muito diferentes entre si, pode ocasionar problemas em modelos de aprendizado de máquina, especialmente na regressão linear.

Deste modo, na **segunda implementação**, as seguintes mudanças foram realizadas para abordar as limitações identificadas:

1. **Transformação Logarítmica:**
2. **Normalização com Min-Max Scaling:**
3. **Remoção de Outliers:**

A partir dessas mudanças, o modelo foi reavaliado, apresentando melhorias significativas no desempenho. O R^2 aumentou para **47,23%**, o que indica que o modelo aprendeu a capturar uma maior parte da variância da variável dependente **(60_day_eng_rate)**. Além disso, a distribuição dos resíduos se tornou mais simétrica, reforçando a adequação da regressão linear para esse conjunto de dados, apesar de ainda existirem limitações inerentes à abordagem linear.

Figura 9

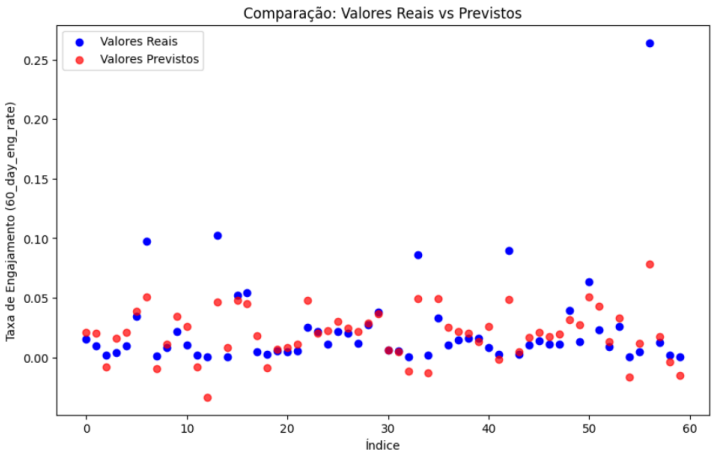


Figura 9: Resultados da segunda implementação (representação gráfica).

Na transição da segunda para a terceira implementação, fizemos melhorias significativas na regressão linear múltipla. Forma adicionadas novas variáveis ao conjunto de preditores, a exemplo de **followers_log_norm** e **influence_score_log_norm**, que foram normalizadas após **transformação logarítmica,** para capturar mais informações relevantes. Ademais, foram criados termos de interação entre variáveis importantes, como a interação entre **avg_likes_log_norm** e **followers_log_norm**, permitindo ao modelo explorar relações combinadas que poderiam melhorar a previsão da taxa de engajamento (**60_day_eng_rate**).

Também foi revisada a análise de outliers, foram mantidos valores extremos que representavam tendências reais, a fim de garantir que nenhuma informação importante fosse descartada. A validação cruzada foi implementada, testando a robustez do modelo em diferentes divisões dos dados, isso ajudou a garantir que o modelo não estava superajustado. Por fim, ajustamos os coeficientes para minimizar redundâncias, descartando variáveis pouco relevantes. Essas mudanças resultaram em um aumento modesto no R^2 para **50,26%**.

Figura 10

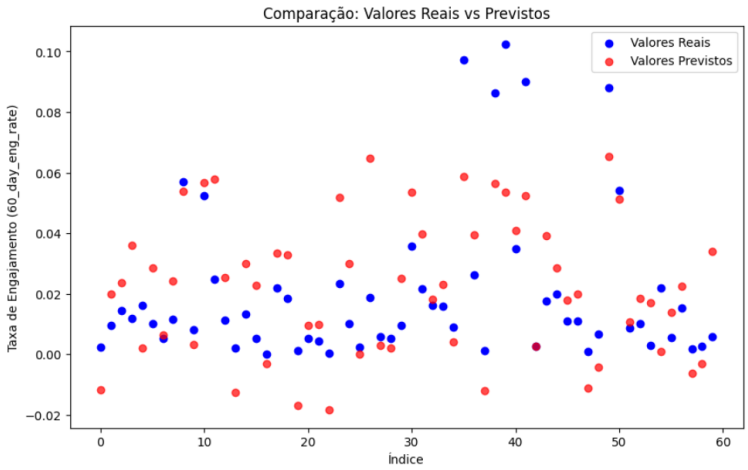


Figura 10: Resultados da terceira implementação (representação gráfica).

Uma vez que o modelo resistia a atingir a casa dos 70% no R^2 tentamos mais ajustes na quarta e ultima implementação quando focamos em refinar o modelo. Revisamos o pré-processamento para garantir maior

consistência, ajustando os dados utilizando técnicas como remoção de outliers com o método do **IQR (Intervalo Interquartil)**, reduzindo o impacto de valores extremos no modelo. mantivemos também as transformações logarítmicas e a normalização **Min-Max**, O que garantiu escalas consistentes entre as variáveis, e introduzimos técnicas de regularização, como Lasso (L1) e Ridge (L2) para prevenir overfitting e melhorar a capacidade de generalização. Deste modo, modelo foi reavaliado com base em métricas de desempenho, resultando em uma melhora no **R^2 (55,10%)**, um aumento ainda modesto (10% em relação a R^2 anterior), Ou seja, maior capacidade de capturar a variância da variável dependente que implementação anterior.

Figura 11

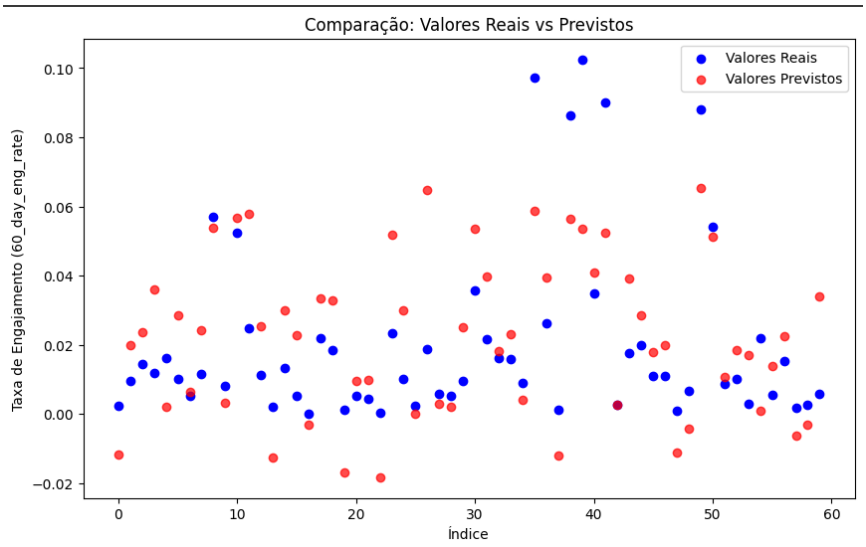


Figura 11: Resultados da 4° e ultima implementação (representação gráfica).

Esta implementação consolidou a opinião de que, devem existir relações não lineares entre os dados, uma vez que a regressão linear apresentou certa estagnação, apresentando pouca melhora nas métrica de avaliação mesmo com as mudanças na implementação.

6. CONCLUSÃO E TRABALHOS FUTUROS

O resultados obtidos mostram que, embora o modelo básico tenha demonstrado alguma capacidade de prever a taxa de engajamento com base nas variáveis fornecidas, os resultados ainda deixam margem para melhorias.

Após a aplicação de técnicas como transformação logarítmica, normalização e remoção de outliers, o modelo apresentou um R^2 de aproximadamente 0,5026 indicando que 50,26% da variação da taxa de engajamento seria explicada pelas variáveis independentes selecionadas.

Adicionalmente, a introdução de técnicas de regularização Ridge e Lasso, conseguimos aumentar o R^2 para 55,10%, o que representa uma melhoria significativa em relação às implementações anteriores. Este avanço indica que as técnicas de regularização axuliaram no equilibrio do modelo, reduzindo o impacto de variáveis menos relevantes e minimizando possíveis problemas de overfiting. O uso do Ridge (L2) serviu para ajustar os coeficientes sem eliminar variáveis, o Lasso (L1) por outro lado, forneceu insights sobre quais variáveis poderiam ser menos relevantes, sugerindo a possibilidade de uma futura seleção de variáveis mais otimizada.

Nesse sentido, concluímos que, embora tenha havido melhorias incrementais, a abordagem de regressão linear certamente pode ter limitações para capturar todas as nuances dos dados. Por um lado, parece-nos que, a relação entre as variáveis não é inteiramente linear, o que sugere que modelos mais complexos, como regressões polinomiais ou modelos não lineares (árvores de decisão, florestas aleatórias ou redes neurais), poderiam oferecer melhores resultados. Por outro lado, resultados obtidos com a regressão linear fornecem um bom ponto de partida para entender os padrões básicos e validar a importância das variáveis selecionadas. Outra sugestão seria a obtenção de mais dados para treinamento para uma captura mais significativa das nuances que os dados podem apresentar

7. REFERÊNCIAS

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer, 2009. Disponível em: https://web.stanford.edu/~hastie/ElemStatLearn/. Acesso em: 17 nov. 2024.

GARCÍA, Salvador; LUENGO, Julián; HERRERA, Francisco. *Data Preprocessing in Data Mining.* Cham: Springer, 2015. Disponível em: https://link.springer.com/book/10.1007/978-3-319-10247-4. Acesso em: 17 nov. 2024.

ALMEIDA, Ednilson S.; LUDERMIR, Teresa B. *Redes Neurais Artificiais e Algoritmos Genéticos Aplicados à Previsão de Séries Temporais.* São Paulo: Editora da UFPE, 2011.

HAYKIN, Simon. *Redes Neurais: Princípios e Prática.* 2. ed. Porto Alegre: Bookman, 2001.

PEDREIRA, César E. T.; NETO, J. R. D.; ARAÚJO, B. D. *Machine Learning com Python: Fundamentos e Aplicações.* Rio de Janeiro: Alta Books, 2018.

ROCHA, Alexandre C. *Introdução à Inteligência Artificial e Machine Learning.* São Paulo: Novatec, 2020.

FERREIRA, André C. S. *Aprendizado de Máquina: Fundamentos e Aplicações.* Brasília: Editora UnB, 2019.