Relatório Técnico: Implementação e Análise do Algoritmo k-Nearest Neighbors (kNN) Aplicado ao Instagram

Lucas Silva de Oliveira e Samy Celes Barrozo

17/11/2024

Resumo

Este projeto proposto para os residentes da Trilha de Ciência de Dados da Restic36 tem como principal objetivo implementar e avaliar o desempenho do algoritmo k-Nearest Neighbors (kNN) em uma base de dados que reúne informações dos maiores influencers do Instagram para explorar relações entre variáveis como número de seguidores, curtidas médias e taxas de engajamento. Nesse cenário, o processo de construção do projeto incluiu etapas de análise exploratória de dados, implementação e validação do modelo, bem como otimização de hiperparâmetros para maximizar seu desempenho.

Os dados foram transformados para possibilitar a condução da análise, sendo que os valores categóricos foram transformados em uma faixa numérica constituída para representar a região dos continentes. Todas as variáveis foram normalizadas e além de tratamento dos valores ausentes (NaN) também foram testados diferentes valores de k e as métricas de distância. A avaliação do modelo foi realizada utilizando MAE, MSE e RMSE, bem como validação cruzada para garantir que os resultados fossem consistentes.

No relatório a análise do impacto de variáveis-chave é descrita de forma detalhada, incluindo a relação entre seguidores e as médias das curtidas, além de visualizar os insights obtidos através de gráficos e tabelas. Finalmente, as conclusões fornecem um desenho geral do desempenho do modelo e seu potencial para resolver problemas semelhantes no contexto real.

Introdução

Dentro da concorrência das redes sociais, como o Instagram, a análise da performance e influência de criadores de conteúdo se mostra imprescindível para as marcas e para os profissionais de marketing. O reconhecimento de padrões no comportamento de métricas como seguidores, engajamento, e curtidas médias pode contribuir para a categorização dos influenciadores, prever tendências e otimizar campanhas publicitárias.

O algoritmo k-Nearest Neighbors (kNN) foi o escolhido para esta análise por dois motivos: simplicidade e desempenho para tarefas de classificação e regressão, além de seu comportamento em detectar padrões locais nos dados. O kNN não requer premissas paramétricas sobre os dados, tornando-o ótimo para o conjunto em questão, o qual exibe relações não lineares entre variáveis, tais como seguidores, curtidas médias e taxas de engajamento. A flexibilidade na escolha das métricas de distância, também favorecida por mecanismos de otimização, garante um modelo adequado às características do problema.

Nesse panorama, o conjunto de dados utilizado, Top Instagram Influencers Data (Cleaned), contém informações detalhadas sobre os principais influenciadores do Instagram, permitindo uma rica perspectiva analítica sobre suas métricas de sucesso. As principais variáveis do dataset incluem:

1. **Rank**: a classificação do influenciador está relacionada ao número de seguidores.
2. **Influence Score**: métrica que reflete o poder de influência do influenciador.
3. **Posts**: número total de posts do influenciador.
4. **Followers**: número total de seguidores do influenciador.
5. **Avg Likes**: média de curtidas nos posts.
6. **60 Day Engagement Rate**: taxa de engajamento nos últimos 60 dias.
7. **New Post Avg Like**: média de curtidas nos posts mais recentes.
8. **Total Likes**: O total de curtidas em todos os posts.
9. **Country**: O país de origem do influenciador.

Além disso, os dados foram enriquecidos, incluindo faixas numéricas com base nos continentes, quanto aos quais foram atribuídos valores como América do Sul (1-9), América do Norte (20-29), e Europa (40-49). Esta estrutura possibilita uma melhora nas análises regionais e compara influenciadores de diferentes continentes, o que confere um plus a este conjunto de dados para observar como métricas de engajamento e alcance se relacionam no contexto das redes sociais que dominam o mundo atual.

Metodologia

A análise exploratória inicial dos dados teve como objetivo compreender as principais

características do dataset dos influencers do Instagram, identificar padrões nas variáveis-chave e destacar possíveis problemas a serem tratados antes da modelagem. Nesse cenário, foram utilizados histogramas, boxplots e gráficos de dispersão para obter insights e observar relações entre variáveis.

Insights iniciais:

- **Distribuição das variáveis (histogramas)**:
    - As colunas posts, followers, avg_likes, total_likes, 60_day_eng_rate, e new_post_avg_like apresentaram **assimetria positiva**, com a maioria dos valores concentrados em intervalos menores e caudas longas para valores maiores.
    - Alguns intervalos sem valores indicam lacunas nos dados, sugerindo que certos padrões ou grupos podem estar ausentes.

- **Identificação de outliers (boxplots)**:
    - Foram detectados **outliers em todas as colunas**, representando influenciadores com valores extremos, como número muito alto de seguidores, curtidas, ou taxa de engajamento.
    - Esses outliers são importantes para análise, mas também podem influenciar negativamente a performance do modelo se não tratados adequadamente.

Correlação entre variáveis

- Foi observado que as variáveis avg_likes, 60_day_eng_rate, e new_post_avg_like possuem a **maior correlação entre si**, indicando uma relação próxima entre a média de curtidas e o engajamento.
- O 60_day_eng_rate **(taxa de engajamento)** foi definido como o **alvo do modelo**, dado seu papel como indicador-chave da performance do influenciador.

Relações observadas em gráficos de dispersão

1. Followers x Avg_likes:
    - Existe uma relação positiva: perfis com mais seguidores tendem a ter uma média de curtidas maior.
2. Avg_likes x 60_day_eng_rate:

    - Perfis com uma maior média de curtidas (avg_likes) apresentam também uma taxa de engajamento (60_day_eng_rate) mais elevada.
3. New_post_avg_like x 60_day_eng_rate:
    - Semelhante à relação anterior, uma maior média de curtidas em novas postagens está associada a um engajamento mais alto.
4. Posts x Followers x Avg_likes:
    - Ao adicionar a variável posts, foi observado que perfis com poucos seguidores:
        - Tendem a ter uma baixa média de curtidas.
        - Geralmente possuem menos postagens, sugerindo menor atividade na plataforma.

Tratamento de valores ausentes

- A abordagem inicial para lidar com valores NaN foi a substituição pela moda. No entanto, isso demonstrou ser prejudicial à distribuição das variáveis.
- **Impacto da substituição pela moda**:
    - A substituição resultou em uma concentração excessiva de valores ao redor da moda, distorcendo a variabilidade natural dos dados.
    - No contexto do modelo KNN, essa distorção interfere nas distâncias calculadas, tornando-as artificialmente semelhantes e, potencialmente, degradando a qualidade do modelo.

Implementação do Algoritmo e Ajustes

Configurações do kNN

O algoritmo k-Nearest Neighbors (kNN) foi implementado para prever a métrica de engajamento de 60 dias (60_day_eng_rate) com base em variáveis independentes altamente correlacionadas, como avg_likes e new_post_avg_like. A abordagem para implementação seguiu as etapas descritas abaixo:

1. **Remoção de Outliers com IQR (Interquartile Range)**:
   Antes da aplicação do modelo, os outliers foram removidos utilizando o IQR. Para cada variável, foram calculados os limites inferior e superior, definidos como:

Limite Inferior = $Q1 - 1.5 \times IQR$

Limite Superior = $Q3 + 1.5 \times IQR$

Dados fora desses limites foram eliminados, resultando em um dataset mais robusto para treinamento do modelo.

2. **Treinamento Inicial do kNN**:
   Foi utilizado o regressor kNN, inicialmente configurado com:
   - n_neighbors = 5 (5 vizinhos mais próximos);
   - Métrica padrão de distância Euclidiana.
3. As variáveis independentes selecionadas foram avg_likes e new_post_avg_like, com o alvo sendo 60_day_eng_rate. A separação dos conjuntos de treino e teste foi realizada com holdout (20% para teste) usando train_test_split.

4. **Validação e Métricas de Avaliação**:
   As métricas calculadas para avaliar o desempenho do modelo incluíram:
   - **Erro Médio Absoluto (MAE)**: mede o erro médio entre os valores reais e previstos.
   - **Erro Quadrático Médio (MSE)** e **Raiz do Erro Quadrático Médio (RMSE)**: Avaliam a magnitude dos erros considerando penalizações maiores para erros extremos.
5. **Resultados Iniciais** (sem otimização):
   - MAE: 0.0018
   - MSE: 0.00000597
   - RMSE: 0.0024

A acurácia média na validação cruzada foi de 71,4%, com um desvio padrão de 14,3%, sugerindo inconsistência em alguns *folds,* mas um desempenho razoável do modelo.

Importante observar que durante a análise dos dados, foi observado que a variável country continha valores NaN. Esses valores foram removidos para permitir o mapeamento adequado dos países a um código numérico baseado em seus respectivos continentes. A lista de países únicos presentes no conjunto de dados foi obtida utilizando:

```
# Tive que usar o dropna porque existe valores na coluna country NaN
countrys = sorted(df['country'].dropna().unique())
countrys
```

Criação do Mapeamento por Continente

Para facilitar a análise, foi criado um dicionário chamado continent_mapping, no qual cada continente foi associado a um conjunto de países com seus respectivos códigos numéricos. Esses códigos foram definidos manualmente com base nos requisitos do projeto e no conjunto de dados disponíveis. O mapeamento ficou estruturado da seguinte forma:

- **South America**:
  - Brazil (1), Colombia (2), Uruguay (3)
- **North America**:
  - Canada (20), Mexico (21), United States (22), Puerto Rico (23), Anguilla (24), British Virgin Islands (25)
- **Europe**:
  - France (40), Germany (41), Czech Republic (42), Italy (43), Netherlands (44), Russia (45), Spain (46), Sweden (47), Switzerland (48), United Kingdom (49), Turkey (50)
- **Asia**:
  - India (60), Indonesia (61), United Arab Emirates (62)
- **Africa**:
  - Côte d'Ivoire (80)
- **Oceania**:
  - Australia (90)

Transformação da Variável

Uma função personalizada, country_to_continent_mapping, foi implementada para realizar a conversão da variável country de valores categóricos para os códigos numéricos definidos. A função percorre o mapeamento e retorna o código numérico correspondente ao país. Caso o país não esteja no dicionário, a função retorna None. O código foi aplicado à coluna country:

```
# Aplica a transformação da coluna 'country' usando a função map_country_to_continent
df['country'] = df['country'].map(country_to_continent_mapping)
```

Após a transformação, a coluna country passou a conter valores numéricos representando os países por continente.

Padronização de Outras Colunas

Além do tratamento da variável country, outras colunas como posts, followers, avg_likes, new_post_avg_like, total_likes e 60_day_eng_rate passaram por processos de transformação e padronização para garantir consistência nos dados. Foram aplicadas as seguintes ações:

- **Conversão de Formatos Numéricos**: Valores com sufixos como m, k ou b foram convertidos para seus equivalentes numéricos.
- **Conversão da Taxa de Engajamento**: A coluna 60_day_eng_rate, originalmente em formato percentual, foi convertida para números decimais.

Otimização do Modelo (sem normalização)

Para melhorar a performance da implementação do algoritmo kNN, foi utilizado o GridSearchCV para encontrar os melhores hiperparâmetros.

1. **Hiperparâmetros Testados**:
    - n_neighbors: Variou entre 1 e 24.
    - Métrica de distância: Testadas as opções euclidean, minkowski, manhattan e chebyshev.
2. **Melhores Resultados Encontrados**:
    - Métrica: euclidean
    - Número de vizinhos: 10
3. **Aplicação do Modelo Otimizado**:
   Após otimização, o modelo apresentou métricas de avaliação similares, mas com melhor estabilidade:
    - MAE: 0.0018
    - MSE: 0.00000612
    - RMSE: 0.0024

4. A acurácia média na validação cruzada subiu para 72,7%, com uma redução no desvio padrão para 13,3%.

Normalização das Variáveis

Foi aplicada a normalização com MinMaxScaler para padronizar as variáveis independentes entre 0 e 1. A normalização foi essencial para evitar que variáveis com escalas diferentes influenciassem a distância calculada no kNN.

1. **Comparação Antes e Depois da Normalização**:
    - Antes: A acurácia média foi de 72,7%.
    - Depois: A acurácia média aumentou para **76,8%**, com um desvio padrão de 13,5%.
    - Houve uma leve melhoria na estabilidade e desempenho do modelo, demonstrando que a normalização foi benéfica.

Resultados

Métricas de Avaliação:

Após a implementação do modelo **kNN Regressor** com diferentes configurações, as métricas de avaliação foram analisadas para verificar o desempenho do modelo. Foram consideradas as seguintes métricas:

1. MAE (Mean Absolute Error): Mede a média dos erros absolutos entre os valores previstos e os reais. Um valor menor de MAE indica melhor desempenho.
2. MSE (Mean Squared Error): Calcula o erro quadrático médio, penalizando erros maiores. Um MSE baixo é desejável, pois indica que o modelo está mais próximo dos valores reais.
3. RMSE (Root Mean Squared Error): É a raiz quadrada do MSE e fornece uma interpretação no mesmo nível dos valores do target, ajudando na análise da precisão.

Resultados obtidos com a aplicação do kNN após a normalização das variáveis independentes:

- Erro médio absoluto (MAE): 0.0019303030303030307

- Erro médio quadrático (MSE): 6.8165381083562905e-06
- Raiz do Erro médio quadrático (RMSE): 0.002610850073894763

Esses resultados indicam que o modelo apresenta um erro muito baixo nas previsões, mostrando que ele é capaz de capturar os padrões no conjunto de dados com eficiência.

Validação Cruzada (kNN após a normalização):

A validação cruzada foi utilizada para avaliar a robustez do modelo no dataset. A métrica analisada foi a acurácia em 5 folds:

- **Acurácia por fold**: [0.6576, 0.9613, 0.7419, 0.8805, 0.6001]
- **Acurácia média**: 0.7683
- **Desvio padrão**: 0.1350

Interpretação:

- A média da acurácia de 76,8% mostra que o modelo apresenta um desempenho razoável.
- O desvio padrão relativamente alto (13,5%) indica variação entre os folds, sugerindo sensibilidade do modelo à escolha dos dados. Isso pode ser atribuído à variabilidade nos padrões de dados entre os subconjuntos.

Assim, pode-se concluir que o modelo **kNN Regressor** foi testado em diferentes configurações (sem normalização, com normalização e otimização por GridSearch) e apresentou os seguintes resultados como destaques:

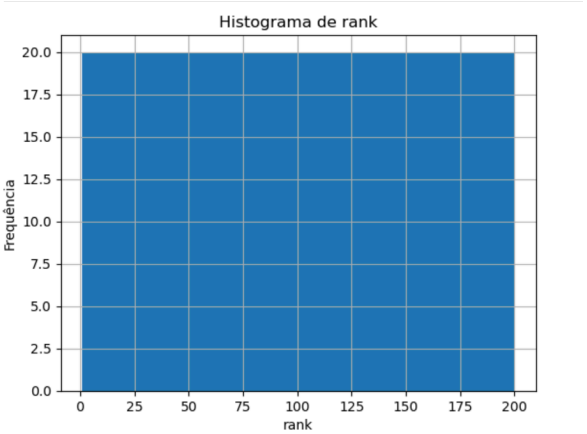
- **Erro Baixo**: MAE, MSE e RMSE muito baixos indicam que o modelo está realizando previsões satisfatórias.
- **Acurácia Razoável**: Com uma acurácia média de 76,8%, o modelo consegue capturar padrões gerais nos dados.
- **Sensibilidade a Dados**: O desvio padrão alto na validação cruzada revela a necessidade de explorar ajustes adicionais nos dados ou no modelo para garantir maior consistência.

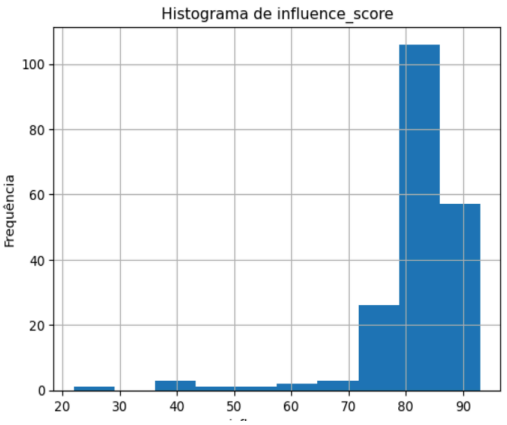
Nesse cenário, estratégias como a seleção cuidadosa das variáveis com alta correlação com o target foram fundamentais para obter um desempenho satisfatório. Apesar da variabilidade

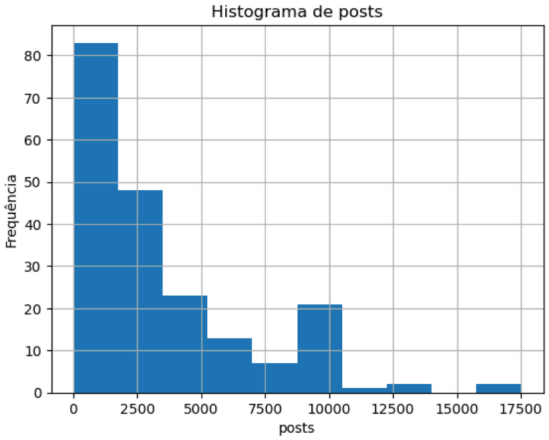
entre os folds, os resultados são promissores, indicando que o modelo é uma abordagem válida para o problema proposto.

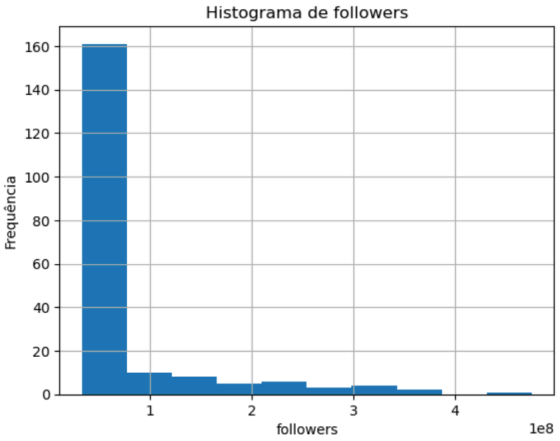
Visualização dos Resultados:

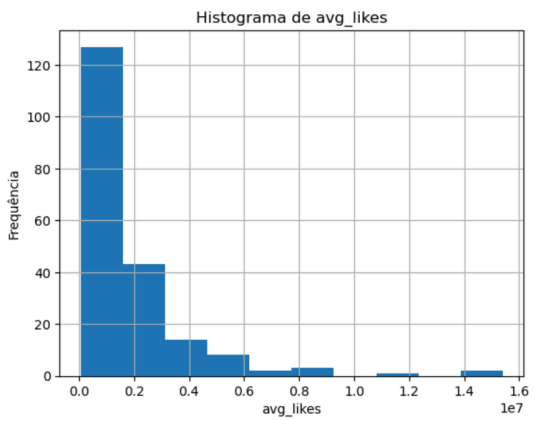
Histogramas para análise exploratória dos dados

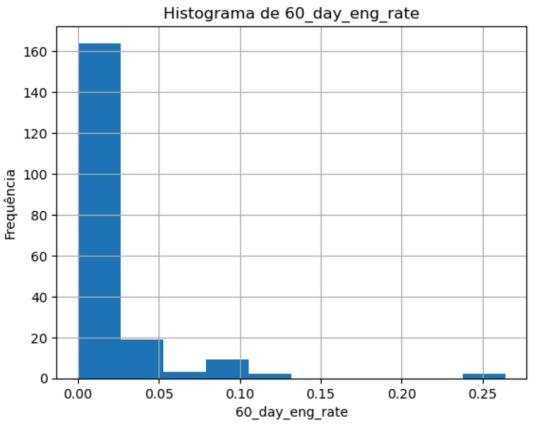


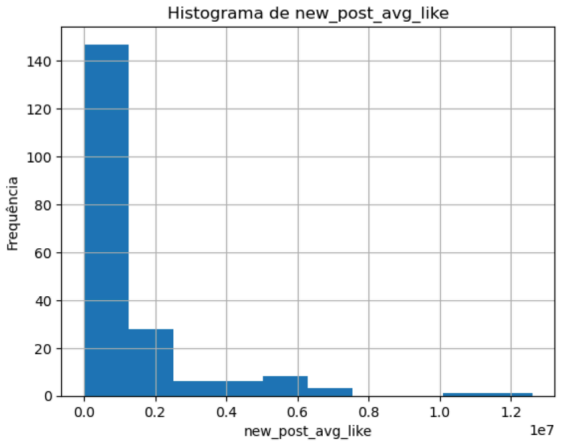


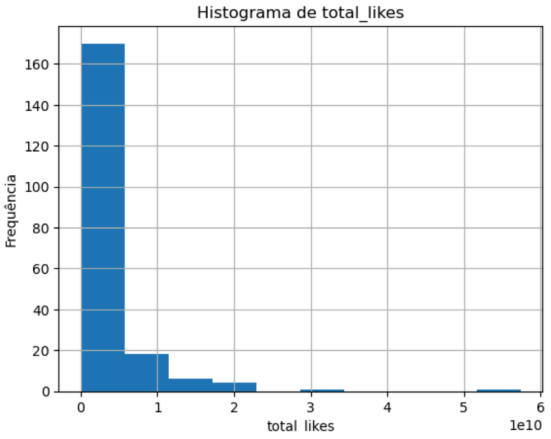


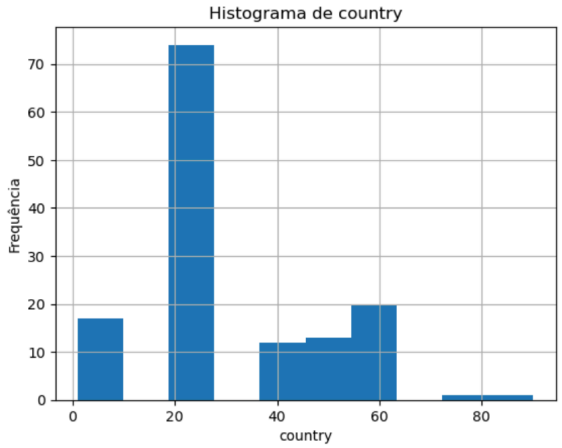



Histograma de avg_likes
Frequência
avg_likes
1e7

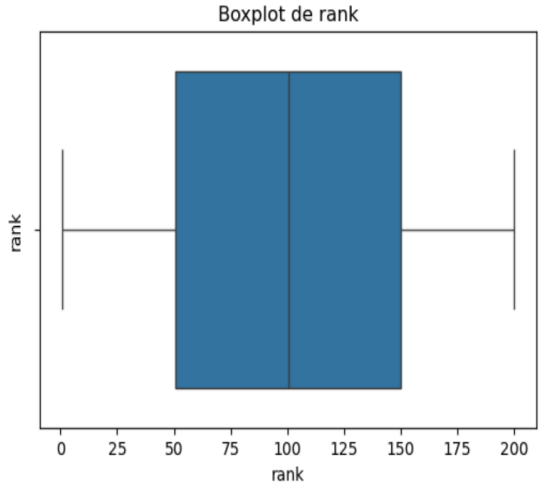

Histograma de 60_day_eng_rate
Frequência
60_day_eng_rate

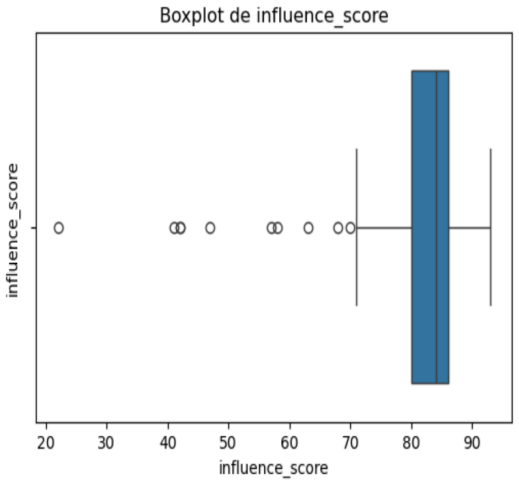

Histograma de new_post_avg_like
Frequência
new_post_avg_like
1e7

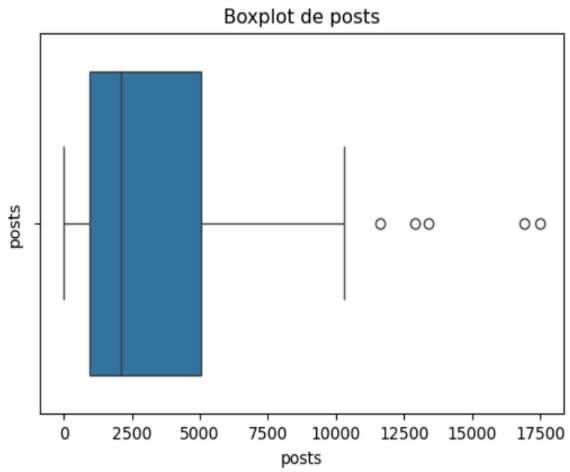

Histograma de total_likes
Frequência
total_likes
1e10

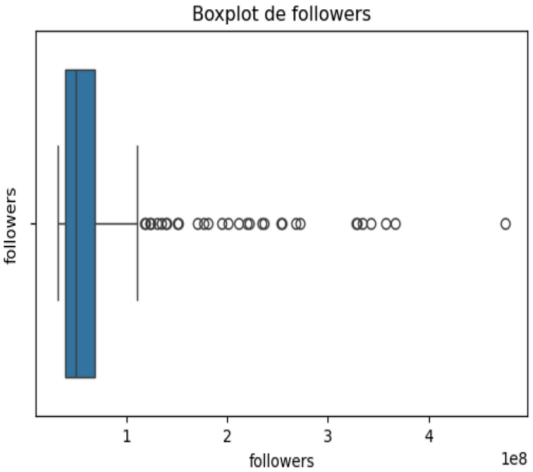

Histograma de country
Frequência
country

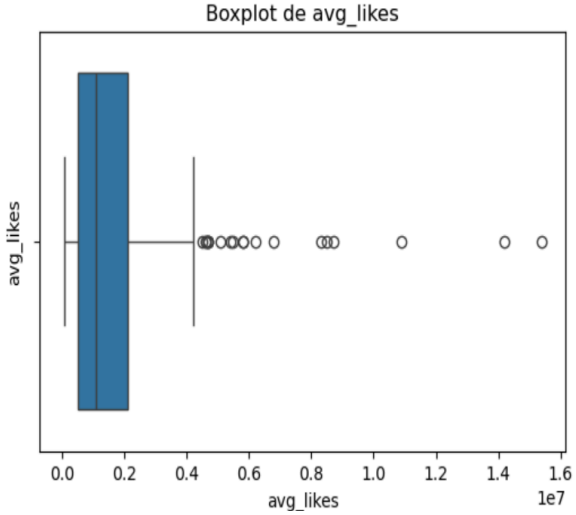
Boxplots para análise da distribuição inicial dos dados

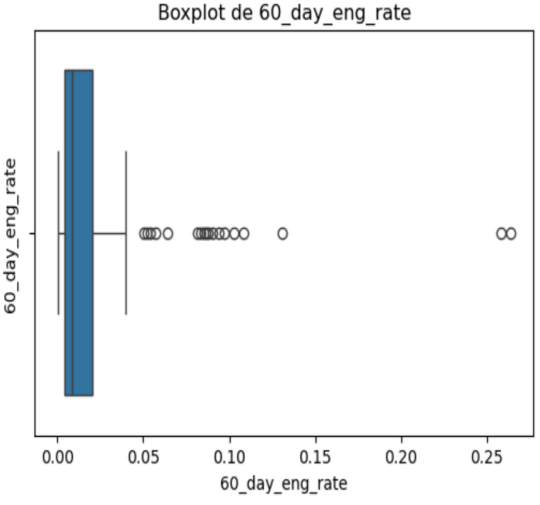


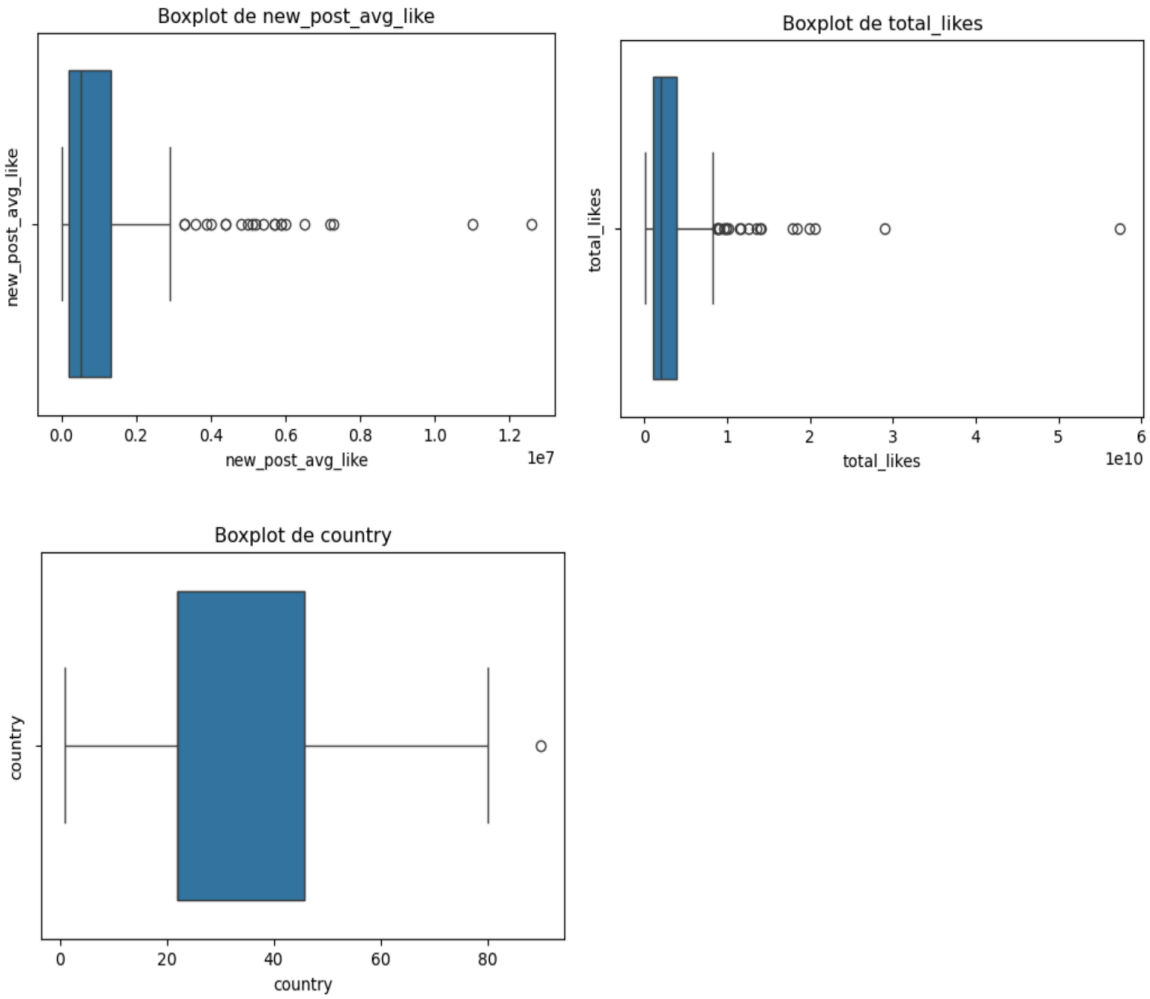










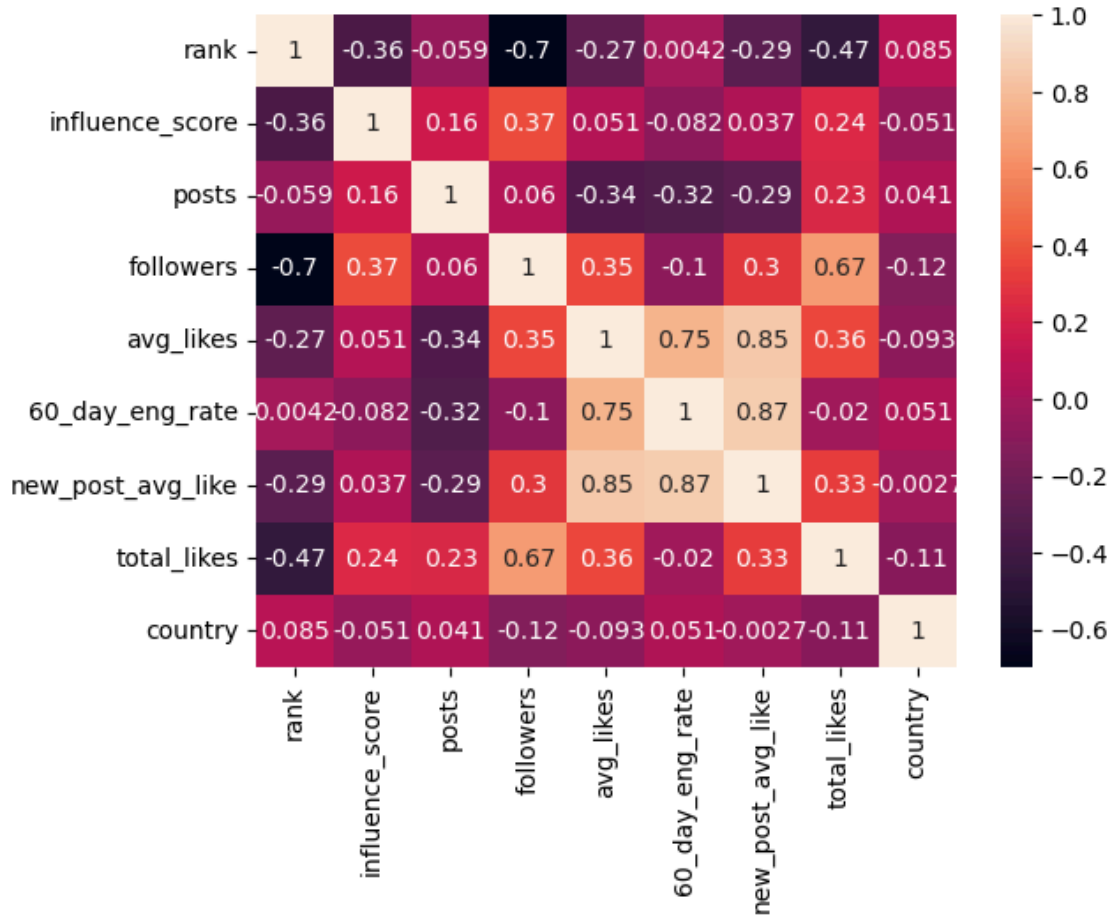


A análise gráfica por meio de histogramas revelou as seguintes características:

- Frequência Concentrada: Há uma maior frequência de valores concentrados em intervalos específicos para as variáveis analisadas, indicando uma tendência de centralização dos dados em faixas delimitadas.
- Ausência de Valores em Certos Intervalos: Alguns intervalos não apresentam dados, sugerindo lacunas ou padrões específicos na distribuição das variáveis.
- Assimetria Positiva: As colunas *posts*, *followers*, *avg_likes*, *total_likes*, *60_day_eng_rate*, e *new_post_avg_like* exibem uma cauda alongada à direita, indicando que a maioria dos valores está concentrada em faixas menores, mas há valores extremos significativamente maiores.

Os boxplots permitiram identificar a presença de outliers em todas as variáveis do dataset. Esses valores extremos podem influenciar métricas como média e desvio padrão, exigindo atenção durante o treinamento do modelo e a análise de resultados.

Matriz de correlação

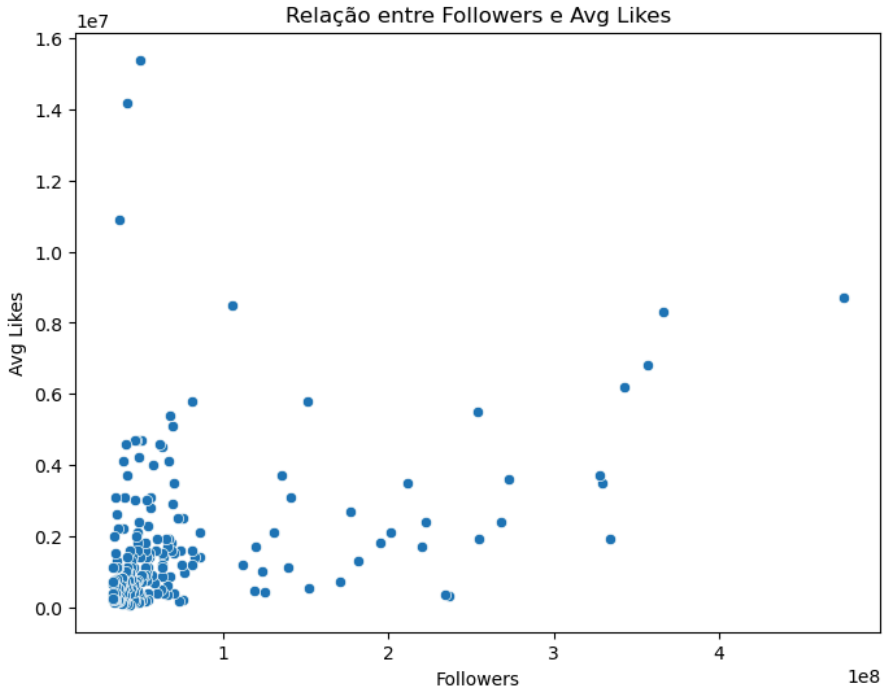


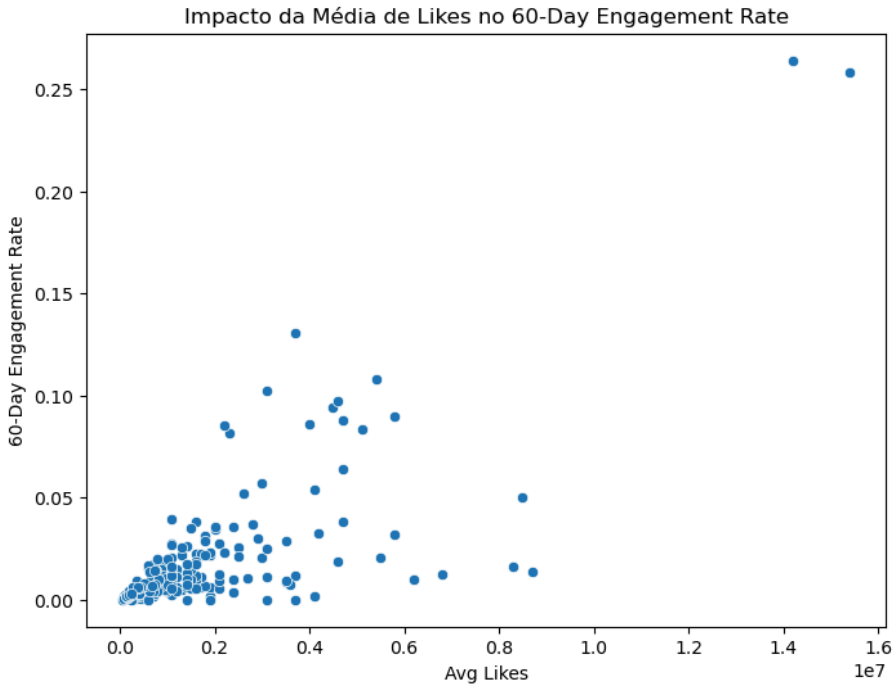
A matriz de correlação gerada permitiu identificar as relações entre as diversas variáveis presentes no dataset. Os principais pontos observados foram:

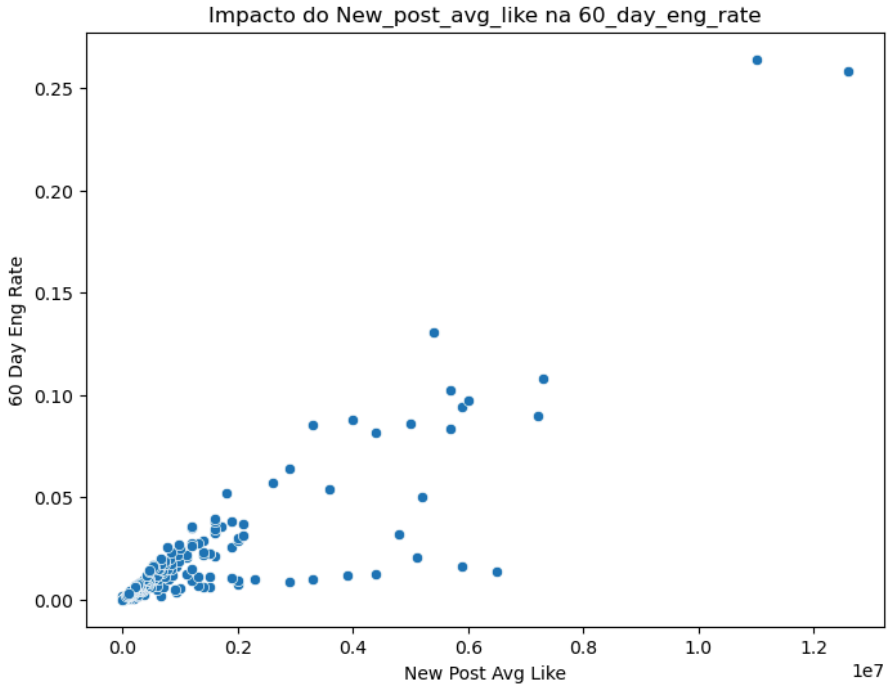
1. **Correlação Significativa:**
    - As colunas **avg_likes**, **60_day_eng_rate**, e **new_post_avg_like** apresentam as maiores correlações entre si, indicando uma relação direta e positiva. Isso sugere que o comportamento de uma dessas variáveis tende a influenciar as demais de maneira consistente.
    - Essa forte correlação pode ser reflexo de uma interação natural entre o número médio de curtidas e a taxa de engajamento, o que reforça a importância dessas variáveis para o modelo.
2. **Seleção do Target:**
    - O **60_day_eng_rate** (taxa de engajamento) foi escolhido como a variável *target* do modelo. Essa decisão é fundamentada na relevância dessa métrica

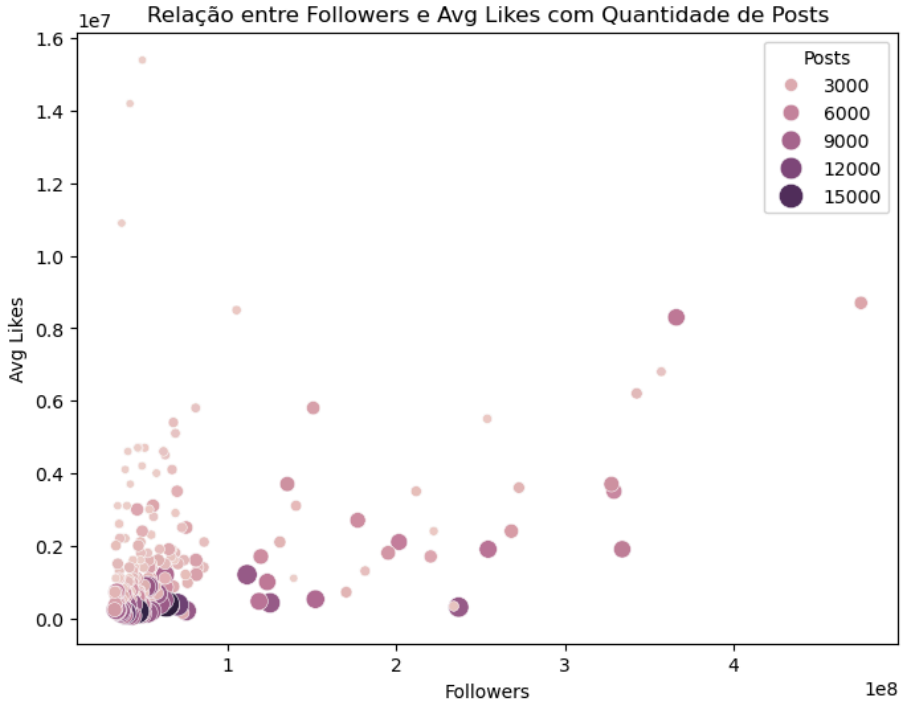
para análises de performance, especialmente em contextos relacionados a redes sociais, marketing ou comportamento de usuários.

Gráficos de Dispersão (Scatter Plot)









No gráfico followers x avg_likes é possível observar uma relação direta entre o número de seguidores (**followers**) e a média de curtidas (**avg_likes**) em posts. Assim, perfis com um maior número de seguidores tendem a apresentar uma média de curtidas mais alta. Essa relação é esperada em plataformas sociais, pois perfis com maior alcance geralmente geram mais interações.

Já no gráfico avg_likes x 60_day_eng_rate**,** observa-se que perfis com uma maior média de curtidas (**avg_likes**) tendem a ter uma taxa de engajamento de 60 dias (**60_day_eng_rate**) também mais elevada, o que reforça a correlação identificada na matriz de correlação. Esse comportamento destaca a relevância da média de curtidas como uma métrica preditiva para o engajamento, sendo um indicativo de público ativo e interações de qualidade.

Similar ao gráfico anterior, o gráfico new_post_avg_like x 60_day_eng_rate demonstra que perfis com uma alta média de curtidas em novos posts (**new_post_avg_like**) também possuem uma taxa de engajamento mais elevada (**60_day_eng_rate**). Essa análise reforça a ideia de que a capacidade de engajamento do perfil está diretamente relacionada ao comportamento de interação nos posts recentes, sendo um indicador de relevância no curto prazo.

Por fim, podemos observar que, conforme o gráfico followers x avg_likes, perfis com poucos seguidores (**followers**) geralmente apresentam uma baixa média de curtidas (**avg_likes**) e também possuem poucos posts (**posts**). À medida que o número de seguidores aumenta, é possível notar uma maior dispersão no número de posts, mas com uma tendência geral de aumento na média de curtidas. Assim, fica claro que perfis menores tendem a ter menos interações e produzem menos conteúdo, enquanto perfis maiores, além de alcançarem mais curtidas, demonstram maior variabilidade no volume de conteúdo produzido.

Tratamento de Valores NaN para implementação do KNN

Através do método df.info(), foi possível identificar que a variável country possui 62 valores ausentes, enquanto a coluna 60_day_eng_rate apresenta apenas 1 valor ausente:

In [174…

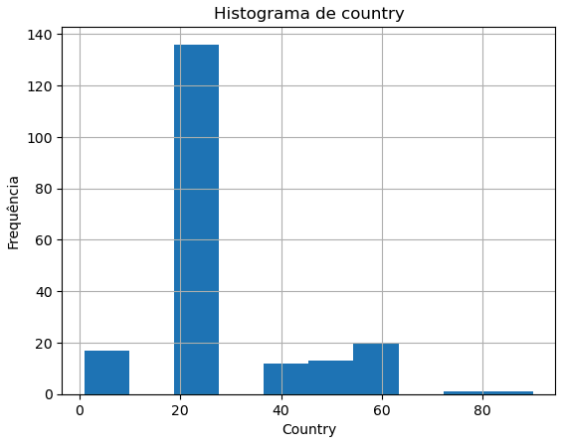
```
# O KNN não pode ser implementado enquanto houver valores NaN no dataset
df.info()
```

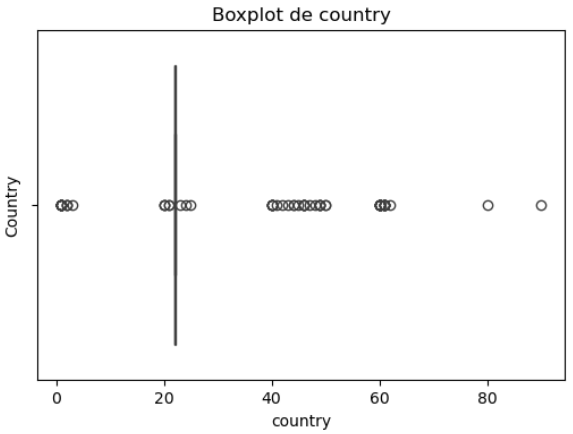
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   rank              200 non-null    int64  
 1   channel_info      200 non-null    object 
 2   influence_score   200 non-null    int64  
 3   posts             200 non-null    float64
 4   followers         200 non-null    float64
 5   avg_likes         200 non-null    float64
 6   60_day_eng_rate   199 non-null    float64
 7   new_post_avg_like 200 non-null    float64
 8   total_likes       200 non-null    float64
 9   country           138 non-null    float64
dtypes: float64(7), int64(2), object(1)
memory usage: 15.8+ KB
```

Tratamento da Coluna 60_day_eng_rate

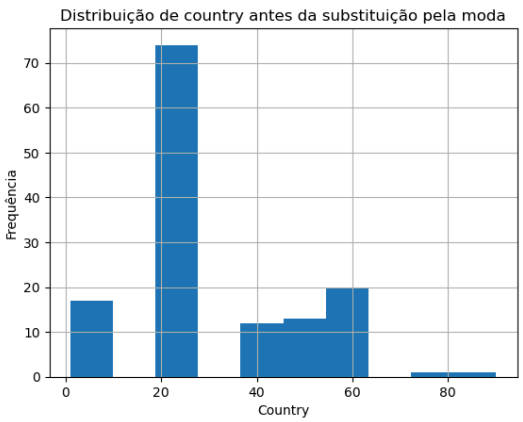
- Optou-se por substituir o valor ausente pela mediana da coluna. Essa decisão foi tomada para minimizar o impacto de outliers, garantindo que o valor imputado esteja próximo da distribuição central da variável.

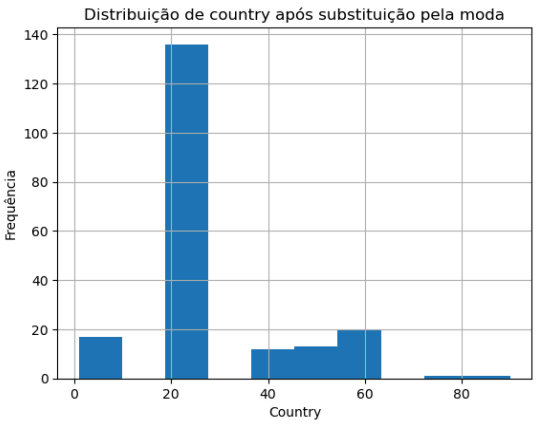
Análise Inicial da coluna country





- Para testar o impacto de preencher os valores ausentes, foi criada a variável country_test_mode, onde os valores NaN foram substituídos pela moda da coluna.





- **Resultados Observados:**
  - Os gráficos revelam que a substituição dos valores NaN pela moda não trouxe benefícios significativos para a análise e, pelo contrário, distorce a distribuição original, prejudicando a variabilidade.

No contexto do KNN, o preenchimento de valores ausentes pela moda pode resultar em distâncias artificiais entre as observações, pois muitas delas passam a compartilhar um valor comum, introduzindo vieses nas previsões do modelo. A solução foi remover todas as linhas que possuem valores NaN na coluna country:

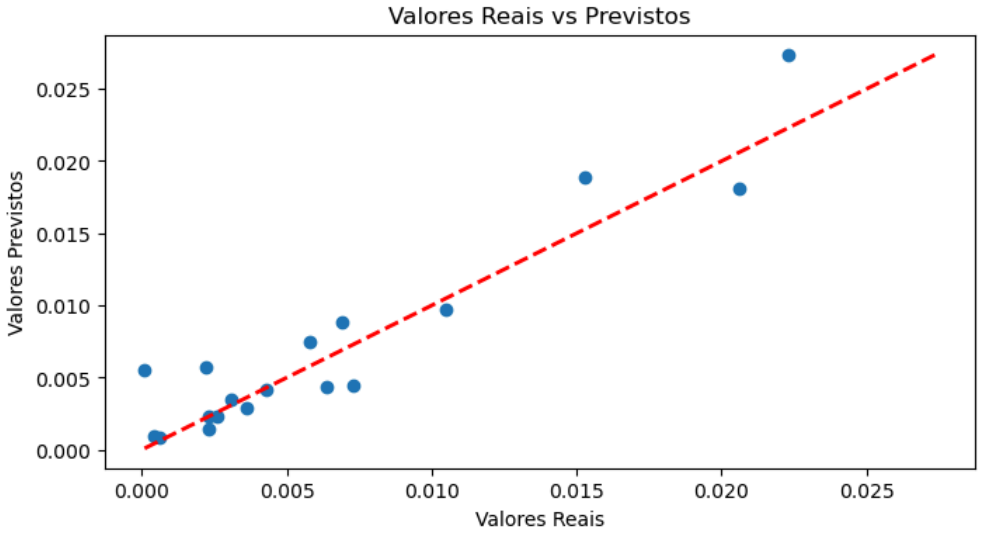
```
# Dropo todas as linhas que têm NaN na coluna 'country'
df = df.dropna(subset=['country'])

df.isnull().sum()
```

Esta exclusão garante que a variabilidade original da variável seja mantida, evitando distorções que possam impactar negativamente o desempenho do modelo. Após o tratamento, o dataset ficou completamente limpo de valores ausentes.

Implementação do KNN sem normalização:

- Erro médio absoluto (MAE): 0.0018066666666666663
- Erro médio quadrático (MSE): 5.972044444444443e-06
- Raiz do Erro médio quadrático (RMSE): 0.002443776676467071



MAE (Erro Médio Absoluto): indica o erro médio absoluto entre os valores reais e os previstos. Um valor baixo como 0.0018 mostra que as previsões são muito próximas dos valores reais. Já o MSE (Erro Médio Quadrático), penaliza erros maiores, mas como o valor é muito baixo (5.972e-06), isso reforça que o modelo tem um desempenho adequado. O RMSE

(Raiz do Erro Médio Quadrático), por outro lado, traduz o erro para a mesma escala do target. O valor 0.0024 confirma a qualidade do ajuste.

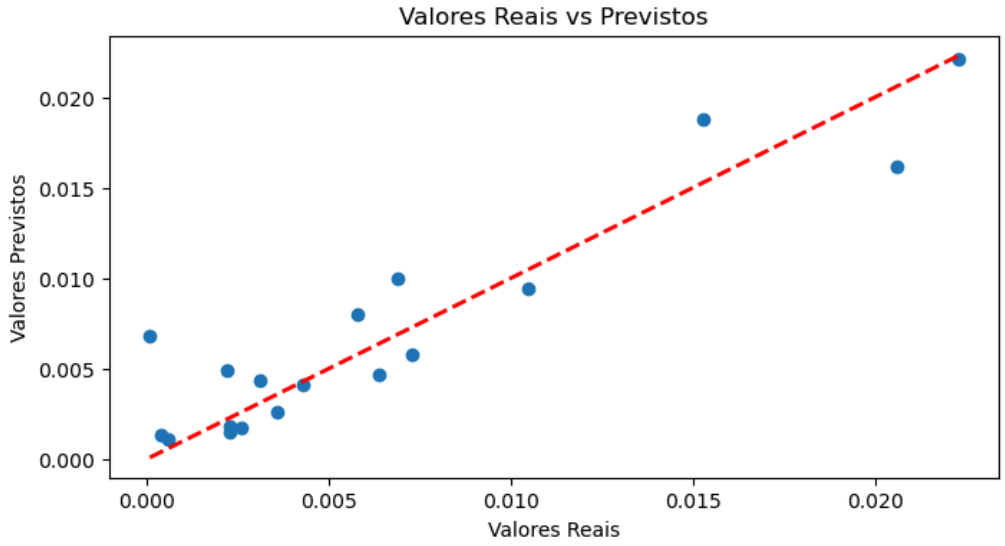
Sobre a Validação Cruzada:

- Acurácia por fold: [0.53132227 0.82405989 0.79609431 0.86794228 0.55141594]
- Acurácia média: 0.714166935046496
- Desvio padrão: 0.14307723161675343

    - Acurácia por fold: A variação entre os folds, de 0.531 a 0.867, pode indicar que há alguma inconsistência nos dados ou que o modelo é sensível a como os dados de treino e teste são divididos.
    - Acurácia média: 71,4% é um resultado moderado, sugerindo que há espaço para melhorias no modelo.
    - Desvio padrão: 0.14 reflete certa variabilidade no desempenho entre os folds.

Otimização sem normalização

Usando GridSearch para k e métrica de distância



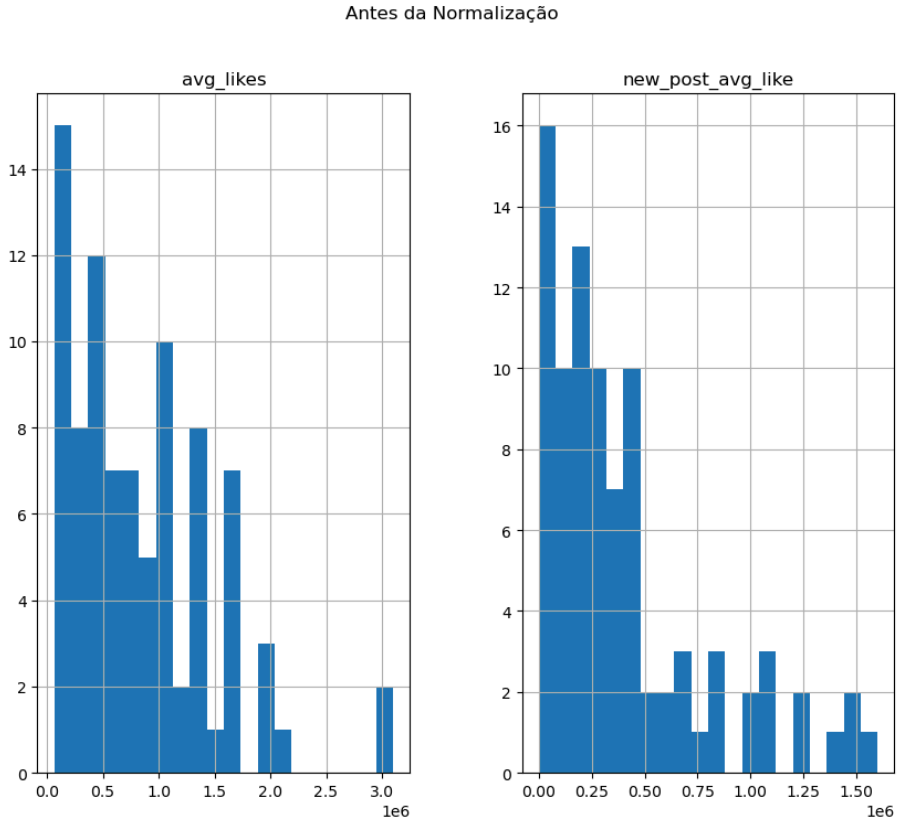
Após realizar a otimização do modelo KNN por meio do GridSearchCV, foi possível identificar os melhores hiperparâmetros para o problema em questão: metric = 'euclidean' e

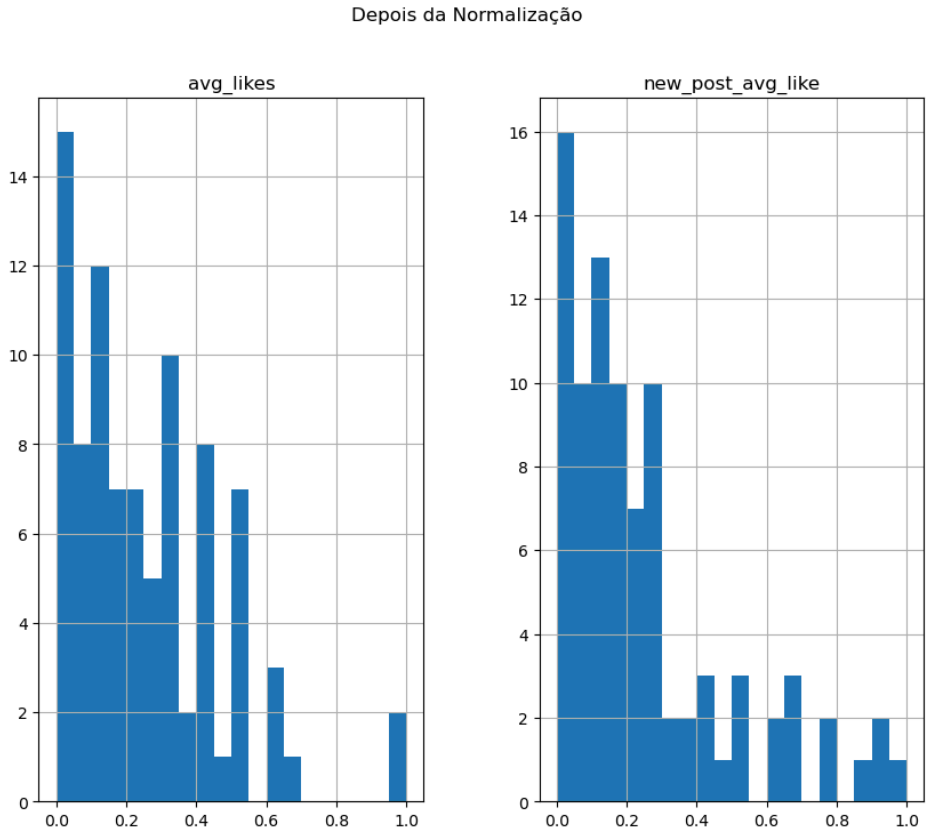
n_neighbors = 10. Esses parâmetros foram utilizados para refinar o modelo, resultando em um ajuste ligeiramente melhor.

As métricas de erro indicaram que o modelo continua com um bom desempenho. O Erro Médio Absoluto (MAE) foi de 0.0018, enquanto o Erro Quadrático Médio (MSE) foi de 6.123e-06 e o Raiz do Erro Quadrático Médio (RMSE) de 0.00247. Esses valores permanecem baixos, evidenciando que o modelo consegue prever valores bastante próximos dos reais.

A validação cruzada revelou uma acurácia média de 72,7%, um pequeno incremento em relação ao modelo inicial sem a normalização (71,4%). Além disso, houve uma redução no desvio padrão, que passou para 0.133, indicando maior consistência entre os folds. Apesar disso, ainda há variabilidade significativa entre os folds, com valores que variam de 0.533 a 0.923. Isso pode ser reflexo de uma possível sensibilidade do modelo a variações nos dados ou à distribuição das amostras.

Com normalização



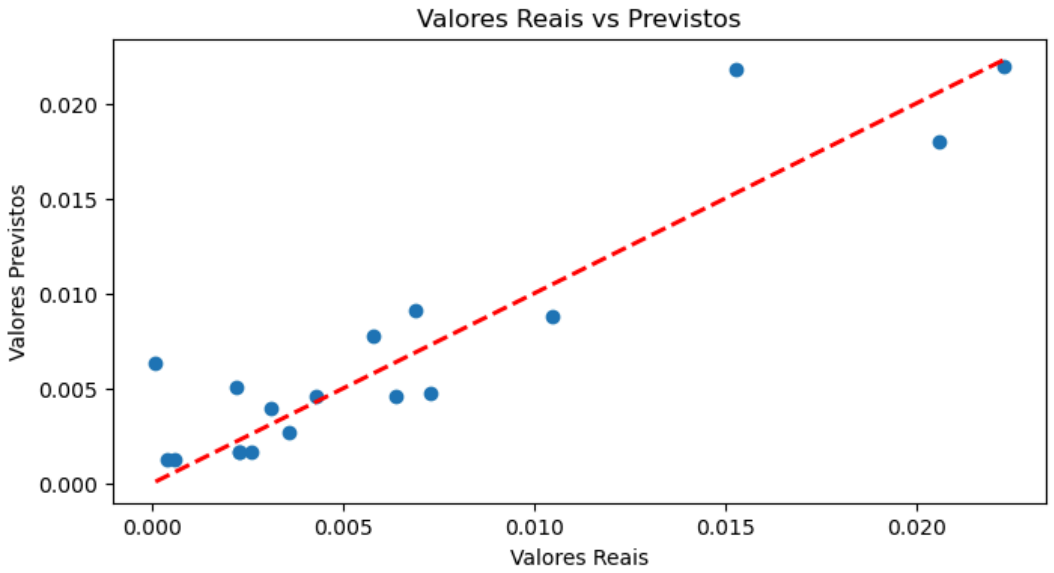


Com a aplicação da normalização das variáveis independentes usando o MinMaxScaler, os dados foram ajustados para um intervalo entre 0 e 1. Isso contribuiu para padronizar a escala das variáveis e eliminar possíveis vieses causados por diferenças de magnitude entre os atributos.

Após a normalização, foi conduzida uma nova otimização com GridSearchCV, identificando os melhores parâmetros como metric = 'euclidean' e n_neighbors = 11. O modelo foi então treinado e testado utilizando as variáveis normalizadas.

Aplicação do KNN

- Erro médio absoluto (MAE): 0.0019303030303030307
- Erro médio quadrático (MSE): 6.8165381083562905e-06
- Raiz do Erro médio quadrático (RMSE): 0.00261085007389476



Validação Cruzada:

- Acurácia média: 0.7682799983672477
- Desvio padrão: 0.13497588046710107
- Acurácia por fold: [0.65755947 0.96133962 0.74187114 0.8804931 0.60013666]

Tendo em vista essas informações, é possível observar que o MAE, MSE e RMSE continuam baixos, indicando que o modelo não está errando significativamente nas previsões. A acurácia média melhorou ligeiramente (em torno de 4%) em comparação com o modelo sem normalização, o que sugere que a normalização ajudou a estabilizar o desempenho.

O alto desvio padrão nos resultados da validação cruzada ainda representa uma preocupação, pois aponta para a variação nos desempenhos entre os folds. Esse desvio pode ser causado por:

- **Diferença nos dados de treino e teste**: O modelo pode estar reagindo de forma diferente a diferentes subconjuntos de dados.
- **Distribuição não uniforme**: Se os dados não forem bem balanceados ou se houver características que influenciam mais fortemente em certos folds, a consistência pode ser afetada.

Discussão

Com base nos resultados apresentados, fica claro que o modelo de **K-Nearest Neighbors (KNN)** apresentou um desempenho satisfatório no projeto em questão, com baixos valores de erro, como demonstrado pelos indicadores **MAE (Erro Médio Absoluto)**, **MSE (Erro Quadrático Médio)** e **RMSE (Raiz do Erro Quadrático Médio)**. Esses valores baixos indicam que o modelo é capaz de prever os valores de forma precisa, com pequenas discrepâncias entre os valores reais e os previstos.

Entretanto, a validação cruzada revelou que, embora a acurácia média seja razoavelmente boa (76.8%), o desvio padrão relativamente alto (0.135) sugere que o desempenho do modelo varia significativamente entre os diferentes subconjuntos de dados. Esse desvio pode indicar instabilidade no desempenho do modelo, que é uma área crítica a ser investigada mais a fundo no futuro. Além disso, o modelo parece ser sensível à forma como os dados são divididos, o que pode ser causado por diferentes distribuições de dados em cada fold. Como o KNN é um modelo baseado em distâncias, a variação no desempenho entre folds pode ser resultado de diferenças nas distâncias entre os pontos, especialmente se houver outliers ou dados desbalanceados.

Limitações Encontradas

1. **Desvio Padrão Alto nas Folds da Validação Cruzada**: O principal ponto a ser destacado é a variação significativa entre as acurácias nos diferentes folds, como demonstrado pelos valores que variam de 60% a 96%. Essa inconsistência sugere que o modelo pode ser muito sensível à divisão dos dados. É possível que alguns folds tenham características que dificultam o aprendizado do modelo, o que é particularmente importante em modelos como o KNN, que dependem fortemente da proximidade dos dados.
2. **Dependência de Distâncias**: O **KNN** é um modelo baseado em distâncias, e sua performance pode ser afetada pela escolha da métrica de distância. Embora a métrica **euclidiana** tenha sido escolhida, ela pode não ser a mais adequada para todos os tipos de dados. Em conjuntos de dados com variáveis de escalas muito diferentes ou com muitas dimensões, as distâncias euclidianas podem se tornar imprecisas, o que prejudica a performance do modelo.
3. **Sensibilidade à Escolha de K**: A escolha do número de vizinhos **(k)** tem um impacto direto no desempenho do modelo. Embora a **pesquisa em grade (GridSearch)** tenha encontrado que o valor de **k = 11** com a métrica euclidiana era o melhor para este conjunto de dados, a escolha de **k** pode ser altamente dependente do problema e pode não ser a melhor em outras situações. A escolha de **k** deve ser feita com base em experimentação e análise do trade-off entre **underfitting** e **overfitting**.

Impacto das Escolhas Feitas no Desempenho do Modelo

1. **Normalização das Variáveis**: A **normalização** das variáveis independentes teve um impacto positivo no modelo. Isso pode ser visto pela melhoria na acurácia média após a normalização. O KNN é sensível à escala das variáveis, e a normalização garantiu que todas as variáveis contribuem igualmente para o cálculo das distâncias. Sem a normalização, variáveis com escalas diferentes poderiam dominar o cálculo das distâncias e prejudicar a performance do modelo.
2. **Pesquisa em Grade (GridSearch)**: A utilização do GridSearch para otimizar os parâmetros k e a métrica de distância foi crucial para melhorar a performance do modelo. O k = 11 foi encontrado como o melhor valor para o número de vizinhos, enquanto a métrica euclidiana apresentou bons resultados. A escolha correta desses

parâmetros evitou um desempenho ruim, mas, como mencionado, a dependência do valor de k ainda requer um processo de afinação cuidadoso.

3. **Divisão dos Dados de Treinamento e Teste**: A divisão dos dados com a função train_test_split (80% para treinamento e 20% para teste) foi uma escolha padrão, mas o modelo pode se beneficiar de diferentes estratégias de validação, como a validação cruzada estratificada, para garantir que a distribuição dos dados seja mais representativa em cada fold. Isso ajudaria a reduzir a variabilidade observada no desvio padrão.

Conclusão e Trabalhos Futuros

Foi um projeto que possibilitou o contato prático com importantes conceitos dentro da ciência de dados como a normalização das variáveis, que foi essencial para o sucesso do modelo KNN, o qual depende do cálculo de distâncias. Sem normalizar, variáveis em escalas diferentes poderiam ter dominado as decisões do modelo, prejudicando as previsões.

Ademais, a escolha do número de vizinhos (k) e da métrica de distância mostrou-se crítica e utilizar o GridSearch para determinar os melhores parâmetros foi um aprendizado prático sobre como otimizar modelos baseados em hiperparâmetros. Não obstante, a análise de validação cruzada revelou inconsistências no desempenho entre diferentes folds, destacando a sensibilidade do modelo aos dados de entrada e a importância de entender as variações para melhorar a generalização. Outrossim, o uso de métricas como **MAE**, **MSE** e **RMSE** permitiu uma avaliação quantitativa das previsões, reforçando a necessidade de medições consistentes para validar a qualidade do modelo.

Importante destacar também a sensibilidade do KNN à distribuição dos dados e presença de outliers. Esse fato reforça a importância de um bom pré-processamento e balanceamento dos dados para modelos baseados em vizinhança.

Sobre possíveis melhorias em trabalhos futuros, observa-se que seria interessante explorar a redução da variabilidade na validação cruzada ao implementar validação cruzada estratificada para garantir que cada fold tenha uma distribuição representativa dos dados, além de experimentar técnicas de balanceamento dos dados, como oversampling ou undersampling, caso existam desbalanceamentos significativos no target.

A exploração de outras métricas de distância ou até métricas baseadas em aprendizado (como Mahalanobis) pode ser encarada como um ponto de melhoria para o modelo. Ademais, estudar a implementação do **KNN ponderado** (Weighted KNN), onde os vizinhos mais próximos têm maior influência na previsão, é uma possibilidade para lidar com outliers e melhorar a robustez do modelo.

A utilização de modelos complementares também é um ponto a ser explorado através da comparação do desempenho do KNN com outros algoritmos, como **Regressão Linear**, **Árvores de Regressão** ou **Modelos Baseados em Ensemble** (como Random Forest ou Gradient Boosting), para verificar se algum oferece melhor desempenho ou maior estabilidade. Além do valor ótimo encontrado pelo GridSearch, analisar a curva de desempenho para diferentes valores de **k** para garantir que o modelo não está sub ajustando (underfitting) ou sobre ajustando (overfitting) é outro ponto que pode potencializar a acurácia do modelo.

Referências

BOCCATO, Levy e ATTUX, Romis. *k-Nearest Neighbors.* [Apresentação de slides]. 2024. Disponível em: https://www.dca.fee.unicamp.br/~lboccato/topico_5_k_nearest_neighbor.pdf. Acesso em: 12 nov. 2024.

CEPEDI. Residência tecnológica: trilha de ciência de dados. Ilhéus: CEPEDI, 2024. Curso online, 560 horas.

FEA dev. Machine Learning usando Python - modelo KNN (AULA PRÁTICA). 2020. [vídeo]. 41 min. Disponível em: https://www.youtube.com/watch?v=xL2RK0QYtHc&t=2055s. Acesso em: 11 nov. 2024.

GRUS, Joel. *Data Science do Zero: Primeiras Regras com o Python*. 1. ed. Rio de Janeiro: Alta Books, 2016. 336 p. ISBN 978-8576089988.