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

Noah Vila Nova Aragão
16/Novembro/2024

Resumo

Durante a 9ª unidade da bolsa Cepedi em Ciência de Dados, foi realizada uma análise com o algoritmo k-Nearest Neighbors (kNN) para criar uma métrica alternativa de influência, o true_influence_score, baseada em variáveis como taxa de engajamento, média de likes recentes e pontuação de influência. Após preparar os dados, incluindo transformação, normalização e tratamento de valores ausentes, o algoritmo foi otimizado usando o GridSearchCV com diferentes métricas de distância e números de vizinhos.

Os resultados mostraram que, embora a métrica de distância Manhattan tenha alcançado o menor erro (27,98%), as taxas de erro permaneceram altas, indicando limitações nos dados para capturar relações claras entre as variáveis. Foram criados gráficos para explorar essas relações, reforçando a importância de revisitar o conjunto de dados e ajustar a abordagem para análises futuras.

Introdução

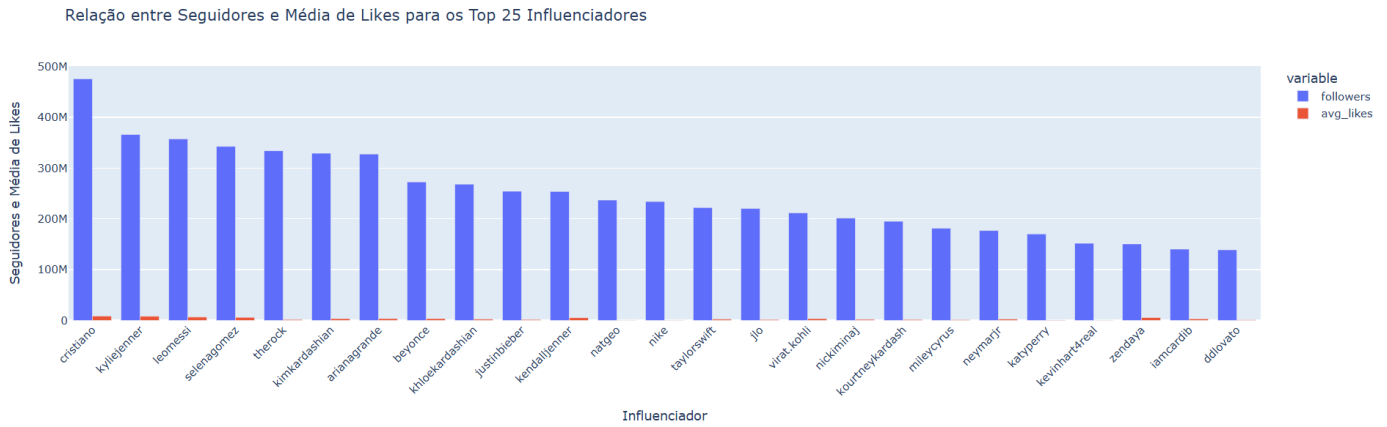
Durante a 9° Unidade da bolsa de estudos da Cepedi voltada a Ciência de Dados, foi passada uma tarefa de duração de 2 semanas para fazer uma implementação e análise do algoritmo KNN. O projeto consistia em usar uma base de dados dos 200 perfis com mais seguidores no instagram, apresentando características como: Nome do perfil, ranking, quantidade de seguidores, quantidade de postagens, quantidade total de “likes”, quantidade mediana de “likes” e quantidade mediana de “likes” de postagens recentes, a porcentagem da popularidade nos últimos 60 dias e uma pontuação de influência. A escolha do algoritmo kNN (k-Nearest Neighbors) foi dada devido a sua simplicidade de aplicação, por ser versátil e também por ter uma natureza não paramétrica, ou seja, não assume nada sobre a distribuição dos dados e é flexível.

Metodologia

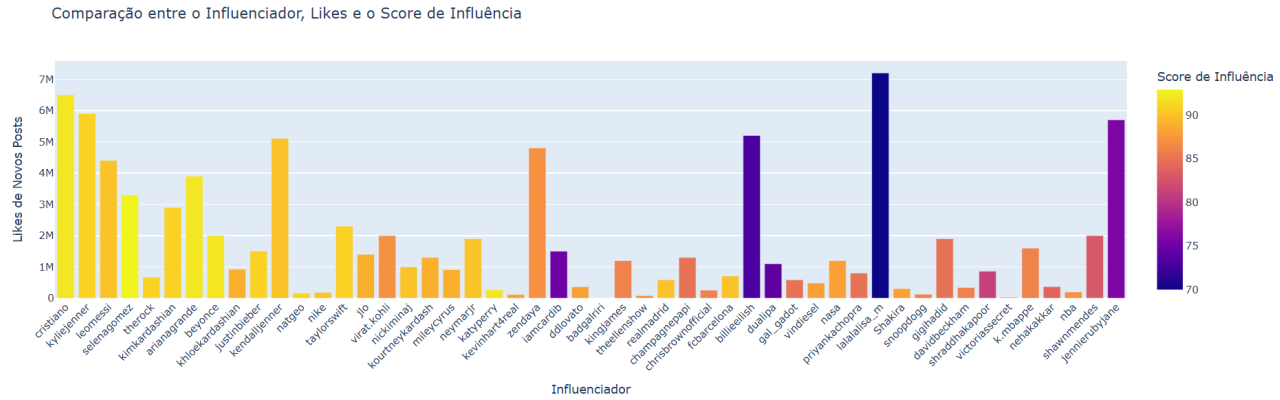
No primeiro passo do projeto, foi colocado que seria a etapa de Definição e Preparação do Problema, em que pegava o acesso a base de dados e preparava os mesmos para analisar o problema. A primeira mudança feita nos dados foi proposta pela orientação do projeto: transformação dos valores na coluna country em faixas numéricas baseadas em continentes. Após fazer essa mudança, a segunda foi para mudar os valores

de influence_score, posts, followers, avg_likes, new_post_avg_like, total_likes que estavam em formato string para o formato float para melhorar uso deles na IA.

Em uma tentativa de entender melhor os dados que estavam sendo analisados, foram criados 2 gráficos para que a análise fosse mais certeira, os gráficos foram feitos usando a biblioteca plotly pelo programador já ter familiaridade, o primeiro foi feito pegando os top 25 do rank, com o eixo x sendo o nome da conta, e o eixo y sendo a quantidade de seguidores e quantidade média de likes de todos os post:



Ao encarar o primeiro gráfico, é bastante perceptível que todos os perfis têm uma quantidade exorbitantemente maior de seguidores do que de média de likes, assim, levantou o questionamento em relação ao influence_score, sendo necessário outro gráfico para entender esse dado melhor. O segundo gráfico tinha como eixo x o nome da conta, e o eixo y com a média de likes de novos posts com a cor sendo o influence score:



Quando o segundo gráfico foi feito, ficou claro que o dado influence_score não condiz com o seu nome, já que uma pontuação de influência deveria ser avaliada principalmente se a pessoa é influente no presente ou não, uma pessoa que anos atrás era influente na internet mas com o tempo foi perdendo sua popularidade deixa de ser influente. Muito provavelmente o influence_score utilizava como cálculo, o rank que o perfil estava junto com quantidade de seguidores. Entretanto não fazia sentido um perfil que estava com a média de likes de posts recentes nos 7 milhões ter um score de influência tão abaixo de perfis que chegavam perto dos 100 mil likes, por isso, o objetivo para o kNN foi traçado, criar uma nova coluna chamada true_influence_score que analisaria se a pessoa atualmente é influente ou não.

Com o objetivo do kNN em mente, foi necessária uma terceira alteração para tratar os valores ausentes, substituindo-os por zero, para evitar problemas durante a análise e

modelagem. Após isso, foi realizado o cálculo do true_influence_score, utilizando uma fórmula ponderada que considerava as variáveis 60_day_eng_rate, new_post_avg_like e influence_score, com pesos definidos como 0.4, 0.3 e 0.3, respectivamente. Essa métrica foi normalizada para uma escala de 0 a 100, facilitando a interpretação e análise comparativa dos influenciadores.

Com os dados devidamente preparados, o próximo passo foi implementar o algoritmo k-Nearest Neighbors (kNN) para prever o true_influence_score com base em variáveis explicativas selecionadas: 60_day_eng_rate, new_post_avg_like e influence_score. Antes de treinar o modelo, os dados foram divididos em conjuntos de treino e teste, utilizando uma proporção de 70% para treino e 30% para teste, garantindo que a avaliação do modelo fosse feita em dados não vistos.Para melhorar o desempenho do algoritmo, as variáveis preditoras foram normalizadas usando o método StandardScaler, que transforma os dados para que tenham média zero e desvio padrão igual a um.

Em seguida, o modelo kNN foi configurado para ser otimizado utilizando o GridSearchCV, que realiza validação cruzada para testar diferentes combinações de hiperparâmetros. Os parâmetros avaliados foram:Número de vizinhos (n_neighbors): Valores variando de 1 a 15; Métricas de distância: Incluindo euclidean, manhattan, chebyshev, minkowski e cosine, cada uma com características específicas para calcular similaridades entre os pontos.

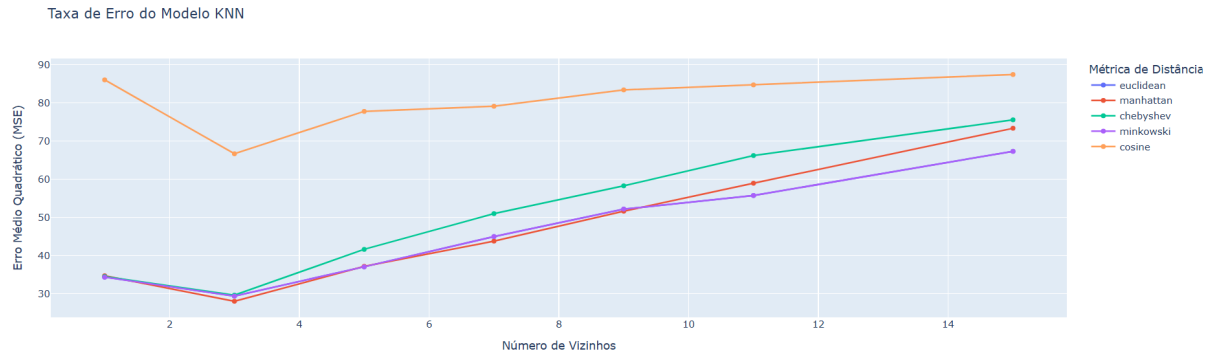
O GridSearchCV foi configurado com 5-fold cross-validation, garantindo que cada modelo fosse treinado e validado em diferentes subconjuntos dos dados de treino, reduzindo o risco de overfitting. O critério de avaliação foi o Mean Squared Error (MSE) negativo, que, após o cálculo, foi convertido para valores positivos para facilitar a análise dos resultados.

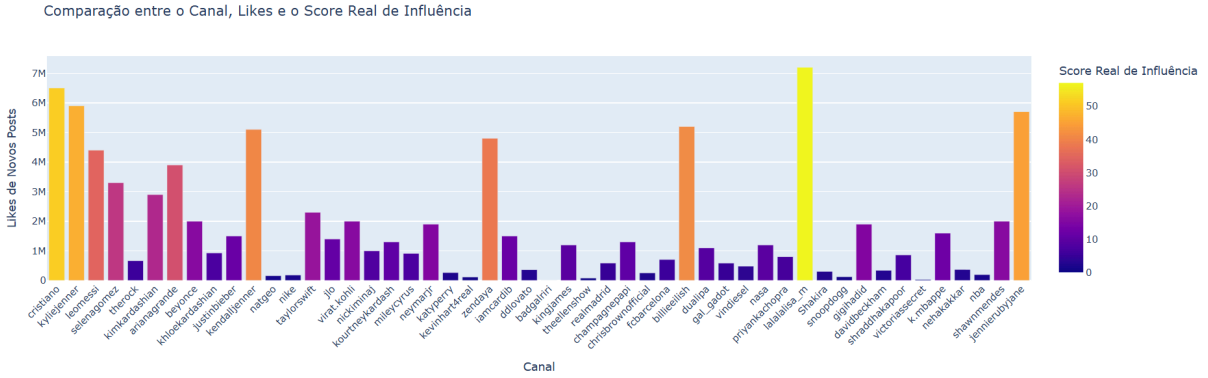
Após a execução do GridSearchCV, o melhor conjunto de parâmetros foi selecionado com base no menor MSE médio obtido nos dados de validação. Todos os resultados do processo foram salvos em um arquivo CSV para análise futura, incluindo o desempenho de cada combinação de parâmetros testada.

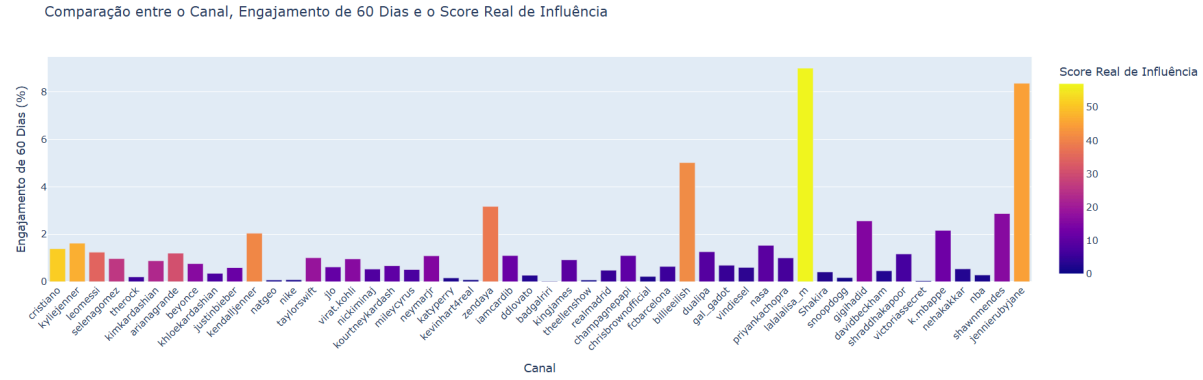
Resultados

Para facilitar a interpretação dos dados presentes no CSV, foram desenvolvidos três gráficos principais. O primeiro gráfico apresenta a taxa de erro para todas as combinações de métricas de distância e números de vizinhos testados, permitindo uma análise detalhada do desempenho do modelo.

Já o segundo e o terceiro gráficos foram construídos com base nos melhores parâmetros identificados, ou seja, a métrica de distância e o número de vizinhos que resultaram na menor taxa de erro. Esses gráficos destacam a relação entre o true_influence_score e as variáveis médias de novos posts e taxas de engajamento dos últimos 60 dias, proporcionando uma visão mais aprofundada sobre como essas variáveis influenciam a pontuação de influência real.







Discussão

No final da análise, observou-se que as taxas de erro foram relativamente altas, mesmo utilizando o melhor número de vizinhos identificado para este dataset, que foi 3. Apesar das tentativas de ajuste e otimização, todas as métricas de distância apresentaram desempenhos bastante próximos no que diz respeito à média de erro quadrático (MSE). A métrica de distância Manhattan destacou-se ligeiramente, alcançando o menor valor de erro com uma taxa de 27,98%, sugerindo que ela é mais adequada para capturar as relações entre as variáveis neste caso específico.

Entretanto, esses resultados indicam limitações importantes no modelo. Primeiro, a alta taxa de erro sugere que os dados podem não apresentar uma relação suficientemente linear ou forte entre as variáveis independentes e a variável de saída para que o kNN seja altamente eficaz. Além disso, o fato de as diferentes métricas terem resultados semelhantes reforça a possibilidade de que o desempenho do modelo seja mais influenciado por

características intrínsecas dos dados, como colinearidade ou ruído, do que pelos parâmetros do algoritmo.

Outro ponto crítico foi o impacto da normalização dos dados. Embora essencial para algoritmos baseados em distância, a normalização não foi suficiente para melhorar significativamente o desempenho, indicando que outros métodos de pré-processamento ou seleção de variáveis poderiam ser necessários. Também é possível que variáveis importantes para o problema tenham ficado de fora da análise ou que algumas variáveis incluídas tenham introduzido ruído ao invés de valor preditivo.

Por fim, a escolha do número de vizinhos foi crucial para minimizar o erro, mas um número baixo de vizinhos, como 3, pode resultar em um modelo mais sensível a outliers e variações nos dados.

Conclusão e Trabalhos Futuros

Embora o algoritmo kNN tenha identificado a configuração mais adequada dentro das opções testadas, os resultados sugerem que ele possui limitações significativas para este conjunto de dados. Modelos mais robustos, como aqueles baseados em árvores de decisão (e.g., Random Forest ou Gradient Boosting), poderiam ser explorados para melhorar o desempenho.

A análise reforça que, apesar de sua simplicidade e utilidade em diversos problemas, o kNN pode não ser a melhor escolha em situações onde os dados apresentam complexidade ou ruído significativo. Assim, a capacidade do modelo de gerar previsões confiáveis depende não apenas da escolha de hiperparâmetros, mas também de uma análise mais aprofundada da preparação e seleção das variáveis.

Por último, mas não menos importante, o momento escolhido para a execução deste projeto na residência de software foi extremamente desfavorável para os residentes. A decisão de realizar o projeto no final do segundo semestre, quando os residentes que são alunos têm que entregar trabalhos e provas finais, e os residentes que atuam no mercado de trabalho precisam cumprir metas profissionais, resultou em sobrecarga e estresse, dificultando a conciliação entre as demandas. Esse período poderia ter sido aproveitado de forma mais produtiva, considerando que os primeiros quatro meses da residência foram dedicados apenas a conteúdos teóricos, o que poderia ter sido melhor utilizado para o desenvolvimento do projeto.

Outro ponto relevante é a falta de organização por parte da coordenação, bem como a ausência de transparência no curso. Durante a escolha das trilhas, a de Ciência de Dados foi destacada como uma opção com foco não tanto em programação, mas em uma abordagem geral sobre o vasto campo da área. No entanto, ao iniciar a residência, os alunos foram informados de que a trilha seria exclusivamente voltada para Inteligência Artificial e que, ao longo do curso, a ênfase seria totalmente teórica. Curiosamente, há apenas um mês do término da bolsa, foi anunciado que as últimas duas unidades seriam práticas, o que contradizia totalmente as expectativas criadas no início do programa.