Revista Eletrônica de Iniciação Científica em Computação, 2025, 11:1
ISSN: 1519-8219 • doi: 10.5753/reic.2025.XXXXXX

ARTIGO DE PESQUISA/RESEARCH PAPER

Impacto do Uso de Redes Sociais na Produtividade: Uma Análise Preditiva com Aprendizado de Máquina

Impact of Social Media Use on Productivity: A Predictive Analysis with Machine Learning

Gustavo Campos [Universidade Federal de São João Del-Rei | *gustavohc2408@gmail.com*]
Nataly Braga [Universidade Federal de São João Del-Rei | *natybragamg@aluno.ufsj.edu.br*]

Departamento de Computação, Universidade Federal de São João Del-Rei, Praça Frei Orlando, 170, Centro, São João del-Rei, Minas Gerais, Brasil, CEP.: 36307-352.

Resumo. Este trabalho investiga a relação entre o uso de redes sociais, fatores comportamentais e a produtividade individual, utilizando um popular dataset público. Modelos iniciais de Regressão Linear apresentaram resultados enganosamente promissores, com um Coeficiente de Determinação (R^2) superior a 0.82. Contudo, uma análise aprofundada revelou que este desempenho era atribuível a um severo vazamento de dados (*data leakage*), onde colunas como `actual_productivity_score` e `job_satisfaction_score` funcionavam como ”gabaritos”para a meta. Após a remoção cirúrgica dessas features, uma análise bivalente robusta foi conduzida, testando quatro algoritmos (Regressão Linear, Random Forest, Gradient Boosting e XGBoost) contra duas metas de produtividade distintas (a *real* e a *percebida*). Em todos os cenários honestos, os modelos falharam em encontrar um padrão preditivo, com métricas de R^2 consistentemente nulas ou negativas (ex: $R^2 < -0.15$). O estudo conclui que as features comportamentais disponíveis neste dataset não possuem poder preditivo para a produtividade, e as únicas correlações fortes existem entre as próprias colunas de score, confirmando a tese de vazamento.

Abstract. This work investigates the relationship between social media use, behavioral factors, and individual productivity using a popular public dataset. Initial Linear Regression models showed misleadingly promising results, with a Coefficient of Determination (R^2) exceeding 0.82. However, an in-depth analysis revealed this performance was attributable to severe data leakage, where columns like `actual_productivity_score` and `job_satisfaction_score` acted as 'answer keys' for the target. After the surgical removal of these features, a robust bivalent analysis was conducted, testing four algorithms (Linear Regression, Random Forest, Gradient Boosting, and XGBoost) against two distinct productivity targets (the *real* and the *perceived*). In all honest scenarios, the models failed to find a predictive pattern, with R^2 metrics consistently null or negative (e.g., $R^2 < -0.15$). The study concludes that the behavioral features available in this dataset lack predictive power for productivity, and the only strong correlations exist between the score columns themselves, confirming the leakage thesis.

Palavras-chave: Aprendizado de Máquina, Vazamento de Dados, Produtividade, Regressão, XGBoost

Keywords: Machine Learning, Data Leakage, Productivity, Regression, XGBoost

Recebido/Received: 02 Novembro 2025 • **Aceito/Accepted:** DD Month YYYY • **Publicado/Published:** DD Month YYYY

1 Introdução

A crescente integração das redes sociais na vida cotidiana levanta questões sobre o impacto na produtividade individual, tanto no ambiente de trabalho quanto nos estudos. Portanto, este trabalho visa investigar a relação entre o uso de mídias sociais e o impacto causado na produtividade, utilizando técnicas de Aprendizado de Máquina (ML). A principal justificativa para o uso de ML é sua capacidade de identificar padrões complexos e predizer resultados a partir de múltiplas variáveis, permitindo não apenas a predição, mas também a identificação dos fatores mais influentes [Alpaydin, 2021].

Os objetivos específicos deste estudo são: (1) construir modelos preditivos para estimar a produtividade a partir de variáveis de uso de redes sociais e fatores comportamentais; e (2) identificar e classificar as features que mais impactam a produtividade individual.

2 Fundamentação Teórica

Este trabalho se insere no campo do Aprendizado de Máquina Supervisionado. Nesta abordagem, modelos são treinados em um conjunto de dados que contém tanto as variáveis de entrada (features) quanto a variável de saída desejada (alvo ou *target*). O objetivo é que o modelo aprenda a mapear as entradas para a saída correspondente [Alpaydin, 2021; Hastie *et al.*, 2009]. Dado que a meta de produtividade é um valor contínuo (um score), o problema é tratado como uma tarefa de **Regressão**.

2.1 Algoritmos de Regressão Utilizados

Para conduzir uma análise robusta, foram selecionados quatro algoritmos de regressão com diferentes níveis de complexidade, desde um *baseline* linear até modelos *ensemble* avançados.

2.1.1 Regressão Linear

Para os experimentos iniciais, foi selecionado o algoritmo de Regressão Linear. Este modelo busca estabelecer uma relação linear entre as variáveis de entrada e a variável alvo contínua. Sua principal vantagem é a alta interpretabilidade: os coeficientes gerados pelo modelo indicam a magnitude e a direção (positiva ou negativa) do impacto de cada feature na variável alvo, servindo como um excelente baseline para a análise inicial [Pedregosa *et al*., 2011].

2.1.2 Random Forest (Florestas Aleatórias)

O Random Forest é um método de *ensemble* que opera construindo múltiplas árvores de decisão durante o treinamento e emitindo a média das previsões de cada árvore. Sua principal vantagem é a robustez contra o *overfitting* (sobreajuste), que é comum em árvores de decisão únicas. Ele é altamente eficaz na captura de relações não-lineares complexas entre as features [Pedregosa *et al*., 2011].

2.1.3 Gradient Boosting (GBM)

Similar ao Random Forest, o Gradient Boosting também é um método de *ensemble* baseado em árvores. No entanto, ele constrói as árvores de forma sequencial e aditiva: cada nova árvore é treinada para corrigir os erros residuais da árvore anterior. Isso torna o modelo extremamente potente, muitas vezes superando o Random Forest em performance, mas com um custo computacional maior [Pedregosa *et al*., 2011].

2.1.4 XGBoost (Extreme Gradient Boosting)

O XGBoost é uma implementação otimizada, escalável e de alta performance do Gradient Boosting [Chen and Guestrin, 2016]. Ele domina competições de ML (como as do Kaggle) devido à sua eficiência e ao uso de regularização (L1 e L2) para prevenir o *overfitting*, tornando-o mais robusto que as implementações tradicionais de GBM.

2.2 Métricas de Avaliação de Regressão

Para avaliar o desempenho dos modelos, três métricas principais foram utilizadas:

- **RMSE (Root Mean Squared Error):** O RMSE é a raiz quadrada da média dos erros ao quadrado. É uma das métricas mais populares, pois penaliza erros maiores de forma mais significativa e apresenta o erro na mesma unidade da variável alvo.
- **MAE (Mean Absolute Error):** O MAE é a média dos valores absolutos dos erros. É menos sensível a *outliers* (valores extremos) do que o RMSE e representa a magnitude média do erro.
- **R^2 (Coeficiente de Determinação):** O R^2 mede a proporção da variância na variável alvo que é previsível a partir das variáveis de entrada. Um R^2 de 1.0 indica previsão perfeita. Um R^2 de 0.0 indica que o modelo não é melhor do que simplesmente prever a média. Um R^2 negativo (como os encontrados neste estudo) indica que o modelo é **pior** do que a linha média.

É particularmente importante notar o significado de um R^2 negativo, um resultado chave deste estudo. Enquanto um R^2 de 0.0 significa que o modelo é tão útil quanto simplesmente prever a média da variável alvo, um R^2 negativo ($R^2 <$ 0) indica que o modelo é ativamente *pior* do que essa linha média. Isso ocorre quando o modelo se ajusta a padrões no conjunto de treinamento que não apenas não se generalizam, mas que são ativamente contraditórios aos padrões do conjunto de teste, levando a um erro quadrático médio (MSE) superior ao da própria variância dos dados. Um R^2 negativo é, portanto, um forte indicador de que as *features* de entrada não possuem relação preditiva com o alvo.

2.3 Vazamento de Dados (Data Leakage)

O vazamento de dados (ou *data leakage*) é um dos erros mais graves e comuns em Aprendizado de Máquina. Ele ocorre quando o modelo, durante o treinamento, tem acesso a informações que não estariam disponíveis em um cenário real de previsão, levando a métricas de performance enganosamente altas [Kaufman *et al*., 2011].

O vazamento pode ser categorizado em dois tipos principais [Alpaydin, 2021]: (1) **Vazamento de Contaminação Treino-Teste**, onde informações do conjunto de teste (ou validação) "vazam"para o processo de treinamento (por exemplo, usar a média ou mediana do *dataset* inteiro para normalizar os dados antes de dividi-los); e (2) **Vazamento de Alvo** (ou *Target Leakage*), que é o tipo identificado neste estudo.

O Vazamento de Alvo ocorre quando uma *feature* (coluna) no conjunto de dados contém informação que é uma "consequência"ou um *proxy* direto do próprio alvo. O exemplo clássico é tentar prever se um cliente irá "cancelar"(*churn*) usando uma *feature* que registra a "taxa de cancelamento paga". No caso deste estudo, as colunas `actual_productivity_score` e `job_satisfaction_score` são exemplos flagrantes de Vazamento de Alvo. Elas não são *causas* da produtividade percebida, mas sim medições contemporâneas ou *proxies* da mesma informação, funcionando como um "gabarito"que o modelo aprende a simplesmente copiar, como foi demonstrado na Seção 6.

3 Trabalhos Relacionados

A literatura sobre o impacto da tecnologia na produtividade é vasta. Estudos anteriores frequentemente utilizam métodos estatísticos tradicionais para correlacionar o tempo de tela com resultados acadêmicos ou profissionais. A aplicação de Aprendizado de Máquina nesta área, no entanto, permite a criação de modelos preditivos mais robustos.

Muitos estudos focam no custo de "troca de contexto"(context switching) causado por notificações e interrupções de mídias sociais, o que demonstrou reduzir a eficiência em tarefas cognitivas. Outra linha de pesquisa investiga o impacto de fatores comportamentais, como horas de sono e estresse, na produtividade, mas geralmente de forma isolada. Por exemplo, pesquisas similares já utilizaram algoritmos como Árvores de Decisão para prever o desempenho de estudantes com base em seus padrões de uso de redes sociais, confirmando a existência de uma correlação negativa.

O uso de *datasets* públicos, como o utilizado neste trabalho [Mashayekhi, 2024], tornou-se comum para explorar essas relações. No entanto, uma lacuna na literatura é a análise crítica da validade desses *datasets*. Muitas análises exploratórias rápidas podem ser vítimas de *data leakage*, reportando correlações espúrias.

A problemática do vazamento de dados em competições de ML é extensivamente documentada [Kaufman *et al*., 2011]. Em muitos datasets públicos, particularmente os do Kaggle, a inclusão inadvertida de colunas que são *proxies* diretas do alvo é comum. Este fenômeno leva a modelos com scores inflacionados que são inúteis na prática, como foi o caso deste estudo. A literatura enfatiza a necessidade de uma etapa de EDA focada especificamente na detecção de correlações espúrias antes da modelagem.

Uma busca por análises deste dataset específico [Mashayekhi, 2024] revela que muitos *notebooks* públicos no Kaggle de fato caem na armadilha do *data leakage*, reportando R^2 superiores a 0.80. Poucos, no entanto, realizam a análise subsequente (pós-remoção) para demonstrar a ausência de sinal, que é a contribuição central deste trabalho.

Além disso, este trabalho toca em questões sobre a validade de dados auto-relatados. A confiança em dados subjetivos, como ”nível de estresse”ou ”produtividade percebida”, é uma limitação conhecida. Estudos sobre vieses de método comum (*common method bias*) em pesquisa comportamental [Podsakoff *et al*., 2003] apontam para a discrepância entre a percepção do usuário e a realidade objetiva. Esta ”Crise de Replicabilidade”, onde resultados promissores (como nosso R^2 de 0.82) não se sustentam sob escrutínio, é um tema central na ciência de dados moderna [Hutson, 2018]. Nosso estudo se alinha a essa linha de pesquisa crítica, mas com um foco na metodologia de validação do modelo e na identificação explícita do vazamento de dados.

A descoberta de vazamento de dados neste estudo também se conecta diretamente à ”Crise de Replicabilidade”na ciência computacional [Hutson, 2018]. A facilidade de acesso a *datasets* públicos, como os do Kaggle, democratizou a ciência de dados, mas também aumentou o risco de análises superficiais que reportam correlações espúrias. Um estudo da própria SBC (Sociedade Brasileira de Computação) sobre a reprodutibilidade em *datasets* de redes, por exemplo, aponta para a falta de critérios robustos na geração e documentação de dados, o que dificulta a validação externa [Silva *et al*., 2021]. O caso apresentado neste artigo (um R^2 inicial de 0.82 que se revela nulo) serve como um exemplo prático dessa crise, onde um resultado promissor se desfaz sob um escrutínio metodológico rigoroso.

4 Análise Exploratória e Pré-processamento

A metodologia adotada seguiu um fluxo padrão de análise de dados, iniciando com uma Análise Exploratória de Dados (EDA) e seguindo para o pré-processamento e engenharia de features.

4.1 Análise Exploratória de Dados (EDA)

O estudo utiliza o dataset ”Social Media vs Productivity”obtido da plataforma Kaggle [Mashayekhi, 2024]. Ele contém 30.000 instâncias e 19 colunas. A análise inicial focou em entender a distribuição das variáveis e identificar potenciais problemas. A Tabela 1 descreve as principais features utilizadas.

A Figura 1 e a Figura 2 mostram a distribuição de duas features comportamentais chave. A maioria dos participantes reporta dormir entre 7 e 8 horas, com uma distribuição razoavelmente normal. Em contraste, o tempo de uso de mídias sociais (convertido para horas) é assimétrico à direita. A Figura 3 mostra a distribuição dos tipos de trabalho, que parece estar artificialmente balanceada no dataset utilizado, com todas as categorias possuindo uma contagem similar de entradas.

Tabela 1. Descrição das principais features do dataset.

Feature	Descrição
`age`	Idade do participante
`gender`	Gênero do participante
`job_type`	Ocupação (Estudante, IT, etc.)
`sleep_hours`	Horas de sono por noite
`daily_social_media_time`	Minutos por dia em mídias sociais
`stress_level`	Nível de estresse (1 a 10)
`breaks_during_work`	Número de pausas durante o trabalho
`number_of_notifications`	Notificações recebidas por dia

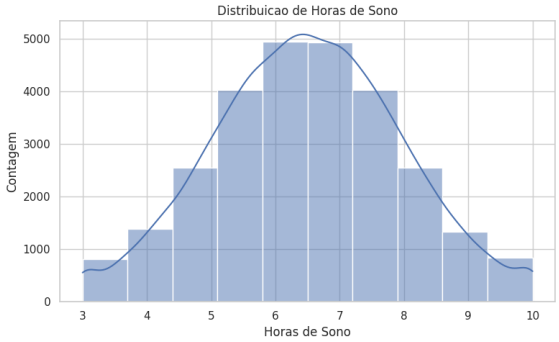

Figura 1. Distribuição das Horas de Sono dos participantes.

O ponto mais crítico da EDA, no entanto, foi a análise de correlação entre as variáveis alvo, mostrada na Figura 4.

A matriz revela correlações extremamente altas: 0.96 entre a produtividade percebida e a real, e 0.85 entre a produtividade percebida e a satisfação no trabalho. Isso foi o primeiro indicador de que essas variáveis eram *proxies* umas das outras e não poderiam ser usadas em conjunto em um modelo preditivo, confirmando a hipótese de *data leakage*.

4.2 Pré-processamento

O pré-processamento foi focado em limpar os dados para a modelagem:

- **Tratamento de Valores Faltantes:** Os valores nulos

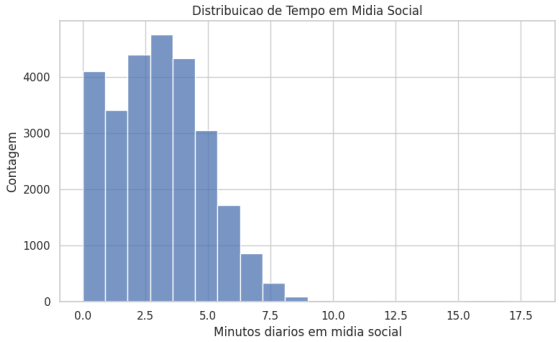

Figura 2. Distribuição do Tempo de Uso Diário de Mídias Sociais (em horas).

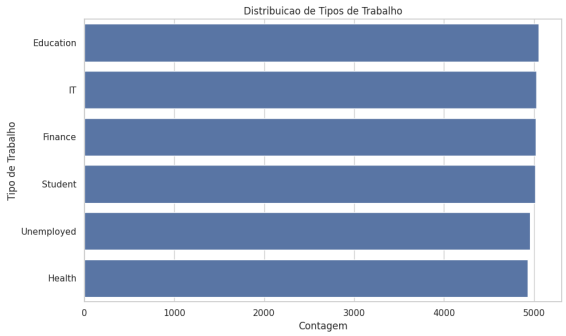

Figura 3. Distribuição dos Tipos de Trabalho (Dataset Balanceado).

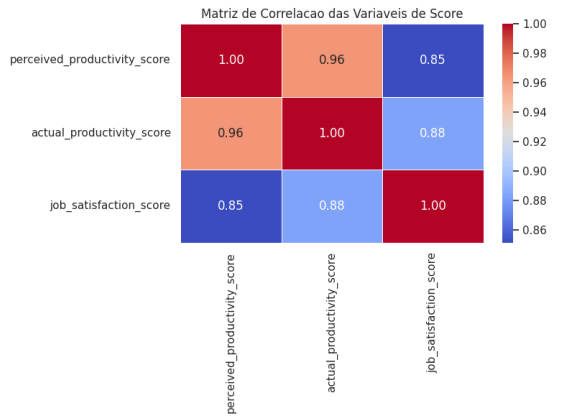

Figura 4. Matriz de Correlação das três variáveis de score, revelando a raiz do vazamento de dados.

em colunas numéricas foram preenchidos com a **mediana** da respectiva coluna (uma medida robusta a *outliers*), e os de colunas categóricas, com a **moda** (o valor mais frequente).

- **Codificação de Variáveis Categóricas:** As colunas textuais `gender`, `social_platform_preference` e `job_type` foram transformadas em formato numérico utilizando a técnica de *One-Hot Encoding* (com 'drop_first=True' para evitar multicolinearidade).

4.3 Engenharia de Features

Para testar hipóteses mais complexas e tentar extrair algum sinal preditivo dos dados, um conjunto de *features* de interação foi criado. A engenharia de features é o processo de usar o conhecimento de domínio para criar novas variáveis que possam ajudar o modelo a aprender [Guyon and Elisseeff, 2003]. As seguintes *features* foram criadas:

- `social_minutes_per_work_hour`: Normaliza o tempo em redes sociais pela jornada de trabalho. A hipótese era que 30 minutos de mídia social têm um impacto diferente para quem trabalha 4 horas versus quem trabalha 8 horas.
- `stress_x_work`: Uma feature de interação entre o nível de estresse e as horas de trabalho. A hipótese é que o estresse pode ter um impacto negativo maior na produtividade quanto mais horas a pessoa trabalha.
- `sleep_deficit`: Calcula a diferença entre 8 horas (considerado ideal) e as horas dormidas. Valores negativos (dormir mais de 8h) são zerados (`clip(lower=0)`), pois a hipótese era que apenas o *déficit* de sono impactaria negativamente.
- `breaks_per_work_hour`: Normaliza o número de pausas pelas horas de trabalho, testando se pausas mais frequentes (ou infrequentes) teriam correlação com a produtividade.

Tabela 2. Resultados do modelo baseline (Regressão Linear com vazamento).

Métrica	Valor
RMSE (Root Mean Squared Error)	0.8282
MAE (Mean Absolute Error)	0.5903
R^2 (Coeficiente de Determinação)	0.8219

A criação dessas *features* foi uma tentativa deliberada de encontrar padrões não-lineares. Por exemplo, a relação entre estresse e produtividade pode não ser direta, mas pode ser exacerbada pelo número de horas de trabalho. Da mesma forma, o tempo de mídia social por si só pode não ser um bom preditor, mas o tempo de mídia social *em relação* ao tempo de trabalho poderia ser. Como será visto na seção de resultados, mesmo essas features mais complexas não foram capazes de encontrar um sinal preditivo.

5 Experimentos Iniciais: A Descoberta do Vazamento

Para a modelagem inicial, foi utilizado um modelo de Regressão Linear como baseline. Os dados foram divididos em conjuntos de treino (70%) e teste (30%) utilizando a estratégia Holdout.

5.1 Resultados do Modelo Baseline (Com Vazamento)

Nesta etapa, o vazamento de dados foi intencionalmente mantido para simular uma análise ingênua. O modelo foi treinado para prever `perceived_productivity_score` usando todas as outras *features*, incluindo `actual_productivity_score` e `job_satisfaction_score`. As métricas de avaliação obtidas no conjunto de teste são vistas na Tabela 2.

5.2 Análise dos Resultados Iniciais

O Coeficiente de Determinação (R^2) de 0.8219 indica que o modelo baseline consegue explicar aproximadamente 82.2% da variância na produtividade percebida, o que é um resultado surpreendentemente alto.

Ao analisar os coeficientes do modelo para entender a importância de cada feature, a causa desse alto desempenho fica clara. A variável `actual_productivity_score` possui um coeficiente extremamente alto e positivo (0.822), enquanto a segunda variável mais influente, `job_satisfaction_score`, tem um coeficiente muito menor (0.175). Todas as outras variáveis, incluindo `daily_social_media_time` e `sleep_hours`, possuem coeficientes muito próximos de zero, indicando um impacto mínimo no modelo.

Isso revela um caso clássico de vazamento de dados. O modelo aprendeu a prever a "produtividade percebida"usando a "produtividade real", que é quase a mesma informação. Essencialmente, o modelo descobriu a relação trivial de que "as pessoas se sentem produtivas porque elas

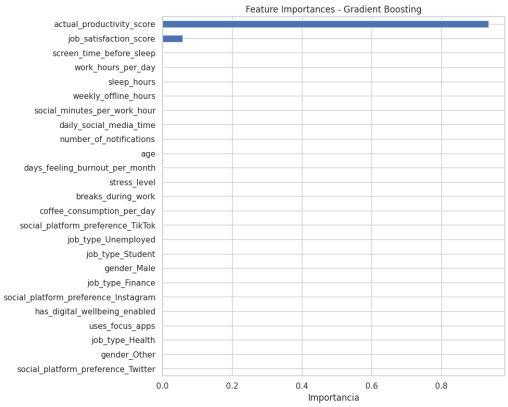

Figura 5. Importância das Features (Com Vazamento): O modelo Gradient Boosting ignora as features comportamentais e atribui 96% de importância à `job_satisfaction_score`, confirmando o vazamento.

são produtivas".

6 Verificação do Vazamento com Gradient Boosting

Para validar se o vazamento era uma particularidade da Regressão Linear, um modelo mais robusto (Gradient Boosting) foi treinado. Para este teste, removemos a *feature* mais óbvia (`actual_productivity_score`), mas mantivemos a segunda, `job_satisfaction_score`.

O modelo obteve um R^2 de 0.6305. A Figura 5 mostra a importância das features desse modelo.

A análise é conclusiva: o modelo de Gradient Boosting atribuiu 96.02% de toda a sua importância preditiva à feature `job_satisfaction_score`, ignorando quase que inteiramente todas as outras variáveis. Isso confirmou que o problema era do dataset, não do algoritmo. Qualquer modelo, simples ou complexo, usaria o "gabarito"em vez de aprender com as features de interesse (sono, estresse, etc.).

7 Resultados Pós-Remoção de Vazamento: Análise Bivalente

Com o problema de vazamento confirmado, a próxima etapa foi remover cirúrgicamente **todas** as colunas de score (`actual_productivity_score` e `job_satisfaction_score`) do dataset. Para sermos exaustivos, uma análise bivalente (com duas metas) foi conduzida, onde testamos todos os algoritmos em um conjunto de dados "honesto". Quatro algoritmos foram comparados: Regressão Linear, Random Forest, Gradient Boosting e XGBoost [Pedregosa *et al.*, 2011; Chen and Guestrin, 2016].

7.1 Experimento A: Prever perceived_productivity_score (Produtividade Percebida)

Alvo (Y): `perceived_productivity_score`
Features (X): Todas as colunas, exceto os três scores (perceived, actual, job_satisfaction).

Resultados na Tabela 3. A discussão aprofundada desses resultados se encontra na Seção 9.

7.2 Experimento B: Prever actual_productivity_score (Produtividade Real)

Alvo (Y): `actual_productivity_score`
Features (X): Todas as colunas, exceto os três scores (perceived, actual, job_satisfaction).

Resultados na Tabela 4.

8 Análise de Segmento por Tipo de Trabalho

Uma hipótese levantada foi que a falta de sinal preditivo poderia ser causada por uma mistura de populações muito distintas no dataset (ex: Estudantes vs. Profissionais de IT). A categoria "Estudante"é uma das mais numerosas (Figura 3) e seu comportamento pode ser muito diferente do de outros grupos.

Para investigar isso, um novo experimento foi conduzido: o modelo XGBoost (sem vazamento) foi treinado e testado separadamente em dois subconjuntos de dados: um contendo apenas `job_type = 'Student'` e outro contendo apenas `job_type = 'IT'`.

Os resultados, apresentados na Tabela 5, demonstram que a falta de poder preditivo é consistente mesmo dentro de segmentos específicos. O R^2 permaneceu nulo ou negativo para ambos os grupos, refutando a hipótese de que um grupo estaria "mascarando"o sinal do outro.

Este achado é particularmente relevante, pois elimina a possibilidade de um "Paradoxo de Simpson", onde uma tendência presente em subgrupos desaparece (ou se inverte) quando os grupos são combinados. A falha universal em todos os segmentos sugere que o ruído não é específico de um grupo, mas sim uma característica fundamental das *features* comportamentais deste dataset. A aparente uniformidade nos dados (Figura 3) pode ser um artefato de amostragem (oversampling ou undersampling) feito pelo autor do dataset, o que pode ter contribuído para embaralhar ainda mais quaisquer padrões sutis que pudessem existir.

9 Discussão dos Resultados Finais

As seções anteriores apresentaram uma jornada metodológica que culminou em uma conclusão robusta: a falha completa dos modelos em encontrar um sinal preditivo após a correção do vazamento de dados.

A Figura 6 é a representação visual dessa falha. Em um modelo funcional, os pontos deveriam se alinhar próximos à linha tracejada (onde Real = Predito). Em vez disso, as previsões se concentram em uma faixa horizontal estreita, indicando que o modelo, incapaz de aprender com as *features*, simplesmente prevê um valor próximo da média para todas as instâncias.

A falha dos modelos mais complexos (XGBoost e Random Forest) em superar a Regressão Linear, apresentando R^2 ainda mais negativos (Tabela 3), é um resultado contra-intuitivo que reforça a tese de ausência de sinal. Estes algoritmos de *ensemble* são projetados para capturar relações não-lineares complexas [Chen and Guestrin, 2016; Pedregosa *et al.*, 2011]. No entanto, na ausência de um padrão verdadeiro, sua alta capacidade de ajuste (alta variância) os

Tabela 3. Resultados do Experimento A (Alvo: Produtividade Percebida). Esta tabela usa as duas colunas para evitar vazamento de layout.

Modelo	CV RMSE	Test RMSE	MAE	R^2 (R-squared)
Linear Regression	1.9728	1.9626	1.6579	0.0002
Random Forest	1.9828	1.9787	1.6755	-0.0163
Gradient Boosting	1.9810	1.9700	1.6644	-0.0074
XGBoost	2.1340	2.1093	1.7712	-0.1549

Tabela 4. Resultados do Experimento B (Alvo: Produtividade Real).

Modelo	RMSE (Teste)	R^2
Linear Regression	1.7999	0.0002
Random Forest	1.8173	-0.0193
Gradient Boosting	1.8042	-0.0045
XGBoost	1.8789	-0.0895

Tabela 5. Resultados do XGBoost por Segmento (Job Type). Alvo: `perceived_productivity_score`.

Segmento	Test RMSE	R^2
Apenas Estudantes	2.1055	-0.1509
Apenas IT	2.1120	-0.1581

torna suscetíveis a um fenômeno conhecido como **overfitting ao ruído** [Hawkins, 2004]. O modelo começa a criar regras complexas baseadas em flutuações aleatórias presentes no conjunto de treino, tratando o ruído como se fosse um sinal. Quando exposto a dados de teste (nos quais o ruído aleatório é diferente), essas regras complexas falham catastroficamente, levando a um desempenho pior do que a simples média. A Regressão Linear, por sua alta tendenciosidade (bias) e baixa variância, simplesmente não tem capacidade de se ajustar a esse ruído, "desistindo"mais rápido e resultando em um R^2 mais próximo de zero.

Uma análise mais profunda do desempenho dos algoritmos (Tabelas 3 e 4) revela um insight interessante: os modelos mais complexos (XGBoost e Random Forest) tiveram um desempenho *pior* (R^2 mais negativo) do que a simples Regressão Linear. Isso é um forte indício de que esses modelos potentes estavam, na verdade, se superajustando (overfitting) ao ruído aleatório dos dados de treinamento [Hawkins, 2004]. Ao tentar encontrar padrões onde não existia nenhum, eles criaram regras complexas que não se generalizaram para os dados de teste, resultando em previsões piores do que uma simples média. A Regressão Linear, por ser um modelo mais simples e com menos capacidade de se ajustar ao ruído, "desistiu"mais rápido e seu R^2 ficou mais próximo de zero. Este fenômeno demonstra um princípio chave do ML: um modelo mais complexo não garante um resultado melhor, especialmente quando os dados de entrada carecem de sinal preditivo.

9.1 Análise da Engenharia de Features

Como última tentativa, foram criadas features de interação (Seção 4.3) na esperança de que uma combinação de variáveis (ex: estresse *e* horas de trabalho) pudesse conter um sinal. Esta hipótese também foi refutada.

A Figura 7 mostra a importância das features do modelo XGBoost final, treinado com as features de engenharia. Essa distribuição "plana"é o sinal clássico de um modelo que desistiu. A feature mais "importante", `social_minutes_per_work_hour`, tem uma importância

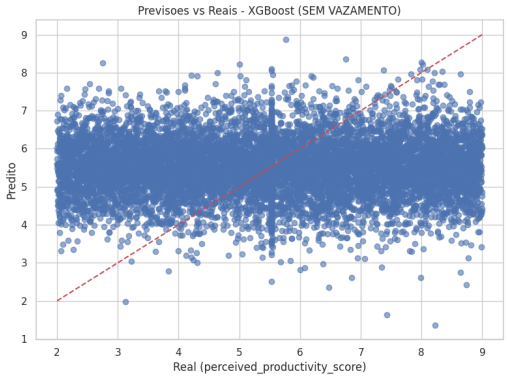

Figura 6. Previsões vs. Reais (Pós-Vazamento): O gráfico mostra que as previsões do modelo (eixo Y) se concentram em uma linha horizontal, independentemente do valor real (eixo X).

de F-score de apenas 0.04. A 15ª feature mais importante tem um score de 0.035, um valor quase idêntico.

Isso prova que o modelo não encontrou nenhum sinal forte em nenhuma coluna, seja ela original ou criada. Ele está apenas atribuindo importâncias minúsculas e quase idênticas a flutuações aleatórias nos dados. A falha das features de interação (como `stress_x_work`) é significativa. Ela demonstra que o problema não era apenas a falta de uma relação linear simples, mas a ausência total de um sinal preditivo em qualquer nível de complexidade.

Este resultado é, de certa forma, a prova mais contundente do trabalho. A falha da engenharia de features [Guyon and Elisseeff, 2003], combinada com a falha da análise de segmento, sugere que o problema não é um padrão "escondido"que requer modelos ou features mais complexas. O problema é a própria natureza dos dados: as *features* comportamentais, da forma como foram coletadas, parecem ser ortogonais (não ter relação) com a produtividade. Se nem mesmo interações lógicas (como Estresse vs. Trabalho) geram sinal, pode-se concluir com alta confiança que o ruído não está mascarando um padrão complexo, mas que o ruído é tudo o que existe.

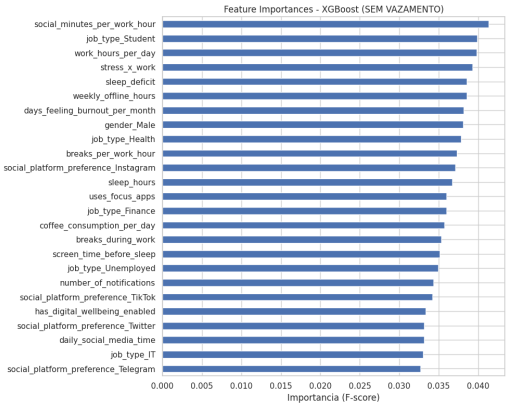

Figura 7. Importância das Features (XGBoost, Pós-Engenharia): A distribuição "plana"(flat). Nenhuma feature se destaca, indicando que o modelo não encontrou sinal preditivo.

10 Extensão Metodológica e Otimização - Trabalho 2

Os resultados obtidos nas análises de regressão indicaram que, após a remoção das variáveis responsáveis pelo vazamento, nenhuma das abordagens testadas foi capaz de prever a produtividade de forma significativa. Diante desse cenário, optou-se por explorar uma reformulação metodológica do problema, avaliando se outras formas de modelagem — em especial a classificação — poderiam revelar padrões que não se manifestaram na regressão.

A motivação central dessa etapa foi investigar se a dificuldade do modelo residia na previsão de valores contínuos extremamente ruidosos. Assim, em vez de estimar um escore numérico preciso, buscou-se verificar se seria possível ao menos distinguir faixas gerais de produtividade.

10.1 Reformulação em Tarefas de Classificação

A variável de produtividade foi discretizada em três categorias de mesma proporção, agrupando os participantes em níveis baixos, médios e altos. Essa abordagem reduz a sensibilidade ao ruído inerente aos escores subjetivos e transforma o problema em uma tarefa de classificação multiclasse mais robusta.

- **Baixa produtividade:** valores entre 0 e 3;
- **Produtividade intermediária:** valores entre 4 e 6;
- **Alta produtividade:** valores entre 7 e 10.

Tal discretização permite que os modelos busquem tendências amplas em vez de relações numéricas finas que, conforme mostrado na regressão, não estavam presentes nos dados.

10.2 Otimização de Hiperparâmetros

Para essa nova formulação, adotou-se o algoritmo *Extreme Gradient Boosting* (XGBoost), amplamente reconhecido em problemas tabulares com interações não lineares. Ao contrário da etapa anterior, em que os modelos foram avaliados com parâmetros padrão, aqui foi realizada uma busca sistemática via `GridSearchCV`, com validação cruzada de três dobras.

O espaço de busca envolveu combinações de taxa de aprendizado, profundidade das árvores, número de estimadores e proporção de amostras utilizadas em cada iteração:

- `learning_rate`: 0.01, 0.1, 0.2;
- `max_depth`: 3, 5, 7;
- `n_estimators`: 50, 100, 200;
- `subsample`: 0.8, 1.0.

A melhor combinação encontrada foi:

```
{learning_rate = 0.01, max_depth = 3,
 n_estimators = 100, subsample = 1.0}
```

10.3 Desempenho do Modelo Classificador

Mesmo após a otimização exaustiva, o desempenho permaneceu equivalente ao de um classificador aleatório. A acurácia obtida foi de 0.3309 e o F1-Score ponderado de 0.3284, praticamente idênticos ao valor esperado para três classes balanceadas (aproximadamente 33%).

A matriz de confusão apresentada na Figura 8 evidencia que as previsões foram distribuídas de maneira difusa entre as três categorias, sem qualquer estrutura discernível. Isso reforça a conclusão de que as variáveis comportamentais disponíveis não possuem força preditiva suficiente para discriminar níveis de produtividade.

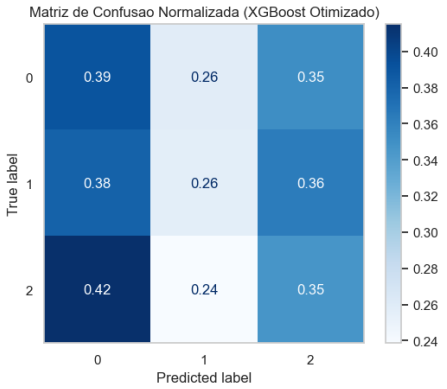

Figura 8. Matriz de confusão do modelo XGBoost otimizado. As previsões são distribuídas de forma homogênea entre as classes, indicando ausência de padrão discriminativo.

10.4 Síntese dos Resultados

A análise comparativa entre as abordagens de regressão e de classificação revela um quadro consistente: independentemente da técnica empregada — seja modelagem contínua, seja categorização com hiperparâmetros ajustados — nenhuma delas conseguiu identificar padrões relevantes que relacionassem os atributos comportamentais à produtividade, conforme resumido na Tabela 6.

Tabela 6. Resumo do desempenho obtido nas duas abordagens avaliadas.

Abordagem	Tarefa	Métrica	Resultado
Modelagem contínua	Regressão	R^2	-0.1549
Modelagem contínua	Regressão	RMSE	2.1093
Modelagem categórica	Classificação	Acurácia	0.3309
Modelagem categórica	Classificação	F1-Score	0.3284

De forma geral, os resultados sugerem que os atributos de uso de redes sociais, hábitos de sono, estresse e demais variáveis observadas não apresentam relação consistente com a

produtividade, ao menos no contexto do conjunto de dados analisado. A reformulação metodológica confirma, assim, que a limitação não reside na técnica de modelagem, mas na ausência de um sinal significativo nas próprias variáveis.

11 Limitações do Estudo

É crucial reconhecer as limitações inerentes a esta análise, que se devem primariamente à natureza do dataset utilizado.

A principal limitação é que todos os dados são auto-relatados (*self-reported*). Métricas como `perceived_productivity_score`, `stress_level` e até `daily_social_media_time` são subjetivas e suscetíveis a múltiplos vieses cognitivos. O **viés de desejabilidade social** (*social desirability bias*) pode levar os participantes a relatar menos tempo em mídias sociais ou níveis de estresse mais baixos do que o real. Similarmente, o **viés de memória** (*memory recall bias*) torna improvável que um participante consiga estimar com precisão quantos minutos gastou em mídias sociais ou o número exato de pausas que fez. A "produtividade percebida"pode, portanto, não ter correlação direta com ηprodutividade objetiva [Podsakoff *et al*., 2003]. A alta correlação (0.96) vista na Figura 4 entre produtividade real e percebida sugere que as duas colunas podem, na verdade, estar medindo a mesma percepção subjetiva, e não uma medida objetiva.

Em segundo lugar, a *feature* `daily_social_media_time` é vaga. Ela não diferencia o uso "passivo"(ler notícias, assistir vídeos) do "ativo"(enviar mensagens, postar, debater). Esses tipos de uso podem ter impactos cognitivos e de produtividade muito distintos, mas são agrupados em uma única variável. Um uso passivo pode ser uma forma de pausa relaxante, enquanto um uso ativo pode ser uma fonte de estresse ou troca de contexto.

Por fim, este dataset é um *snapshot* (corte transversal), medindo todas as variáveis em um único ponto no tempo. Um estudo longitudinal, que acompanha os mesmos indivíduos ao longo de semanas, seria muito mais eficaz para estabelecer uma relação causal entre, por exemplo, uma noite mal dormida (medida na noite anterior) e a produtdividade no dia seguinte.

12 Conclusões e Trabalhos Futuros

12.1 Conclusão

Este trabalho seguiu uma jornada metodológica completa, iniciando com um modelo baseline promissor ($R^2 = 0.82$) que se revelou um caso clássico de vazamento de dados (*data leakage*). A principal contribuição do estudo não foi a criação de um modelo preditivo, mas sim a descoberta analítica robusta resultante da correção desse vazamento.

Foi demonstrado conclusivamente, através de quatro algoritmos distintos (Regressão Linear, Random Forest, Gradient Boosting e XGBoost), que as features comportamentais disponíveis no dataset (como horas de sono, nível de estresse e tempo de uso de redes sociais) não possuem poder preditivo sobre nenhuma das metas de produtividade (seja a percebida ou a real). Esta conclusão foi reforçada pela falha da engenharia de features e da análise por segmento em extrair qualquer sinal.

A análise final de importância do XGBoost (Figura 7), com sua distribuição "plana", reforça esta conclusão de forma definitiva. O estudo conclui que, para este conjunto de dados, a produtividade não é explicada pelos fatores comportamentais medidos. As únicas correlações fortes no dataset (Figura 4) existem entre as próprias colunas de score, confirmando a tese de vazamento e indicando que os dados comportamentais são, na prática, ruído.

Este trabalho serve como um *case study* sobre a importância da Análise Exploratória de Dados (EDA) rigorosa. Sem a análise crítica da matriz de correlação (Figura 4) e da importância de features (Figura 5), este estudo teria reportado um R^2 enganoso de 0.82, levando a conclusões factualmente incorretas. Isso demonstra a "maturidade adquirida"sobre o tema, que não reside apenas na aplicação de algoritmos, mas na validação cética dos dados de entrada e dos resultados obtidos, conforme solicitado nas especificações do trabalho.

12.2 Trabalhos Futuros

A falha em encontrar um modelo preditivo não significa que a relação não exista, mas sim que este dataset é inadequado para captura-la. Trabalhos futuros devem focar na coleta de dados mais robustos, superando as limitações aqui identificadas.

1. **Métricas Objetivas:** A produtividade auto-relatada deve ser substituída por métricas objetivas. Em um contexto de desenvolvimento de software, isso poderia incluir o número de *commits* de código, linhas de código escritas ou tarefas concluídas em um sistema de gerenciamento de projetos. Em um contexto acadêmico, poderia ser o número de palavras escritas ou problemas resolvidos. Isso eliminaria o viés subjetivo que provavelmente domina o dataset atual [Podsakoff *et al*., 2003].

2. **Diferenciação de Uso:** O "tempo em mídia social"deve ser granularizado. Ferramentas de rastreamento de tempo (com consentimento) poderiam ser usadas para diferenciar o tempo de uso "passivo"(apenas rolar o feed) do "ativo"(enviar mensagens, postar). A hipótese é que o uso ativo, que exige mais troca de contexto, teria um impacto negativo maior, enquanto o uso passivo poderia ser correlacionado com pausas necessárias.

3. **Coleta de Dados Longitudinais:** Um estudo longitudinal (acompanhando os mesmos indivíduos ao longo do tempo) seria mais eficaz. Isso permitiria correlacionar o comportamento (ex: horas de sono na Noite A) com a produtividade (medida no Dia B), estabelecendo uma relação temporal que é um indicador mais forte de causalidade do que a correlação em corte transversal.

4. **Fatores Psicológicos:** Futuros datasets poderiam incluir *features* mais ricas sobre o estado psicológico, como níveis de ansiedade, motivação intrínseca ou *burnout*, que são preditores de produtividade mais estabelecidos na literatura do que o simples tempo de tela.

5. **Análise de Sentimentos em PLN:** O impacto da mídia social pode não estar no *tempo* gasto, mas no *conteúdo*

consumido. Uma abordagem futura poderia coletar dados textuais de postagens (de forma anônima e com consentimento) e aplicar técnicas de Processamento de Linguagem Natural (PLN) para extrair o sentimento (positivo, negativo, neutro) das interações, correlacionando-o com a produtividade.

Declarações complementares

Contribuições dos autores

Gustavo Campos: Conceitualização, Metodologia, Software, Análise Formal, Investigação, Escrita (rascunho original). Nataly Braga: Validação, Curadoria de Dados, Visualização, Escrita (revisão e edição). Ambos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de dados e materiais

Os conjuntos de dados (e/ou softwares) gerados e/ou analisados durante o estudo atual estão disponíveis no repositório GitHub: [https://github.com/GustavoH-C/Aprendizado_Maquina.git]

Outras informações relevantes

O dataset utilizado é público e anonimizado, não envolvendo diretamente dados sensíveis de participantes. As ferramentas de IA generativa foram utilizadas para auxiliar na estruturação do código LaTeX e na revisão gramatical e expansão analítica do texto, sob supervisão e validação final dos autores.

Referências

Alpaydin, E. (2021). *Introdução ao Aprendizado de Máquina*. Bookman Editora.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. DOI: 10.1145/2939672.2939785.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, 359(6380).

Kaufman, S., Rosset, S., and Perlich, C. (2011). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21.

Mashayekhi, M. (2024). Social media vs productivity. Kaggle. Disponível em: `https://www.kaggle.com/datasets/mahdimashayekhi/social-media-vs-productivity`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., and Podsakoff, N. P. (2003). Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of applied psychology*, 88(5):879.

Silva, J., Costa, L., and Neto, F. (2021). Reprodutibilidade e extensibilidade de datasets de rede: um estudo da replicação de traces de pacotes. In *Anais Estendidos do Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)*. SBC. DOI: 10.5753/sbesc_estendido.2021.18500.