

Overfitting e Underfitting em Modelos de Aprendizado de Máquina: Conceitos, Desafios e Perspectivas

Alyson Fumagalli¹, Eliel da Silva¹

¹Instituto Federal de São Paulo (IFSP) — Campus São Paulo, Brasil

alyson.fumagalli@aluno.ifsp.edu.br, eliel.silva@aluno.ifsp.edu.br

Resumo

Este trabalho investiga os fenômenos de *overfitting* e *underfitting* em modelos de aprendizado de máquina, destacando suas causas, implicações práticas e estratégias de mitigação. São discutidos conceitos fundamentais como generalização, troca viés-variância, validação cruzada, curvas de aprendizado e técnicas de regularização. Exemplos reais em domínios como reconhecimento facial, detecção de fraude e previsão de demanda ilustram como modelos mal ajustados impactam diretamente a confiabilidade e o custo operacional de sistemas de IA. A partir disso, o estudo apresenta abordagens consolidadas para mitigar esses problemas, além de analisar o estado atual do tema no meio acadêmico e na indústria, ressaltando a crescente relevância de modelos sobreparametrizados e os desafios teóricos em torno da generalização. O trabalho conclui apontando a necessidade de aprofundar a compreensão sobre esses fenômenos para garantir modelos robustos e responsáveis em aplicações críticas.

Index Terms: Overfitting; Underfitting; Machine Learning; Generalização; Regularização.

Abstract

This work examines the phenomena of overfitting and underfitting in machine learning models, emphasizing their causes, practical implications, and mitigation strategies. Fundamental concepts such as generalization, the bias-variance trade-off, cross-validation, learning curves, and regularization techniques are discussed. Real-world examples from facial recognition, fraud detection, and demand forecasting demonstrate how poorly adjusted models directly affect the reliability and operational costs of AI systems. The study then presents consolidated approaches to mitigate these issues and analyzes the current state of the field in both academia and industry, highlighting the growing relevance of overparameterized models and ongoing theoretical challenges related to generalization. The paper concludes by underscoring the need for deeper research to ensure robust and responsible models in critical applications.

Index Terms: Overfitting; Underfitting; Machine Learning; Generalization; Regularization.

1. Introdução

Com os avanços da tecnologia digital, interações online têm sido cada vez mais comuns no cotidiano. Redes sociais, plataformas de *streaming*, lojas virtuais e vários outros ambientes digitais são responsáveis por gerar volumes massivos de dados que crescem conforme o passar dos anos. Estima-se, inclusive, que ao fim de 2025 o volume global de dados aumente para 181 zettabytes de acordo com o Statista [1].

Esse cenário demonstra-se muito promissor para a área de Inteligência Artificial (IA), especialmente para o ramo de *machine learning* (ML) no qual o objetivo é analisar e encontrar algum tipo de valor nessa vasta quantidade de dados. No entanto, para que isso seja possível, é necessário encontrar um equilíbrio entre generalização e aprendizado.

Diante disso, surgem os conceitos de *overfitting* (“sobreajuste”) e *underfitting* (“subajuste”), que correspondem respectivamente a um modelo de ML que aprende demais ou de menos, prejudicando sua capacidade de generalização e, consequentemente, sua performance em dados reais.

A importância desse tema reside no fato de que tais problemas estão entre as principais causas de falhas em sistemas inteligentes aplicados no mundo real. Quando um modelo não generaliza adequadamente, serviços críticos — como mecanismos de recomendação, sistemas de detecção de fraude, diagnósticos automatizados e previsões estratégicas — tornam-se menos confiáveis, podendo induzir decisões incorretas, enviadas ou economicamente custosas.

Assim, compreender e mitigar esses fenômenos não é apenas uma preocupação técnica, mas uma necessidade fundamental para garantir robustez, segurança e responsabilidade no desenvolvimento de soluções baseadas em dados.

O presente estudo tem por objetivo, portanto, explorar esses conceitos relacionados ao contexto de *machine learning*, destacando sua relevância prática, seus desafios, as principais técnicas de mitigação e — por fim — uma análise sobre o estado atual do tema.

2. Revisão Teórica

A revisão teórica é uma etapa fundamental para compreender os conceitos essenciais relacionados ao aprendizado de máquina e os desafios associados ao *overfitting* e *underfitting*. Nesta seção serão abordados o conceito de modelos de *machine learning*, a generalização, as características do *overfitting* e *underfitting*, o *trade-off* entre viés e variância, bem como a regularização.

Esses tópicos fornecem a base necessária para entender as estratégias de mitigação e os avanços recentes na área.

2.1. Modelos de Machine Learning

O *Machine Learning* é um campo da Inteligência Artificial (IA) que explora a construção de modelos computacionais capazes de identificar padrões, realizar previsões e tomar decisões com intervenção humana mínima e baseando-se em conjuntos de dados finitos [2].

Para a construção de um bom modelo de ML, deve-se coletar um conjunto de dados relevante para o problema a ser resolvido. Esses dados servirão como “combustível” para que um algoritmo selecionado possa aprender padrões em cima

desse conjunto. Após estar devidamente treinado, o modelo poderá identificar padrões em novos dados e fazer previsões ou classificações com base no aprendizado.

2.2. Generalização em Machine Learning

A **generalização** em *machine learning* descreve a capacidade de um modelo treinado conseguir realizar previsões, classificações ou decisões ao se deparar com conjuntos de dados nunca antes vistos.

Esse é um conceito importante de acordo com Domingos (2012) pois a generalização define o quão boa será a performance do modelo treinado em situações reais de análise de dados [3]. Além disso, ele é afetado diretamente pelo *overfitting* e pelo *underfitting*.

2.3. Overfitting e Underfitting

Segundo Bashir et al. (2020), *overfitting* ocorre quando um modelo aprende detalhes e ruídos específicos do conjunto de treinamento, tornando-se excessivamente ajustado a ele. Com isso, o modelo apresenta ótimo desempenho nos dados de treino, mas falha em generalizar para novos dados [4].

Já o *underfitting* ocorre quando o modelo é simples demais para capturar as relações existentes nos dados, resultando em baixo desempenho tanto no treino quanto nos testes [4].

Uma forma mais simples de compreender esses conceitos se dá por meio de uma analogia com o ensino escolar, como aponta Patel (2025) [5]. Imagine que um estudante está se preparando para uma prova:

- Se o estudante apenas memoriza as respostas específicas das questões da prova anterior, ele pode se sair muito bem nessa prova, mas terá dificuldades em responder perguntas diferentes no futuro. Isso é semelhante ao *overfitting*.
- Por outro lado, se o estudante não estuda o suficiente e não compreende os conceitos básicos, ele terá dificuldades em responder tanto as perguntas da prova anterior quanto as novas. Isso é semelhante ao *underfitting*.
- Finalmente, se o estudante entende os conceitos e pratica com uma variedade de perguntas, ele estará bem preparado para qualquer prova. Isso representa um modelo bem ajustado.

Para ilustrar visualmente a diferença entre *underfitting*, ajuste adequado e *overfitting*, considere a Figura 1. Nela, três cenários são comparados lado a lado: um modelo simples demais, um modelo equilibrado e um modelo excessivamente complexo.

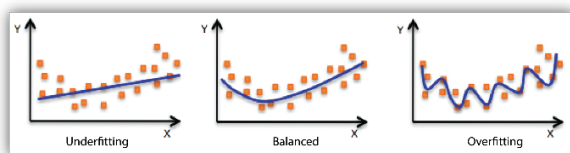


Figura 1: Comparação visual entre *underfitting*, ajuste balanceado e *overfitting*.

No primeiro caso, o modelo é incapaz de capturar o padrão dos dados (subajuste). No cenário central, observa-se um equilíbrio adequado entre simplicidade e capacidade de generalização. Já no último caso, o modelo tenta seguir cada variação dos dados de treino, prejudicando a generalização.

Para identificar a existência de *overfitting* ou *underfitting*, algumas técnicas comuns incluem a validação cruzada e a análise de curvas de aprendizado.

2.3.1. Validação Cruzada

Gomes (2025) entende que **validação cruzada** (*cross-validation*) é uma técnica estatística usada para avaliar o desempenho de um modelo e estimar sua capacidade de generalização. O método mais comum é o *k-fold cross-validation*, onde o conjunto de dados é dividido em k subconjuntos; o modelo é treinado em $k - 1$ partes e testado na parte restante, repetindo-se o processo k vezes [6] conforme observado na Figura 2.

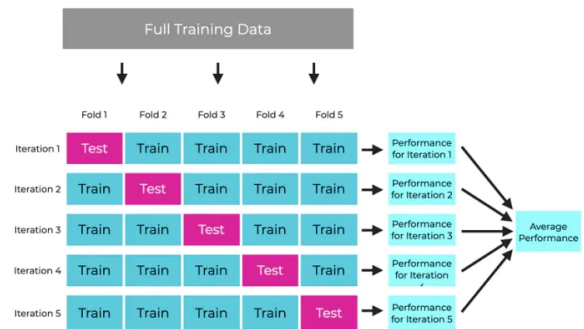


Figura 2: Esquema de validação cruzada (*k-fold cross-validation*).

Esse procedimento reduz o risco de que o desempenho do modelo dependa de uma única divisão entre treino e teste, fornecendo uma estimativa mais estável e confiável. Além de ajudar a medir a generalização, a validação cruzada é essencial para comparar modelos e detectar sinais precoces de *overfitting*.

2.3.2. Curvas de aprendizado

Para Brownlee (2019), **curvas de aprendizado** são representações gráficas que mostram a evolução do desempenho do modelo em função da quantidade de dados ou de iterações de treinamento [7].

Normalmente, plota-se o erro (ou acurácia) nos conjuntos de treino e validação ao longo do tempo. Essas curvas são ferramentas visuais eficazes para diagnosticar *overfitting* e *underfitting*:

- Se o erro de treino é baixo e o de validação é alto, há indício de *overfitting*.
- Se ambos são altos, há *underfitting*.
- Se ambos convergem para valores baixos e próximos, o modelo está bem ajustado.

As curvas possibilitam a identificação de problemas existentes como observado na Figura 3, permitindo ajustar parâmetros, regularização e quantidade de dados para alcançar o equilíbrio ideal entre aprendizado e generalização.

2.4. Troca de Viés-Variância

Lee (s.d.) aponta que **viés** diz respeito à diferença existente entre as previsões médias do modelo e os valores corretos previstos [8]. Assim, modelos com alto viés acabam introduzindo erros por conta de suposições simplificadas e tendem ao *underfitting*.

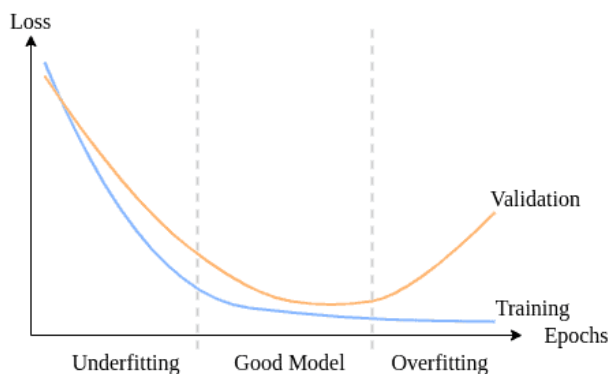


Figura 3: Exemplos de curvas de aprendizado indicando *underfitting*, ajuste adequado e *overfitting*.

A **variância**, por outro lado, representa o quanto as previsões de um modelo mudam para diferentes conjuntos de dados de teste [8]. Modelos com alta variância são mais sensíveis a ruídos e tendem ao *overfitting*.

Essas duas características são inversamente proporcionais e o desafio central para um bom modelo de *machine learning* é encontrar um ponto de equilíbrio entre ambas, conhecido como **troca de viés-variância** (ou *bias-variance trade-off*), em que se busca criar um modelo que seja suficientemente complexo para capturar padrões, mas simples o bastante para generalizar.

2.5. Regularização

A **regularização** abrange uma série de técnicas voltadas para corrigir o *overfitting* em modelos de *machine learning* de forma a aumentar a capacidade de generalização às custas de um aumento nos erros de treinamento, que corresponde a aumentar o viés de um modelo de *machine learning* como explica Goodfellow et al. (2016) [9].

Essas técnicas atuam penalizando a complexidade do modelo, incentivando soluções mais simples que se ajustem melhor aos dados de validação.

3. Desafios

A análise dos desafios relacionados ao *overfitting* e ao *underfitting* exige compreender como esses fenômenos se manifestam em aplicações reais e quais limitações impõem ao uso de modelos preditivos. Nesta seção, são discutidos casos representativos que evidenciam o impacto direto desses problemas em diferentes domínios, além das consequências práticas que afetam a confiabilidade, a robustez e o custo operacional de sistemas baseados em aprendizado de máquina.

3.1. Casos de Overfitting em Machine Learning

O *overfitting* pode emergir de maneiras diversas dependendo do domínio, mas geralmente está associado a modelos excessivamente ajustados às características específicas do conjunto de treino. A seguir, são apresentados exemplos em que essa limitação leva a falhas de generalização, comprometendo a efetividade de sistemas que dependem de previsões precisas em cenários reais.

3.1.1. Modelos de reconhecimento facial que só funcionam bem com rostos do conjunto de treino

Em sistemas de reconhecimento facial, é comum observar que modelos treinados sob condições controladas (iluminação, pose, resolução, rostos das mesmas etnias) apresentam excelente desempenho nos dados de treino, mas têm queda significativa em cenários reais variados — com diferentes idades, iluminação, resolução ou mesmo expressões faciais.

Por exemplo, um estudo feito por Zhalgas et al. (2025) mostra que modelos robustos sob condições ideais caem de rendimento quando são executados em cima de uma pose, iluminação ou qualidade degradada [10].

Esse problema indica um *overfitting* implícito: o modelo “memorizou” muitos dos efeitos do conjunto de treino (por exemplo, rostos frontais, boa iluminação, dataset específico) e não generalizou para condições variadas. Em contextos sensíveis como vigilância ou segurança, isso pode levar a falhas graves (falso-negativo ou falso-positivo).

3.1.2. Modelos de detecção de fraude que falham com padrões novos

No domínio de detecção de fraude financeira, por exemplo, os modelos treinados com dados históricos podem funcionar bem para os tipos de fraude já vistos — mas como os fraudadores mudam seus métodos constantemente, os modelos “aprendem” mal os padrões que vão se repetir e acabam falhando com novos padrões.

Uma revisão feita por Compagnino et al. (2025) afirma que modelos convencionais de ML muitas vezes falham por conta da mudança constante de conceito (*concept drift*) e das fraudes raras (altamente desbalanceadas) [11].

Portanto, mesmo que um modelo apresente acurácia alta em testes internos, se foi ajustado demais ao histórico, ele pode “quebrar” diante de novas estratégias de fraude — custo elevado em finanças, reputação e conformidade.

3.1.3. Modelos de previsão de demanda que não se adaptam a mudanças de mercado

Em previsão de demanda para varejo, modelos treinados com dados históricos podem capturar bem os padrões sazonais e tendências passadas. No entanto, quando ocorrem mudanças abruptas no mercado — como uma pandemia, mudanças econômicas ou novas preferências dos consumidores — esses modelos podem falhar em prever a demanda futura.

Isso ocorre porque o modelo pode ter se ajustado demais aos dados históricos, sem conseguir generalizar para novas condições de mercado. Por exemplo, durante a pandemia de COVID-19, muitos modelos de previsão de demanda falharam em antecipar as mudanças rápidas no comportamento do consumidor, resultando em excesso ou falta de estoque [12].

3.2. Casos de Underfitting em Machine Learning

Assim como o excesso de complexidade prejudica um modelo, a simplicidade exagerada também impede que padrões relevantes sejam capturados. Os exemplos desta subseção ilustram situações em que o *underfitting* surge por limitações estruturais ou modelagens inadequadas, resultando em desempenho insuficiente mesmo em tarefas aparentemente simples.

3.2.1. Modelos lineares simples tentando prever fenômenos complexos

Quando se utiliza um modelo linear para prever um fenômeno com relações não lineares ou muitas interações entre variáveis, o modelo pode ser simples demais para capturar os padrões reais — resultando em baixo desempenho tanto no treino quanto no teste.

Esse é um caso clássico de *underfitting*. Por exemplo: um modelo linear para prever preços de casas que ignora localização, estado, tendências de mercado e interação entre variáveis acaba apresentando erro alto, configurando *underfitting*. Assim, o problema não é “aprender demais” mas “aprender de menos”: o modelo falha em entender o que importa.

3.2.2. Redes neurais rasas para reconhecimento de imagens complexas

Em tarefas de reconhecimento de imagens, redes neurais rasas (com poucas camadas) podem não ser capazes de capturar as complexidades visuais presentes em imagens reais.

Por exemplo, uma rede neural rasa tentando classificar imagens de animais pode não conseguir aprender características complexas como texturas, formas e contextos, resultando em baixo desempenho tanto no conjunto de treino quanto no de teste.

Nesse caso, o modelo é incapaz de capturar os padrões necessários para realizar a tarefa, caracterizando *underfitting*.

3.3. Consequências do Overfitting e Underfitting

As implicações práticas do *overfitting* e do *underfitting* vão além de métricas acadêmicas: elas impactam diretamente a tomada de decisão, a confiança dos usuários e os custos associados ao desenvolvimento e manutenção de sistemas de IA. Esta subseção explora as principais consequências desses problemas, destacando por que compreender e mitigá-los é essencial em aplicações reais.

3.3.1. Impacto na tomada de decisões erradas em ambientes críticos

Quando modelos sofrem de *overfitting* ou *underfitting* em ambientes críticos — como diagnóstico médico, sistemas de crédito, detecção de intrusão ou segurança pública — as consequências são muito reais: falsos positivos ou negativos, diagnósticos incorretos, liberação de crédito quando não se deveria ou bloqueio indevido de usuários, falhas de segurança. Por exemplo, um modelo de detecção de fraude que não generaliza adequadamente pode permitir fraudes novas, gerando perdas financeiras e vazamentos de dados [10].

3.3.2. Perda de confiança em sistemas de IA

Se o modelo começa a falhar repetidamente (por exemplo, recomendando produtos irrelevantes ou fazendo mau reconhecimento facial), os usuários perdem confiança no sistema. Uma recomendação que sempre “acerta” no histórico mas falha para novos usuários gera frustração, menor engajamento e possível abandono da plataforma.

3.3.3. Tempo e recursos gastos com modelos mal ajustados

Modelos que estão mal ajustados — seja por serem *overfit* ou *underfit* — geram retrabalho: mais tempo investido em ajustar hiperparâmetros, recolher dados, re-treinar modelos, realizar validações extras, corrigir falhas em produção. Isso implica

custos adicionais de hardware, equipe, manutenção e possíveis prejuízos por decisões incorretas.

4. Mitigando Overfitting e Underfitting

Ao desenvolvermos e testarmos modelos preditivos, estamos sempre em busca de um resultado generalizado: que o modelo consiga prever resultados precisos em novos dados, após treinamento com um conjunto de dados finitos. Nesse contexto, existem maneiras e técnicas para evitar ou reduzir os efeitos trazidos pelo *overfitting* e *underfitting*, os quais serão delineados a seguir.

4.1. Mitigando Underfitting

4.1.1. Aumento de Complexidade do Modelo

O modo mais simples, porém eficiente, de mitigar o *underfitting* é aumentar a capacidade do modelo. Na prática isso pode significar muitas coisas, como:

- Aumentar a quantidade de dados de treino
- Mudar o modelo (de um linear para um quadrático por exemplo)
- Incorporando mais características ou variáveis para o modelo e parametrizando melhor (engenharia de características).

Na prática, o fundamento para essa abordagem é um profundo conhecimento da base de dados e uma boa dose de tentativa e erro, realizando ajustes e comparando com dados de teste. Ironicamente, também deve-se ter cuidado para não deixar o algoritmo complexo demais e “virar a roda”, passando de *underfitting* para *overfitting* [13, 14].

4.1.2. Engenharia de Características (Feature Engineering)

A engenharia de características é um processo pelo qual selecionamos características de um conjunto de dados ao treinar um modelo, no intuito de melhorar a sua performance. Isso inclui tanto a adição e criação de novas características, quanto a simplificação, removendo-as. Aos mais perceptivos, fica claro que as técnicas de engenharia de características abarcam os dois problemas [15]. As técnicas incluem, por exemplo, a criação de features polinomiais, termos de interação ou a aplicação de métodos de seleção de features para focar nos pontos mais potentes.

4.1.3. Regularização Excessiva

Como citado anteriormente, uma causa um tanto estranha do *underfitting* é justamente os esforços empenhados em mitigar o *overfitting*. Generalizando, muitos dos meios de mitigar o *overfitting* se tratam de simplificar o modelo para que não seja afetado por ruídos nos dados de treino. No entanto, esse procedimento realizado em excesso, pode acabar simplificando o modelo demasiadamente, fazendo-o que não perceba os padrões dos dados e passando de *overfitting* à *underfitting* [16, 17].

4.2. Regularização: Mitigando Overfitting

Pelas técnicas de regularização, os parâmetros recebem penalidades, reduzindo a variância dos modelos e, consequentemente, resultando em um modelo mais otimizado e enxuto. [18]

- **L1 (Lasso):** Adiciona uma penalidade proporcional à soma dos valores absolutos dos pesos. Sua propriedade mais notável é a capacidade de induzir esparsidade, zerando efetivamente os pesos de features consideradas irrelevantes, re-

alizando assim uma forma de seleção de features automática. [19]

- **L2 (Ridge):** Adiciona um termo de penalidade proporcional à soma dos quadrados dos pesos do modelo. Isso incentiva pesos menores, resultando em um modelo “mais suave” e menos sensível a pequenas flutuações nos dados de entrada. [20]
- **Dropout:** Introduzido por Srivastava et al. (2014), o Dropout é uma das técnicas de regularização mais impactantes desenvolvidas para redes neurais. O mecanismo central é mais profundo do que simplesmente “adicionar ruído”. Durante o treinamento, cada neurônio (e suas conexões) têm uma probabilidade p de ser temporariamente desligado (ignorado) em cada passagem de dados (forward/backward pass). A consequência crucial é que os neurônios não podem depender da presença de outros neurônios específicos para corrigir seus erros. Isso os força a aprender características que são individualmente robustas e úteis, prevenindo a “co-adaptação”. A visão teórica mais poderosa do Dropout é que ele funciona como uma forma de combinação de modelos (ensemble). Uma rede neural com n neurônios pode ser vista como uma coleção de 2^n redes diluídas (thinned networks). O Dropout treina efetivamente uma amostra dessa vasta coleção de redes, todas compartilhando pesos. No momento do teste, usar a rede completa com os pesos de saída escalados (multiplicados por p) é uma aproximação de calcular a média das previsões de todo esse ensemble exponencial. Sua eficácia foi demonstrada ao alcançar resultados estado-da-arte no MNIST, superando métodos de regularização padrão da época. [21]
- **Data Augmentation:** O aumento de dados (Data Augmentation) é outra estratégia eficaz, especialmente em tarefas como visão computacional. Ela se baseia em aumentar a quantidade de dados de treino fazendo pequenas alterações nos dados originais ajudando o modelo a generalizar melhor. No caso de visão computacional, a expansão artificial dos dados de treinamento por meio de inversão, rotação ou corte de imagens. [16]
- **Early Stopping (Parada Antecipada):** Talvez a técnica mais simples e onipresente, o Early Stopping monitora o desempenho do modelo em um conjunto de validação separado durante o treinamento. O treinamento é interrompido no momento em que a perda de validação para de diminuir (ou começa a aumentar consistentemente), evitando que o modelo entre na fase de overfitting. Sua principal fraqueza é a sensibilidade ao ruído. Devido à natureza estocástica do treinamento (especialmente com otimizadores como o SGD), as curvas de perda de validação podem ser ruidosas, tornando difícil distinguir um platô temporário ou um pequeno aumento de ruído da divergência final do overfitting. [22]

5. Estado Atual do Tema

O estado atual do tema revela uma divisão fascinante. De um lado, a prática industrial (“Mercado”) se concentra em otimizar e gerenciar robustamente o trade-off viés-variância clássico. Do outro, a pesquisa teórica de fronteira (“Teórico”) está ativamente desconstruindo esse mesmo trade-off, revelando fenômenos contraintuitivos que redefinem nossa compreensão da generalização.

5.1. Comercialmente

Vemos que no âmbito comercial, apesar de haver sim uma busca por inovação, há um foco e a necessidade muito maior de se dispor de modelos e tecnologias consolidadas. Em outras palavras, o básico bem feito é o que predomina [23]. Os principais provedores de nuvem (AWS, Google, Microsoft) definem as melhores práticas da indústria por meio de suas documentações e ferramentas de ML automatizado (AutoML).

- Amazon Web Services (AWS): A documentação da AWS recomenda um portfólio de cinco estratégias principais para prevenir o overfitting: (1) Early Stopping, (2) Pruning (seleção de features), (3) Regularização (L_1/L_2), (4) Ensemble (Bagging/Boosting) e (5) Data Augmentation [24].
- Google Cloud: O Machine Learning Crash Course do Google enfatiza fortemente a detecção do overfitting através da análise visual de curvas de perda. O sinal de alerta canônico é a divergência entre a perda de treinamento (que continua caindo) e a perda de validação (que começa a subir). A prevenção foca na qualidade dos dados e no gerenciamento da complexidade do modelo [25].
- Microsoft Azure: A plataforma Azure Automated ML oferece uma visão prática da divisão de responsabilidades. O Azure AutoML aplica automaticamente regularização (L_1 , L_2 , ElasticNet), otimização de hiperparâmetros, limitações de complexidade do modelo (especialmente para árvores de decisão) e cross-validation (validação cruzada). Isso libera o usuário para focar em tarefas que a plataforma não pode automatizar: (1) Obter mais dados de treinamento, (2) Prevenir target leakage (vazamento de dados) e (3) Usar menos features (engenharia de features manual) [26].

5.2. Academicamente

No âmbito acadêmico, o estado atual do tema é marcado por avanços teóricos que desafiam as noções tradicionais de viés-variância e generalização. Pesquisas recentes revelam que modelos altamente complexos, como redes neurais profundas, podem generalizar bem, mesmo quando estão em regimes de *overparameterization* (sobreparametrização), onde o número de parâmetros excede o número de dados de treinamento [27]. A prática moderna de deep learning é dominada por modelos sobreparametrizados — modelos que têm muito mais parâmetros (pesos) do que pontos de dados de treinamento. Redes neurais massivas e LLMs são rotineiramente treinados para alcançar erro de treinamento zero ou quase zero, um estado conhecido como interpolação [28]. De acordo com a curva em “U” clássica (viés-variância), este regime de interpolação deveria ser o pico do overfitting “maligno”, resultando em um desempenho de generalização desastroso. No entanto, empiricamente, o oposto foi observado: esses modelos interpolados não apenas generalizavam bem, mas frequentemente superavam os modelos “bem regulados” que ficavam no “ponto ideal” clássico. Essa desconexão entre a teoria clássica e a prática moderna tornou-se um dos problemas de pesquisa mais urgentes no campo [29].

6. Conclusão

Este estudo explorou overfitting e underfitting como obstáculos centrais à generalização em aprendizado de máquina. A revisão teórica estabeleceu a ligação desses fenômenos ao trade-off viés-variância e a importância de diagnósticos (validação cruzada, curvas de aprendizado). A análise de falhas práticas em sistemas críticos, como reconhecimento facial e detecção

de fraude, reiterou o impacto custoso da má generalização. Foram sistematizadas as técnicas de mitigação: para underfitting, destacam-se o aumento da complexidade do modelo e a engenharia de características; para overfitting, métodos como regularização (L1/L2), Dropout, Data Augmentation e Early Stopping são fundamentais. Identificou-se uma dicotomia central no estado da arte: a indústria (AutoML) foca em robustecer e automatizar técnicas clássicas para confiabilidade, enquanto a academia desafia a teoria tradicional. O fenômeno da sobreparametrização, onde modelos que interpolam os dados de treino atingem alta generalização, exemplifica essa revisão do consenso. Conclui-se que, embora o gerenciamento pragmático do trade-off viés-variância seja uma necessidade prática contínua, a compreensão teórica da generalização, especialmente em deep learning, está em plena evolução e longe de ser um problema solucionado. Consequentemente, podemos afirmar que há um certo perigo para áreas críticas que dependem de modelos de ML, caso a pesquisa não avance para reconciliar teoria e prática. Assim, este trabalho contribui para o entendimento dos desafios e soluções em overfitting e underfitting, destacando a necessidade de pesquisas futuras que aprofundem a compreensão teórica e aprimorem as práticas industriais.

7. Referências

- [1] K. Bartley, “Big data statistics: How much data is there in the world?” Blog, 2025. [Online]. Available: <https://rivery.io/blog/big-data-statistics-how-much-data-is-there-in-the-world>
- [2] GOOGLE CLOUD, “O que é machine learning (ml)?” Article, s.d. [Online]. Available: <https://cloud.google.com/learn/what-is-machine-learning?hl=pt-BR>
- [3] P. Domingos, “A few useful things to know about machine learning,” Article, 2012. [Online]. Available: <https://cacm.acm.org/research/a-few-useful-things-to-know-about-machine-learning>
- [4] D. Bashir, G. D. Montanez, S. Sehra, P. S. Segura, and J. Lauw, “An information-theoretic perspective on overfitting and underfitting,” 2020. [Online]. Available: <https://arxiv.org/abs/2010.06076>
- [5] S. Patel, “Overfitting vs. underfitting: A visual guide for data science beginners,” Article, 2025. [Online]. Available: <https://medium.com/@psaumya567/overfitting-vs-underfitting-a-visual-guide-for-data-science-beginners-4f8b3b961186>
- [6] P. C. T. Gomes, “Cross validation: Como avaliar modelos em machine learning?” Article, 2025. [Online]. Available: <https://www.datageeks.com.br/cross-validation>
- [7] J. Brownlee, “How to use learning curves to diagnose machine learning model performance,” Article, 2019. [Online]. Available: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance>
- [8] F. Lee, “What is bias-variance tradeoff?” Article, s.d. [Online]. Available: <https://www.ibm.com/think/topics/bias-variance-tradeoff>
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [10] A. Zhalgas, B. Amirgaliyev, and A. Sovet, “Robust face recognition under challenging conditions: A comprehensive review of deep learning methods and challenges,” *Applied Sciences*, vol. 15, no. 17, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/17/9390>
- [11] A. A. Compagnino, Y. Maruccia, S. Caviuoti, G. Riccio, A. Tutone, R. Crupi, and A. Pagliaro, “An introduction to machine learning methods for fraud detection,” *Applied Sciences*, vol. 15, no. 21, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/21/11787>
- [12] INFOR, “Épocas de crise e supply chain: pandemia exige novos modelos de projeção de demanda,” Blog, 2020. [Online]. Available: <https://www.infor.com/pt-br/blog/seasons-of-crisis-and-supply-chain-pandemic-requires-new-demand-projection-models>
- [13] PecanAI, “Mastering model complexity: Avoiding underfitting and overfitting pitfalls,” Blog, June 2024, acesso em: 23 out. 2025. [Online]. Available: <https://www.pecan.ai/blog/machine-learning-model-underfitting-and-overfitting/>
- [14] Ultralytics, “Overfitting,” Blog, acesso em: 15 out. 2025. [Online]. Available: <https://www.ultralytics.com/glossary/overfitting>
- [15] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, and D. S. Turaga, “Learning feature engineering for classification,” in *Ijcai*, vol. 17, 2017, pp. 2529–2535.
- [16] T. Mucci, “Overfitting vs. underfitting: Finding the balance,” Blog, acesso em: 20 out. 2025. [Online]. Available: <https://www.ibm.com/think/topics/overfitting-vs-underfitting>
- [17] Ultralytics, “Underfitting,” Blog, acesso em: 15 out. 2025. [Online]. Available: <https://www.ultralytics.com/glossary/underfitting>
- [18] T. Kotsilieris, I. Anagnostopoulos, and I. E. Livieris, “Special issue: Regularization techniques for machine learning and their applications,” *Electronics*, vol. 11, no. 4, 2022. [Online]. Available: <https://www.mdpi.com/2079-9292/11/4/521>
- [19] L. D. Carderelli, “O que é regularização l1 e l2,” Blog, 2021. [Online]. Available: <https://medium.com/data-hackers/o-que-%C3%A9-regulariza%C3%A7%C3%A3o-l1-l2-6697ada36a51>
- [20] N. Srivastava, “Regularization in deep learning: L1, l2 e dropout,” Blog, 2022. [Online]. Available: <https://www.e2enetworks.com/blog/regularization-in-deep-learning-l1-l2-dropout>
- [21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, 2014.
- [22] H. Li, G. K. Rajbahadur, D. Lin, C.-P. Bezemer, and Z. M. Jiang, “Keeping deep learning models in check: A history-based approach to mitigate overfitting,” *IEEE Access*, vol. 12, pp. 70 676–70 689, 2024.
- [23] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, “Hidden technical debt in machine learning systems,” *Advances in neural information processing systems*, vol. 28, 2015.
- [24] Amazon, “What is overfitting?” Blog. [Online]. Available: <https://aws.amazon.com/what-is/overfitting/>
- [25] Google, “Overfitting,” Blog. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/overfitting/overfitting?hl=pt-br>
- [26] Microsoft, “Prevent overfitting and imbalanced data with automated ml,” Blog. [Online]. Available: <https://learn.microsoft.com/en-us/azure/machine-learning/concept-manage-ml-pitfalls?view=azureml-api-2>
- [27] C. Fang, J. Ding, Q. Huang, T. Tong, and Y. Sun, “4 – the overfitting iceberg,” Blog, 2020. [Online]. Available: <https://blog.ml.cmu.edu/2020/08/31/4-overfitting/>
- [28] M. Lafon and A. Thomas, “Understanding the double descent phenomenon in deep learning,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.10459>
- [29] M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.1903070116>