

Predição de Vagas de Empregos no Mercado de IA e Dados Utilizando Redes Neurais Artificiais

Guilherme Gabriel Saldanha Pereira
Engenharia de Computação
Universidade Federal Rural do Semi-Árido (UFERSA)
Pau dos Ferros, Brasil
guilherme.pereira@alunos.ufersa.edu.br

Rosana Cibely Batista Rego
Departamento de Engenharias e Tecnologia (DETEC)
Universidade Federal Rural do Semi-Árido (UFERSA)
Pau dos Ferros, Brasil
rosana.rego@ufersa.edu.br

Resumo—O mercado de Inteligência Artificial apresenta alta variabilidade salarial e estrutural entre regiões e empresas. Este trabalho propõe o uso de redes neurais artificiais do tipo *Perceptron* de Múltiplas Camadas (MLP) para prever indicadores de vagas: salário, modalidade de trabalho e tamanho da empresa. O modelo previu salários com erro médio absoluto de 11,49%, porém apresentou limitações significativas nas classificações multiclasse, com acurácias de 33,6% para modalidade de trabalho (equivalente a predição aleatória) e 45,8% para tamanho da empresa. As principais limitações incluem: (i) arquitetura única não otimizada para cada tarefa; (ii) ausência de tratamento de desbalanceamento de classes; (iii) falta de engenharia de *features* específica para problemas categóricos. Os resultados evidenciam a eficácia de MLPs para regressão salarial, mas indicam a necessidade de modelos especializados para classificação em mercados voláteis.

Index Terms—Redes Neurais Artificiais, Aprendizado de máquina, Mercado de Trabalho, Inteligência Artificial

I. INTRODUÇÃO

O mercado global de Inteligência Artificial (IA) e Ciência de Dados apresenta um crescimento acelerado e, ao mesmo tempo, fragmentado. Novas metodologias de negócios e de trabalho são desenvolvidas constantemente, acompanhadas pela criação de cargos especializados. Consequentemente, os profissionais enfrentam uma realidade complexa, na qual as remunerações variam drasticamente entre países, empresas e modalidades de trabalho, dificultando a avaliação objetiva de oportunidades.

Esse contexto ganha ainda mais ênfase ao analisar o índice de contratações em IA monitorado pelo *AI Policy Observatory* da OCDE [1], conforme ilustrado na Figura 1.

A partir da Figura 1, observa-se que os índices não representam valores brutos, mas sim o ritmo de crescimento da área de IA em relação ao mercado total de cada nação. A métrica normalizada permite comparar a aceleração do setor entre diferentes economias.

Ao analisar países que mantêm uma trajetória ascendente, como o Brasil, infere-se que tais mercados ainda não atingiram o teto de contratação. Consequentemente, fatores críticos como a modalidade de trabalho e o tamanho das companhias permanecem fluidos. Além disso, a disparidade visualizada entre as curvas evidencia que cada país atravessa um estágio distinto de maturidade, gerando discrepâncias significativas nos indicadores de emprego.

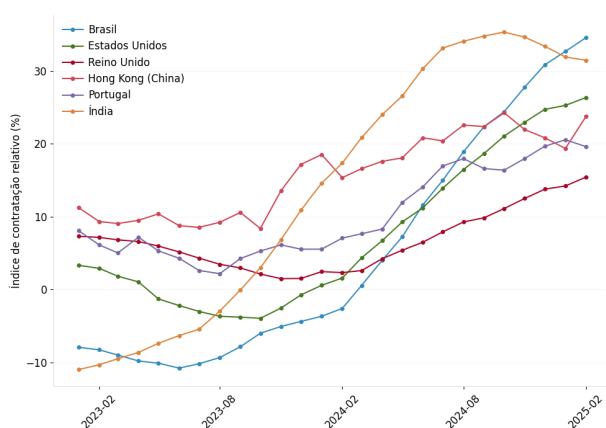


Figura 1. Índice relativo de contratação para vagas de IA ao longo do tempo (Fonte: OCDE)

Essa volatilidade reflete-se diretamente na variação salarial do setor. Tomando o cargo de Engenharia de IA como exemplo, observa-se na Figura 2 flutuações significativas. Dados extraídos do relatório da JobPikr [2] indicam que essas variações são fundamentadas na maturidade econômica e na demanda específica de mercados como o dos Estados Unidos em comparação ao Reino Unido.

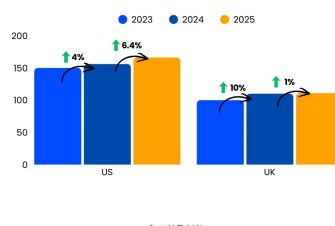


Figura 2. Variação percentual nas faixas salariais entre EUA e UK (Fonte: JobPikr)

Observa-se crescimento nos Estados Unidos e desaceleração no Reino Unido entre 2023 e 2025, evidenciando instabilidade mesmo em mercados maduros.

Este trabalho propõe uma rede neural para prever três indicadores de emprego:

- Salário (em dólares);
- Modalidade de trabalho;
- Tamanho da companhia.

O conjunto de dados utilizado para o treinamento do modelo proposto foi o *Global AI & Data Jobs Salary Dataset* [9]. Este *dataset* contém uma coleção de registros de vagas de tecnologia coletados entre os anos de 2020 e 2026, abrangendo cargos nas áreas de IA, *Machine Learning*, Engenharia de Software, Ciência e Análise de Dados.

Além das variáveis *target* já mencionadas, a base de dados inclui atributos que variam desde métricas de desempenho, como velocidade de promoção, horas trabalhadas por semana e anos de experiência, até indicadores de mercado, como percentual salarial, crescimento de carreira e dificuldade de contratação.

II. TRABALHOS RELACIONADOS

A literatura sobre a predição de indicadores no mercado de trabalho tem explorado diversas abordagens para lidar com a volatilidade do setor tecnológico com o passar do tempo. Kablaoui e Salman [3] demonstraram a eficácia de modelos baseados em redes neurais artificiais para estimar salários no setor de tecnologia utilizando *Python*, ressaltando que atributos como nível de experiência do profissional juntamente localização geográfica são os principais preditores de rendimento, o que corrobora a escolha das *features* utilizadas neste projeto, incluindo estas já citadas.

Kamalov et al. [4] demonstram que funções de ativação não-lineares (*ReLU*) e otimização via *Adam* facilitam a convergência em dados de alta dimensionalidade, justificando a estrutura do modelo MLP adotado.

Souza e Lima [5] mostram que, embora *Single-Task Learning* permita especialização eficiente para regressão, problemas de classificação multiclasse exigem balanceamento rigoroso de classes para superar a predição aleatória, fenômeno observado neste experimento.

A escolha pela arquitetura *Multilayer Perceptron* justifica-se pela sua capacidade superior de mapeamento de funções não-lineares. Enquanto autores como Lothe et al. [11] exploraram modelos de regressão linear e algoritmos de aprendizado supervisionado tradicionais em seus estudos de predição salarial, a MLP surge como uma alternativa robusta para lidar com a complexidade dos dados.

Essa abordagem permite uma modelagem que se mostra promissora para capturar a interdependência entre o salário, a modalidade de trabalho e o tamanho da companhia, especialmente em um mercado global de tecnologia marcado pela volatilidade e pela heterogeneidade, conforme discutido por Nimbalkar et al. [10].

III. METODOLOGIA

O projeto foi dividido em três fases principais:

- 1) Pré-processamento de dados
- 2) Construção do modelo

3) Avaliação do modelo por meio das métricas

O pré-processamento incluiu:

- Tratamento de dados faltantes
- *Encoding* de variáveis categóricas
- Divisão *treino/teste*
- Seleção de *features*

A variável *real_salary_usd* foi criada combinando salário e bônus respectivos a vaga. O tratamento de *outliers* foi omitido para preservar amostras, conforme Figura 3.

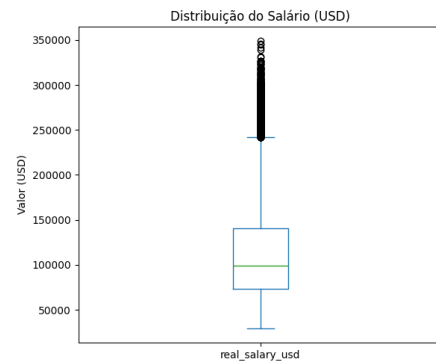


Figura 3. Boxplot da coluna referente aos salários da vaga

O algoritmo de seleção de características determina o número de entradas (*n_entradas*) para cada instância do modelo.

IV. MODELO DE REDE NEURAL

Esta seção descreve a arquitetura da rede e os métodos utilizados em sua construção, conforme ilustrado na Figura 4.

A. Perceptron de Múltiplas Camadas

A forma como o conjunto de dados percorre a rede para gerar uma predição é definida pela propagação direta (*forward propagation*). Para um neurônio individual, essa operação é representada pelo somatório ponderado de suas entradas, seguido pela adição de um termo de ajuste:

$$Y_{pred} = \left(\sum_{i=1}^n X_i \cdot W_i \right) + b \quad (1)$$

Onde:

- *n* é o número total de características (entradas) fornecidas ao neurônio.
- *X_i* representa o valor da *i*-ésima entrada (coluna do conjunto de dados).
- *W_i* representa o peso sináptico associado à *i*-ésima entrada, indicando sua importância para o modelo.
- *b* representa o fator de *bias*, que permite deslocar a função de ativação para melhor ajuste aos dados.
- *Y_{pred}* representa o valor resultante da combinação linear antes da aplicação da função de ativação.

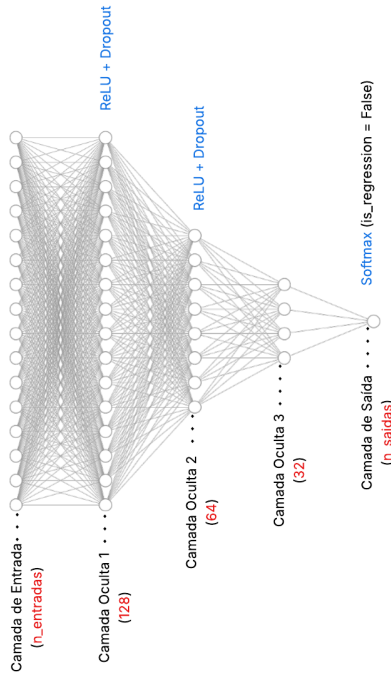


Figura 4. Arquitetura do modelo MLP

B. Aprendizagem Supervisionada

O modelo foi treinado utilizando como base o conceito de aprendizagem supervisionada, onde os rótulos representam os indicadores de interesse e a rede aprende por meio da comparação entre os valores reais oriundos da saída da rede neural e os valores ideais vindos das colunas *target* que são: *salary_usd*, *work_mode* e *company_size*.

O modelo irá aprender utilizando a técnica de retropropagação, a qual consiste na atualização dos pesos a cada nova iteração por meio do vetor de negativo do gradiente formado pela variação de cada parâmetro interno da rede (isto é, peso e *bias*). A retropropagação pode ser representada como a variação dos parâmetros internos da rede, sendo ilustrada abaixo da seguinte forma:

$$\theta_{novo} = \theta_{antigo} - \alpha \cdot \frac{\partial L}{\partial \theta} \quad (2)$$

Onde:

- θ : Representa os parâmetros internos da rede (pesos W ou *bias* b).
- α : Taxa de aprendizado (*learning rate*), que controla o tamanho do passo de ajuste.
- L : Função de custo (MSE para regressão ou Entropia Cruzada para classificação).
- $\frac{\partial L}{\partial \theta}$: Derivada parcial da perda em relação ao parâmetro, indicando a direção e magnitude do ajuste necessário para reduzir o erro.

C. Single Task Learning

Foram criados três modelos baseados na mesma estrutura. Logo, dada uma classe que define toda a arquitetura do modelo da rede neural, foram criadas três instâncias, cada uma responsável por realizar a predição de um indicador.

- Salário: regressão (1 neurônio na camada de saída, registra um único valor)
- Modalidade de trabalho: classificação (3 neurônios na camada de saída, para cada classe)
- Tamanho da empresa: classificação (5 neurônios na camada de saída, para cada classe)

As três instâncias do MLP compartilham a mesma estrutura base (Figura 4), cada modelo é adaptado ao seu problema específico, variando o número de neurônios de saída e funções de ativação.

Limitação: A utilização de uma arquitetura única para todos os problemas (regressão e classificação) impede a otimização individual de hiperparâmetros, comprometendo o desempenho nas tarefas de classificação. O atributo booleano *is_regression* apenas alterna entre configurações de saída, sem ajustes estruturais profundos.

D. Funções de Custo e Ativação

Cada problema utiliza funções de custo e de ativação diferentes pois como foi dito anteriormente, problemas de regressão e de classificação para o aprendizado supervisionado terão abordagens diferentes.

- **Função de Custo de Regressão: MSE (Erro quadrático médio)**

O erro médio quadrado será a função de custo que irá mensurar o erro médio que o indicador de salário está tendo na saída da rede, sendo este representado por:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

Onde:

- n representa o total de amostras que forma computadas na entrada da rede
- y_i é o valor do rótulo para aquela amostra específica
- $\hat{y}_i$ é o valor da predição gerado pela propagação para frente

O MSE foi escolhido por penalizar quadraticamente desvios significativos entre salário real e predito, aumentando a sensibilidade do modelo a discrepâncias.

- **Função de Custo de Classificação: Cross-Entropy (Entropia Cruzada)**

A Entropia Cruzada é a função de custo utilizada para os problemas de classificação (modalidade de trabalho e tamanho da empresa). Ela mensura a discrepância entre a distribuição de probabilidade prevista pelo modelo e a distribuição real dos rótulos. Sua expressão matemática é:

$$L = - \sum_{i=1}^K y_i \log(\hat{y}_i) \quad (4)$$

Onde:

- K : Número total de classes.
- y_i : Valor do rótulo real (0 ou 1) para a classe i , seguindo a codificação *one-hot*.
- $\hat{y}_i$: Probabilidade prevista pelo modelo para a classe i (saída da função *Softmax*).
- L : Perda (*loss*) calculada para a amostra.

Probabilidades baixas na classe correta são penalizadas exponencialmente, forçando ajustes agressivos nos pesos. É adequada para problemas com fronteiras de decisão sutis.

• **Função de Ativação: Engenharia de Computação, Universidade Federal Rural do Semi-Árido (UFERSA): ReLU**

A unidade linear retificada (*ReLU*) é aplicada entre a transição das camadas lineares no modelo. Ela é essencial para a arquitetura desta rede neural tendo em vista que é a partir desta função de ativação que introduz o fator de não-linearidade para o modelo. Sua expressão matemática é:

$$f(x) = \max(0, x) \quad (5)$$

Onde:

- x representa o valor de entrada vindo da transformação linear (soma ponderada de pesos e *bias*).
- $f(x)$ indica o valor de saída da ativação, que será 0 para qualquer entrada negativa e o próprio valor x para entradas positivas.

• **Função de Ativação: Softmax**

Utilizada exclusivamente na camada de saída para os problemas de classificação (modalidade de trabalho e tamanho da empresa), a função *Softmax* é definida por:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (6)$$

Onde:

- z_i representa o valor bruto de saída (*logit*) para a classe i .
- K indica número total de classes do problema.
- $\sigma(\mathbf{z})_i$ trata-se da probabilidade da amostra pertencer à classe i .

A *Softmax* transforma *logits* em uma distribuição de probabilidade (soma = 1.0), indicando o nível de confiança na predição.

E. Otimizador Adam

O otimizador *Adam* é utilizado para atualização dos pesos para todos os indicadores. As equações que regem o seu funcionamento são:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (7)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (8)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (9)$$

Onde:

- g_t representa o gradiente da função de custo no instante t .
- m_t é a estimativa do primeiro momento (média móvel da direção dos gradientes).
- v_t é a estimativa do segundo momento (média móvel da magnitude dos gradientes ao quadrado).
- β_1 e β_2 são os coeficientes de decaimento para as médias móveis (*hiperparâmetros*).
- α representa a taxa de aprendizado (*learning rate*).
- ϵ é uma constante pequena para evitar a divisão por zero.
- θ_t representa os parâmetros (pesos e *bias*) atualizados da rede.

F. Dropout

O *Dropout* é uma técnica de regularização utilizada para reduzir o *overfitting* do modelo, interrompendo a adaptação excessiva entre os neurônios. A operação pode ser definida como:

$$y_i = \frac{x_i}{1 - p} \cdot m_i \quad (10)$$

Onde:

- x_i representa o sinal de entrada do neurônio i .
- p representa a taxa de *dropout*.
- m_i é um vetor binário (*mask*) aleatório.
- $1/(1 - p)$ é o fator de escala aplicado.
- y_i representa o sinal de saída do neurônio i após a regularização.

Durante a fase de teste, o *dropout* é desativado para utilizar toda a capacidade da rede.

V. RESULTADOS

É possível extrair as métricas dos resultados gerados a partir do conjunto de dados de teste de cada indicador. Além disso, por meio do SHAP (*SHapley Additive exPlanations*), torna-se possível visualizar as variáveis selecionadas pelo algoritmo de *feature selection* mencionado na Fase 1, bem como identificar aquelas que exerceram maior impacto na predição de cada um dos indicadores.

A. Salário

- MAE (Erro Médio Absoluto): 12,597.62 USD
- RMSE (Raiz do Erro Quadrático Médio): 16,782.90 USD

O MAE indica erro médio de 12.597,62 USD. Considerando a faixa salarial de 40.000 a 200.000 dólares (Figura 3) e média de 109.574,66 USD, o erro relativo é de 11,49%. Na prática, este percentual implica uma margem de flutuação de aproximadamente 12,6 mil dólares anuais em relação ao valor real, podendo ser interpretado como uma flutuação aceitável sob o contexto de contratações e retenções de talentos, uma vez que se aproxima das faixas de variação comumente utilizadas em negociações salariais reais no setor de tecnologia.

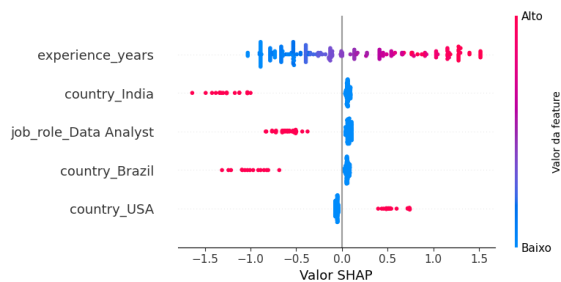


Figura 5. Gráfico de resumo SHAP para o indicador de salário

O RMSE de 16.782,90 USD é maior que o MAE devido à penalização quadrática de outliers. **Limitação:** O modelo não foi otimizado para salários extremos e não incorpora fatores contextuais (experiência, localização específica), limitando sua aplicabilidade prática.

O gráfico de resumo SHAP (Figura 5) indica que os anos de experiência (*experience_years*) representam o fator de maior impacto na determinação da faixa salarial. Além disso, a relevância de múltiplas variáveis ligadas à localidade reforça que a localização geográfica é um determinante crítico para a predição deste indicador. Isso explica a disparidade visualizada entre os índices de contratação para vagas de IA de cada país da OCDE e os relatórios da JobPikr mencionados anteriormente.

B. Modalidade de Trabalho

Tabela I
MÉTRICAS EXTRAÍDAS DO INDICADOR DE MODO DE TRABALHO

Métrica	Valor
Acurácia	0,3394
Precisão (macro)	0,3388
F1-Score (macro)	0,3046

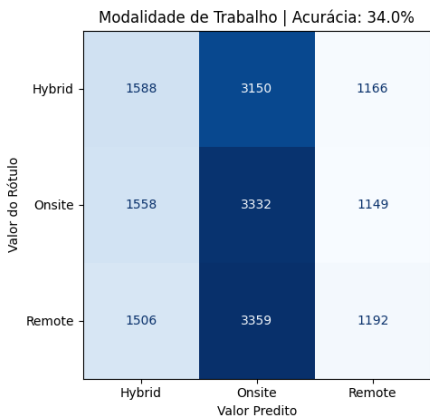


Figura 6. Matriz confusão do indicador de modalidade de trabalho

A acurácia de 33% equivale à predição aleatória para três classes. O modelo não aprendeu padrões úteis para modalidade

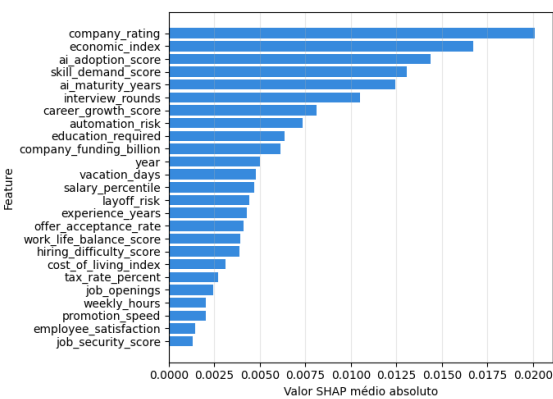


Figura 7. Gráfico de resumo SHAP para o indicador de modelo de trabalho

de trabalho. A matriz de confusão evidencia um viés para a classe "Onsite", indicando uma tendência do modelo em favorecer essa categoria. Entre as principais limitações identificadas, destacam-se: o desbalanceamento severo de classes que não foi tratado adequadamente, a utilização de uma arquitetura genérica que pode não ser a mais apropriada para a tarefa em questão, a ausência de *features* suficientemente discriminativas e a possível falta de correlação entre as variáveis disponíveis e o alvo a ser predito.

C. Tamanho da Companhia

Tabela II
MÉTRICAS EXTRAÍDAS DO INDICADOR DE TAMANHO DA COMPANHIA

Métrica	Valor
Acurácia	0,4584
Precisão (macro)	0,4168
F1-Score (macro)	0,4168

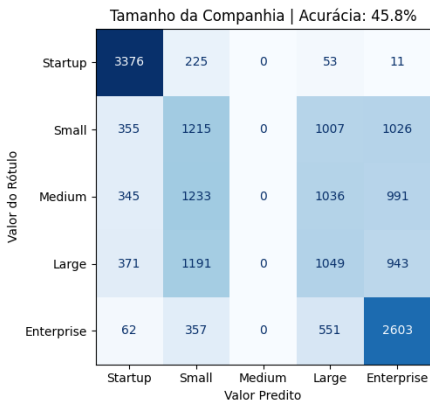


Figura 8. Matriz confusão do indicador de tamanho da companhia

Com cinco classes, predição aleatória teria 20% de acurácia. O modelo atingiu 45,8%. O *F1-Score* de 40% indica aprendizado de padrões, sem viés severo de classe. Mas a acurácia

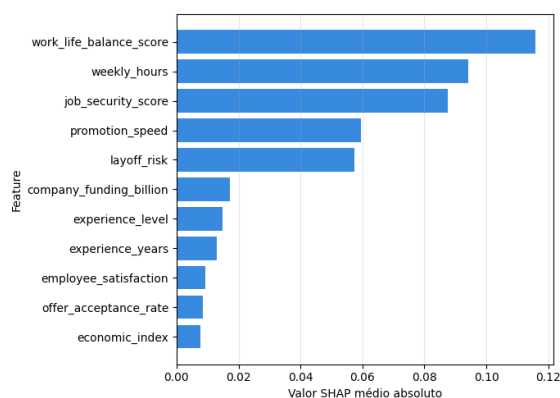


Figura 9. Gráfico de resumo SHAP para o indicador de tamanho da companhia

ainda é modesta para aplicação prática, com ambiguidade significativa entre classes intermediárias de tamanho. A ausência de otimização específica e engenharia de *features* limita o desempenho máximo.

A análise da matriz de confusão (Figura 8) revela que o modelo apresenta dificuldades em distinguir empresas de médio e grande porte. Este comportamento sugere que os atributos das vagas, como faixas salariais e exigências de experiência tornam-se homogêneos ao ponto do modelo não conseguir distinguir as classes de forma correta.

Como discutido na introdução, a fragmentação no mercado global de IA cria uma linha tênue entre diferentes tamanhos corporativos, onde muitas empresas de médio porte já adotam estruturas de contratação e de remuneração comparáveis às de grandes corporações, dificultando a classificação categórica baseada apenas em indicadores de emprego.

VI. CONCLUSÃO

O modelo MLP apresentou desempenho satisfatório para regressão salarial (erro de 11,49%), demonstrando capacidade filtrar as inconsistências globais e fornecer uma estimativa robusta. Contudo, falhou nas classificações multiclasse: modalidade de trabalho (33%, equivalente a predição aleatória) e tamanho da empresa (45,8%, modesto). As principais limitações técnicas identificadas incluem o uso de uma arquitetura genérica não otimizada para cada tarefa, o que acaba favorecendo problemas de regressão, a ausência de tratamento adequado para o desbalanceamento de classes, a falta de engenharia de *features* específica para problemas categóricos, a não incorporação de fatores contextuais relevantes e a ausência de otimização individual dos hiperparâmetros.

No que se refere às limitações do domínio, a volatilidade do mercado de IA (Figura 1) torna a predição categórica mais desafiadora, uma vez que as empresas ainda não estabeleceram padrões rígidos para tamanho da companhia e modalidade de trabalho, ao passo que salários tendem a seguir métricas mais consolidadas. Como trabalhos futuros, destacam-se o desenvolvimento de modelos especializados para cada tarefa,

a aplicação de técnicas de balanceamento como SMOTE, a exploração de arquiteturas alternativas, como *Random Forest* e *XGBoost* para classificação, além da coleta de *features* adicionais mais discriminativas.

AGRADECIMENTOS

Os autores agradecem à disciplina de Redes Neurais do curso de Engenharia de Computação da Universidade Federal Rural do Semi-Árido (UFERSA), no âmbito da qual este trabalho foi desenvolvido.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que utilizaram ferramentas de inteligência artificial, incluindo ChatGPT, Claude e Gemini exclusivamente para auxiliar na melhoria da linguagem e legibilidade deste manuscrito. O uso dessas ferramentas foi realizado sob supervisão humana, e os autores revisaram e editaram cuidadosamente o conteúdo, assumindo total responsabilidade pelo texto final publicado.

REFERÊNCIAS

- [1] OECD.AI, "AI Policy Observatory: Relative AI hiring index," 2026. [Online]. Disponível em: <https://oecd.ai>. Acesso em: 14 abr. 2026.
- [2] JobPikr, "AI Salary Benchmark Report 2026." Disponível em: <https://www.jobspikr.com/report/ai-salary-benchmark-2026/>
- [3] R. Kablaoui and A. Salman, "Machine Learning Models for Salary Prediction Dataset using Python," *2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1-5, 2022.
- [4] F. Kamalov, A. Nazir and M. Safaraliev, "Comparative analysis of activation functions in neural networks," *2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pp. 1-4, 2021.
- [5] R. Souza and A. Lima, "A Comparison of Multi-task Learning and Single-Task Learning Approaches," *Lecture Notes in Computer Science*, vol. 9, no. 1, pp. 22-34, 2023.
- [6] S. Haykin, "Redes Neurais: Princípios e Prática," 2ª ed. Porto Alegre: Bookman, 2001.
- [7] Deep Learning Book, "Como Funciona o Dropout?," 2018. [Online]. Disponível em: <https://www.deeplearningbook.com.br/como-funciona-o-dropout/>. Acesso em: 14 abr. 2026.
- [8] IBM, "O que é backpropagation?," 2024. [Online]. Disponível em: <https://www.ibm.com/br-pt/think/topics/backpropagation>. Acesso em: 15 abr. 2026.
- [9] M. K. Thalla, "Global AI and Data Jobs Salary Dataset," 2024. [Online]. Disponível em: <https://www.kaggle.com/datasets/mohankrishnathalla/global-ai-and-data-jobs-salary-dataset>. Acesso em: 05 mai. 2026.
- [10] P. Nimbalkar, A. Jadhav, P. Patil, S. Chaudhari e D. Surve, "Salary Sense: A Model Pool Driven Developer Salary Prediction and Career Intelligence System," *2026 International Conference on Computing, Communication, Control and Cyber-Physical Systems (I5CPS)*, Mumbai, India, 2026. [Online]. Disponível em: IEEE Xplore.
- [11] D. M. Lothe, P. Tiwari, N. Patil, S. Patil e V. Patil, "Salary Prediction Using Machine Learning," *International Journal of Advance Scientific Research and Engineering Trends*, vol. 6, no. 5, pp. 2456-0774, Maio 2021.