UNIVERSIDADE FEDERAL DE ALFENAS

ASSUCENA QUÉREN LOPES

ANÁLISE DE DESEMPENHO DE MODELOS SUPERVISIONADOS APLICADOS À CLASSIFICAÇÃO MULTIRRÓTULO

ALFENAS/MG

2025

ASSUCENA QUÉREN LOPES

ANÁLISE DE DESEMPENHO DE MODELOS SUPERVISIONADOS APLICADOS À CLASSIFICAÇÃO MULTIRRÓTULO

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas. Área de concentração: Aprendizado de Máquina.

Orientador: Profª. Dra. Mariane Moreira de Souza

ALFENAS/MG

2025

Sistema de Bibliotecas da Universidade Federal de Alfenas
Biblioteca Unidade Educacional Santa Clara

Lopes, Assucena Quéren.
Análise de desempenho de modelos supervisionados aplicados à classificação multirrótulo / Assucena Quéren Lopes. - Alfenas, MG, 2025.
57 f. : il. -

Orientador(a): Mariane Moreira de Souza.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Alfenas, Alfenas, MG, 2025.
Bibliografia.

1. Classificação Multirrótulo. 2. Avaliação de Modelos. 3. Aprendizado de Máquina. 4. Análise Comparativa. 5. Aprendizado Supervisionado. I. de Souza, Mariane Moreira, orient. II. Título.

Ficha gerada automaticamente com dados fornecidos pelo autor.

ASSUCENA QUÉREN LOPES

ANÁLISE DE DESEMPENHO DE MODELOS SUPERVISIONADOS APLICADOS À CLASSIFICAÇÃO MULTIRRÓTULO

O(A) Presidente da banca examinadora abaixo assina a aprovação do Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas. Área de concentração: Aprendizado de Máquina.

Aprovada em: 15 de Dezembro de 2025

Profa. Dra. Mariane Moreira de Souza — Assinatura:
Universidade Federal de Alfenas

gov.br
Documento assinado digitalmente
MARIANE MOREIRA DE SOUZA
Data: 22/12/2025 20:57:28-0300
Verifique em https://validar.iti.gov.br

Prof. Rafael José de Alencar Almeida — Assinatura:
IF Sudeste MG

Prof.ª Dr.ª Isabelle Cristinne Pinto Sampaio Costa — Assinatura:
Universidade Federal de Alfenas

RESUMO

Este trabalho investiga o desempenho de diferentes algoritmos supervisionados aplicados à classificação multirrótulo, com foco em compreender como modelos tradicionais se comportam quando avaliados de forma comparativa e estatisticamente fundamentada. Foram analisados cinco classificadores amplamente utilizados na literatura, *XGBoost*, *Random Forest*, *Multilayer Perceptron*, *Regressão Logística e k-Nearest Neighbors*, todos adaptados à abordagem *Binary Relevance* e avaliados em múltiplas métricas relevantes para cenários multirrótulo.Os resultados evidenciaram diferenças consistentes entre os modelos. O *XGBoost* apresentou desempenho elevado em métricas sensíveis ao equilíbrio entre rótulos, como *F1-macro* e *Hamming Loss*, destacando-se por sua capacidade de capturar relações complexas entre atributos. O *Random Forest* obteve os valores mais altos em métricas influenciadas por frequências de rótulos, como *F1-micro* e *Jaccard*, refletindo sua estabilidade em cenários com classes majoritárias. O *Multilayer Perceptron* apresentou desempenho intermediário e maior variabilidade entre execuções, enquanto *Regressão Logística* e k-NN foram sistematicamente inferiores aos demais modelos. Os resultados obtidos indicam que determinados modelos apresentaram desempenho mais consistente do que outros no contexto específico desta investigação, considerando as métricas avaliadas e as características da base *Yeast*. Conclui-se que esses dois métodos constituem alternativas mais eficazes para classificação multirrótulo no conjunto avaliado, cada qual apresentando vantagens específicas conforme o critério de desempenho adotado.

Palavras-chave: classificação multirrótulo; avaliação de modelos; aprendizado de máquina; análise comparativa; aprendizado supervisionado.

ABSTRACT

This work presents a comparative study of traditional supervised learning algorithms applied to the multilabel classification problem using the Yeast dataset. The objective is to evaluate how different models behave in a scenario characterized by label imbalance and complex multirrelational dependencies. Five algorithms were analyzed (XGBoost, Random Forest, Multilayer Perceptron, Logistic Regression, and k-Nearest Neighbors) using the Binary Relevance transformation and One-vs-Rest estimation scheme. Each model underwent hyperparameter optimization based on cross-validation and metric-specific threshold selection. Afterward, the best configurations were evaluated over 50 repetitions with different random seeds to assess stability and robustness. The results indicate that performance varies depending on the evaluation metric: XGBoost stands out in F1-macro and Hamming Loss, while Random Forest performs better in F1-micro and Jaccard. These findings highlight how different learning biases influence model behavior in multilabel settings and reinforce the importance of comprehensive evaluation using multiple metrics. The study thus contributes to a deeper understanding of model performance in multilabel classification and provides empirical evidence useful for future studies and methodological improvements.

Keywords: multilabel classification; supervised learning; model comparison; evaluation metrics; machine learning.

SUMÁRIO

1 INTRODUÇÃO

Em diversos domínios que lidam com a coleta e interpretação de informações, especialmente com a expansão de dispositivos conectados e sistemas baseados em Internet das Coisas (IoT), tornou-se cada vez mais comum que uma mesma instância possa estar associada a diversas categorias simultaneamente, o que caracteriza as tarefas conhecidas como classificação multirrótulo. Esse tipo de estrutura aparece em áreas como anotação de textos, categorização de imagens, recomendação de conteúdo, recuperação de informação e estudos em bioinformática, nas quais diferentes aspectos de um mesmo objeto precisam ser identificados conjuntamente (TSOUMAKAS; KATAKIS, 2007; ZHANG; ZHOU, 2014). A possibilidade de múltiplos rótulos por instância diferencia esse problema da classificação tradicional e exige abordagens capazes de lidar com essa forma particular de representação.

Ao longo dos últimos anos, diferentes métodos foram propostos para tratar a classificação multirrótulo, variando desde adaptações de algoritmos convencionais até técnicas específicas desenvolvidas para esse cenário. Entre essas abordagens, destacam-se os métodos baseados na transformação do problema, que permitem aplicar classificadores amplamente conhecidos em um contexto multirrótulo sem a necessidade de modificar profundamente suas estruturas internas (ZHANG; ZHOU, 2018; CHARTE *et al*., 2018). O uso dessas estratégias possibilita investigar, de forma sistemática, como diferentes modelos respondem quando aplicados a problemas em que mais de um rótulo deve ser previsto.

Nesse contexto, conjuntos de dados utilizados com frequência na literatura contribuem para estudos comparativos que buscam compreender o comportamento dos modelos em ambientes controlados. O conjunto *Yeast*, oriundo de dados biológicos de expressão gênica, é um desses casos. Sua estrutura, composta por múltiplos rótulos por instância e número moderado de atributos, favorece análises que exploram diferentes métodos de classificação multirrótulo (ELISSEEFF; WESTON, 2001; SZYMANSKI; KAJDANOWICZ, 2017).

Dessa forma, compreender como diferentes algoritmos se comportam ao lidar com esse tipo de dados se torna uma etapa relevante para identificar características,

limitações e potenciais aplicações dessas técnicas. Nesse sentido, este trabalho se propõe a analisar o desempenho de distintos modelos aplicados à classificação multirrótulo no conjunto de dados *Yeast*, buscando oferecer uma visão clara e organizada sobre a atuação dessas abordagens nesse tipo de tarefa.

1.1 CONTEXTUALIZAÇÃO

O projeto apresentado neste trabalho está inserido dentro de uma trajetória de pesquisa mais ampla relacionada ao estudo de técnicas de classificação multirrótulo. Embora o foco deste trabalho esteja restrito à análise comparativa de algoritmos tradicionais aplicados ao conjunto de dados *Yeast* em um cenário supervisionado, ele representa uma etapa preliminar dentro de um projeto amplo, que busca contribuições com o estado da arte, e que terá duração aproximada de quatro anos. Entretanto, o estudo apresentado nesta monografia é parte fundamental para a continuidade e desenvolvimento do todo.

Este projeto com duração estimada de 4 anos, está previsto em uma proposta de doutorado que visa explorar métodos de classificação multirrótulo em ambientes de fluxo contínuo de dados (*data stream*), nos quais as instâncias chegam de forma sequencial e potencialmente infinita, exigindo modelos capazes de adaptação constante. Em tais cenários, desafios como variação temporal, atualização dinâmica de modelos, adaptação a mudanças de distribuição e ausência de rótulos imediatos tornam o problema significativamente mais complexo.

Nesse sentido, o presente trabalho cumpre um papel importante ao fornecer uma base experimental estruturada sobre o comportamento de algoritmos clássicos em tarefas multirrótulo. Ao analisar modelos consolidados e método de transformação de problema, este trabalho estabelece referências conceituais e empíricas que servirão de apoio para etapas futuras da pesquisa, nas quais técnicas mais avançadas e cenários mais dinâmicos serão explorados. Assim, contribuindo para consolidar os conhecimentos fundamentais que permitirão, posteriormente, estender essa investigação para contextos de *data stream* e metodologias adaptativas.

1.1.1 Transformação de Problema e Adaptação de Algoritmo

Os métodos empregados para resolver tarefas de classificação multirrótulo podem ser agrupados em duas perspectivas principais: transformação do problema e adaptação de algoritmos. Essas abordagens representam estratégias distintas para lidar com a atribuição simultânea de múltiplos rótulos a uma mesma instância, e são discutidas em diferentes estudos sobre aprendizagem multirrótulo, escalabilidade e métodos avançados de modelagem (READ *et al*., 2012; PEREIRA *et al*., 2018).

A primeira abordagem, conhecida como transformação do problema, consiste em converter a tarefa multirrótulo em um conjunto de problemas de classificação tradicionais, possibilitando o uso direto de algoritmos amplamente conhecidos. Um exemplo representativo é o método *Binary Relevance*, no qual um classificador independente é treinado para cada rótulo. Abordagens transformadas se destacam pela simplicidade conceitual e pela flexibilidade, permitindo avaliar o comportamento de diferentes algoritmos sob o mesmo esquema de decomposição (ZHANG; ZHOU, 2014).

A segunda abordagem, denominada adaptação de algoritmos, modifica diretamente um método de aprendizagem para operar de forma nativa com múltiplos rótulos, sem converter o problema. Abordagens adaptadas tendem a capturar propriedades estruturais do conjunto de rótulos e a lidar com situações dinâmicas, incluindo o surgimento de novos rótulos ao longo do fluxo (YUE; ZHANG; WU, 2018).

De modo geral, a distinção entre essas duas categorias está no foco da adaptação: na transformação do problema, ajusta-se a tarefa ao algoritmo, permitindo o uso direto de modelos tradicionais; enquanto na adaptação de algoritmos, ajusta-se o algoritmo ao problema, incorporando de forma explícita particularidades do cenário multirrótulo. Ambas as abordagens são relevantes e complementares, e sua escolha depende dos objetivos do estudo, das características do conjunto de dados e do tipo de análise que se deseja realizar (PEREIRA *et al*., 2018).

1.2 OBJETIVOS

1.2.1 Objetivo geral

Contribuir para o estudo da classificação multirrótulo por meio da análise e comparação do desempenho de diferentes algoritmos aplicados ao conjunto de dados *Yeast*, investigando como essas abordagens se comportam em tarefas que envolvem múltiplos rótulos por instância.

1.1.2 Objetivos específicos

Para alcançar esse objetivo geral, o trabalho se organiza em um conjunto de objetivos específicos que orientam a investigação e delimitam os aspectos essenciais a serem analisados:

a) Examinar o desempenho de diferentes algoritmos tradicionais de aprendizagem de máquina quando adaptados ao contexto multirrótulo, utilizando a abordagem Binary Relevance.
b) Comparar os resultados obtidos por múltiplos algoritmos a partir de métricas amplamente utilizadas em classificação multirrótulo.
c) Identificar padrões, diferenças e tendências observadas no comportamento dos modelos ao lidarem com instâncias que possuem múltiplos rótulos.
d) Discutir as contribuições e limitações dos algoritmos avaliados no contexto da classificação multirrótulo.
e) Sugerir possíveis direções para estudos futuros que aprofundem a análise de modelos aplicados à classificação multirrótulo.

2 TRABALHOS RELACIONADOS

A classificação multirrótulo tem sido amplamente estudada desde o início dos anos 2000, motivada pela necessidade de modelar dados nos quais uma instância pode estar simultaneamente associada a diferentes categorias. Um dos trabalhos pioneiros e mais influentes é o de Elisseeff e Weston (2001), que introduziu métodos de margem voltados para cenários multirrótulo e consolidou o uso da base *Yeast* como referência experimental. Esse estudo inaugurou uma linha de investigação focada na avaliação sistemática de algoritmos multirrótulo em ambientes controlados.

Entre as contribuições centrais da área está o trabalho de Tsoumakas e Katakis (2007), que apresentou uma das abordagens mais utilizadas em experimentos comparativos: o *Binary Relevance* (BR). Nesse estudo, os autores revisaram formalmente diferentes estratégias para lidar com múltiplos rótulos e destacaram o BR como método base, amplamente empregado para avaliar algoritmos tradicionais devido à sua simplicidade, escalabilidade e compatibilidade com classificadores diversos. Desde então, o BR tornou-se a abordagem predominante em estudos que buscam comparar modelos de forma direta, sem incorporar técnicas especializadas ou dependências entre rótulos.

Com o avanço da área, surgiram trabalhos que exploram o impacto das relações entre rótulos e da estrutura multirrótulo sobre o desempenho dos modelos. Entre eles, Read *et al*. (2011) se destaca ao propor o *Classifier Chains*, método que explora dependências entre rótulos por meio de modelagem sequencial. Embora mais sofisticado do que o BR, o estudo serviu para demonstrar que dependências estruturais podem influenciar resultados experimentais, reforçando a importância de métodos comparativos que mantenham a independência entre rótulos quando o objetivo é analisar apenas o comportamento dos classificadores.

No campo da avaliação, pesquisas como as de Pereira *et al*. (2018) analisaram a relação entre métricas de desempenho multirrótulo, destacando que diferentes medidas capturam aspectos distintos da tarefa e que escolhas inadequadas podem levar a interpretações equivocadas. Esses estudos mostram que a comparação entre algoritmos exige cuidado metodológico, especialmente em bases como o *Yeast*, que apresentam desbalanceamento e correlação entre rótulos.

Além desses trabalhos centrais, diferentes pesquisas examinaram o

comportamento de algoritmos clássicos aplicados ao cenário multirrótulo por meio de transformações como o BR. Estudos empíricos, incluindo os de Charte *et al*. (2018) e de Zhang e Zhou (2014), reforçam o papel de algoritmos tradicionais, como k-NN, árvores de decisão, regressões lineares e redes neurais, em experimentos comparativos, servindo de referência para análises de desempenho sob condições uniformes.

Em conjunto, esses trabalhos formam o núcleo da literatura em classificação multirrótulo e fundamentam experimentos que investigam o comportamento de algoritmos tradicionais por meio da transformação *Binary Relevanc*e. É nesse contexto que o presente estudo se insere: avaliando modelos clássicos sob uma abordagem consolidada e amplamente utilizada na literatura, contribuindo para a compreensão comparativa do desempenho desses algoritmos em uma base multirrótulo estabelecida

3 ANÁLISE DESCRITIVA DA BASE DE DADOS

3.1 BASE DE DADOS *YEAST*

O conjunto de dados *Yeast* é amplamente utilizado em estudos de classificação multirrótulo e tornou-se uma referência importante em trabalhos que analisam o comportamento de algoritmos tradicionais nesse tipo de tarefa. Originalmente, o *dataset* foi proposto em pesquisas de biologia computacional voltadas à predição da localização subcelular de proteínas, utilizando medidas derivadas de expressão gênica e propriedades físico-químicas (ELISSEEFF; WESTON, 2001). Desde então, passou a integrar coleções de *benchmark* empregadas na literatura para investigações sobre métodos multirrótulo devido à sua estrutura compacta, diversidade de rótulos e características que permitem explorar diferentes aspectos desse tipo de classificação.

A base possui 1.500 instâncias, cada uma descrita por 8 atributos numéricos contínuos, que incluem informações como composição de aminoácidos, *scores* de hidrofobicidade e outras medidas projetadas para capturar características relevantes de proteínas. Diferentemente de problemas tradicionais de classificação, cada instância no *Yeast* pode estar associada simultaneamente a um subconjunto dos 14 rótulos disponíveis, que correspondem a possíveis localizações subcelulares. Essa

estrutura faz com que o conjunto apresente características típicas de problemas multirrótulo, como a presença de dependências entre classes, variação na frequência dos rótulos e múltiplas categorias ativas por observação.

Do ponto de vista experimental, o *Yeast* é especialmente adequado para estudos comparativos porque combina:

a) número moderado de instâncias, permitindo experimentos repetidos de forma eficiente;
b) atributos contínuos, que podem beneficiar ou prejudicar diferentes algoritmos;
c) desbalanceamento entre rótulos, comum em cenários reais;
d) estrutura multirrótulo clara, que exige métricas e abordagens específicas.

Assim, antes de analisar a cardinalidade, distribuição dos rótulos e demais aspectos estruturais da base, é importante compreender essa composição geral, pois ela orienta a escolha dos modelos e influencia a interpretação dos resultados experimentais apresentados nas seções seguintes.

3.1.1 Cardinalidade e densidade

A cardinalidade e a densidade dos rótulos são métricas fundamentais para caracterizar a estrutura de um conjunto multirrótulo (PEREIRA *et al*., 2018). A cardinalidade representa o número médio de rótulos ativos por instância, enquanto a densidade é a cardinalidade dividida pelo número total de rótulos do conjunto. Essas medidas fornecem uma visão direta da complexidade do *dataset* e da sobreposição entre rótulos, elementos importantes para interpretar o desempenho dos algoritmos utilizados.

Nos experimentos realizados com a base *Yeast*, foram obtidos os seguintes valores:

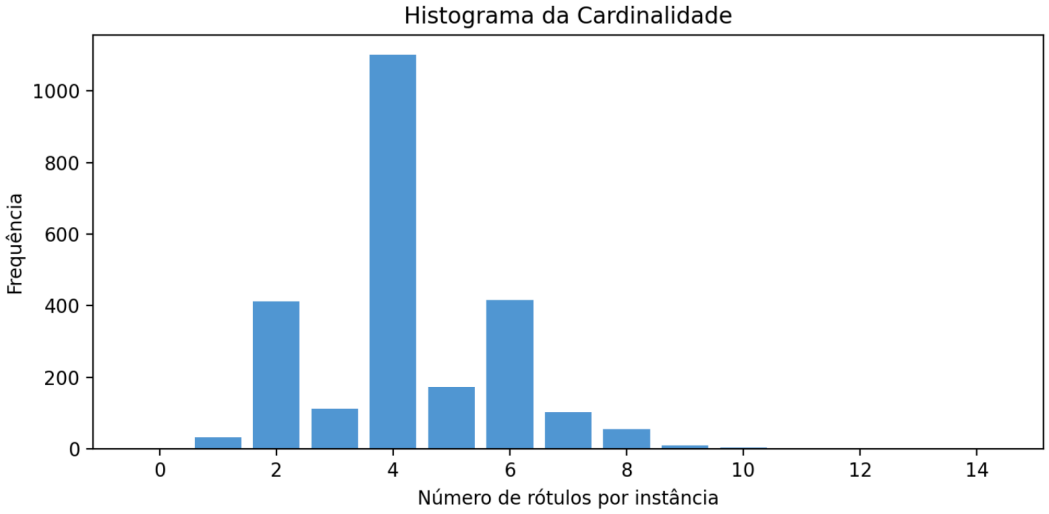
1. Cardinalidade: 4,2371
2. Densidade: 0,3026

Esses números indicam que cada instância possui, em média, quatro rótulos, aproximadamente, ativos em um total de catorze classes possíveis, caracterizando

uma base com sobreposição moderada entre rótulos, característica coerente com estudos anteriores sobre o Yeast (ELISSEEFF; WESTON, 2001).

A Figura 2 apresenta o histograma da cardinalidade. Observa-se que a distribuição é acentuadamente concentrada em torno dos valores 3, 4 e 5, com o valor 4 sendo o mais frequente. Essa concentração cria um perfil típico de bases multirrótulo intermediárias: não há predominância de instâncias com apenas um rótulo, tampouco há volumes significativos de instâncias contendo muitos rótulos. Há ainda instâncias menos frequentes com cardinalidades mais altas (6, 7 e 8 rótulos), mas representadas em menor escala, o que reforça a heterogeneidade da base.

Figura 1 - Histograma da cardinalidade da base *Yeast*.



Fonte: Autor (2025).

3.1.2 Distribuição dos rótulos

A análise da frequência individual dos rótulos constitui um passo fundamental na caracterização de conjuntos multirrótulo, uma vez que distribuições assimétricas podem afetar de maneira significativa o comportamento dos algoritmos de classificação (PEREIRA *et al*., 2018). Em bases reais, é comum que alguns rótulos ocorram com alta frequência enquanto outros sejam raros, configurando cenários de desbalanceamento que impõem desafios adicionais aos modelos.

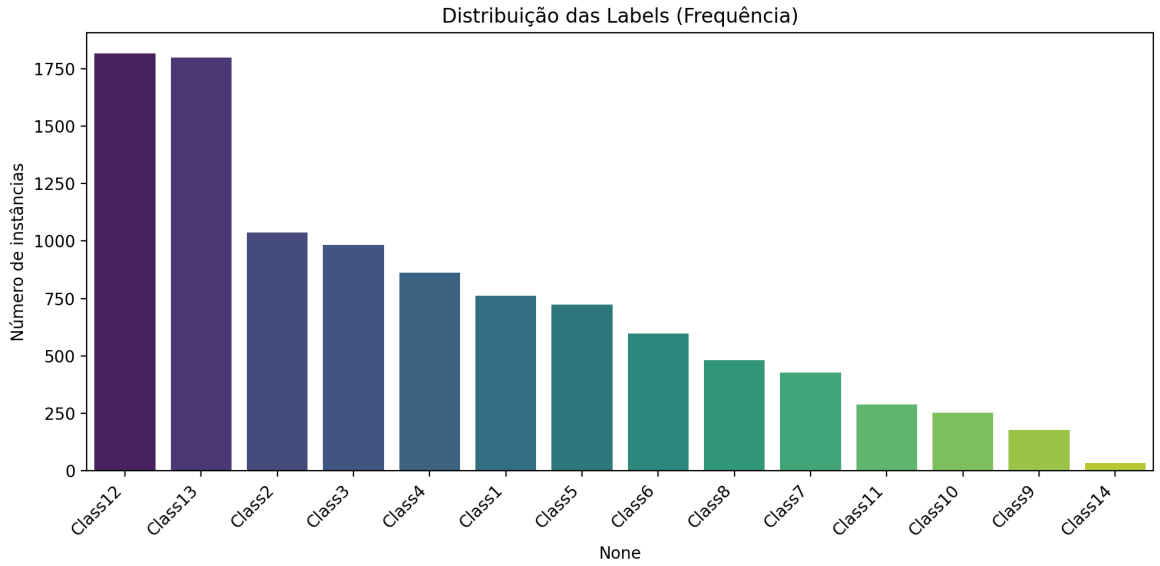
A Figura 2 apresenta a distribuição dos 14 rótulos da base Yeast. Observa-se um desbalanceamento marcante: as classes “Class12” e “Class13” são amplamente

predominantes, ultrapassando 1.800 ocorrências cada uma, enquanto outras categorias apresentam ocorrências substancialmente menores, como “Class9”, “Class10”, “Class11” e especialmente “Class14”, que é a menos frequente. Esse padrão é coerente com descrições anteriores da base *Yeast* encontradas na literatura, que destacam sua assimetria e heterogeneidade (ELISSEEFF; WESTON, 2001).

Em estudos envolvendo classificação multirrótulo, técnicas de reamostragem são frequentemente discutidas como forma de mitigar o impacto de distribuições desbalanceadas. Entretanto, neste trabalho optou-se por não aplicar métodos de balanceamento, por duas razões principais. Primeiro, o objetivo deste estudo é comparar o comportamento de algoritmos tradicionais sob a distribuição real do conjunto, preservando suas características originais. Alterar a frequência relativa dos rótulos comprometeria a comparabilidade entre os modelos e dificultaria a interpretação dos resultados. Segundo, a literatura aponta que técnicas de reamostragem podem disturbar propriedades essenciais de conjuntos multirrótulo, como estrutura de correlação, cardinalidade e dependência entre rótulos (PEREIRA *et al*., 2018), podendo introduzir vieses artificiais. Além disso, grande parte dos estudos clássicos e contemporâneos utiliza a base *Yeast* sem procedimentos de balanceamento, exatamente para manter condições uniformes de comparação (ELISSEEFF; WESTON, 2001; READ *et al*., 2012; CHARTE *et al*., 2018).

Assim, a análise da distribuição dos rótulos realizada aqui reflete fielmente a estrutura original da base, permitindo avaliar como diferentes algoritmos se comportam em um cenário multirrótulo realista e desbalanceado. A compreensão dessa distribuição é fundamental para interpretar as diferenças de desempenho entre classes majoritárias e minoritárias e para contextualizar os resultados apresentados nas seções subsequentes.

Figura 2- Distribuição da frequência dos rótulos na base Yeast



Fonte: Autor (2025).

3.1.3 Correlação entre rótulos

Em problemas de classificação multirrótulo, é comum que determinados rótulos apresentem padrões de ocorrência conjunta, refletindo relações estruturais entre as classes. A análise dessas interdependências é relevante porque o desempenho dos algoritmos pode variar significativamente conforme o grau de correlação observado entre os rótulos (PEREIRA *et al*., 2018). Bases reais tendem a apresentar esse comportamento heterogêneo, tornando essencial compreender suas associações internas.

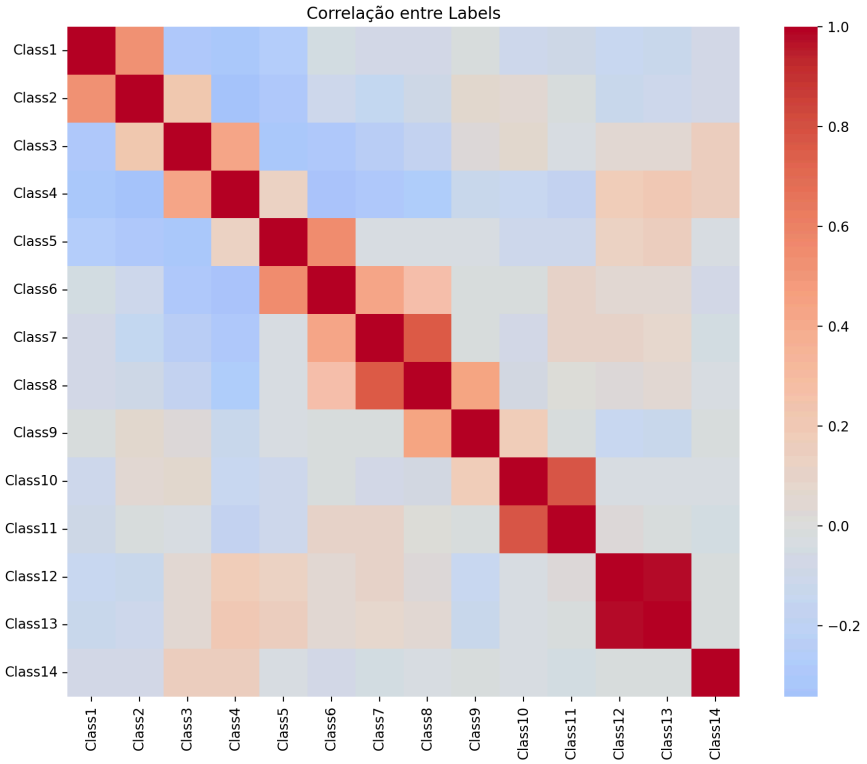
A Figura 3 apresenta a matriz de correlação entre os rótulos da base Yeast. Observa-se que alguns pares de classes exibem correlações positivas moderadas, evidenciadas pelos tons avermelhados da matriz. Isso indica que certos rótulos tendem a ocorrer simultaneamente em um número significativo de instâncias. Por outro lado, há regiões com valores próximos de zero ou negativos, representadas pelos tons azulados, sugerindo independência ou fraca associação entre outros pares de rótulos. Essa combinação de padrões reflete a natureza complexa do problema biológico modelado pela base, conforme já relatado na literatura (ELISSEEFF; WESTON, 2001).

Apesar da abordagem aplicada neste estudo, *Binary Relevance*, tratar cada rótulo de forma independente (READ *et al*., 2012), a presença de correlações

estruturadas possui implicações metodológicas importantes. Em bases que apresentam dependências fortes entre os rótulos, métodos alternativos de transformação do problema podem ser mais adequados. Abordagens como *Classifier Chains* exploram diretamente a correlação ao modelar rótulos sequencialmente, incorporando predições anteriores como atributos do próximo classificador (READ *et al.*, 2011). Da mesma forma, algoritmos especificamente adaptados para o cenário multirrótulo, como aqueles baseados em árvores, redes neurais ou *boosting* multirrótulo (SHI; SAHOO; HOI, 2012), podem capturar dependências complexas entre classes e, portanto, potencialmente apresentar desempenho superior nesses contextos.

Assim, embora o objetivo deste estudo seja avaliar o comportamento de algoritmos tradicionais sob a independência imposta pelo *Binary Relevance*, a matriz de correlação demonstra que a base Yeast contém padrões estruturados que poderiam ser explorados por métodos mais expressivos. Essa observação é relevante para pesquisas futuras que desejem aprofundar a modelagem multirrótulo incorporando explicitamente relações entre classes.

Figura 3 - Matriz de Correlação entre os rótulos da base *Yeast*



Fonte: Autor (2025).

4 METODOLOGIA

4.1 CONJUNTO DE DADOS E PRÉ PROCESSAMENTO

O conjunto utilizado neste estudo encontra-se dividido em dois arquivos: um subconjunto de treinamento e um subconjunto de teste, ambos contendo os 14 rótulos originais e o mesmo conjunto de atributos preditores. Os dados incluem variáveis numéricas derivadas de características biológicas, e cada instância pode estar associada simultaneamente a múltiplas classes.

Para este trabalho, foram considerados todos os atributos numéricos presentes no conjunto de dados, preservando integralmente a estrutura original. As colunas correspondentes aos rótulos foram identificadas e separadas das variáveis preditoras, permitindo a construção das matrizes de atributos (X) e rótulos binários (Y) necessárias para os experimentos.

Antes da modelagem, foi aplicado um procedimento de padronização dos atributos utilizando o método *StandardScaler*, que transforma cada variável para média zero e desvio padrão unitário. Esse procedimento é recomendado em algoritmos sensíveis à escala dos atributos, como redes neurais, métodos baseados em distância e modelos lineares penalizados, contribuindo para estabilidade numérica e convergência mais eficiente. A padronização foi incorporada diretamente ao pipeline de treinamento por meio de uma etapa inicial aplicada exclusivamente às variáveis preditoras, garantindo que a transformação fosse ajustada apenas com dados de treinamento em cada *fold* da validação cruzada.

A estrutura de treinamento e teste adotada reflete a divisão disponível na própria base, permitindo avaliar o desempenho dos modelos em um conjunto de dados não utilizado em nenhuma etapa da busca de hiperparâmetros. Essa separação entre as etapas de otimização e avaliação final é essencial para evitar sobreajuste ao processo de validação e assegurar uma comparação objetiva entre diferentes algoritmos de classificação multirrótulo.

4.2 ESTRATÉGIA MULTIRRÓTULO

O BR converte o problema multirrótulo em um conjunto de problemas independentes de classificação binária, treinando um modelo distinto para cada rótulo. Dessa forma, cada classificador aprende a prever a presença ou ausência do respectivo rótulo de forma isolada, com base nas mesmas variáveis de entrada (TSOUMAKAS; KATAKIS, 2007).

A adoção dessa abordagem atende ao propósito deste estudo, que visa comparar o desempenho de diferentes algoritmos clássicos em um ambiente multirrótulo sem incorporar mecanismos adicionais que explorem relações entre os rótulos. O uso do BR permite avaliar o comportamento intrínseco de cada modelo sob condições controladas, evitando a interferência de técnicas mais sofisticadas, como métodos de captura de dependências entre rótulos ou *ensembles* especializados que poderiam introduzir vieses na comparação ou dificultar a interpretação dos resultados.

A implementação do BR foi realizada por meio do estimador *OneVsRestClassifier*, fornecido pela biblioteca *scikit-learn* (PEDREGOSA *et al*., 2011). Essa estrutura operacionaliza o treinamento de um modelo para cada rótulo e gerencia internamente o processo de ajuste e predição, mantendo a independência entre as tarefas binárias. Além disso, como cada classificador retorna probabilidades para cada instância, foi possível aplicar posteriormente um procedimento de seleção de limiar (*threshold*) específico por métrica, o que contribui para uma avaliação mais precisa do desempenho multirrótulo.

O uso do BR, aliado ao *OneVsRestClassifier*, constitui assim uma escolha metodológica adequada para comparar algoritmos de naturezas distintas em condições homogêneas, permitindo isolar efeitos atribuídos aos próprios classificadores e analisar de forma transparente sua atuação no contexto da classificação multirrótulo.

4.3 MODELOS AVALIADOS

4.3.1 *XGBoost*

O *XGBoost* (*eXtreme Gradient Boosting*), proposto por Chen e Guestrin (2016), introduziu avanços significativos em eficiência computacional, regularização e paralelização, o que o tornou uma referência em experimentos empíricos e competições de aprendizado de máquina.

O algoritmo segue o princípio fundamental do *boosting*: construir um conjunto de classificadores fracos adicionados sequencialmente, em que cada novo modelo busca corrigir os erros cometidos pelos modelos anteriores. No caso do *XGBoost*, esses classificadores fracos são árvores de decisão, cujas estruturas são ajustadas para minimizar uma função objetivo composta por dois termos: um termo de perda, derivado da discrepância entre predições e rótulos verdadeiros, e um termo de regularização, que penaliza a complexidade das árvores. Essa formulação pode ser expressa da seguinte forma:

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Em que $\ell(\cdot)$ representa a função de perda escolhida e $\Omega(f_k)$ controla o tamanho e a estrutura das árvores adicionadas ao ensemble. O uso explícito da regularização ℓ_1 e ℓ_2 diferencia o *XGBoost* de versões anteriores do *boosting*, contribuindo para maior robustez e mitigação de sobreajuste (CHEN; GUESTRIN, 2016).

Outro aspecto relevante é o emprego do método de segunda ordem para otimização, no qual o algoritmo utiliza não apenas o gradiente da função de perda, mas também sua hessiana aproximada. Esse recurso permite atualizações mais precisas durante o processo de construção das árvores, favorecendo convergência mais rápida e estabilidade numérica. Além disso, o *XGBoost* incorpora técnicas como *shrinkage* (aprendizado por etapas pequenas), amostragem por colunas (*colsample*), amostragem por instâncias (*subsample*) e o algoritmo de construção de árvores baseado em histogramas, que possibilita dividirem-se dados de forma eficiente mesmo em grandes bases.

A combinação desses elementos faz do *XGBoost* um modelo adequado para

capturar relações não lineares e interações entre atributos, apresentando desempenho competitivo em uma ampla variedade de cenários supervisionados. No contexto da classificação multirrótulo empregada neste trabalho, o *XGBoost* foi utilizado como estimador base dentro da estratégia *One-vs-Rest*, permitindo que um conjunto de classificadores binários independentes fosse treinado para cada rótulo. Essa configuração explora a capacidade do modelo de aprender fronteiras de decisão complexas, ao mesmo tempo em que preserva a simplicidade estrutural da abordagem *Binary Relevance*.

O uso do *XGBoost* neste estudo se justifica por sua eficiência, capacidade de generalização e relevância consolidada na literatura contemporânea, tornando-o um candidato adequado para comparação com outros modelos clássicos de aprendizagem supervisionada.

4.3.2 *Random Forest*

O *Random Forest* é um método de aprendizado supervisionado baseado em *ensemble* que combina múltiplas árvores de decisão independentes para produzir uma predição agregada. Proposto por Breiman (2001) como uma evolução dos métodos de *bagging*, o algoritmo busca reduzir a variância característica das árvores individuais por meio da agregação de modelos treinados de maneira diversificada. Essa diversidade decorre de dois mecanismos fundamentais: (i) a amostragem *bootstrap* dos dados em cada árvore e (ii) a seleção aleatória de subconjuntos de atributos durante as divisões internas das árvores.

No processo de treinamento, cada árvore recebe uma amostra gerada por *bootstrap*, na qual instâncias são sorteadas com reposição a partir do conjunto original (EFRON, 1979). Esse procedimento cria conjuntos de treinamento levemente distintos, nos quais algumas instâncias podem aparecer mais de uma vez, enquanto outras podem não ser selecionadas. Tal reamostragem reduz a correlação entre as árvores e contribui para a diminuição da variância do *ensemble* (BREIMAN, 2001). Um efeito colateral útil desse processo é que, em média, cerca de 36,8% das instâncias não aparecem em cada amostra *bootstrap*, originando os exemplos *out-of-bag*, comumente utilizados como mecanismo de avaliação interna.

O segundo eixo de aleatoriedade ocorre no processo de construção das divisões internas das árvores. Em vez de considerar todos os atributos disponíveis, o

Random Forest avalia apenas um subconjunto aleatório a cada divisão, conforme discutido por Ho (1998). Esse procedimento impede que as árvores sigam estruturas muito semelhantes e aumenta a diversidade do ensemble, reforçando seu poder de generalização.

A predição para uma nova instância é obtida por votação majoritária no caso de classificação, o que tende a reduzir a variância do estimador sem aumento substancial do viés. Essa característica confere ao *Random Forest* uma robustez natural contra sobreajuste, especialmente quando comparado a uma única árvore de decisão, cujas predições são altamente sensíveis a pequenas perturbações nos dados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Adicionalmente, o modelo é capaz de capturar relações não lineares e interações entre atributos sem a necessidade de transformações prévias, e costuma apresentar desempenho estável mesmo em cenários com presença de ruído ou atributos pouco informativos.

No contexto da classificação multirrótulo, o *Random Forest* foi utilizado em conjunto com a estratégia *One-vs-Rest* (TSOUMAKAS; KATAKIS, 2007), construindo-se uma floresta independente para cada rótulo. Essa integração permite que o modelo produza escores contínuos, como probabilidades estimadas, que posteriormente são convertidos em decisões binárias por meio da aplicação de limiares, atendendo às exigências do cenário multirrótulo (ZHANG; ZHOU, 2014). Embora não seja um método projetado originalmente para esse tipo de problema, sua estabilidade, flexibilidade e forte desempenho empírico o tornam uma escolha adequada para análises comparativas envolvendo múltiplos algoritmos.

Por essas razões, o *Random Forest* desempenha papel relevante na investigação conduzida, oferecendo uma base sólida para avaliar o comportamento de modelos baseados em árvores em um cenário multirrótulo denso e com distribuição desbalanceada de rótulos, como ocorre no conjunto *Yeast* (PEREIRA *et al*., 2018; ZHANG; ZHOU, 2014).

4.3.3 *Multilayer Perceptron* (MLP)

O *Multilayer Perceptron* (MLP) é uma arquitetura de rede neural artificial composta por múltiplas camadas de unidades não lineares organizadas de maneira totalmente conectada. Esse modelo pertence à classe dos métodos de aprendizado supervisionado parametrizados e é capaz de aproximar funções complexas por meio

de composições sucessivas de transformações lineares e funções de ativação não lineares. A formulação teórica que fundamenta o MLP remonta ao trabalho de Rumelhart, Hinton e Williams (1986), que introduziram o algoritmo de retropropagação do erro (*backpropagation*) como mecanismo de treinamento.

Um MLP típico é composto por três partes principais: (i) uma camada de entrada, responsável por receber o vetor de atributos; (ii) uma ou mais camadas ocultas, formadas por neurônios que aplicam transformações não lineares; e (iii) uma camada de saída, cujos neurônios produzem as estimativas finais. Para um neurônio *j* em uma camada intermediária, a saída é dada por:

$$h_j = \sigma\left(\sum_i w_{ij} x_i + b_j\right)$$

Em que w_{ij} denota o peso associado à conexão entre as unidades e i, b_j é o termo de viés e $\sigma(\cdot)$ é uma função de ativação não linear, como ReLU, sigmoide ou tangente hiperbólica. A combinação sucessiva dessas transformações permite ao MLP aproximar funções altamente complexas, conforme demonstrado por teoremas de universalidade (HORNIK; STINCHCOMBE; WHITE, 1989).

O treinamento do MLP consiste em ajustar os parâmetros w_{ij} e b_j por meio da minimização de uma função de perda, usualmente otimizada com variantes de descida do gradiente. O algoritmo de *backpropagation* calcula os gradientes de maneira eficiente ao propagar o erro da saída para as camadas anteriores. Embora eficaz, esse processo pode ser sensível à inicialização dos pesos, à profundidade da rede e à taxa de aprendizado, características que tornam o comportamento do MLP altamente dependente da escolha adequada de hiperparâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016).

Modelos MLP são particularmente adequados para capturar relações não lineares entre atributos, uma vez que as funções de ativação introduzem flexibilidade para modelar interações complexas. Entretanto, ao contrário de métodos baseados em árvores, redes neurais tendem a exigir padronização dos atributos e podem demandar maior número de iterações de treinamento, além de apresentarem maior sensibilidade a hiperajustes inadequados.

No contexto da classificação multirrótulo, o MLP foi empregado neste estudo

por meio da estratégia *One-vs-Rest*, treinando-se uma rede independente para cada rótulo. Essa configuração permite que cada rede produza escores contínuos, tipicamente probabilidades geradas pela aplicação de uma função sigmoidal na camada de saída. Esse formato facilita a posterior aplicação de limiares de decisão, necessários para conversão dos escores probabilísticos em predições binárias (ZHANG; ZHOU, 2014).

Além disso, o MLP oferece uma perspectiva complementar aos demais modelos avaliados: enquanto métodos como *Random Forest* e k-NN são não paramétricos, e a Regressão Logística é um modelo linear, o MLP representa a classe de modelos neurais capazes de captar dependências não lineares profundas. Essa diversidade metodológica é importante para uma análise comparativa ampla e informada, especialmente em um cenário multirrótulo como o da base Yeast, que apresenta padrões complexos e classes de frequências distintas (PEREIRA *et al*., 2018; ZHANG; ZHOU, 2014).

Por sua flexibilidade, capacidade de modelagem não linear e ampla adoção na literatura de aprendizado supervisionado, o MLP constitui um componente essencial na avaliação comparativa conduzida neste trabalho.

4.3.4 Regressão Logística

A Regressão Logística é um modelo estatístico amplamente utilizado para tarefas de classificação binária, fundamentado na modelagem da probabilidade de ocorrência de um evento a partir de uma combinação linear dos atributos. Sua formulação clássica baseia-se na função logística (sigmoide), que transforma uma combinação linear em uma probabilidade no intervalo $[0, 1]$. Dado um vetor de atributos $x \in \mathbb{R}^p$, a probabilidade estimada de que a classe positiva ocorra é expressa por:

$$P(y = 1 \mid x) = \sigma(w^\top x + b) = \frac{1}{1 + e^{-(w^\top x + b)}}$$

Em que w representa o vetor de pesos ajustado durante o treinamento e b é o termo de viés. Essa formulação permite interpretar o modelo em termos de razões de chances e contribuições individuais de cada atributo, característica que torna a regressão logística especialmente útil em cenários que demandam interpretabilidade (HOSMER; LEMESHOW; STURDIVANT, 2013).

O treinamento consiste na maximização da verossimilhança logística, frequentemente realizada por métodos numéricos como gradiente descendente ou algoritmos quasi-Newton. Para melhorar a estabilidade do treinamento e evitar sobreajuste, são comumente adicionados termos de regularização, como penalidades ℓ_1 ou ℓ_2, resultando em variantes regularizadas do modelo (NG, 2004). A regularização ℓ_1 induz esparsidade nos coeficientes, enquanto a ℓ_2 controla a magnitude dos parâmetros, reduzindo sensibilidade ao ruído.

Embora seja um modelo linear, a Regressão Logística apresenta desempenho competitivo em diversos domínios, especialmente quando as relações entre atributos e rótulos podem ser adequadamente representadas por uma combinação linear. Entre suas vantagens destacam-se: eficiência computacional, robustez a sobreajuste em cenários moderados e facilidade de interpretação dos coeficientes.

No presente estudo, a Regressão Logística foi empregada em conjunto com a estratégia *One-vs-Rest* (TSOUMAKAS; KATAKIS, 2007), que transforma o problema multirrótulo em um conjunto de tarefas de classificação binária independentes. Para cada rótulo, ajusta-se um modelo logístico distinto, cuja saída probabilística é convertida em decisão binária mediante aplicação de um limiar estabelecido conforme o procedimento experimental adotado. Essa configuração torna a Regressão Logística adequada para o cenário multirrótulo, uma vez que modelos lineares costumam apresentar estabilidade e baixa variância, características desejáveis em experimentos comparativos.

Além disso, a inclusão da Regressão Logística neste estudo oferece um contraponto relevante aos demais modelos avaliados. Por ser um classificador linear, ela permite identificar casos em que a separabilidade linear é suficiente e ajuda a contextualizar o desempenho de modelos mais complexos, como MLP, *Random Forest* e *XGBoost*. Em um conjunto de dados como a base *Yeast*, que apresenta padrões estruturais variados e rótulos com frequências distintas, a Regressão Logística funciona como uma referência metodológica importante, ajudando a caracterizar o comportamento de classificadores tradicionais em um cenário multirrótulo (ZHANG; ZHOU, 2014; PEREIRA *et al*., 2018).

Por sua eficiência, interpretabilidade e ampla adoção na literatura de aprendizado supervisionado, a Regressão Logística constitui um componente fundamental da análise comparativa conduzida neste trabalho.

4.3.5 K-Nearest Neighbors (k-NN)

O algoritmo *k-Nearest Neighbors* (k-NN) é um método de aprendizado supervisionado baseado em instâncias, pertencente à classe dos classificadores não paramétricos. Introduzido por Cover e Hart (1967), o k-NN fundamenta-se no princípio de que a classificação de uma nova instância pode ser inferida a partir das observações mais próximas no espaço de atributos, segundo uma medida de distância previamente definida. Diferentemente de modelos paramétricos, o k-NN não realiza um processo explícito de treinamento; em vez disso, todo o conjunto de treinamento é mantido em memória e empregado no momento da predição.

Dada uma nova instância x, o algoritmo identifica o conjunto $N_k(x)$ formado pelos k exemplos rotulados mais próximos de xxx no conjunto de treinamento. A proximidade é usualmente definida por métricas como distância Euclidiana $(p = 2)$ ou *Manhattan* $(p = 1)$. Uma vez obtidos os vizinhos, a predição é realizada por votação majoritária, ponderada ou não. Na forma básica, a classe predita é dada por:

$$\hat{y} = \text{mode}\{y_i : x_i \in N_k(x)\}$$

Em versões ponderadas, atribui-se maior influência a vizinhos mais próximos por meio de pesos inversamente proporcionais à distância:

$$\hat{y} = \arg\max_{c} \sum_{x_i \in N_k(x)} w_i \cdot \mathbb{I}(y_i = c)$$

,

com $w_i = \frac{1}{d(x,x_i)}$, por exemplo. A flexibilidade do k-NN permite que o modelo capture fronteiras de decisão complexas, uma vez que a classificação é determinada localmente, refletindo estruturas regionais do espaço de atributos.

Entretanto, o desempenho do k-NN depende de forma significativa da definição do parâmetro k, da escolha da métrica de distância e da distribuição dos dados. Valores muito pequenos de k tornam o modelo sensível ao ruído, enquanto valores elevados podem suavizar excessivamente as fronteiras de decisão. Além disso, o k-NN pode sofrer com a maldição da dimensionalidade, na qual distâncias em espaços de alta dimensão tendem a perder capacidade discriminativa (BISHOP, 2006). Por essa razão, modelos baseados em distância costumam apresentar

melhor desempenho quando os atributos são previamente padronizados, como foi adotado neste estudo.

No contexto da classificação multirrótulo, o k-NN foi empregado por meio da estratégia *One-vs-Rest* (TSOUMAKAS; KATAKIS, 2007): para cada rótulo ℓ, o algoritmo estima a probabilidade de pertencimento pela fração de vizinhos positivos em $N_k(x)$:

$$\hat{P}(y_\ell = 1 \mid x) = \frac{1}{k} \sum_{x_i \in N_k(x)} \mathbb{I}(y_{i\ell} = 1)$$

Esses escores contínuos são posteriormente convertidos em decisões binárias mediante aplicação de um limiar escolhido experimentalmente. Essa abordagem torna o k-NN compatível com o *pipeline* multirrótulo adotado neste trabalho, ao mesmo tempo em que preserva suas características intrínsecas de simplicidade e localidade.

A inclusão do k-NN neste estudo fornece um contraponto relevante aos demais modelos avaliados. Por ser um método não paramétrico e baseado exclusivamente na estrutura local dos dados, ele permite investigar a capacidade de classificadores de vizinhança em capturar padrões multirrótulo em cenários com diferentes densidades e frequências de rótulos, como ocorre no conjunto *Yeast* (ZHANG; ZHOU, 2014; PEREIRA *et al*., 2018). Além disso, sua natureza simples e interpretável contribui para a diversidade metodológica da análise comparativa.

Por sua flexibilidade, transparência e ampla utilização em estudos empíricos, o k-NN representa uma peça complementar importante na avaliação conduzida, permitindo contrastar o comportamento de métodos baseados em distância com modelos lineares, neurais e baseados em árvores.

4.4 SELEÇÃO DE MODELOS

4.4.1 Validação cruzada k-fold

A validação cruzada é um procedimento estatístico amplamente utilizado para estimar o desempenho de modelos supervisionados a partir de amostras finitas de dados. Entre suas variantes, a validação cruzada do tipo *k-fold* é uma das mais empregadas, consistindo em particionar o conjunto de dados em *k* subconjuntos mutuamente exclusivos, denominados *folds*, de tamanhos aproximadamente iguais (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Em cada iteração, um dos *folds* é utilizado como conjunto de validação, enquanto os *k−1* restantes compõem o conjunto de treinamento. O processo é repetido *k* vezes, de modo que cada subconjunto atue exatamente uma vez como validação. Ao final, as métricas obtidas em cada iteração são agregadas, usualmente por meio da média, produzindo uma estimativa mais estável do desempenho do modelo.

Esse procedimento reduz a variância associada à avaliação, uma vez que o modelo é testado em múltiplas partições do conjunto de dados, mitigando a dependência de uma única divisão treino–validação. Além disso, o *k-fold* é especialmente útil em conjuntos de dados de tamanho moderado, pois permite utilizar eficientemente todas as instâncias disponíveis tanto para treinamento quanto para validação (ARLOT; CELISSE, 2010).

No caso da classificação multirrótulo, abordagens tradicionais de *k-fold* podem gerar partições com distribuições inadequadas de combinações de rótulos. Para contornar esse problema, este trabalho emprega o método *IterativeStratification*, proposto por Sechidis, Tsoumakas e Brown (2016). Esse método produz partições estratificadas capazes de preservar, em cada fold, proporções semelhantes das combinações de rótulos presentes no conjunto original, tornando a avaliação mais consistente em cenários multirrótulo.

4.4.2 Limiar (Threshold)

Modelos supervisionados aplicados à classificação multirrótulo geralmente produzem saídas contínuas, como probabilidades estimadas ou *decision scores*, que representam a evidência numérica atribuída pelo modelo ao pertencimento de uma instância a cada rótulo. Tais valores, porém, não constituem diretamente uma decisão categórica. Para transformá-los em predições binárias, é necessário definir um limiar de decisão (*threshold*), mecanismo amplamente discutido na literatura de aprendizagem supervisionada e multirrótulo (ZHANG; ZHOU, 2014).

O limiar opera como um ponto de corte: valores iguais ou superiores ao limite são interpretados como predição positiva, enquanto valores inferiores são convertidos em predição negativa. Esse processo desempenha papel fundamental na determinação do equilíbrio entre diferentes aspectos do desempenho preditivo, como precisão, cobertura e trade-offs entre falsos positivos e falsos negativos (SURESH; GUTTAG, 2021). A adequação do limiar é particularmente relevante em cenários multirrótulo devido à heterogeneidade entre os rótulos, que podem apresentar distribuições distintas, diferentes prevalências ou níveis variados de desbalanceamento, tornando thresholds fixos, como o convencional 0,5, potencialmente subótimos (PEREIRA *et al*., 2018)

Estudos na área argumentam que a escolha cuidadosa do limiar pode melhorar significativamente o desempenho em métricas sensíveis à estrutura multirrótulo, como *F1* e *Jaccard* (ZHANG; ZHOU, 2014; TSOUMAKAS; KATAKIS, 2007). Dessa forma, o limiar tem função central na etapa de inferência: ele ajusta a forma como o modelo transforma evidências contínuas em atribuição efetiva de rótulos, influenciando diretamente o comportamento final do classificador.

4.4.3 *Grid Search*

A busca de hiperparâmetros foi conduzida por meio de uma busca em grade (*grid search*), técnica que consiste em avaliar sistematicamente combinações pré-definidas de valores para os hiperparâmetros de cada modelo. Esse procedimento é amplamente utilizado em estudos de comparação empírica por oferecer reprodutibilidade e controle explícito sobre o espaço de busca (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

No presente trabalho, as grades de hiperparâmetros foram definidas com abrangência moderada, contemplando valores comumente recomendados na literatura e na documentação das bibliotecas correspondentes (BREIMAN, 2001; CHEN; GUESTRIN, 2016; PEDREGOSA *et al*., 2011). O objetivo foi garantir que cada modelo tivesse a oportunidade de operar em configurações representativas, sem, no entanto, explorar espaços demasiadamente amplos ou complexos.

Para o *XGBoost*, foram incluídos hiperparâmetros centrais como *n_estimators*, *max_depth* e *learning_rate*, que influenciam a capacidade do *ensemble* de árvores em capturar padrões não lineares (CHEN; GUESTRIN, 2016).

O *Random Forest* foi explorado por meio de variações em *n_estimators*, *max_depth*, *min_samples_leaf* e max_features, parâmetros associados ao controle de variância e à robustez do método (BREIMAN, 2001).

Para o *Multilayer Perceptron* (MLP), foram consideradas diferentes profundidades e larguras de rede (*hidden_layer_sizes*), além de valores de regularização (*alpha*) e taxas de aprendizado (*learning_rate_init*), que influenciam o comportamento do processo de otimização (RUMELHART; HINTON; WILLIAMS, 1986).

No caso da Regressão Logística, hiperparâmetros relacionados à regularização (C), ao tipo de penalidade (*penalty*) e ao algoritmo de otimização (*solver*) foram incluídos devido ao impacto na estabilidade e na convergência do modelo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Para o *k-Nearest Neighbors* (k-NN), foram explorados parâmetros que controlam a noção de proximidade entre instâncias, como o número de vizinhos (*n_neighbors*), o tipo de ponderação (*weights*) e a métrica de distância (p) (COVER; HART, 1967).

As grades utilizadas foram dimensionadas de forma a permitir variação suficiente para capturar diferentes regimes de comportamento dos modelos, mas sem constituir um espaço de busca excessivamente amplo. Dessa forma, evitou-se tanto a necessidade de custos computacionais elevados quanto o risco de avaliações instáveis decorrentes de combinações altamente específicas de hiperparâmetros. A extensão moderada dessa busca permitiu uma comparação equilibrada entre os algoritmos, refletindo práticas comuns em estudos experimentais de classificação multirrótulo.

4.4.4 Fluxo de execução do experimento

A Figura 4 representa a execução do experimento, o qual seguiu um fluxo padronizado, onde cada combinação de hiperparâmetros definida para os modelos foi avaliada de forma consistente em ambiente multirrótulo. A busca em grade foi conduzida sobre o conjunto de valores estabelecidos previamente para cada algoritmo, conforme recomendações da literatura e das implementações utilizadas (CHEN; GUESTRIN, 2016; BREIMAN, 2001; PEDREGOSA *et al*., 2011). Cada configuração era então submetida ao procedimento de validação cruzada estratificada multirrótulo com cinco partições, implementada por meio do método *IterativeStratification*, garantindo que a avaliação ocorresse sob partições comparáveis (SECHIDIS; TSOUTSOURAS; BROWN, 2016).

Em cada *fold*, o *pipeline* composto pela padronização dos atributos e pelo estimador *OneVsRest* era ajustado ao conjunto de treinamento e aplicado ao conjunto de validação. As probabilidades produzidas para cada rótulo serviam de base para a etapa subsequente, na qual se realizava uma varredura de limiares no intervalo entre 0,05 e 0.9, com incrementos de 0,05. Para cada valor de limiar, calcularam-se as métricas consideradas no estudo, permitindo identificar, dentro de cada fold, o limiar que proporciona o melhor desempenho segundo cada métrica avaliada (PEREIRA *et al*., 2018; ZHANG; ZHOU, 2014).

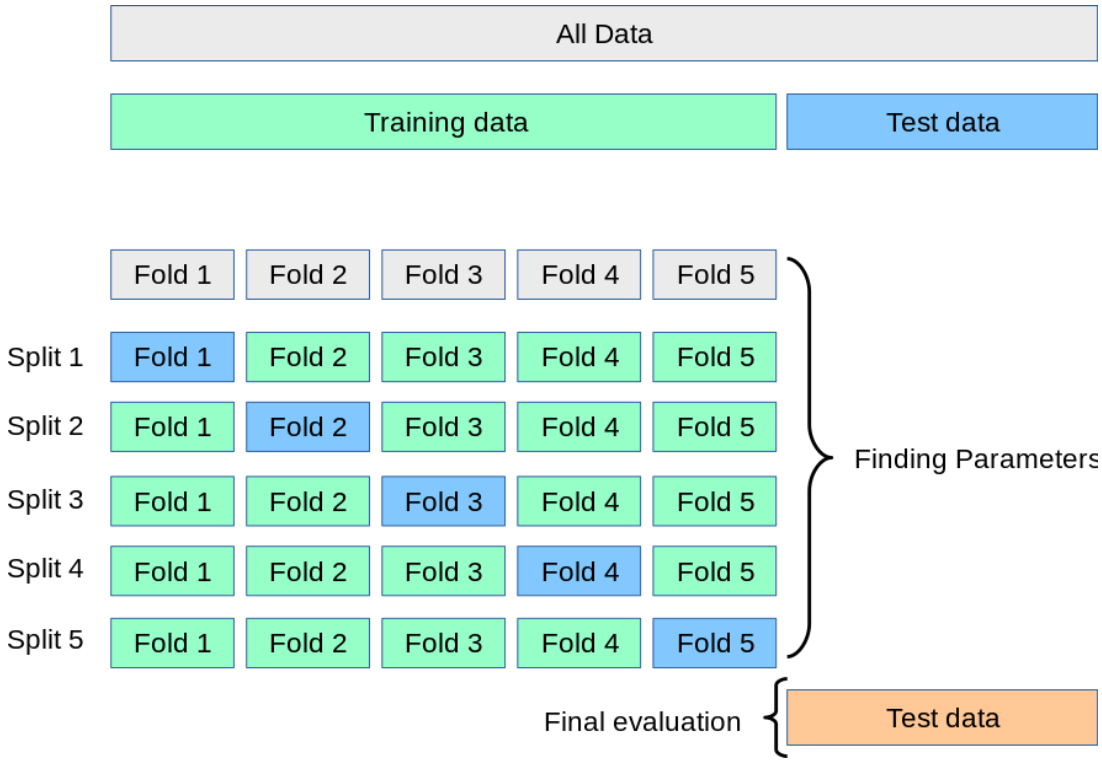
Concluídas as iterações da validação cruzada, eram calculadas as médias dos melhores resultados e as médias dos melhores limiares correspondentes, produzindo estimativas mais estáveis do comportamento de cada configuração. Para cada algoritmo e para cada métrica, selecionou-se a combinação de hiperparâmetros que apresentou o melhor desempenho médio ao longo dos *folds*. Essas configurações foram registradas individualmente, permitindo sua utilização posterior na etapa de avaliação final em conjunto de teste independente. Essa estratégia busca reduzir o risco de *overfitting* a valores pontuais de limiar e proporcionar um limiar médio mais estável, derivado da estrutura dos dados observada nos diferentes *folds*.

Como resultado desse processo, obteve-se, para cada um dos modelos avaliados, o conjunto de hiperparâmetros que apresentou o melhor desempenho

médio em cada métrica, acompanhado do limiar correspondente que otimizou essa mesma métrica ao longo da validação cruzada.

Ao término do processo de otimização, foram obtidos, para cada algoritmo avaliado (*XGBoost, Random Forest, Multilayer Perceptron*, Regressão Logística e k-NN) quatro conjuntos distintos de hiperparâmetros, cada um correspondente a uma métrica utilizada como critério de otimização (F1-macro, F1-micro, Hamming Loss e Jaccard). Dessa forma, cada modelo foi ajustado especificamente para maximizar (ou minimizar, no caso da Hamming Loss) o desempenho segundo uma métrica particular, resultando em configurações especializadas que refletem diferentes perspectivas de avaliação no cenário multirrótulo. Esses conjuntos de hiperparâmetros foram posteriormente empregados na etapa de avaliação final, assegurando que a comparação entre os algoritmos considerasse o melhor desempenho possível de cada um sob cada critério analisado.Esse material serviu de base para a análise comparativa apresentada nas seções posteriores.

Figura 4- Representação do processo de validação cruzada k-fold empregado para avaliar cada combinação de hiperparâmetros no cenário multirrótulo.



Fonte: DeepLearning.AI Comunity (2023)

4.5 MÉTRICAS ADOTADAS

A avaliação de modelos em tarefas de classificação multirrótulo requer métricas capazes de capturar diferentes dimensões do desempenho, considerando simultaneamente tanto a qualidade das decisões para cada rótulo quanto a fidelidade à estrutura conjunta dos rótulos atribuídos a cada instância. Neste trabalho, adotaram-se cinco métricas amplamente reconhecidas na literatura: *F1-macro, F1-micro, Hamming Loss* e *Jaccard (média por amostra)*. A escolha desse conjunto se justifica por sua complementaridade, uma vez que cada métrica destaca um aspecto distinto do comportamento dos modelos, permitindo uma análise abrangente, equilibrada e coerente com o objetivo de comparar algoritmos tradicionais em um cenário multirrótulo (ZHANG; ZHOU, 2014; PEREIRA *et al*., 2018).

A métrica *F1-macro* avalia separadamente o desempenho em cada rótulo e, em seguida, calcula a média aritmética dos valores obtidos. Sua formulação utiliza a média harmônica entre precisão e revocação. Seja P_ℓ a precisão e R_ℓ a revocação do rótulo ℓ, o F1 por rótulo é definido como:

$$F1_\ell = \frac{2 \cdot P_\ell \cdot R_\ell}{P_\ell + R_\ell}$$

E o *F1-macro* como:

$$F1_{\text{macro}} = \frac{1}{L} \sum_{\ell=1}^{L} F1_\ell$$

Ao atribuir peso igual a todos os rótulos, independentemente da frequência, o *F1-macro* é particularmente adequado para bases em que há forte desbalanceamento entre rótulos, como ocorre na base Yeast (TSOUMAKAS; KATAKIS, 2007). Sua maximização permite identificar algoritmos que mantêm desempenho consistente mesmo em rótulos menos frequentes.

Complementarmente, a métrica *F1-micro* agrega globalmente as contagens de verdadeiros positivos (*TP*), falsos positivos (*FP*) e falsos negativos (*FN*) antes do

cálculo do F1. Ela é dada por:

$$F1_{\text{micro}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Diferentemente do *F1-macro*, o *F1-micro* é mais influenciado por rótulos frequentes, sintetizando o desempenho global do modelo em todas as decisões multirrótulo (ZHANG; ZHOU, 2014). Por esse motivo, sua maximização fornece uma visão complementar que ajuda a comparar o comportamento geral dos algoritmos.

A métrica *Hamming Loss* quantifica o erro médio por rótulo, avaliando a proporção de decisões incorretas entre todas as combinações instância–rótulo. Sua formulação é dada por:

$$\text{HammingLoss} = \frac{1}{N \cdot L} \sum_{i=1}^{N} \sum_{\ell=1}^{L} \mathbb{I}\left[Y_{i\ell} \neq \hat{Y}_{i\ell}\right]$$

Por medir erro e não acerto, essa métrica deve ser minimizada. Sua sensibilidade a erros individuais por rótulo a torna particularmente útil para identificar algoritmos que cometem pequenas imprecisões de forma recorrente, oferecendo uma perspectiva mais granular do desempenho (TSOUMAKAS; KATAKIS, 2007).

A métrica *Jaccard* (média por amostra) avalia o grau de sobreposição entre o conjunto de rótulos verdadeiros e o conjunto de rótulos previstos para cada instância. Para cada instância *i*, o índice é definido como:

$$J_i = \frac{\left|Y_i \cap \hat{Y}_i\right|}{\left|Y_i \cup \hat{Y}_i\right|}$$

A pontuação final corresponde à média desses valores ao longo de todas as instâncias. O *Jaccard* é particularmente sensível tanto à omissão quanto ao excesso de rótulos previstos, tornando-se uma métrica adequada para investigar algoritmos que tendem a superpredizer ou subpredizer o número de rótulos atribuídos (PEREIRA *et al*., 2018). Por representar similaridade entre conjuntos, deve ser maximizado.

A combinação dessas métricas está diretamente alinhada ao propósito deste estudo. Enquanto *F1-macro* e *Hamming Loss* permitem observar o desempenho por rótulo, *F1-micro* resume o comportamento global, e *Jaccard* avaliam o quão fielmente os conjuntos de rótulos previstos refletem os conjuntos verdadeiros. Essa diversidade garante uma análise robusta e evita conclusões baseadas em uma única perspectiva de desempenho, permitindo comparar de forma abrangente algoritmos clássicos aplicados ao cenário multirrótulo.

4.6 AVALIAÇÃO NO CONJUNTO DE TESTE

Após a etapa de seleção de hiperparâmetros, realizou-se a avaliação final dos modelos no conjunto de teste, utilizando exclusivamente as configurações previamente identificadas como ótimas para cada métrica durante o processo de validação cruzada. Esse procedimento segue a recomendação de separar rigorosamente as fases de seleção e avaliação, evitando que o desempenho observado seja influenciado por escolhas ajustadas ao conjunto de treinamento ou às partições da validação cruzada (CAWLEY; TALBOT, 2010).

Para cada algoritmo considerado, *XGBoost*, *Random Forest*, *Multilayer Perceptron*, *Regressão Logística* e *k-Nearest Neighbors*, e para cada uma das quatro métricas otimizadas, foi utilizada a combinação de hiperparâmetros selecionada na etapa anterior, juntamente com o limiar médio associado à respectiva métrica. O uso de limiares fixados exclusivamente a partir dos *folds* de validação garante que o conjunto de teste permaneça completamente independente durante todo o processo, preservando a validade estatística da avaliação.

A avaliação final foi conduzida mediante 50 repetições independentes para cada par (modelo, métrica), variando-se a semente de aleatoriedade em cada repetição. Essa estratégia, amplamente empregada em experimentos envolvendo modelos sensíveis à inicialização ou à ordem dos dados (DIETTERICH, 1998; DEMŠAR, 2006), permite estimar a estabilidade dos algoritmos, reduzindo a influência de variações estocásticas e possibilitando análises mais robustas. Em cada repetição, o *pipeline* contendo padronização dos atributos e o estimador *OneVsRest* foi ajustado ao conjunto completo de treinamento e posteriormente aplicado ao conjunto de teste. A partir das probabilidades estimadas, aplicou-se o

limiar selecionado para cada métrica, produzindo as predições binárias utilizadas no cálculo dos indicadores.

As quatro métricas descritas anteriormente, *F1-macro*, *F1-micro*, *Hamming Loss*, e *Jaccard* (média por amostra), foram calculadas em cada repetição, resultando em uma distribuição de 50 valores por combinação de modelo e métrica. Esses resultados foram registrados integralmente, permitindo posteriormente a construção de estatísticas resumidas, tais como média, desvio-padrão e amplitude interquartil, bem como a comparação entre modelos ao longo de diferentes dimensões de desempenho.

Esse procedimento, ao combinar reavaliação independente no conjunto de teste, múltiplas repetições e uso de configurações previamente selecionadas, corrobora para que a comparação entre os algoritmos seja conduzida de forma rigorosa, reprodutível e estatisticamente sólida. A estrutura adotada permite distinguir diferenças reais de desempenho de flutuações decorrentes de fatores aleatórios, fornecendo uma base confiável para a análise apresentada nas seções posteriores.

5 RESULTADOS E DISCUSSÕES

5.1 ANÁLISE DE ESTABILIDADE DOS MODELOS

A análise de estabilidade tem como objetivo avaliar o quanto o desempenho dos modelos varia quando submetidos a diferentes inicializações aleatórias, refletindo sua robustez e consistência estatística. Para isso, cada modelo teve sua melhor configuração, definida previamente pela otimização em cada métrica, executada 50 vezes, alterando-se apenas a semente aleatória. Essa abordagem permite observar não apenas o valor médio das métricas, mas também a dispersão e a presença de possíveis outliers, fornecendo uma visão mais completa sobre a confiabilidade dos modelos em cenários de classificação multirrótulo. Nos próximos tópicos, são apresentados *boxplot* detalhados para cada métrica, evidenciando o comportamento comparativo entre os cinco modelos avaliados.

5.1.1 Comparação dos modelos otimizados para métrica F1-Macro

A Figura 5 apresenta a distribuição dos valores de *F1-macro* obtidos nas 50 execuções independentes para cada algoritmo, considerando que todos foram previamente otimizados de acordo com essa métrica. O *F1-macro*, por atribuir igual peso a cada rótulo independentemente de sua frequência, é particularmente importante no contexto da base *Yeast*, caracterizada por forte desbalanceamento entre classes (TSOUMAKAS; KATAKIS, 2007; ZHANG; ZHOU, 2014). Assim, modelos que alcançam melhores valores de F1-macro tendem a apresentar maior capacidade de aprendizado em rótulos raros, aspecto crucial para a comparação conduzida.

Os resultados corroboram que o *XGBoost* apresenta um desempenho médio elevado dentre os modelos analisados, com mediana superior a 0,48 e baixa variabilidade entre as execuções. Essa estabilidade sugere que o método de *boosting* gradiente, ao combinar árvores fracas de maneira sequencial, foi capaz de capturar padrões relevantes tanto em rótulos frequentes quanto nos menos representados, mantendo coerência mesmo sob variações aleatórias introduzidas pelas *seeds*. Tal comportamento é consistente com a literatura, que aponta o *XGBoost* como um método robusto para cenários com fronteiras de decisão complexas (CHEN; GUESTRIN, 2016).

O *Random Forest* apresenta distribuição mais estável, com mediana próxima de 0,468. A reduzida variabilidade é coerente com a natureza do algoritmo, que combina múltiplas árvores construídas a partir de amostras *bootstrap* e subconjuntos aleatórios de atributos, resultando em *ensembles* conhecidos por sua robustez (BREIMAN, 2001).

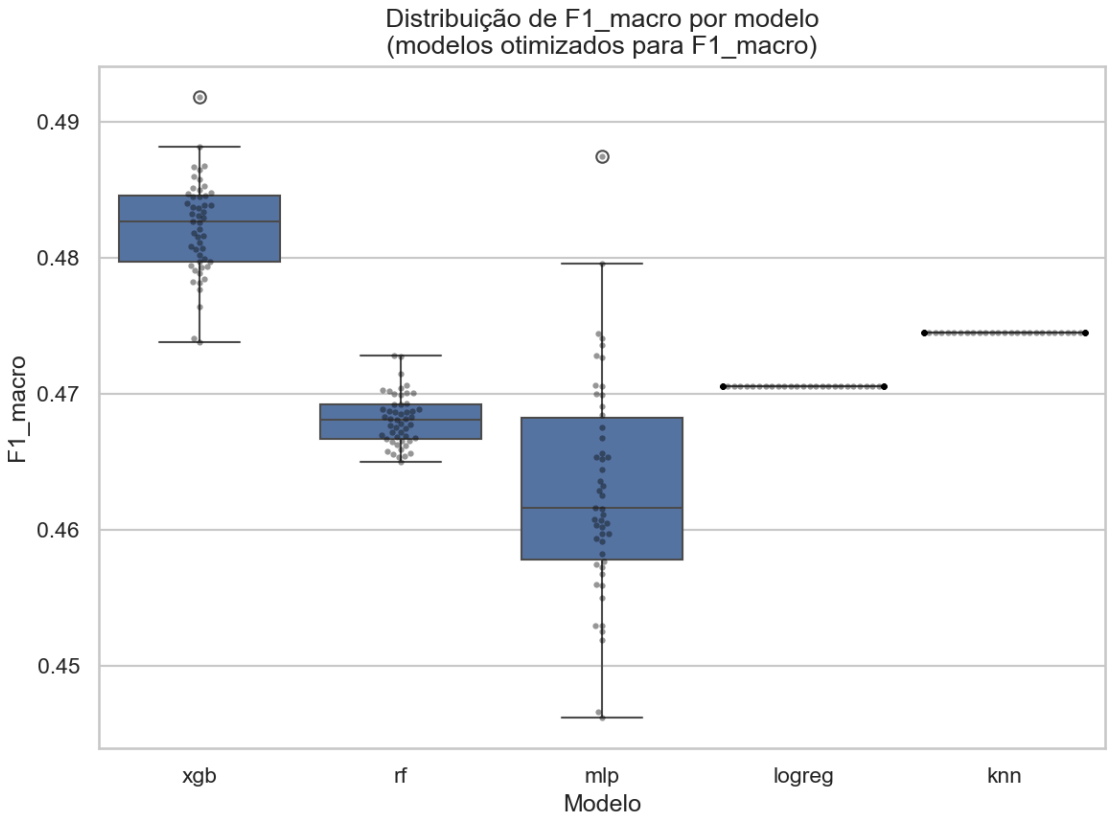
O MLP exibe a maior amplitude interquartil dentre os métodos analisados, com valores variando aproximadamente entre 0,447 e valores próximos aos do *Random Forest.* Essa oscilação é consistente com propriedades de redes neurais treinadas por otimização não convexa, nas quais diferentes inicializações podem conduzir a mínimos locais distintos (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Regressão Logística e o k-NN exibem medianas em torno de 0,47. No entanto, diferentemente do MLP, esses dois métodos apresentam variabilidade praticamente nula, o que reflete sua natureza determinística ou quase determinística: o k-NN não depende de inicialização aleatória e a Regressão

Logística, embora dependa, tende a convergir para soluções muito semelhantes dadas as características lineares do modelo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

De maneira geral, os padrões observados revelam diferenças relevantes na estabilidade das famílias de algoritmos: métodos baseados em *ensembles* de árvores apresentam comportamento estável, redes neurais exibem maior sensibilidade à aleatoriedade e métodos lineares ou baseados em vizinhança produzem distribuições estreitas, ainda que com valores mais modestos. A análise do *F1-macro*, portanto, fornece uma visão inicial sobre o comportamento dos modelos sob repetição controlada, permitindo observar suas características estruturais de estabilidade e variabilidade no cenário multirrótulo.

Figura 5 - Boxplot das 50 repetições dos modelos otimizados para métrica F1-macro



Fonte: Autor (2025)

5.1.2 Comparação dos modelos otimizados para métrica *F1-Micro*

A Figura 6 apresenta a distribuição dos valores de *F1-micro* obtidos nas 50 execuções independentes para cada algoritmo, considerando que todos foram otimizados especificamente para essa métrica. Diferentemente do *F1-macr*o, o *F1-micro* agrega contagens globais de verdadeiros positivos, falsos positivos e falsos negativos antes de calcular a pontuação, o que faz com que rótulos mais frequentes exerçam maior influência sobre o resultado final (ZHANG; ZHOU, 2014). Assim, o *F1-micro* tende a refletir predominantemente o desempenho em classes majoritárias, sendo especialmente útil para avaliar o comportamento dos modelos em bases com distribuição desigual de rótulos, como é o caso da *Yeast* (TSOUMAKAS; KATAKIS, 2007).

Os resultados observados para o *Random Forest* mostram medianas ligeiramente superiores às do *XGBoost*, com valores próximos a 0,682 e baixa variabilidade ao longo das execuções. Esse comportamento está alinhado à literatura, que descreve *ensembles* de árvores como métodos particularmente estáveis em cenários supervisionados, devido à combinação de modelos construídos de forma independente e à redução de variância promovida pela agregação (BREIMAN, 2001). A pequena dispersão sugere que alterações na semente aleatória impactam marginalmente o desempenho final.

O *XGBoost*, por sua vez, apresenta uma distribuição também concentrada, com mediana em torno de 0,680. A proximidade entre os valores observados para *Random Forest* e *XGBoost* indica que ambos os métodos foram capazes de capturar de forma consistente os padrões associados aos rótulos mais frequentes, o que é coerente com sua capacidade de modelar interações não lineares e fronteiras complexas por meio de árvores de decisão (CHEN; GUESTRIN, 2016).

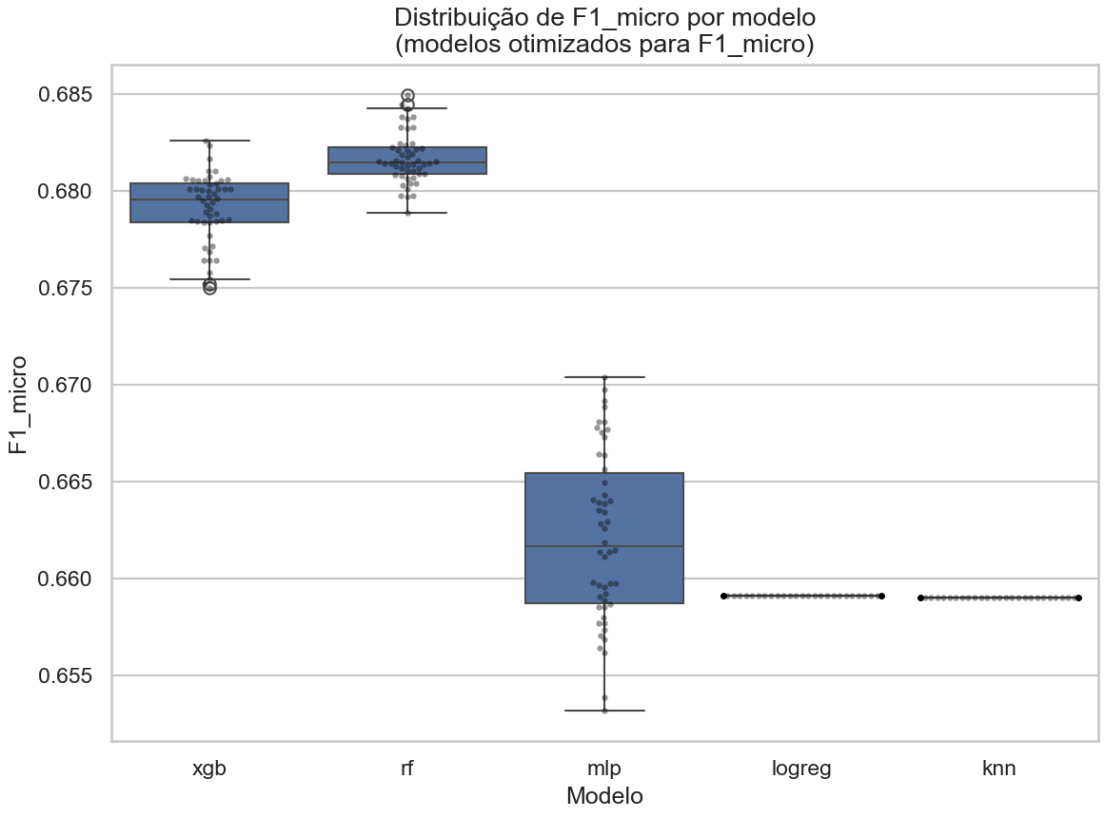
O MLP apresenta novamente a maior variabilidade entre os modelos, com valores oscilando aproximadamente entre 0,653 e 0,670. Essa amplitude reflete a sensibilidade de redes neurais a fatores como inicialização dos pesos e dinâmica do processo de otimização, características amplamente documentadas em ambientes de dados moderados em tamanho (GOODFELLOW; BENGIO; COURVILLE, 2016). Embora parte das execuções atinja valores próximos aos observados para *XGBoost* e *Random Forest*, a dispersão maior indica menor previsibilidade no desempenho.

A Regressão Logística e o k-NN exibem distribuições praticamente

degeneradas, com todos os valores concentrados ao redor de 0,659. Esse comportamento é decorrente da natureza determinística desses modelos no contexto escolhido: o k-NN não depende de inicialização aleatória e tende a produzir resultados idênticos sob as mesmas condições de treino e teste, enquanto a Regressão Logística frequentemente converge para soluções muito semelhantes em problemas de baixa complexidade não linear (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

De maneira geral, os padrões observados na métrica *F1-micro* revelam diferenças importantes entre as famílias de modelos: métodos baseados em árvores apresentam estabilidade e valores próximos entre si, redes neurais exibem maior sensibilidade às condições de treinamento e métodos lineares ou baseados em vizinhança mostram alta previsibilidade, embora com valores modestos. Essa análise complementa os resultados do *F1-macro* ao destacar o comportamento dos modelos sob uma perspectiva global e fortemente influenciada pelos rótulos frequentes.

Figura 6: Boxplot das 50 repetições dos modelos otimizados para F1-micro



Fonte: Autor (2025)

5.1.3 Comparação dos modelos otimizados para métrica *Hamming Loss*

A Figura 7 apresenta a distribuição dos valores de *Hamming Loss* obtidos nas 50 execuções para cada modelo, considerando que todos foram otimizados especificamente para essa métrica. Diferentemente das medidas derivadas de F1, a *Hamming Loss* deve ser minimizada pois quantifica o erro médio por rótulo, penalizando igualmente omissões e atribuições indevidas em cada posição da matriz multirrótulo (TSOUMAKAS; KATAKIS, 2007). Por ser uma métrica sensível a erros individuais e não apenas ao conjunto completo de rótulos, ela fornece uma perspectiva complementar sobre o comportamento dos algoritmos em relação às decisões binárias tomadas para cada classe. Como se trata de uma métrica de erro, valores menores indicam melhor desempenho (ZHANG; ZHOU, 2014).

Os valores observados para o *XGBoost* apresentam-se concentrados em torno de aproximadamente 0,191, com baixa dispersão entre as execuções. Essa redução consistente no erro médio por rótulo sugere que, no cenário em avaliação, o modelo conseguiu estabelecer fronteiras de decisão estáveis para a maior parte das classes. Ensembles de boosting são frequentemente associados a erro reduzido em situações com interações não lineares entre atributos, o que pode favorecer a diminuição da *Hamming Loss* (CHEN; GUESTRIN, 2016).

O *Random Forest* apresenta valores ligeiramente superiores, com mediana próxima de 0,204 e pequena variabilidade. Embora a métrica seja maior do que a observada para o *boosting*, o comportamento estável se mantém, coerente com propriedades conhecidas do método relacionadas à redução de variância pela agregação de múltiplas árvores treinadas de forma independente (BREIMAN, 2001).

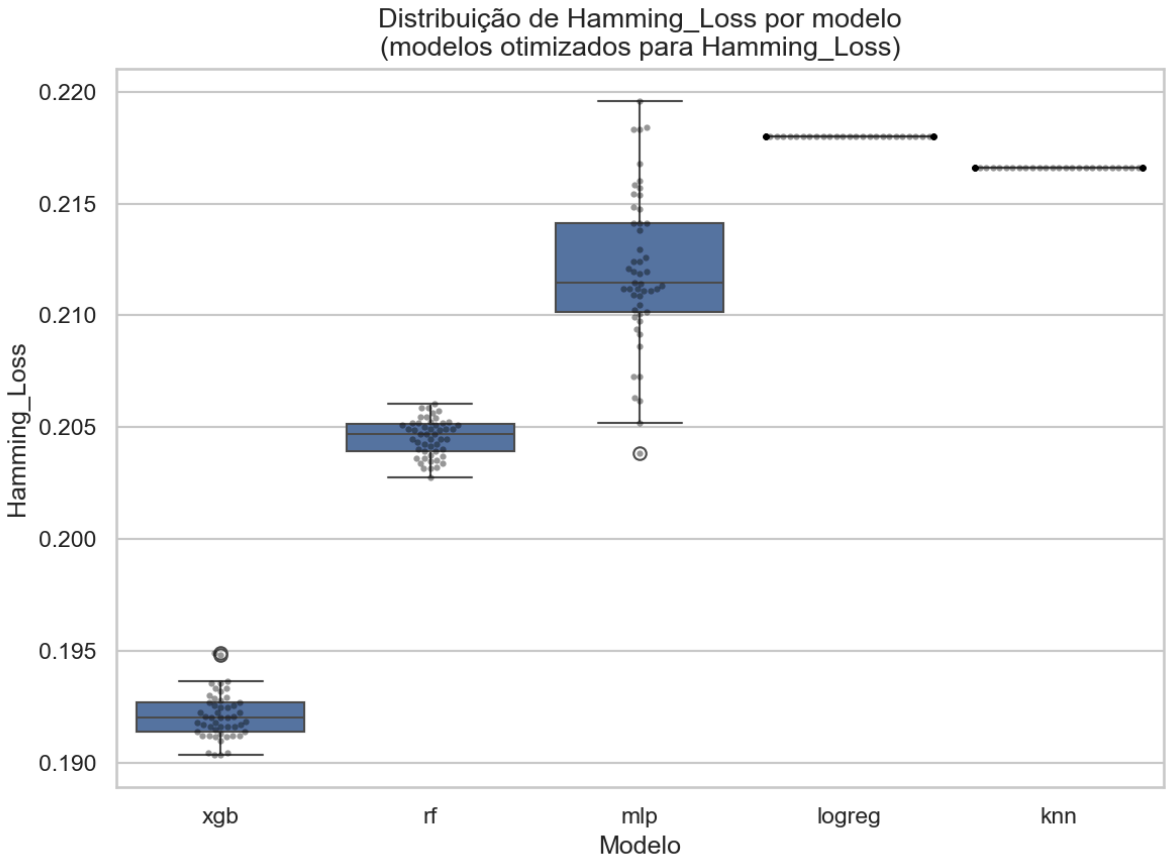
O MLP apresenta valores mais elevados de *Hamming Loss*, em torno de 0,211, com variabilidade moderada. A dispersão observada é compatível com a sensibilidade de redes neurais à inicialização e à trajetória de otimização, sobretudo em bases de tamanho limitado. Esse padrão indica que, nesse conjunto, o modelo pode ter enfrentado maior dificuldade para ajustar simultaneamente os limites de decisão de todos os rótulos, resultando em um erro médio mais alto.

A Regressão Logística e o k-NN exibem valores praticamente constantes entre as execuções, com *Hamming Loss* próxima de 0,217 e 0,216, respectivamente. Essa ausência de dispersão reflete novamente o caráter determinístico desses algoritmos no contexto empregado. Seus valores mais

elevados indicam maior frequência de discordâncias entre as predições binárias e os rótulos verdadeiros, o que pode decorrer da limitação de modelos lineares ou baseados em vizinhança em capturar fronteiras de decisão mais complexas presentes na base *Yeast*.

A análise da Hamming Loss complementa as métricas anteriores ao enfatizar erros individuais por rótulo, permitindo uma avaliação detalhada da consistência dos modelos. Os resultados mostram diferenças claras entre as famílias de algoritmos, com *ensembles* de árvores apresentando valores reduzidos de erro e métodos lineares ou baseados em distância exibindo maior taxa de discordâncias ponto a ponto. Esse panorama reforça a importância de considerar múltiplas métricas na interpretação dos resultados em cenários multirrótulo.

Figura 7 - Boxplot das 50 repetições dos modelos otimizados para *Hamming Loss*



Fonte: Autor (2025)

5.1.4 Comparação dos modelos otimizados para métrica *Jaccard*

A Figura 8 apresenta a distribuição dos valores da métrica *Jaccard* (média por amostra) nas 50 execuções realizadas para cada modelo, considerando que todos foram otimizados previamente para essa métrica. O índice de *Jaccard* avalia a similaridade entre os conjuntos de rótulos previstos e verdadeiros para cada instância, sendo definido pela razão entre a interseção e a união desses conjuntos (PEREIRA *et al*., 2018). Trata-se de uma métrica particularmente informativa em contextos multirrótulo, pois penaliza tanto omissões quanto inclusões indevidas, capturando de forma equilibrada a sobreposição entre os conjuntos.

Os resultados mostram que *XGBoost* e *Random Forest* apresentam distribuições bastante próximas, com medianas em torno de 0,558–0,560 e baixa variabilidade entre as execuções. Esse comportamento sugere que ambos os métodos conseguiram manter um nível consistente de concordância entre os conjuntos de rótulos previstos e verdadeiros, aspecto compatível com a capacidade de ensembles baseados em árvores de capturar dependências não lineares entre atributos (BREIMAN, 2001; CHEN; GUESTRIN, 2016). A proximidade entre as distribuições também indica que, para essa métrica específica, pequenas diferenças estruturais entre *boosting* e *bagging* não resultaram em grandes discrepâncias no padrão de similaridade entre conjuntos.

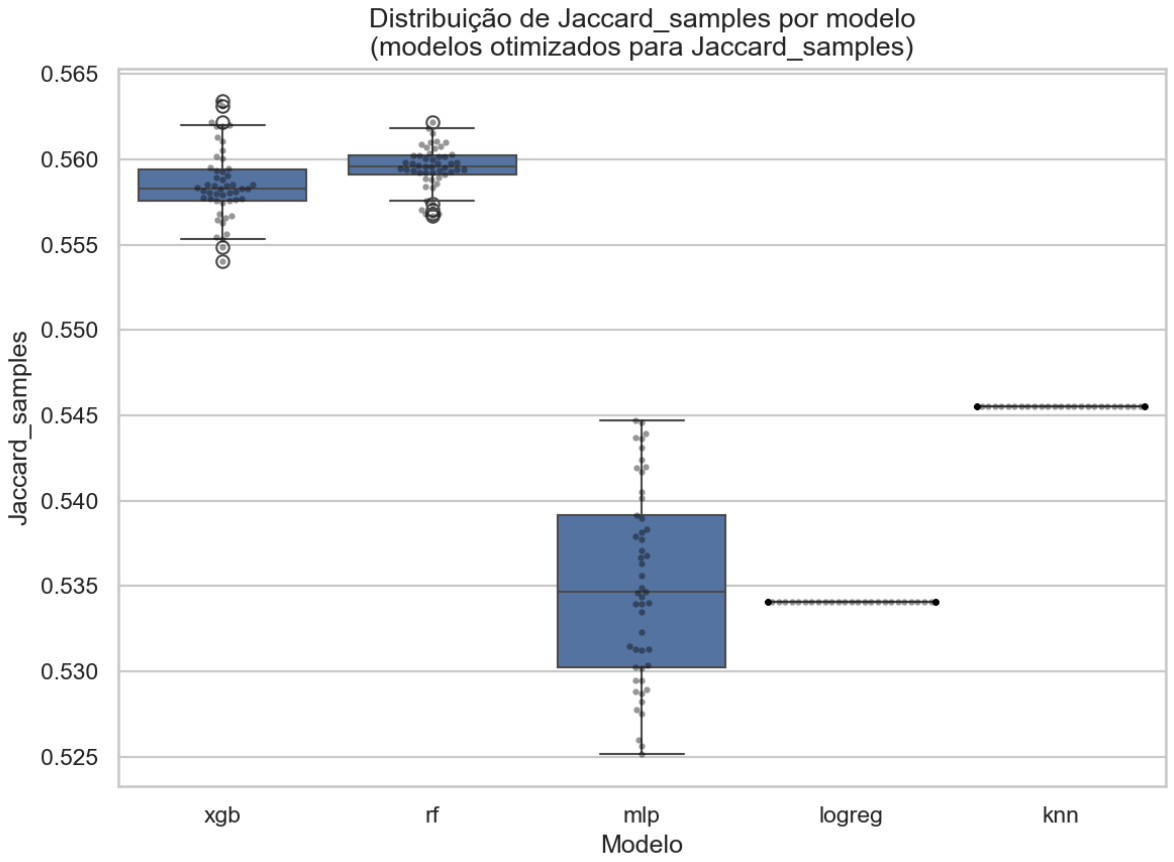
O MLP apresenta valores sistematicamente mais baixos, com mediana aproximada de 0,535 e variabilidade moderada. A dispersão observada reflete novamente a conhecida sensibilidade das redes neurais a fatores como inicialização dos pesos, processo iterativo de otimização e arquitetura, o que pode levar a diferentes níveis de aproximação entre os conjuntos previstos e reais (GOODFELLOW; BENGIO; COURVILLE, 2016). Em particular, o *Jaccard* tende a penalizar mais intensamente casos em que o modelo prevê conjuntos maiores ou menores do que o verdadeiro, o que pode explicar parte da oscilação do MLP nessa métrica.

A Regressão Logística e o k-NN exibem comportamentos praticamente determinísticos, com valores concentrados em torno de 0,534 e 0,545, respectivamente. Essa ausência de variabilidade decorre do caráter não estocástico desses classificadores no contexto empregado, que reproduzem previsões muito

semelhantes independentemente da *seed* inicial. Os valores mais modestos, em comparação com os observados para os *ensembles* de árvores, sugerem certa limitação desses métodos em representar adequadamente as relações estruturais entre os rótulos verdadeiros.

A análise da métrica *Jaccard* complementa as avaliações anteriores ao focalizar diretamente a similaridade entre conjuntos de rótulos completos, oferecendo uma visão mais global do comportamento preditivo. A métrica evidencia diferenças entre famílias de modelos quanto à capacidade de reproduzir simultaneamente múltiplas ativações, um aspecto central em tarefas multirrótulo. Os padrões observados reforçam a utilidade de considerar métricas complementares na comparação entre algoritmos aplicados à base *Yeast*.

Figura 8 - Boxplot das 50 repetições dos modelos otimizados para *Jaccard*



Fonte: Autor (2025)

5.2 ANÁLISE ESTATÍSTICA DOS MODELOS

Esta seção apresenta uma análise descritiva dos valores obtidos nas 50 execuções independentes realizadas para cada combinação de modelo e métrica otimizada. Na Tabela 1, 2, 3 e 4 são reportadas a média, a mediana, o desvio padrão, os valores mínimo e máximo e o coeficiente de variação (CV), permitindo avaliar tanto o nível de desempenho quanto a estabilidade dos modelos frente às variações induzidas pelas sementes aleatórias. O conjunto de medidas estatísticas auxilia na interpretação dos *boxplots* apresentados na seção anterior, oferecendo uma visão quantitativa complementar sobre a dispersão, centralidade e robustez dos resultados (FREUND; PERLES, 2000).

De modo geral, observa-se que os modelos *k-NN e Regressão Logística* apresentam variância praticamente nula em todas as métricas avaliadas, refletindo seu comportamento determinístico ou quase determinístico no contexto experimental empregado. Esses resultados são consistentes com a natureza desses algoritmos: o k-NN não depende de inicialização aleatória e tende a produzir predições idênticas dado o mesmo conjunto de treinamento, enquanto a Regressão Logística, por ser um método linear com superfície de otimização convexa, converge tipicamente para soluções muito semelhantes em diferentes inicializações (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Essa estabilidade, entretanto, não implica necessariamente desempenho superior, mas evidencia previsibilidade estatística elevada.

Para o *Random Forest*, os resultados mostram desvios padrão moderados (entre 0.0008 e 0.0018) e coeficientes de variação baixos, indicando bom nível de estabilidade entre as repetições. Esse comportamento é coerente com as propriedades do método, que utiliza amostragem bootstrap e seleção aleatória de atributos, mas mantém consistência global por meio da agregação de árvores independentes (BREIMAN, 2001).

O modelo *XGBoost* apresenta desvios padrões igualmente baixos (0.0010 a 0.0035), evidenciando estabilidade semelhante à do *Random Forest*, embora com valores médios superiores em várias métricas, especialmente *F1-macro* e *Hamming Loss.* Essa combinação de desempenho consistente e variabilidade reduzida alinha-se ao comportamento típico de métodos de boosting, que, ao atualizar iterativamente modelos fracos, tendem a reduzir a variância sem elevar excessivamente o viés (CHEN; GUESTRIN, 2016).

Em contraste, o MLP é o modelo que apresenta maior variabilidade entre as execuções, como evidenciado pelos desvios padrão mais elevados, especialmente em *F1-macro* (0.0080) e *Jaccard* (0.0056). Essa oscilação é coerente com a sensibilidade de redes neurais à inicialização dos pesos e ao processo iterativo de otimização, fatores que podem conduzir a mínimos locais distintos (GOODFELLOW; BENGIO; COURVILLE, 2016). Apesar disso, o MLP atinge, na média, valores competitivos em algumas métricas, embora com previsibilidade menor em comparação aos *ensembles* de árvores.

Tabela 1 - Estatística descritiva dos modelos para métrica ***F1-macro***

modelo	média	mediana	desvio padrão
knn	0.4745	0.4745	1.1215e-16
reg. logística	0.4710	0.4710	5.6075e-17
mlp	0.4629	0.4616	0.0080
random forest	0.4681	0.4681	0.0019
XGboost	0.4822	0.4827	0.0035

Fonte: Autor (2025).

Tabela 2 - Estatística descritiva dos modelos para métrica ***F1-micro***

modelo	média	mediana	desvio padrão
knn	0.6590	0.6590	0.0000
reg. logística	0.6591	0.6591	1.1214e-16
mlp	0.6621	0.6616	0.0044
random forest	0.6817	0.6815	0.0013
XGboost	0.6791	0.6796	0.0018

Fonte: Autor (2025).

Tabela 3 - Estatística descritiva dos modelos para métrica ***Hamming Loss***

modelo	média	mediana	desvio padrão
knn	0.2166	0.2166	5.6075e-17
reg. logística	0.2180	0.2180	0.0000
mlp	0.2120	0.2114	0.0034
random forest	0.2045	0.2047	0.0008
XGboost	0.1921	0.1920	0.0010

Fonte: Autor (2025).

Tabela 4 - Estatística descritiva dos modelos para métrica ***Jaccard***

modelo	média	média	desvio padrão
knn	0.5455	0.5455	0.0000
reg. logística	0.5341	0.5341	0.0000
mlp	0.5351	0.5346	0.0056
random forest	0.5595	0.5596	0.0012
XGboost	0.5585	0.5583	0.0021

Fonte: Autor (2025).

5.3 ANÁLISE DE NORMALIDADE E TESTES DE SIGNIFICÂNCIA

A análise estatística teve como objetivo complementar as avaliações gráficas realizadas previamente, verificando se as diferenças observadas entre os modelos podem ser atribuídas ao acaso ou se refletem padrões sistemáticos de desempenho. Para isso, as 50 repetições associadas a cada combinação de modelo e métrica foram analisadas segundo critérios de normalidade e, posteriormente, comparadas por meio de testes estatísticos adequados.

A primeira etapa consistiu em avaliar se as distribuições das repetições apresentavam comportamento compatível com a normalidade. Esse procedimento é fundamental, pois a escolha de testes comparativos depende das características estatísticas dos dados. Para essa finalidade, aplicou-se o teste de *Shapiro–Wilk*, recomendado para amostras pequenas e médias e amplamente utilizado em experimentos de aprendizado de máquina para avaliar a adequação de testes paramétricos (SHAPIRO; WILK, 1965). Os resultados mostraram que 19 dos 20 modelos apresentaram valores de *p* superiores a 0,05, o que indica ausência de evidências contra a normalidade. Alguns casos específicos, especialmente no k-NN e na Regressão Logística, apresentaram variância zero, produzindo distribuições degeneradas. Esse comportamento é esperado, dada a natureza determinística dessas técnicas em determinadas circunstâncias, e não compromete a análise, apenas limita a aplicabilidade de testes estatísticos comparativos nessas situações.

Considerando esse conjunto de evidências, adotou-se o *teste t de Welch* como procedimento padrão para comparação entre modelos. A escolha fundamenta-se em duas razões principais: a normalidade foi atendida na maior parte das distribuições e o *teste de Welch* não pressupõe homogeneidade de variâncias, sendo adequado quando diferentes modelos apresentam variabilidades distintas (WELCH, 1947). Nos casos em que a variância foi nula, nenhum teste paramétrico ou não paramétrico é aplicável, uma vez que não há dispersão suficiente para definir diferença estatística. Ainda assim, esses casos foram devidamente identificados e interpretados no contexto da análise geral.

O *teste t de Welch* foi aplicado de forma pareada, comparando sempre duas distribuições referentes à mesma métrica. O teste avalia a hipótese nula de que as médias dos dois modelos são iguais, contra a hipótese alternativa de que elas diferem. A interpretação baseia-se no *p-valor*: valores inferiores a 0,05 indicam

rejeição da hipótese de igualdade das médias. O sinal do estatístico *t* indica qual modelo apresentou maior média, permitindo identificar direções de diferença quando existentes.

Os resultados obtidos para a métrica *F1_macro* revelaram diferenças estatisticamente significativas em todas as comparações possíveis. O *XGBoost* apresentou médias elevadas às dos demais modelos, enquanto o *Random Forest* tem média mais elevada que MLP, Regressão Logística e k-NN. O MLP, por sua vez, possui média elevada a Regressão Logística e k-NN, que compuseram o grupo de desempenho menos competitivo. Esses resultados são coerentes com a natureza da métrica, que atribui peso igual a todos os rótulos, favorecendo modelos com maior capacidade de aprendizado em classes menos frequentes.

Para a métrica *F1_micro*, alguns modelos apresentaram variância zero, impossibilitando determinados testes. Nas comparações possíveis, o *Random Forest* apresentou médias superiores às do MLP, Regressão Logística e *XGBoost*. O *XGBoost* possui média maiores que MLP e Regressão Logística, enquanto o MLP possui média elevada a apenas Regressão Logística. Esses resultados refletem o fato de que o *F1_micro* é mais influenciado por classes frequentes, favorecendo modelos com maior estabilidade nessas classes.

Na métrica *Hamming Loss*, em que valores menores indicam melhor desempenho, observou-se que o *XGBoost* apresentou os menores erros entre os modelos comparados, seguido pelo *Random Forest*. O MLP obteve valores intermediários, enquanto Regressão Logística e k-NN apresentaram os maiores valores, em alguns casos sem variabilidade entre repetições. Esse comportamento é consistente com a capacidade dos métodos baseados em árvores de capturar padrões complexos e reduzir erros individuais por rótulo.

Por fim, na métrica *Jaccard* (média por amostra), algumas distribuições com variância zero também limitaram a aplicação de testes. Nas comparações possíveis, o *Random Forest* apresentou valores elevados aos do *XGBoost* e do MLP, e o *XGBoost* apresentou valores mais competitivos que o MLP. Essa métrica avalia diretamente a similaridade entre os conjuntos de rótulos previstos e verdadeiros, favorecendo modelos cuja estrutura de decisões permite maior alinhamento entre conjuntos.

Considerando o conjunto dos resultados, é possível sintetizar que, no contexto estudado, o *XGBoost* se destaca em métricas que valorizam equilíbrio

entre rótulos ou redução de erros individuais, como F1_macro e *Hamming Loss*. O *Random Forest*, por outro lado, apresenta desempenho competitivo em métricas sensíveis à frequência das classes, como *F1_micro* e *Jaccard*. O MLP ocupa uma posição intermediária, com maior variabilidade entre execuções. Já a Regressão Logística e o k-NN apresentaram desempenho menos competitivo e variabilidade mínima, refletindo limitações desses métodos no cenário multirrótulo considerado.

Assim, a análise estatística confirma que as diferenças observadas nos *boxplots* correspondem a padrões sistemáticos de desempenho, não sendo resultado apenas de variação aleatória. Os achados contribuem para consolidar a compreensão comparativa dos modelos no contexto da base *Yeast*, respeitando as particularidades impostas por sua estrutura multirrótulo e desbalanceamento.

6 CONCLUSÃO

Os resultados obtidos ao longo da avaliação permitem concluir que a escolha do algoritmo influencia de maneira significativa o desempenho final, e que diferentes modelos tendem a apresentar vantagens distintas a depender da métrica considerada. Observou-se que métodos capazes de capturar relações não lineares e de ajustarem-se a padrões complexos foram os que apresentaram melhor desempenho global no conjunto de experimentos, ainda que por razões distintas. O *XGBoost* demonstrou maior adequação em métricas que enfatizam o tratamento equilibrado dos rótulos e a redução de erros por rótulo, ao passo que o *Random Forest* apresentou desempenho mais favorável em medidas influenciadas pela frequência relativa das classes. Esses achados reforçam conclusões presentes na literatura, segundo as quais diferentes famílias de algoritmos respondem de forma distinta às características estruturais dos dados multirrótulo.

Modelos de menor complexidade, como Regressão Logística e k-NN, apresentaram desempenho mais limitado no conjunto de experimentos conduzidos, sugerindo que técnicas que não exploram relações não lineares ou decisões hierárquicas podem ter dificuldade em capturar a estrutura multirrótulo da base Yeast. Já o *Multilayer Perceptron* apresentou desempenho intermediário, porém com maior variabilidade, o que evidencia a sensibilidade de redes neurais a inicializações e ao processo de otimização, especialmente em conjuntos de dados de tamanho

moderado.

No conjunto, os resultados obtidos contribuem para a compreensão das potenciais vantagens e limitações de métodos clássicos quando aplicados à classificação multirrótulo. As conclusões apresentadas reforçam a importância de selecionar modelos e métricas alinhados às características específicas do problema, uma vez que diferentes visões de desempenho podem levar a interpretações distintas sobre a adequação dos algoritmos.

7 TRABALHOS FUTUROS

Os resultados obtidos neste estudo fornecem uma base para diferentes direções de investigação, especialmente considerando que este trabalho integra a etapa inicial de um projeto mais amplo voltado ao desenvolvimento de métodos para classificação multirrótulo em fluxos contínuos de dados. A seguir, apresentam-se possibilidades de continuidade que se beneficiam diretamente dos achados e limitações identificados.

Uma primeira linha de aprofundamento consiste em explorar métodos de classificação multirrótulo que incorporem explicitamente dependências entre rótulos, tais como *Classifier Chains* (READ *et al*., 2011) ou variantes probabilísticas que modelam correlações estruturais. Como observado na análise descritiva da base *Yeast*, existe correlação significativa entre rótulos, o que sugere que abordagens além do *Binary Relevance* podem oferecer ganhos de desempenho, particularmente em métricas sensíveis à similaridade entre conjuntos de rótulos (ZHANG; ZHOU, 2014).

Outra continuidade natural envolve avaliar métodos especialmente projetados para classificação multirrótulo. Esses modelos podem servir de comparação adicional, ampliando o escopo empírico estabelecido neste estudo e permitindo uma análise mais abrangente das propriedades de cada paradigma.

Considerando a motivação central do macro projeto, uma direção de particular relevância consiste em migrar o estudo para o cenário de *data streams*, incorporando a evolução temporal das distribuições e o fenômeno de *concept drift* (GAMA *et al*., 2004). Métodos supervisionados tradicionais, como os avaliados neste trabalho, não são projetados para adaptação contínua, o que reforça a necessidade

de investigar algoritmos capazes de operar incrementalmente, como *Hoeffding Trees* adaptativas, modelos baseados em janelas deslizantes, ou sistemas com detecção de mudança integrada.

Em paralelo, destaca-se a importância de investigar cenários com ausência ou atraso de rótulos, que são comuns em aplicações reais de fluxos contínuos e têm sido discutidos em pesquisas de rotulagem tardia e detecção de novidades (FARIA *et al*., 2016; CERRI *et al*., 2022). Assim, futuras pesquisas podem incluir o desenvolvimento de mecanismos que façam uso de informações não supervisionadas, como densidade, reconstrução, incerteza ou divergência entre modelos, para orientar a adaptação de classificadores em ambientes onde a rotulagem imediata não é viável.

Por fim, seria relevante expandir o conjunto de bases avaliadas, incorporando *datasets* multirrótulo com características distintas, incluindo maior número de rótulos, dependência estrutural mais pronunciada ou cardinalidade mais elevada. Essa ampliação permitiria analisar a generalização dos achados e verificar até que ponto os padrões identificados na base *Yeast* se replicam em diferentes contextos.

REFERÊNCIAS

ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. *Statistics Surveys*, v. 4, p. 40–79, 2010.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, p. 5–32, 2001.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco: ACM, 2016.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27, 1967.

EFRON, B. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, v. 7, n. 1, p. 1–26, 1979.

ELISSEEFF, A.; WESTON, J. A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, v. 14, p. 681–687, 2001.

GIBAJA, E.; VENTURA, S. A tutorial on multi-label learning. *ACM Computing Surveys*, v. 47, n. 3, p. 1–38, 2015.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2. ed. New York: Springer, 2009.

HERRERA, F. Dealing with imbalanced multi-label classification: Measures and random resampling algorithms. *Neurocomputing*, v. 273, p. 489–499, 2018.

HO, T. K. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 832–844, 1998.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine*

Learning Research, v. 12, p. 2825–2830, 2011.

PEREIRA, R. B. *et al.* Correlation analysis of performance measures for multi-label classification. *Information Processing & Management*, v. 54, n. 3, p. 359–377, 2018.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 1986.

SECHIDIS, K.; TSOUTSOURAS, K.; BROWN, G. On the stratification of multi-label data. In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*. 2016.

SZYMANSKI, P.; KAJDANOWICZ, T. A repository of multi-label classification datasets. *arXiv preprint*, arXiv:1705.03045, 2017.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: an overview. In: MAIMON, O.; ROKACH, L. (org.). *Data Mining and Knowledge Discovery Handbook*. Boston: Springer, 2007. p. 667–685.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Mining multi-label data. In: MAIMON, O.; ROKACH, L. (org.). *Data Mining and Knowledge Discovery Handbook*. Boston: Springer, 2010. p. 667–685.

ZHANG, M.; ZHOU, Z. A brief introduction to multi-label learning. In: FU, Y. (org.). *Deep Learning and Artificial Intelligence*. Cham: Springer, 2018. p. 1–10.

ZHANG, M.; ZHOU, Z. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, v. 26, n. 8, p. 1819–1837, 2014.