Universidade Federal de São João del-Rei

Ciência da Computação

Análise e Classificação de Doenças da Tireoide com XGBoost

22 de setembro de 2025

Relatório de Mineração de Dados

Kariny Abrahão (212050013)

Professor:
Leonardo Rocha

Sumário

1 Introdução **2**

2 Atividades Práticas **3**
2.1 Atividade 1 - Pesquisa bibliográfica . . . 3
2.2 Atividade 2 - Seleção e Análise Exploratória Inicial dos Dados . . . 4
2.3 Atividade 3 - Pré-processamento, Balanceamento dos Dados e Análise Exploratória . . . 5
2.3.1 Remoção de colunas redundantes e irrelevantes . . . 5
2.3.2 Limpeza e padronização da variável target . . . 6
2.3.3 Tratamento de valores ausentes . . . 6
2.3.4 Análise exploratória e remoção de outliers . . . 6
2.3.5 Imputação de valores nulos utilizando KNN . . . 7
2.3.6 Balanceamento das classes com SMOTE . . . 9
2.3.7 Implementação do XGBoost . . . 9

3 Resultados **10**
3.1 Importância das Features . . . 10
3.2 Evolução do Treinamento . . . 10
3.2.1 Métricas de Avaliação Final . . . 11

4 Conclusão **11**

1 Introdução

A glândula tireoide desempenha um papel fundamental na regulação do metabolismo humano por meio da produção dos hormônios triiodotironina (T3) e tiroxina (T4). Alterações no funcionamento dessa glândula podem gerar duas condições clínicas principais: **hipotireoidismo** e **hipertireoidismo**.

O **hipotireoidismo** ocorre quando há uma produção insuficiente de hormônios tireoidianos, o que pode levar a sintomas como fadiga, ganho de peso, intolerância ao frio e depressão. Já o **hipertireoidismo** é caracterizado pelo excesso de hormônios tireoidianos, causando perda de peso não intencional, ansiedade, taquicardia e intolerância ao calor. Ambas as condições, quando não diagnosticadas ou tratadas adequadamente, podem comprometer significativamente a qualidade de vida dos pacientes e até gerar complicações mais graves.

Na prática clínica, o diagnóstico dessas disfunções é realizado principalmente pela análise de exames laboratoriais. O **TSH (hormônio estimulador da tireoide)** é considerado o marcador mais sensível: níveis elevados de TSH sugerem **hipotireoidismo**, enquanto níveis suprimidos (muito baixos) indicam **hipertireoidismo**. Além disso, a dosagem dos hormônios tireoidianos T3 e T4 auxilia na confirmação do quadro: no hipotireoidismo observa-se T4 reduzido, enquanto no hipertireoidismo os níveis de T3 e T4 costumam estar aumentados. Outros índices derivados, como o **FTI (Free Thyroxine Index)** e o **T4U (T4 uptake)**, também são utilizados para refinar a interpretação.

Apesar de bem estabelecidos na prática médica, esses exames nem sempre apresentam cortes claros entre as diferentes condições, havendo sobreposição de valores entre indivíduos saudáveis e doentes. Esse cenário torna o diagnóstico desafiador em alguns casos, principalmente quando existem valores limítrofes ou resultados laboratoriais inconsistentes.

Nesse contexto, técnicas de **mineração de dados e aprendizado de máquina** vêm sendo exploradas como ferramentas de apoio ao diagnóstico de doenças da tireoide. Essas técnicas permitem lidar com grandes conjuntos de dados, aplicar estratégias de pré-processamento para tratar inconsistências e treinar modelos de classificação capazes de distinguir entre indivíduos saudáveis, com hipotireoidismo ou com hipertireoidismo.

O presente trabalho tem como objetivo realizar um estudo de caso utilizando a base de dados *Thyroid Disease Data*, disponível no Kaggle [1] para responder:

Como técnicas de aprendizado de máquina, em especial o XGBoost, podem auxiliar no diagnóstico automatizado de doenças da tireoide (hipotireoidismo e hipertireoidismo) a partir de exames laboratoriais?

---

2 Atividades Práticas

Durante o desenvolvimento deste trabalho prático, foram realizadas as seguintes atividades:

- Pesquisa bibliográfica sobre doenças da tireoide (hipotireoidismo e hipertireoidismo) e revisão de artigos que utilizam aprendizado de máquina aplicado ao diagnóstico médico.
- Seleção e análise inicial dos dados.
- Pré-processamento dos dados, análise exploratória dos dados, tratamento de valores, padronização e balanceamento das classes utilizando oversampling *(SMOTE)*.
- Implementação do algoritmo XGBoost.
- Avaliação do modelo utilizando métricas como acurácia, precisão, recall e f1-score.
- Geração de gráficos e visualizações para análise das importâncias das variáveis no modelo.

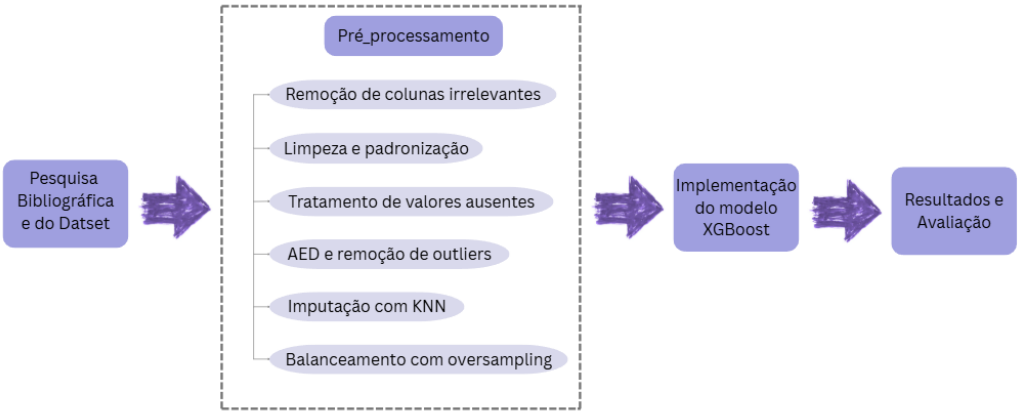


Figura 1: Fluxograma resumido das atividades.

2.1 Atividade 1 - Pesquisa bibliográfica

Para fundamentar o pré-processamento e as técnicas de classificação aplicadas neste trabalho, foram utilizadas como referência diversas fontes, entre notebooks e artigos científicos. Em particular:

- Notebooks do Kaggle, como "XGBoost Multi-Class Classification" de EmmanuelFwerr, "Thyroid Disease Detection" de iamArslanKhalid, e "Thyroid Classification" de ZiadAbdelaziz, que fornecem exemplos práticos de análise exploratória de dados (EDA), tratamento de valores ausentes, encoding e avaliação com métricas variadas.
- O artigo "*Enhanced Diagnosis of Thyroid Diseases Through Advanced Machine Learning Methodologies*" (Oture, Iqbal & Wang, 2025), que compara várias técnicas de ML e DL aplicadas ao mesmo tipo de dados, ressaltando uso de oversampling/undersampling e identificando TSH como biomarcador importante. [2]

Essas referências ajudaram a definir escolhas no meu pipeline, como quais variáveis manter, quais métricas observar, como tratar valores ausentes e como balancear classes para evitar viés no modelo.

2.2 Atividade 2 - Seleção e Análise Exploratória Inicial dos Dados

A segunda atividade teve como objetivo compreender melhor o conjunto de dados *Thyroid Disease Data*, disponível no Kaggle [1], que contém informações clínicas e laboratoriais de pacientes, incluindo variáveis contínuas e categóricas. O dataset possui 9172 registros, quantidade considerada suficiente para a condução de nosso estudo, permitindo a exploração de padrões relevantes e o treinamento de modelos de aprendizado de máquina com validade estatística.

O foco dessa etapa foi detalhar cada uma das features, entendendo o significado de seus valores e como elas poderiam impactar os diagnósticos médicos.

A Tabela 1 apresenta a descrição das colunas do dataset e os tipos de valores que elas contêm. A coluna **target** contém os diagnósticos médicos dos pacientes, que estão detalhados na Tabela 2.

Neste estágio, com a ajuda da análise detalhada das features, avaliamos o conjunto de dados e concluímos que ele é suficiente e adequado para o objetivo deste trabalho, que consiste no diagnóstico automatizado de hipotireoidismo e hipertireoidismo.

Tabela 1: Descrição das variáveis do conjunto de dados *Thyroid Disease Data*.

Variável	**Descrição**
age	Idade do paciente (inteiro)
sex	Sexo com o qual o paciente se identifica (string)
on_thyroxine	Se o paciente está em uso de tiroxina (booleano)
query_on_thyroxine	Se o paciente acredita estar em uso de tiroxina (booleano)
on_antithyroid_meds	Se o paciente está em uso de medicamentos antitireoidianos (booleano)
sick	Se o paciente está doente (booleano)
pregnant	Se o paciente está grávida (booleano)
thyroid_surgery	Se o paciente já passou por cirurgia na tireoide (booleano)
I131_treatment	Se o paciente está em tratamento com I131 (booleano)
query_hypothyroid	Se o paciente acredita ter hipotireoidismo (booleano)
query_hyperthyroid	Se o paciente acredita ter hipertireoidismo (booleano)
lithium	Se o paciente faz uso de lítio (booleano)
goitre	Se o paciente apresenta bócio (booleano)
tumor	Se o paciente possui tumor (booleano)
hypopituitary	Se o paciente apresenta hipopituitarismo (booleano)
psych	Se o paciente apresenta condição psiquiátrica (booleano)
TSH_measured	Se o TSH foi medido no sangue (booleano)
TSH	Nível de TSH no sangue a partir de exame laboratorial (float)
T3_measured	Se o T3 foi medido no sangue (booleano)
T3	Nível de T3 no sangue a partir de exame laboratorial (float)
TT4_measured	Se o TT4 foi medido no sangue (booleano)
TT4	Nível de TT4 no sangue a partir de exame laboratorial (float)
T4U_measured	Se o T4U foi medido no sangue (booleano)
T4U	Nível de T4U no sangue a partir de exame laboratorial (float)
FTI_measured	Se o FTI foi medido no sangue (booleano)
FTI	Nível de FTI no sangue a partir de exame laboratorial (float)
TBG_measured	Se o TBG foi medido no sangue (booleano)
TBG	Nível de TBG no sangue a partir de exame laboratorial (float)
referral_source	Fonte de encaminhamento do paciente (string)
target	Diagnóstico médico (string)
patient_id	Identificador único do paciente (string)

Tabela 2: Descrição dos códigos de diagnóstico presentes na coluna *target*.

Letra	Diagnóstico
Condições de hipertireoidismo	
A	Hipertireoidismo
B	T3 tóxico
C	Bócio tóxico
D	Tóxico secundário
Condições de hipotireoidismo	
E	Hipotireoidismo
F	Hipotireoidismo primário
G	Hipotireoidismo compensado
H	Hipotireoidismo secundário
Proteína de ligação	
I	Proteína de ligação aumentada
J	Proteína de ligação diminuída
Saúde geral	
K	Doença não tireoidiana concomitante
Terapia de reposição	
L	Consistente com terapia de reposição
M	Sub-reposição
N	Super-reposição
Tratamento antitireoidiano	
O	Medicamentos antitireoidianos
P	Tratamento com I131
Q	Cirurgia
Diversos	
R	Resultados de exames discordantes
S	TBG elevado
T	Hormônios tireoidianos elevados

2.3 Atividade 3 - Pré-processamento, Balanceamento dos Dados e Análise Exploratória

Após a análise inicial, iniciou-se o pré-processamento dos dados, etapa fundamental para garantir que o modelo de aprendizado de máquina pudesse ser treinado de forma eficaz. As ações realizadas incluíram:

2.3.1 Remoção de colunas redundantes e irrelevantes

O primeiro passo no pré-processamento foi analisar a relevância de cada coluna do dataset. Colunas do tipo `*_measured` foram removidas, pois indicavam apenas se um exame havia sido realizado e não forneciam informação adicional relevante para o modelo. A coluna `patient_id`, por se tratar de um identificador único, também foi descartada por não possuir utilidade preditiva. Da mesma forma, a coluna `referral_source` foi removida, por não agregar valor significativo na previsão do diagnóstico. Por fim, a coluna `TBG` foi excluída, uma vez que apresentava uma quantidade extremamente elevada de valores nulos (8823 de 9172, aproximadamente 96%) e não é considerada fundamental para os diagnósticos de hipotireoidismo ou hipertireoidismo.

Essas remoções simplificaram o dataset, reduziram a complexidade do modelo e eliminaram colunas que poderiam introduzir ruído ou viés desnecessário.

2.3.2 Limpeza e padronização da variável target

Posteriormente, iniciamos a exploração e limpeza da variável **target**, que representa o alvo do modelo. Primeiramente, foram removidos espaços em branco e todas as letras foram padronizadas para maiúsculas, evitando erros ou perda de informação.

Em seguida, filtramos apenas os diagnósticos relevantes para o estudo: as letras que representam o hipertireoidismo (A, B, C, D), o hipotireoidismo (E, F, G, H), e os casos negativos, representados pelo caractere `"-"`. Todas as demais linhas contendo outros diagnósticos foram removidas, a fim de evitar ruídos e possíveis vieses no modelo.

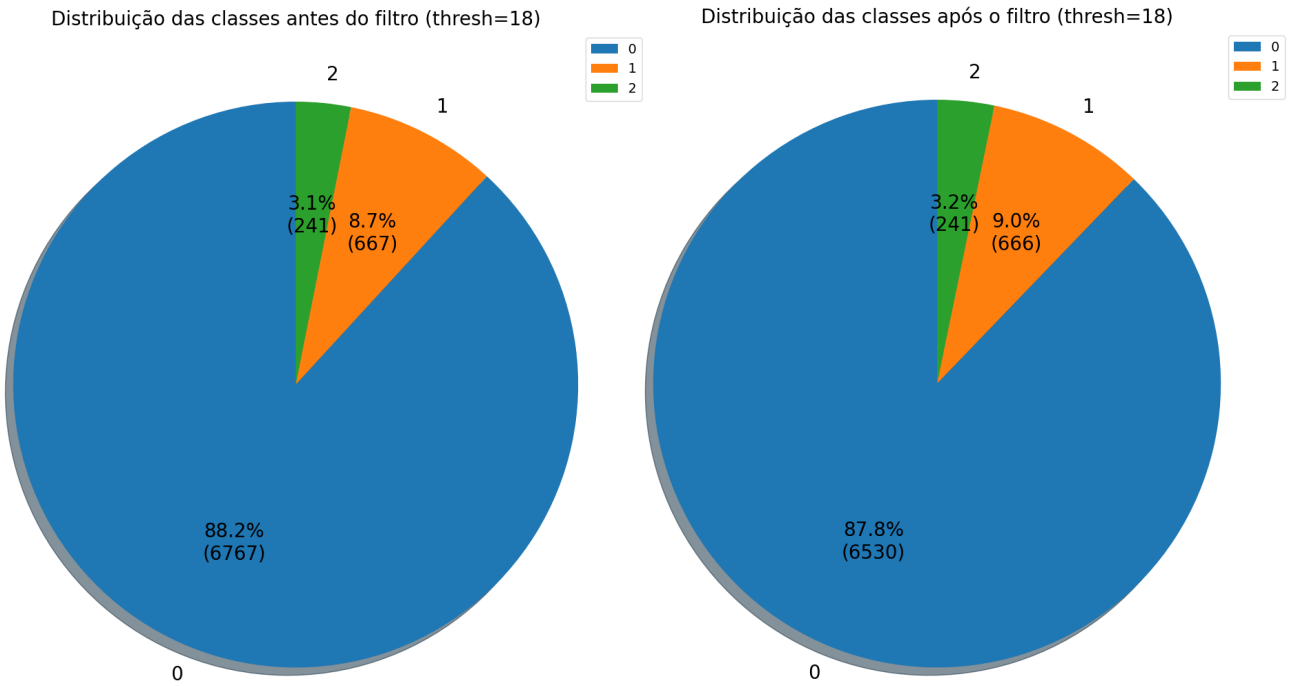
Por fim, a coluna **target** foi mapeada numericamente: 0 para os casos negativos, 1 para o hipotireoidismo e 2 para o hipertireoidismo, tornando-a adequada para o treinamento do modelo de classificação.

2.3.3 Tratamento de valores ausentes

Inicialmente, foram preenchidos alguns valores ausentes na coluna `sex`. Para os casos em que o sexo estava nulo, verificou-se a coluna `pregnant`: quando o valor era verdadeiro, atribuiu-se diretamente o sexo feminino, já que a condição de gravidez não seria possível para indivíduos do sexo masculino.

Além disso, utilizando o parâmetro `thresh=18` na função `dropna`, de forma a remover linhas que continham mais de quatro valores ausentes (dado que havia um total de 22 atributos, incluindo a variável alvo). Esse procedimento eliminou instâncias incompletas que poderiam comprometer a qualidade do modelo, ao mesmo tempo este procedimento resultou indiretamente em uma redução da classe majoritária (0), funcionando de forma semelhante a uma técnica de *undersampling* não intencional, o que contribuiu para melhorar o balanceamento do conjunto de dados.

Na Figura 2, observa-se a distribuição das classes antes e após a aplicação do filtro de valores nulos. Nota-se a queda da classe negativa (0) e a elevação proporcional das classes de hipotireoidismo (1) e hipertireoidismo (2), contribuindo para um balanceamento mais adequado do conjunto de dados.



(a) Distribuição antes do filtro (thresh=18)

(b) Distribuição após o filtro (thresh=18)

Figura 2: Distribuição das classes antes e depois da aplicação do filtro de valores nulos.

2.3.4 Análise exploratória e remoção de outliers

Para entender melhor a distribuição das variáveis numéricas em relação ao *target*, foram gerados gráficos de dispersão (*strip plots*) para cada atributo numérico, com cores diferenciando as classes do *target* (0 = outros, 1 = hipotireoidismo, 2 = hipertireoidismo).

Os gráficos permitiram identificar valores extremos em algumas colunas, o que possibilitou a definição de limites plausíveis para cada variável. Dessa forma, foram impostas restrições nos dados numéricos para remover outliers e evitar que valores atípicos viessem a enviesar o modelo. Os limites aplicados foram:

- **TSH**: 0 – 500
- **T3**: 0 – 20
- **TT4**: 0 – 400
- **T4U**: 0 – 2
- **FTI**: 0 – 300
- **Idade**: 0 – 100 anos

A Figura 3 apresenta os gráficos gerados, onde cada ponto representa um paciente e as cores indicam a classe do *target*. Observa-se que a aplicação desses limites contribuiu para remover valores extremos que poderiam impactar negativamente o desempenho do modelo de classificação.

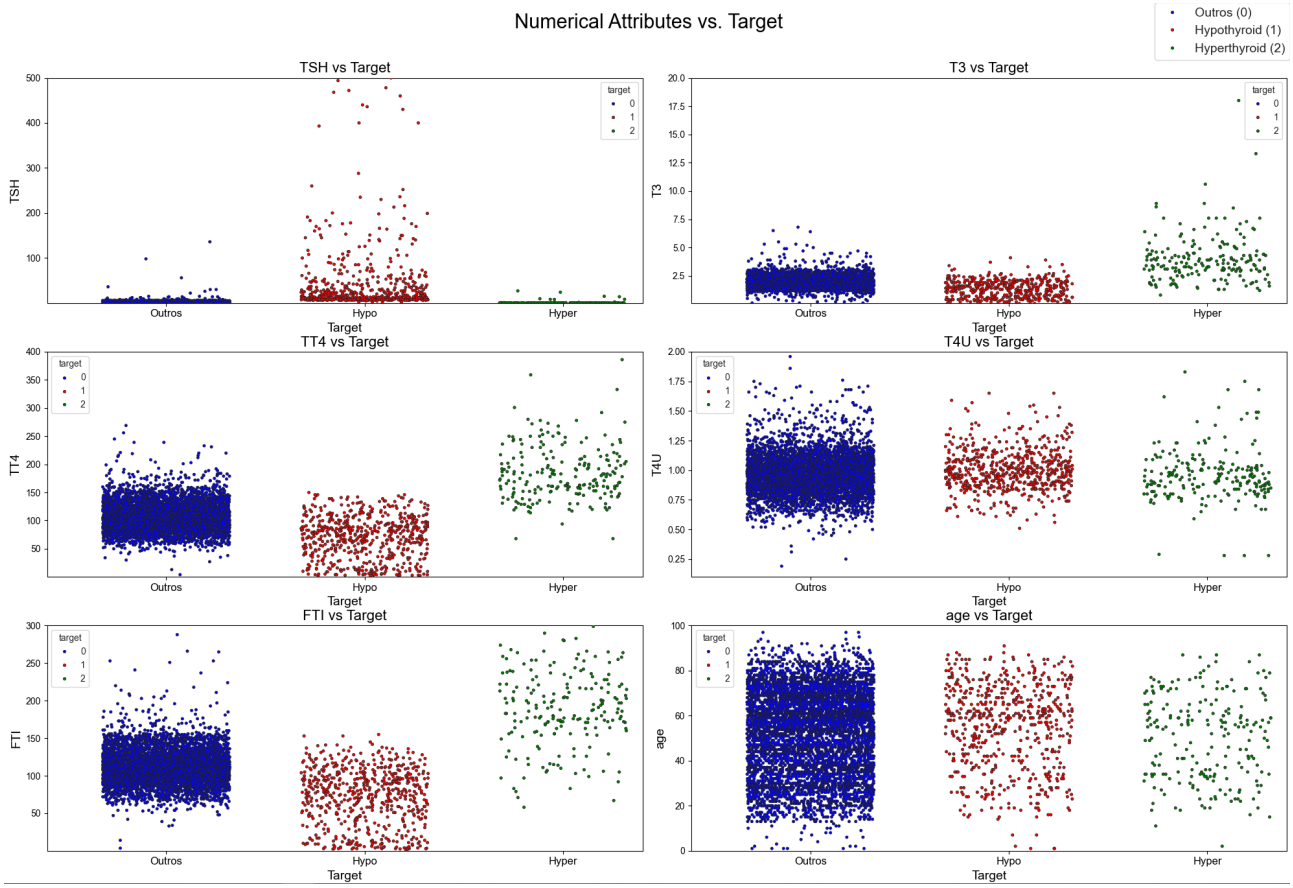


Figura 3: Distribuição das variáveis numéricas em função do *target*, com limites aplicados para remoção de outliers.

2.3.5 Imputação de valores nulos utilizando KNN

Após a remoção de outliers e a análise das variáveis, foi realizada a imputação dos valores nulos restantes utilizando o método de KNN (*K-Nearest Neighbors*). Para isso, considerou-se os 10 vizinhos mais próximos de cada registro, atribuindo pesos maiores aos vizinhos mais próximos. Esse procedimento permitiu preencher tanto os valores nulos das variáveis numéricas quanto das categóricas, preservando a estrutura dos dados.

Antes da aplicação do KNN, foi necessário transformar as variáveis categóricas, que possuíam valores binários como “t”/“f” ou “0”/“1”, para um formato adequado, garantindo que o algoritmo pudesse processá-las corretamente.

Para analisar o impacto da imputação, foram geradas curvas de distribuição das principais variáveis contínuas (**TSH**, **T3**, **TT4**, **T4U**, **FTI**) antes e depois do KNN. As figuras 4 a 8 apresentam essas distribuições. Observa-se que, embora a curva do dado original tenha ficado ligeiramente abaixo da curva após a imputação, o formato geral das distribuições se manteve, indicando que a imputação preservou a característica dos dados sem introduzir distorções significativas.

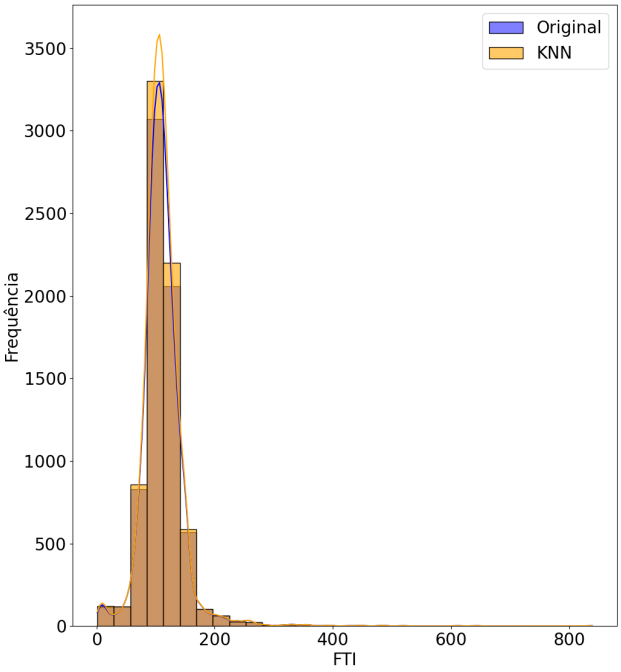


Figura 4: Distribuição da variável FTI.

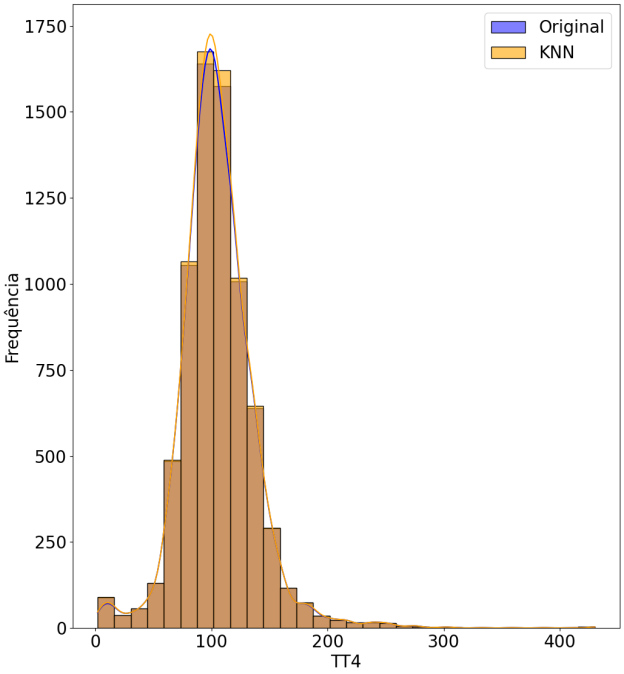


Figura 5: Distribuição da variável TT4.

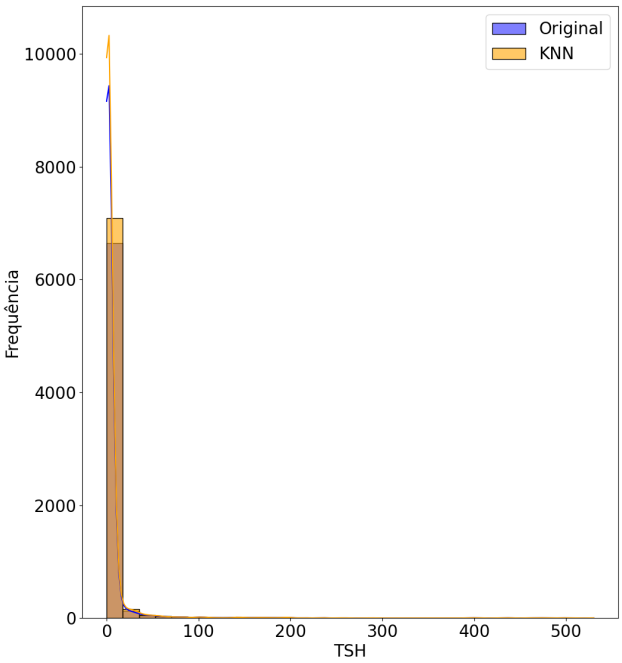


Figura 6: Distribuição da variável TSH.

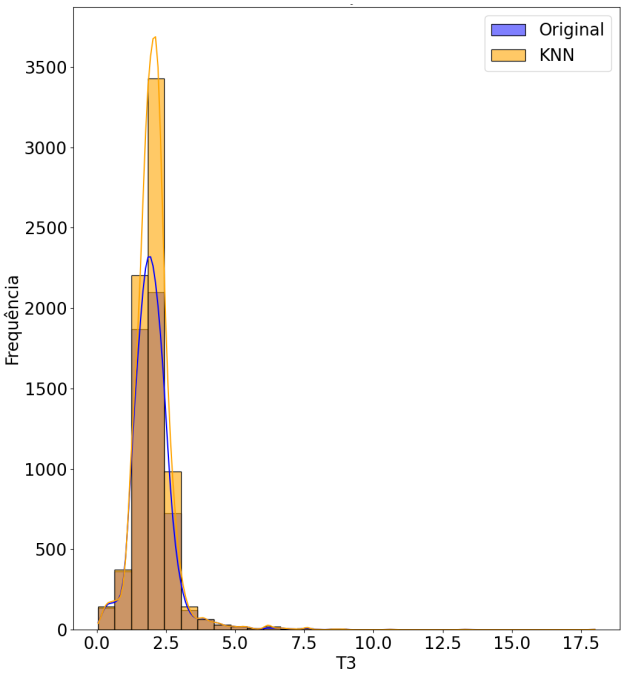


Figura 7: Distribuição da variável T3.

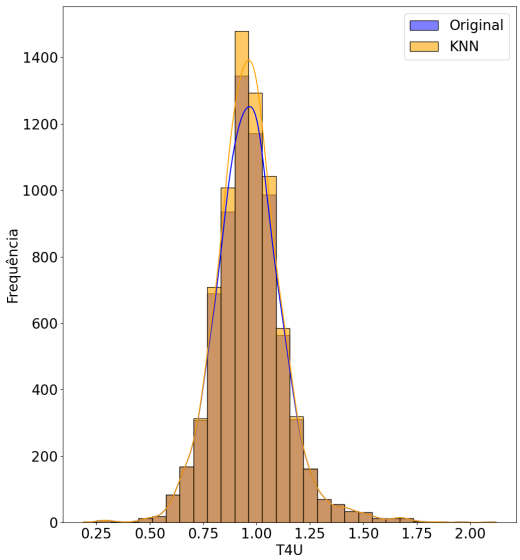


Figura 8: Distribuição da variável T4U.

2.3.6 Balanceamento das classes com SMOTE

Após a divisão inicial dos dados em treino e teste, com 33% reservados para avaliação, foi aplicado o método *SMOTE (Synthetic Minority Over-sampling Technique)* ao conjunto de treino. A função `train_test_split` do `scikit-learn` foi utilizada com o parâmetro `stratify=y` para garantir que a proporção de cada classe fosse mantida em ambos os conjuntos.

O *SMOTE* é uma técnica de oversampling que cria amostras sintéticas das classes minoritárias, de modo a balancear o conjunto de treino e permitir que o modelo aprenda melhor padrões de todas as classes. O parâmetro `random_state=44` foi definido para tornar o processo reprodutível.

2.3.7 Implementação do XGBoost

Para a etapa final do estudo de caso, foi implementado um modelo *XGBoost* para classificação dos pacientes em três classes: negativos, hipotireoidismo e hipertireoidismo. Inicialmente, buscou-se otimizar os hiperparâmetros do modelo utilizando *GridSearchCV*, mas os melhores valores foram definidos manualmente como:

- `colsample_bytree = 0.7`
- `learning_rate = 0.1`
- `max_depth = 7`
- `n_estimators = 100`
- `subsample = 1.0`
- `objective = 'multi:softmax'`
- `num_class = 3`
- `eval_metric = ['merror', 'mlogloss']`

Para lidar com o desbalanceamento das classes, foi utilizado o vetor de *sample_weight*, calculado a partir da função `compute_sample_weight` com o parâmetro `class_weight='balanced'`. Esse vetor ajusta o peso de cada amostra, dando maior importância para as classes minoritárias durante o treinamento.

O modelo foi treinado utilizando o conjunto de treino reamostrado pelo SMOTE e avaliado no conjunto de teste original, preservando a distribuição real das classes.

3 Resultados

Após a implementação do modelo XGBoost, foi realizada a avaliação de desempenho e análise da importância das features. O modelo foi treinado utilizando as amostras balanceadas pelo SMOTE e aplicando o `sample_weight` para dar maior peso às classes minoritárias.

3.1 Importância das Features

A importância das variáveis foi calculada pelo próprio modelo XGBoost, gerando um ranking que indica quais features mais contribuíram para a predição. Observou-se que, antes da utilização do `sample_weight`, a feature de maior importância era o **FTI**. Após aplicar os pesos para equilibrar as classes, a feature mais importante passou a ser o **TSH**, alinhando-se melhor com achados reportados na literatura.

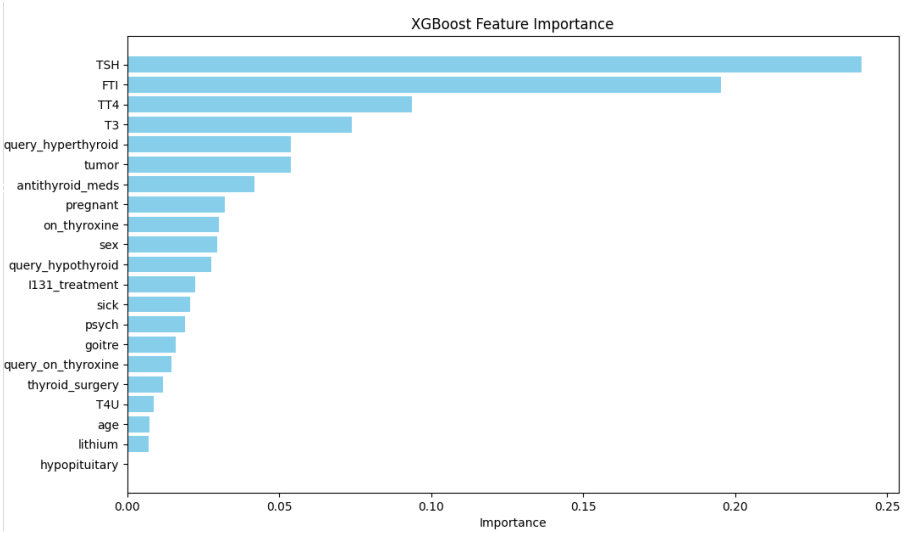


Figura 9: Importância das features calculada pelo XGBoost.

3.2 Evolução do Treinamento

O desempenho durante o treinamento foi monitorado pelas métricas de logloss e erro de classificação (`merror`) para o conjunto de treino e teste.

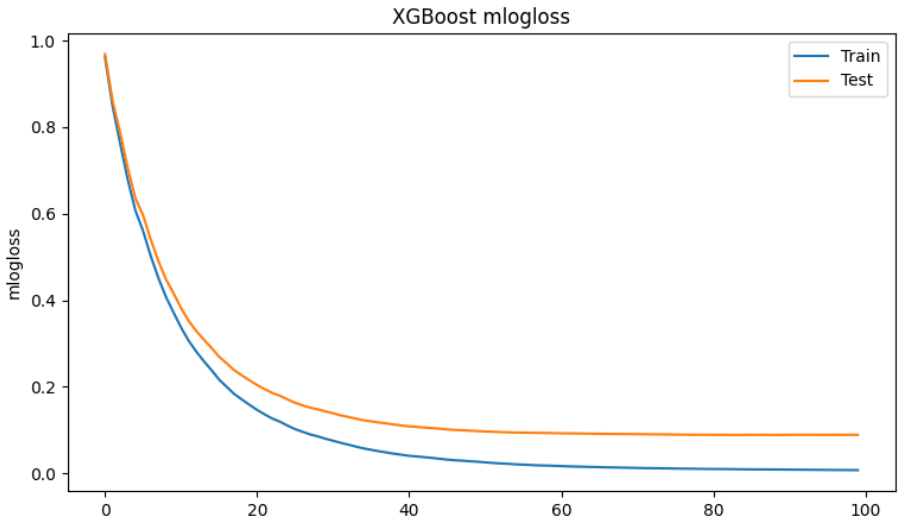


Figura 10: Evolução do `mlogloss` durante o treinamento do XGBoost.

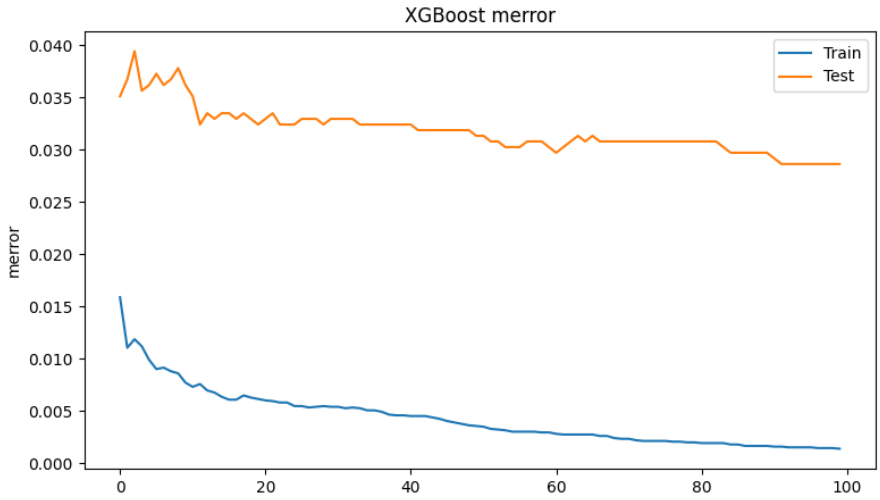


Figura 11: Evolução do `merror` durante o treinamento do XGBoost.

3.2.1 Métricas de Avaliação Final

O modelo foi testado no conjunto de teste, apresentando os seguintes resultados:

- **Accuracy:** 0.97
- **Balanced Accuracy:** 0.96
- **Micro F1-score:** 0.97
- **Macro F1-score:** 0.92
- **Weighted F1-score:** 0.97

A matriz de confusão e o relatório de classificação mostraram que o modelo conseguiu identificar de forma consistente os casos de hipotireoidismo e hipertireoidismo, mesmo diante do desbalanceamento inicial das classes.

```
------------------ Confusion Matrix -----------------

[[1586   39    7]
 [   3  163    0]
 [   4    0   50]]
```

Figura 12: Matriz de confusão do modelo XGBoost no conjunto de teste.

```
--------------- Classification Report ---------------

              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1632
           1       0.81      0.98      0.89       166
           2       0.88      0.93      0.90        54

    accuracy                           0.97      1852
   macro avg       0.89      0.96      0.92      1852
weighted avg       0.98      0.97      0.97      1852

---------------------- XGBoost ----------------------
```

Figura 13: Relatório de classificação do modelo XGBoost no conjunto de teste.

Em resumo, o modelo XGBoost com ajuste de pesos para as classes minoritárias apresentou resultados satisfatórios para a classificação automatizada das doenças da tireoide, com a importância das features condizente com achados clínicos.

4 Conclusão

O presente estudo teve como objetivo investigar como técnicas de aprendizado de máquina, em especial o XGBoost, podem auxiliar no diagnóstico automatizado de doenças da tireoide, considerando hipotireoidismo e hipertireoidismo, a partir de exames laboratoriais.

Durante o desenvolvimento do trabalho, foi possível observar que o conjunto de dados *Thyroid Disease Data* do Kaggle, contendo informações clínicas e laboratoriais de 9.172 pacientes, era suficiente e adequado para treinar um modelo de classificação confiável. Por meio de uma etapa sistemática de pré-processamento, que incluiu limpeza de dados, tratamento de valores nulos, padronização e balanceamento das classes com SMOTE, garantiu-se que o modelo fosse treinado com dados de qualidade e representativos.

A implementação do XGBoost, aliada à busca de hiperparâmetros com GridSearch e ao uso de pesos de amostra (*sample_weight*) para lidar com o desbalanceamento das classes, permitiu identificar corretamente padrões relevantes nos exames laboratoriais. Observou-se que, após aplicar o *sample_weight*, a importância das variáveis no modelo passou a refletir melhor os achados clínicos descritos na literatura: o TSH tornou-se a variável mais relevante, alinhando-se ao que é considerado um biomarcador sensível para disfunções tireoidianas.

Os resultados obtidos, avaliados por métricas como acurácia, recall, f1-score e análise da matriz de confusão, demonstram que o XGBoost foi capaz de distinguir de forma eficiente entre pacientes saudáveis, hipotireoideos e hipertireoideos. Além disso, a visualização das importâncias das features fornece uma interpretação útil para profissionais de saúde, mostrando quais exames contribuem mais para o diagnóstico automatizado.

Portanto, conclui-se que técnicas de aprendizado de máquina, especialmente o XGBoost, são ferramentas promissoras para o apoio ao diagnóstico de doenças da tireoide. Elas permitem transformar dados laboratoriais em previsões confiáveis, auxiliando médicos a identificar disfunções tireoidianas de forma mais rápida e objetiva, e potencialmente melhorando o cuidado clínico.

Referências

[1] Emmanuel Fwerr. *Thyroid Disease Data*. https://www.kaggle.com/datasets/emmanuelfwerr/thyroid-disease-data. Acessado em: 21 set. 2025. 2023.

[2] Osasere Oture, Muhammad Zahid Iqbal e Xining (Ning) Wang. "Enhanced Diagnosis of Thyroid Diseases Through Advanced Machine Learning Methodologies". Em: *Sci* 7.2 (2025), p. 66. DOI: 10.3390/sci7020066.