UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ALVARO PEDROSO QUEIROZ

UMA ABORDAGEM DE CLASSIFICAÇÃO DE SUBTIPOS DE LEUCEMIA PARA IDENTIFICAÇÃO DE GENES DIFERENCIALMENTE EXPRESSOS UTILIZANDO DADOS GENÉTICOS ALIADO À TÉCNICAS DE APRENDIZADO DE MÁQUINA

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2024

ALVARO PEDROSO QUEIROZ

UMA ABORDAGEM DE CLASSIFICAÇÃO DE SUBTIPOS DE LEUCEMIA PARA IDENTIFICAÇÃO DE GENES DIFERENCIALMENTE EXPRESSOS UTILIZANDO DADOS GENÉTICOS ALIADO À TÉCNICAS DE APRENDIZADO DE MÁQUINA

A leukemia subtype classification approach to identify differentially expressed genes using genetic data combined with machine learning techniques

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Danilo Sipoli Sanches

CORNÉLIO PROCÓPIO
2024

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio

ALVARO PEDROSO QUEIROZ

UMA ABORDAGEM DE CLASSIFICAÇÃO DE SUBTIPOS DE LEUCEMIA PARA IDENTIFICAÇÃO DE GENES DIFERENCIALMENTE EXPRESSOS UTILIZANDO DADOS GENÉTICOS ALIADO À TÉCNICAS DE APRENDIZADO DE MÁQUINA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Computação Aplicada.

Data de aprovação: 18 de Dezembro de 2023

Dr. Danilo Sipoli Sanches, Doutorado - Universidade Tecnológica Federal do Paraná

Dra. Elisangela Aparecida Da Silva Lizzi, Doutorado - Universidade Tecnológica Federal do Paraná

Dra. Glaucia Maria Bressan, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcio Dorn, Doutorado - Universidade Federal do Rio Grande do Sul (Ufrgs)

Dedico este trabalho aos meus pais, amigos e familiares que sempre me apoiaram.

AGRADECIMENTOS

Em primeiro lugar, a Deus, que fez com que meus objetivos fossem alcançados, durante todos os meus anos de estudos.

Aos meus pais, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho.

Ao professor Danilo Sipoli Sanches, por ter sido meu orientador e ter desempenhado tal função com dedicação e amizade.

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

Aos meus colegas de turma, por compartilharem comigo tantos momentos de descobertas e aprendizado e por todo o companheirismo ao longo deste percurso.

À instituição de ensino UTFPR, essencial no meu processo de formação profissional, pela dedicação, e por tudo o que aprendi ao longo dos anos do curso.

Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer (EINSTEIN, Albert, 1933).

RESUMO

Queiroz, A. P. UMA ABORDAGEM DE CLASSIFICAÇÃO DE SUBTIPOS DE LEUCEMIA PARA IDENTIFICAÇÃO DE GENES DIFERENCIALMENTE EXPRESSOS UTILIZANDO DADOS GENÉTICOS ALIADO À TÉCNICAS DE APRENDIZADO DE MÁQUINA. 113 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2024.

A leucemia é uma das principais doenças cancerígenas prejudiciais que confere mortalidade e morbidade em diferentes faixas etárias. O desafio do diagnóstico é causado por diversos fatores, sendo a classificação incorreta dos subtipos da doença um dos principais deles. Logo, torna-se fundamental descobrir os distúrbios genéticos ocorridos que ocasionou uma determinada doença. Nesse contexto, o uso de aprendizado de máquina pode ser aplicado para resolução de problemas relacionados à leucemia. Dessa forma, têm-se por objetivo criar uma ferramenta para aplicação de um modelo de aprendizado de máquina que seja interpretável e capaz de identificar genes diferencialmente expressos e classificar subtipos de leucemia. Para tal, foi proposto um *pipeline* baseado na metodologia CRISP-DM, com a finalidade de preparar dados genéticos e treinar modelos classificadores multi-classe. Assim, foram utilizadas diferentes abordagens e classificadores para determinar modelos otimizados de aprendizado de máquina com alta precisão. Os modelos utilizados possuem abordagens canônicas e hierárquicas, além de utilizarem técnicas de seleção de características para seu treinamento. Resultados altamente precisos foram obtidos nos experimentos realizados em relação a resultados obtidos na literatura, sendo possível comparar diferentes abordagens, técnicas e seleções de recursos. Por fim, uma aplicação foi criada abordando os conceitos homologados para a criação de modelos de aprendizado de máquina de forma intuitiva e para interpretabilidade dos resultados, utilizou-se a biblioteca SHAP para estabelecer os principais genes para classificação de forma global e as contribuições de cada gene para a classificação de um determinada amostra.

Palavras-chave: Aprendizado de máquina, Leucemia, Explicabilidade

ABSTRACT

Queiroz, A. P. A leukemia subtype classification approach to identify differentially expressed genes using genetic data combined with machine learning techniques. 113 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2024.

Leukemia is one of the main specific cancer diseases that confer mortality and morbidity in different age groups. The diagnostic challenge is caused by several factors, with the incorrect classification of disease subtypes being one of the main ones. Therefore, it becomes essential to discover the genetic disorders that occurred that caused a certain disease. In this context, the use of machine learning can be applied to solve problems related to leukemia. Therefore, we aim to create a tool for applying a machine learning model that is interpretable and capable of identifying differentially expressed genes and classifying subtypes of leukemia. To this end, a pipeline based on the CRISP-DM methodology was proposed, with the purpose of preparing genetic data and training multi-class classified models. Therefore, different approaches and classifiers were used to determine optimized machine learning models with high accuracy. The models used have canonical and hierarchical approaches, in addition to using feature selection techniques for their training. Highly accurate results were obtained in experiments carried out in comparison with results obtained in the literature, making it possible to compare different approaches, techniques and specific resources. Finally, an application was created addressing the approved concepts for creating machine learning models in an intuitive way and given the need to interpret the results, the SHAP library was used to establish the main genes for global classification and how contributions of each gene to the classification of a given sample.

Keywords: Machine Learning, Leukemia, Explainability

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE SIGLAS

AB	AdaBoosting
BIOINFO	Bioinformática
ACM	Association for Computing Machinery
ALL	Acute Lymphocytic Leukemia
AUC	Area Under the Curve
cDNA	Ácido Desoxirribonucleico Complementar
CRISP	CRoss-Industry Standard Process
DM	Data Mining
DNA	Ácido Desoxirribonucleico
DT	Decision Tree
FDA	Food and Drug Administration
FN	False Negatives
FP	False Positives
GEO	Gene Expression Omnibus
IA	Inteligência Artificial
IEE	Institute of Electrical and Electronics Engineers
KDD	Knowledge Discovery Databases
KNN	K-nearest Neighbors Algorithm
LCL	Local Classification per Level
LCN	Local Classification per Node
LCPN	Local Classification per Parent Node
LLA	Leucemia Linfoblástica Aguda
LLC	Leucemia Linfocítica Crônica
LMA	Leucemia Mieloide Aguda
LMC	Leucemia Mieloide Crônica
LR	Logistic Regression
MDS	Myelodysplastic Syndrome
ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naive Bayes
NCBI	National Center for Biotechnology Information
OMS	Organização Mundial da Saúde
PPG	Programa de Pós-Graduação
RF	Random Forest
RNA	Ácido ribonucleico
ROC	Receiver operating characteristic
SEMMA	Sample, Explore, Modify, Model, Assess
SHAP	SHapley Additive exPlanations
SNC	Sistema Nervoso Central
SVM	Support Vector Machine
TN	True Negatives

TP	True Positives
UTFPR	Universidade Tecnolóica Federal do Paraná
XAI	eXplainable Artificial Intelligence
ZERROR	Algoritmo de Regra Zero

LISTA DE SÍMBOLOS

ϕ	Fi
$\in$	Pertence
$\sum$	Somatório
$\mathbb{R}$	Conjunto dos Reais

SUMÁRIO

1 INTRODUÇÃO

O câncer é a doença que causa a segunda maior morte no mundo (FAUZI et al., 2021). E possui muitos tipos distintos, sendo classificado conforme a sua localização e, caso o câncer ocorra nas células do sangue, esse é denominado de leucemia (HAMIDAH et al., 2020).

A leucemia é uma das principais doenças cancerígenas prejudiciais que confere mortalidade e morbidade em diferentes faixas etárias. Ela tem início na medula óssea e espalha-se através das células sanguíneas (SARDER et al., 2020).

Uma quantidade considerável de trabalhos foi publicada na literatura sobre aplicações de aprendizado de máquina para detecção e classificação de Leucemia (ALSALEM et al., 2018a; ALSALEM et al., 2018b; SALAH et al., 2019). Esses estudos abordam métodos apropriados que permitem a classificação e detecção de leucemia e seus subtipos, com a finalidade de realizar o diagnóstico precoce de forma precisa (ALSALEM et al., 2018a). Por consequência, a crescente quantidade de dados gerados na literatura médica, permite que a integração de ferramentas que utilizem os conceitos de Inteligência Artificial (IA) sejam amplamente utilizadas (SALAH et al., 2019).

As publicações recentes que utilizam técnicas de aprendizado de máquina (subcampo da IA) para classificação de subclasses de leucemia abordaram diferentes técnicas para classificação como: K-nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB), dentre outros algoritmos consolidados na literatura. Tais técnicas, apresentaram valores significantes de acurácia (acima de 94%), além de apresentarem os principais componentes utilizados pelos algoritmos para a realização da classificação, trazendo novas percepções de informações biológicas que permitem maior interpretabilidade

do problema (VANITHA et al., 2016; CASTILLO et al., 2019; HAMIDAH et al., 2020; FAUZI et al., 2021).

1.1 PROBLEMATIZAÇÃO

O desafio do diagnóstico de leucemia é causado por diversos fatores. Um dos principais fatores é a classificação incorreta dos subtipos da doença (SALAH et al., 2019).

Assim sendo, para a realização do diagnóstico, é necessário encontrar biomarcadores da doença. Logo, torna-se fundamental descobrir os distúrbios genéticos ocorridos em uma disfunção ou mutação que ocasionou uma determinada doença. Porém, encontrar genes relacionados a doenças experimentalmente é um processo demorado devido ao grande número de genes a serem analisados (VASIGHIZAKER et al., 2019).

1.2 JUSTIFICATIVA

Na área médica é crucial entender o mecanismo de uma doença com a finalidade de buscar informações e diagnósticos precoces, proporcionando tratamentos eficazes ao paciente (SARDER et al., 2020). Recentemente, as aplicações de IA na área médica se estendem tanto para pesquisas clínicas, quanto aos procedimentos clínicos de diferentes doenças como os tumores (ALLEGRA et al., 2022).

Nesse contexto, o uso de aprendizado de máquina pode ser aplicado para resolução de problemas relacionados à leucemia (FAUZI et al., 2021). A criação de sistemas de detecção e classificação de leucemia é necessária para proporcionar tratamentos adequados aos pacientes, minimizando os perigos causados pela doença (ALSALEM et al., 2018b). Salienta-se que a realização de um diagnóstico rápido e preciso para esse câncer é vital para a recuperação do paciente (ALSALEM et al., 2018b).

Da mesma forma, o uso de abordagens computacionais para descobertas

biológicas pode acelerar experimentos e prever novos biomarcadores de doenças, além de diminuir custos para encontrar as melhores abordagens de tratamento para os pacientes (VASIGHIZAKER et al., 2019).

1.3 OBJETIVOS

1.3.1 OBJETIVOS GERAIS

Este trabalho pretende criar uma ferramenta para aplicação de um modelo de aprendizado de máquina interpretável capaz de identificar genes diferencialmente expressos através de dados de contagem de genes de pacientes que possuem leucemia e a qual subtipo de leucemia pertencem.

1.3.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Desenvolver um modelo de aprendizado de máquina para classificação de subtipos de leucemia;
- Aplicar diferentes abordagens (canônicos e hierárquicos) de aprendizado de máquina e tratamento de dados;
- Utilizar de técnicas de explicabilidade/interpretabilidade de modelos de aprendizado de máquina para identificação da contribuição dos genes na classificação de um determinado subtipo de leucemia;
- Desenvolver uma ferramenta intuitiva capaz de abordar o problema e apresentar de forma gráfica os resultados obtidos do modelo e as contribuições dos genes;

1.3.3 CONTRIBUIÇÕES

Este trabalho busca contribuir com o desenvolvimento de uma metodologia que busca testar diferentes abordagens de aprendizado de máquina, buscando o

máximo de métricas de desempenho de precisão de classificação de subtipos de leucemia.

Também irá permitir a interpretabilidade dos modelos desenvolvidos através da criação de uma ferramenta que permite visualizar de forma gráfica as contribuições que cada gene possui para a definição de um determinado subtipo de leucemia.

1.4 ORGANIZAÇÃO DO TEXTO

Neste capítulo foram apresentados o contexto em que este trabalho se insere, as motivações para a sua realização e os objetivos a serem alcançados.

No Capítulo 2 relatam-se os principais conceitos sobre o aprendizado de máquina e sobre a doença de leucemia, enfatizando-se principalmente os subtipos da doença que serão classificadas e como metologias baseadas em inteligência artificial podem ser inseridas na problemática. Também é traçada uma visão geral a respeito dos trabalhos relacionados que abordaram técnicas de aprendizado de máquina para classificação de diferentes subtipos de leucemia.

O Capítulo 3 descreve as tecnologias utilizadas para o desenvolvimento deste trabalho, abordando ferramentas, técnicas e metodologias. Os experimentos realizados podem ser encontrados no Capítulo 4, detalhando a metodologia realizada para comparação de abordagens de aprendizado de máquina. Sendo assim, discutem-se os resultados dos experimentos no Capítulo 5.

A criação da ferramenta desenvolvida é apresentada no Capítulo 6, abordando as etapas desenvolvidas para a criação de modelos interpretáveis baseado no *pipeline* homologado. As conclusões do trabalho são abordadas no Capítulo 7, bem como os possíveis trabalhos futuros. Por fim, no Capítulo 8 são feitas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

A aplicação de aprendizado de máquina para detecção e classificação de Leucemia possui uma quantidade considerável de trabalhos publicados na literatura. A leucemia pode ser classificada em subtipos que seus diagnósticos precoces permitem a realização de tratamentos adequados (ALSALEM et al., 2018a), (ALSALEM et al., 2018b), (SALAH et al., 2019). Sendo assim, os principais conceitos sobre a doença de leucemia, seus tipos e subtipos serão abordados nesse capítulo. Ademais, também serão apresentadas as metologias baseadas em aprendizado de máquina que permitem a classificação nesse determinado problema. Por fim, exibi-se uma visão geral dos trabalhos relacionados que abordam a temática de classificação de diferentes subtipos de leucemia.

2.1 LEUCEMIA

O câncer é a segunda doença que mais causa morte no mundo, ficando atrás apenas das doenças cardiovasculares (CASTILLO et al., 2019). Existem vários tipos de câncer, que podem ser classificados conforme a sua localização no corpo humano (HAMIDAH et al., 2020). A leucemia, juntamente com o linfoma e mieloma, é uma das três formas de câncer no sangue existentes (CASTILLO et al., 2019).

As leucemias são um conjunto de doenças malignas do sangue e da medula óssea que ocasionam risco de vida (JULIUSSON; HOUGH, 2016). Os indivíduos que possuem leucemia produzem um número anormal de glóbulos brancos imaturos, que colapsam a medula óssea e inibem a criação de células sanguíneas vitais do sistema imunológico (CASTILLO et al., 2019).

Os fatores que podem contribuir para o desenvolvimento da leucemia estão

atrelados a questões genéticas e ambientais (HUTTER, 2010). A doença pode se apresentar em todas as idades, onde os fatores biológicos associados a doença mais agressivos tendem a aumentar conforme o aumento da idade, em contrapartida, as estratégias de tratamento diminuem (JULIUSSON; HOUGH, 2016).

Diferentes tipos de leucemia possuem distribuições de idade muito diferentes. Existem dois tipos principais de leucemia (crônica e aguda), onde cada um dividido em duas categorias (JULIUSSON; HOUGH, 2016), sendo estas: Leucemia Mieloide Aguda (LMA), Leucemia Linfoblástica Aguda (LLA), Leucemia Mieloide Crônica (LMC) e Leucemia Linfocítica Crônica (LLC), detalhados nas próximas subseções.

A classificação da Organização Mundial da Saúde (OMS) de tumores dos tecidos hematopoiéticos e linfoides teve revisões publicadas em 2016, onde refletem opiniões de hematopatologistas, hematologistas, oncologistas e geneticistas, apresentando assim conceitos de classificação e subtipos de leucemia do ponto de vista genético (ARBER et al., 2016).

2.1.1 LEUCEMIA MIELOIDE AGUDA (LMA)

A LMA caracteriza-se pelo crescimento descontrolado de células indiferenciadas denominadas de “blastos” no sangue e na medula óssea, afetando a produção de hemácias e insuficiência da medula óssea. Não possui causa evidente, mas pode estar relacionado a exposição a benzeno, irradiações ionizantes, anemia e síndrome de Down (HAMERSCHLAK, 2008), (VAKITI; MEWAWALLA, 2021).

O tratamento da LMA envolve terapia de indução inicial e terapia pós-remissão. Vale ressaltar que o diagnóstico precoce com análise rápida de anormalidades citogenéticas e moleculares é fundamental para a adaptação da melhor terapia para o paciente (PELCOVITS; NIROULA, 2020).

Atualmente, o método de sub-tipagem de LMA leva em consideração as causas citogenéticas de LMA, conforme expressado no Quadro 1.

Quadro 1: Subtipos de LMA

LMA e neoplasias relacionadas	Detalhamento dos subtipos
Leucemia mieloide aguda com anormalidades genéticas recorrentes	LMA com t(8;21)(q22;q22) - RUNX1-RUNX1T1
	LMA com inv(16)(p13.1q22) ou t(16;16)(p13.1;q22) - CBFB-MYH11
	Leucemia Promielocítica Aguda (LPA) com PML-RARA
	LMA com t(9;11)(p22;q23) - MLLT3-KMT2A
	LMA com t(6;9)(p23;q34) - DEK-NUP214
	LMA com inv(3)(q21q26.2) ou t(3;3)(q21;q26.2) - GATA2, MECOM
	LMA (megacarioblástica) com t(1;22)(p13;q13) - RBM15-MKL1
	LMA com NPM1 mutado
	LMA com mutações bialélicas de CEBPA
Leucemia mieloide aguda com alterações relacionadas à mielodisplasia	
Neoplasias mieloides relacionadas à terapia	
Leucemia mieloide aguda, sem outra especificação	LMA com diferenciação mínima
	LMA sem maturação
	LMA com maturação
	Leucemia mielomonocítica aguda
	Leucemia monoblástica/monocítica aguda
	Leucemia eritroide pura
	Leucemia megacarioblástica aguda
	Leucemia basofílica aguda
	Panmielose aguda com mielofibrose
sarcoma mieloide	
Proliferações mieloides relacionadas à síndrome de Down	Mielopoiese anormal transitória (TAM)
	Leucemia mieloide associada à síndrome de Down

Fonte: Arber et al. (2016)

2.1.2 LEUCEMIA LINFOBLÁSTICA AGUDA (LLA)

A LLA caracteriza-se pela produção descontrolada de blastos de características linfoides do tipo B ou T de forma anormal e imatura e no bloqueio

da produção normal de glóbulos vermelhos, brancos e plaquetas. Acredita-se que a ocorrência da doença esteja relacionado após dano ao DNA. (HAMERSCHLAK, 2008), (PUCKETT; CHAN, 2022).

O tratamento bem-sucedido da LLA envolve a administração de um regime de multi-medicamentos dividido em várias fases (ou seja, indução, consolidação e manutenção) e inclui terapia direcionada ao sistema nervoso central (SNC). O objetivo da terapia de indução é alcançar a remissão completa e restaurar a hematopoiese normal (PUCKETT; CHAN, 2022), (TERWILLIGER; ABDUL-HAY, 2017).

A abordagem de análise genômica subdivide a LLA em mais de 30 subtipos genéticos. A caracterização das anormalidades genéticas em células de LLA é importante para identificação de fatores genéticos desfavoráveis e incorporar a melhor teria que possa reduzir o risco do paciente (INABA; PUI, 2021). O Quadro 2 e Quadro 3 apresentam de forma detalhada os subtipos supracitados, bem como os riscos e abordagens terapêuticas para as células de características do tipo B e tipo T respectivamente.

2.1.3 LEUCEMIA MIELOIDE CRÔNICA (LMC)

A LMC é causada pela translocação dos cromossomos 9 e 22 para criar o que é chamado de cromossomo Filadélfia (Ph). O cromossomo Ph é uma anormalidade que envolve os cromossomos de números 9 e 22. A LMC afeta tanto o sangue periférico quanto a medula óssea e as causas que levam a essa alteração são geralmente desconhecidas (EDEN; COVIELLO, 2022),(HAMERSCHLAK, 2008), (SESSIONS, 2007).

Existem 4 tratamentos de primeira linha aprovados pela Food and Drug Administration (FDA) para LMC em fase crônica, sendo inibidores de tirosina quinase comercialmente disponíveis, como *imatinib* de primeira geração e *dasatinib* de segunda geração, *nilotinib* e *bosutinib* (EDEN; COVIELLO, 2022).

Quadro 2: Subtipos de Leucemia Linfoblástica Aguda tipo B

Subtipo	Risco Genético	Tratamento
ETV6/RUNX1	Baixo	Redução de intensidade, baseada em MRD
Hyperdiploidy	Baixo	Redução de intensidade, baseada em MRD
DUX4-rearranged	Baixo	Intensidade de dose padrão, baseada em MRD
TCF3/PBX1	Intermediário	Intensidade de dose padrão, baseada em MRD, terapia intratecal intensiva
PAX5alt	Intermediário	Intensidade de dose padrão, baseada em MRD
PAX5 p.Pro80Arg	Intermediário	Intensidade de dose padrão, baseada em MRD, inibidores de JAK
ZNF384-rearranged	Intermediário	Intensidade de dose padrão, baseada em MRD
iAMP21	Intermediário	Intensificação da terapia
NUTM1-rearranged	Intermediário	Intensidade de dose padrão, baseada em MRD
Near-haploid	Alto	Intensificação da terapia, baseada em MRD, inibidores de BCL-2
Low-hypodiploid	Alto	Intensificação da terapia, baseada em MRD, inibidores de BCL-2
BCR/ABL1	Alto	Inibidores ABL1, inibidores BCL-2
BCR/ABL1-like; JAK-STAT activating mutation	Alto	Inibidores de JAK, inibidores de BCL-2
BCR/ABL1-like; ABL1-class	Alto	Inibidores ABL1, inibidores BCL-2
KMT2A (MLL)-rearranged	Alto	Inibidores de DOT1L, inibidores de menina, inibidores de proteassoma, inibidores de histona desacetilase, inibidores de BCL-2
MEF2D-rearranged	Alto	Inibidores de histona desacetilase, inibidores de proteassoma
TCF3-HLF	Alto	Inibidores de BCL-2
ETV6/RUNX1-like	Alto	Intensificação da terapia, baseada em MRD

Fonte: Inaba e Pui (2021)

2.1.4 LEUCEMIA LINFOCÍTICA CRÔNICA (LLC)

A LLC se trata do tipo mais frequente de leucemia em adultos, que se caracteriza pela expansão de células B monoclonais no sangue periférico, medula

Quadro 3: Subtipos de Leucemia Linfoblástica Aguda tipo T

Subtipo	Tratamento
Non-early T-cell precursor	Intensidade de dose padrão, baseada em MRD, nelarabina, inibidores de BCL-2
JAK-STAT activating mutation	Intensidade de dose padrão, baseada em MRD, nelarabina, inibidores de JAK, inibidores de BCL-2
ABL1 fusions (e.g., NUP214-ABL1)	Intensidade de dose padrão, baseada em MRD, inibidores ABL1, nelarabina, inibidores BCL-2
Early T-cell precursor ALL	Intensidade de dose padrão, baseada em MRD, inibidores de JAK, inibidores de BCL-2

Fonte: Inaba e Pui (2021)

óssea, linfonodos e baço (BOSCH; DALLA-FAVERA, 2019), (SCARFÒ et al., 2016).

Os pacientes diagnosticados com LLC não necessitam de tratamento, visto que se trata de uma doença heterogênea. As formas de tratamento atuais não podem curar a LLC, com exceção do transplante alogênico de células-tronco hematopoiéticas (MUKKAMALLA et al., 2022).

Assim sendo, nota-se que a identificação do subtipo de leucemia se torna uma etapa fundamental para a realização do tratamento adequado que permita ser bem-sucedido. Para a realização de tal tarefa com alta precisão e de forma transparente, pretende-se aplicar conceitos de aprendizado de máquina para solução do problema.

2.2 APRENDIZADO DE MÁQUINA

A IA busca imitar computacionalmente a inteligência humana através de diferentes algoritmos. Recentemente, as aplicações médicas com uso da IA se estendem a pesquisas clínicas, medicina translacional e procedimentos clínicos diferentes, como os tumores. Nesse contexto, um dos principais subcampos da IA utilizado na área médica é o aprendizado de máquina (ALLEGRA et al., 2022). Esse subcampo da IA é composto por algoritmos que se baseiam em informações prévias, conseguindo generalizar uma função matemática que permite realizar previsões sobre novos conjuntos de dados (THEOBALD, 2018).

A revolução do aprendizado de máquina ocorreu por consequência de diversos motivos. Primeiramente, o aumento do volume de dados e o poder

de computação disponível permitiram a realização de feitos interessantes, mesmo usando abordagens (conceitualmente) antigas. Assim sendo, o campo de pesquisa na área está em constante expansão, com desenvolvimentos de métodos que permitem escalar melhor e possuir maior investimento em recursos de dados e desenvolvimentos de sistemas (SKIENA, 2017).

O termo aprendizado de máquina ou *machine learning* (ML) foi estabelecido por volta de 1960, composto pela palavra aprendizado que se refere à tarefa de aprender uma atividade ou identificar um padrão de eventos, e a palavra máquina referindo-se a um computador, robô ou outro dispositivo (LIU, 2017).

Um algoritmo de aprendizado de máquina consiste em treinar computadores a realizar tarefas de forma inteligente utilizando dados de exemplos ou experiências passadas (NAQA; MURPHY, 2015). Esses algoritmos se alteram ou se adaptam automaticamente por repetição e se tornam melhores na realização da tarefa desejada. O processo de adaptação é denominado de treinamento, em que são fornecidas amostras de dados de entrada, criando um modelo matemático generalizado que produz o resultado desejado a partir de novos dados (NAQA; MURPHY, 2015).

O campo é composto por três categorias de aprendizado: supervisionado, não-supervisionado e por reforço (THEOBALD, 2018), sendo assim, a aplicação de determinado aprendizado é dependente do problema a ser resolvido.

2.2.1 APRENDIZADO SUPERVISIONADO

No aprendizado supervisionado, também denominado reconhecimento de padrões, se dá a construção de um procedimento de classificação a partir de um conjunto de dados para os quais as classes verdadeiras são conhecidas. Portanto, o procedimento será aplicado a sequência contínua de casos, em que a cada novo caso, deve ser atribuído um rótulo de um conjunto de classes predefinido, com base em atributos ou recursos observados (TAYLOR et al., 1994).

A aprendizagem supervisionada é um paradigma constante em problemas de classificação e regressão.

A regressão treina e prevê uma resposta de valor contínuo, enquanto a classificação tenta encontrar a classe apropriada (rótulo) (LIU, 2017). A entrada são vetores de características x_i, cada um com um rótulo associado de classe ou valor alvo y_i, representando a supervisão (SKIENA, 2017).

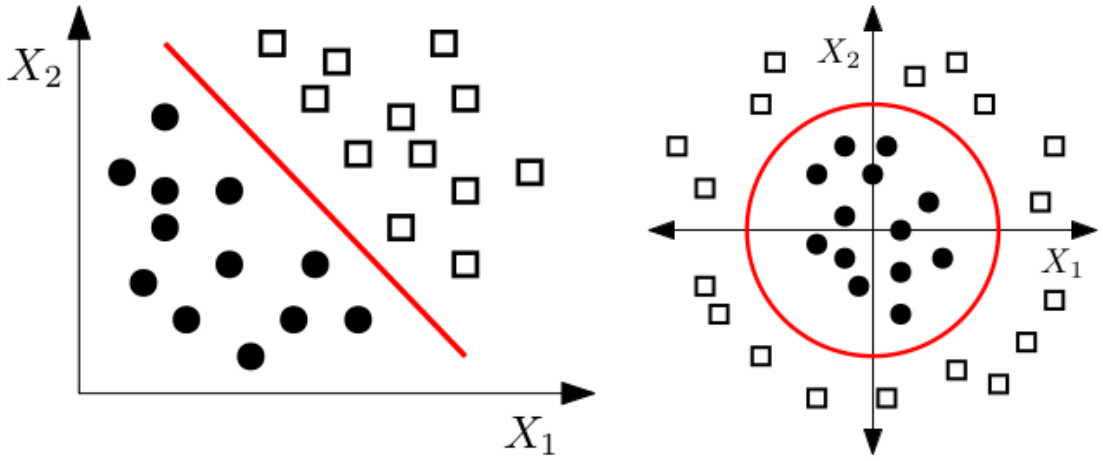
Nesse cenário, quando o problema em questão possui dados de aprendizagem que possuem descrição, metas ou saídas desejadas além dos sinais indicativos, o objetivo da aprendizagem torna-se encontrar uma regra geral que possibilita mapear as entradas em saídas correspondentes. Para esses tipos de dados de aprendizado, denomina-se dados rotulados e a regra aprendida é usada para rotular novos dados que possuem saídas desconhecidas. Os rótulos são geralmente fornecidos por sistemas de registro de eventos e humanos especialistas (LIU, 2017).

2.2.1.1 CLASSIFICAÇÃO

O problema da atribuição de um rótulo a um item a partir de um conjunto de possibilidades é bastante frequente, por exemplo, prever o vencedor de uma determinada competição esportiva, ou decidir o gênero de um determinado filme. Tais problemas, possuem características de classificação, visto que cada um envolve a seleção de um rótulo das opções possíveis (SKIENA, 2017).

Dessa forma, a tarefa de classificação ocorre em uma gama de atividades humanas, onde o termo pode abranger qualquer contexto, em que alguma decisão ou previsão seja feita com base em informações atualmente disponíveis. Logo, o procedimento de classificação é um método formal que consegue realizar tais julgamentos em novas situações (TAYLOR et al., 1994). A Figura 1 expressa um exemplo da separação de classes que deve ser abordado por um algoritmo de classificação.

Figura 1: Exemplo de separação de classes [1]



Fonte: Skiena (2017)

Assim sendo, o uso da técnica de aprendizado de máquina pode ser utilizado para a realização da classificação dos diferentes subtipos de leucemia e por consequência entender quais são as características mais importantes que permitem realizar a diferenciação entre os mesmos.

2.2.2 APRENDIZADO NÃO SUPERVISIONADO

No aprendizado não supervisionado, as variáveis e padrões de dados não possuem classificação, logo, a máquina deve descobrir padrões ocultos e criar rótulos através de algoritmos não supervisionados (THEOBALD, 2018).

Os métodos não supervisionados tentam encontrar estrutura nos dados, fornecendo rótulos ou valores (classificações) sem nenhum padrão confiável. São utilizados para exploração e dar sentido a um conjunto de dados (SKIENA, 2017).

O algoritmo k-means é um exemplo popular de aprendizado não supervisionado, onde se trata de agrupar pontos de dados que possuem características semelhantes (THEOBALD, 2018).

[1] Exemplo de como um algoritmo de classificação deve se comportar para a separação de classes. O primeiro exemplo trata-se de um problema linearmente definido. O segundo trata-se de um problema não linearmente separável.

2.2.3 APRENDIZADO POR REFORÇO

O aprendizado por reforço consiste em melhorar continuamente o modelo, através de *feedbacks* de iterações anteriores (diferente de aprendizados supervisionados e não supervisionados, em que atingem um ponto final indefinido após um modelo ser formulado a partir dos segmentos de dados de treinamento e teste) (THEOBALD, 2018). Os dados de aprendizagem fornecem *feedbacks* para que o sistema adapta-se às condições dinâmicas para atingir um determinado objetivo. O sistema avalia seu desempenho com base as respostas de *feedback* e reage de acordo. Os casos mais conhecidos incluem carros autônomos e sistemas especialistas em xadrez (LIU, 2017).

2.3 TRABALHOS RELACIONADOS

Numerosos estudos confirmaram a necessidade da utilização de sistemas que fornecem resultados de precisão com resposta rápida para detecção e classificação de leucemia (ALSALEM et al., 2018b). Deste modo serão apresentados no Quadro 4, alguns desses trabalhos que exploram tal problemática.

A estratégia de busca de trabalhos relacionados foi pautada nas seguintes diretrizes:

- O trabalho propõe métodos de classificação de leucemia;
- A classificação deve ser multi-classe;
- A classificação deve ser relacionada a distinguir tipos e subtipos de leucemia;
- O classificador deve utilizar técnicas de aprendizado de máquina;
- Os dados devem ser de expressão gênica;
- Só devem ser utilizadas como classes, tipos e subtipos de leucemia;

Nos trabalhos relacionados pode-se notar a utilização de diferentes técnicas de aprendizado de máquina para classificação de subclasses de leucemia, tais como

Quadro 4: Trabalhos Relacionados

Referência	Multi-classes	Técnicas	Resultados
Vanitha et al. (2016)	LMA; LLA T-Cell; LLA B-Cell;	Multiclass Support Vector Machine (mSVM); k-Nearest Neighbor (k-NN);	mSVM apresentou melhores resultados comparados a técnica de k-NN, obtendo acurácia de 100%.
Castillo et al. (2019)	LMA; LLA; LMC; LLC;	Support Vector Machines (SVM); Random Forest (RF); k-Nearest Neighbor (k-NN); Naive Bayes (NB);	Com cerca de 40 genes expressivos, e a técnica k-NN, obteve-se 98.87% de acurácia e 99.05% de f1-score.
Hamidah et al. (2020)	BCR-ABL; E2A-PBX; Hyperdiploid$>$50 chromosomes; MLL; T-ALL; TEL-AML1;	One-Against-One Multiclass Support Vector Machine (OAO-MSVM);	O modelo gerou 94% de acurácia, 96% de precisão, 95% de recall e 95% de F1 Score.
Fauzi et al. (2021)	AML; ALL T-Cell; ALL B-Cell;	Fuzzy Support Vector Machine (FSVM);	FSVM com PCA como seleção de recursos (60 features) resultou em 96,924% com treinamento em 80% dos dados.
Schmidt et al. (2022)	18 subtipos: TCF3-PBX1; HLF; IKZF1 N!59Y; PAX5 P80R; Ph; Ph-like; PAX5alt; DUX4; ETV6-RUNX1; ETV6-RUNX1-like; MEF2D; NUTM1; High hyperdiploid; Low hyperdiploid; Near haploid; Low hypodiploid; BCL2/MYC; iAMP21; 5 meta-subtipos: ZNF384 group; KMT2A group; Ph group; ETV6-RUNX1 group; High Sig group;	Logistic Regression	Verificou-se que ALLSorts tem uma precisão geral de 92%. No entanto, o desempenho da classificação foi desequilibrado entre os subtipos.

Fonte: Autoria Própria (2023)

K-nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB) e Logistic Regression (LR). As técnicas obtiveram resultados significantes, obtendo acurácias acima de 94%.

Vale destacar que nas pesquisas de Castillo et al. (2019) e Fauzi et al. (2021) também abordam os genes que representam maior importância para a classificação das subclasses de leucemia conforme os modelos de aprendizado de máquina gerados, gerando informações adicionais para pesquisas futuras de especialistas na área.

O trabalho de (SCHMIDT et al., 2022) apresenta o ALLSorts, um classificador de subtipo de RNA-Seq para leucemia linfoblástica aguda de células B. O classificador consegue distinguir 18 subtipos de LLA com alta precisão.

Os estudos realizados em (FELTES et al., 2019) apresentam o Curared Microarray Database (CuMiDa), onde possuem conjuntos de dados de câncer curados de estudos do Gene Expression Omnibus (GEO), exclusivamente para aprendizado de máquina, contendo também conjuntos de dados de multi-classes de leucemia.

O Quadro 5 apresenta um resumo sobre as implementações de aprendizados de máquina em dados de leucemia de multi-classe que também serão utilizadas para validação do pipeline criado e os resultados do estudo utilizados como resultados de linha de base. As técnicas de aprendizado de máquina utilizadas para todos os projetos citados são: algoritmo de regra zero (ZERROR), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Decision Tree (DT), Naive Bayes (NB), Random Forest (RF), K-Nearest Neighbors (KNN).

Entretanto, em tais estudos não se é utilizado ferramentas de interpretabilidade desses recursos, como valores estatísticos e formas de visualização gráfica, que permitiriam obter uma maior compreensibilidade do aprendizado do modelo e assim poder verificar como cada gene se relaciona para a realização da classificação de uma determinada subclasse, sendo assim um diferencial proposto desse presente trabalho.

Quadro 5: Projetos de multi-classes de leucemia do CuMiDa

Projeto	Multi-classes	Resultados
GSE9476	AML, Bone_Marrow, Bone_Marrow_CD34, PB, PBSC_CD34	SVM atingiu uma acurácia de 98,4%
GSE71449	JMML_LIN28_high, JMML_LIN28_low, MML_LIN28_high, normal	Decision Tree atingiu uma acurácia de 71,1%
GSE28497	B-CELL_ALL, B-CELL_ALL_ETV6-RUNX1, B-CELL_ALL_HYPERDIP, B-CELL_ALL_MLL, B-CELL_ALL_T-ALL, B-CELL_ALL_TCF3-PBX1, B-CELL_ALL_HYPO	SVM atingiu uma acurácia de 87,9%

Fonte: Autoria Própria (2023)

3 METODOLOGIA

Conforme os estudos realizados na área da classificação de subtipos de leucemia, propõe-se o desenvolvimento de um classificador baseado em aprendizado de máquina, para se obter um modelo que possa identificar os diferentes subtipos de leucemia e poder, através da interpretabilidade do modelo, identificar biomarcadores (genes) que contribuem para a diferenciação dos mesmos. Sendo assim, este capitulo apresenta as tecnologias e metodologias que serão adotadas para que o objetivo possa ser alcançado.

3.1 TECNOLOGIAS E FERRAMENTAS

Para o desenvolvimento do projeto, serão apresentadas a linguagem de programação utilizada, o ambiente de desenvolvimento integrado e as principais bibliotecas que permitem a realização da ciência de dados no problema supracitado.

3.1.1 PYTHON

A linguagem de programação tradicionalmente utilizada pela comunidade de cientistas de dados é a linguagem Python. Nela é possível obter uma variedade de bibliotecas e recursos de linguagem que permitem de forma facilitada a manipulação e visualização de dados, além da utilização de técnicas de aprendizado de máquina. Vale ressaltar que, por ser uma linguagem interpretada, o processo de desenvolvimento é rápido e agradável (SKIENA, 2017).

3.1.2 JUPYTER-NOTEBOOK

Um projeto de ciência de dados abrange vários domínios além de código, como a manipulação e visualização de dados, resultados computacionais e análises de aprendizados obtidos durante o processo (SKIENA, 2017). Dessa forma, o uso do Jupyter-Notebook como ambiente de desenvolvimento integrado se dá ao fato do mesmo possuir ferramentas eficientes que unem texto e código, auxiliando na organização, anotação e detalhamento de funcionalidades, execução de blocos de código e visualização de resultados gráficos na própria ferramenta, se tornando assim um instrumento primoroso para realizar experimentos.

3.1.3 PLATAFORMA ANACONDA

A plataforma Anaconda é uma das mais populares na área de ciência de dados, possuindo mais de 150 pacotes pré-instalados, e 250 pacotes de código aberto que podem ser instalados (KADIYALA; KUMAR, 2017).

As bibliotecas da plataforma Anaconda são de extrema importância para a utilização na área da ciência de dados, haja vista, a necessidade da manipulação dos dados com o *pandas*, a utilização de estruturas de dados e manipulação destas com *numpy* e *scipy*, a criação de gráficos e visualizações de dados em geral com o *matplotlib* e o *seaborn* e o uso de funções matemáticas com o *math*.

3.1.4 BIBLIOTECA SCIKIT-LEARN

A biblioteca Scikit-Learn é uma ferramenta desenvolvida em linguagem Python, que permite a utilização de numerosas técnicas de análise de dados e pré-processamento, além de abranger diversas técnicas de classificação para criação de modelos de aprendizado de máquina (PEDREGOSA et al., 2011).

3.1.5 BIBLIOTECA SHAP

A biblioteca SHAP permite explicar previsões individuais baseado em valores de Shapley ótimos da teoria de jogos. É possível calcular a contribuição de

cada recurso para uma previsão, além de disponibilizar formas gráficas de análise (LUNDBERG; LEE, 2017).

3.1.6 BIBLIOTECA STREAMLIT

A biblioteca Streamlit permite a criação de interfaces, exibição de textos, visualização de dados e gerenciamento de um aplicativo *WEB* de forma simplificada e altamente intuitiva (KHORASANI et al., 2022).

3.2 MATERIAIS E MÉTODOS

O processo de mineração de dados consiste em inferir conhecimento a partir de grandes volumes de dados. Logo, é preferível obter uma metodologia que possa extrair informações de um conjunto de dados e transformá-las em uma estrutura compreensível, com a descoberta de padrões utilizando sistemas computacionais inteligentes (KESAVARAJ; SUKUMARAN, 2013).

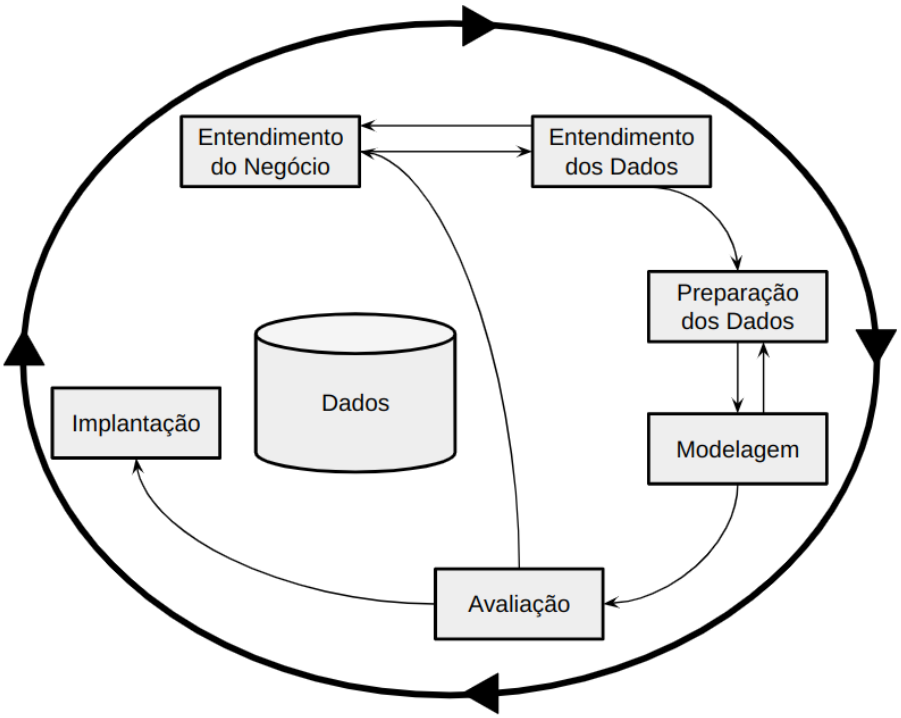
Na literatura é possível encontrar diferentes metodologias que permitem compreender as informações dos dados. Uma das metodologias mais famosas é o *Knowledge Discovery Databases* (KDD), que consiste em nove etapas que permite na descoberta de conhecimento através dos dados. A metodologia *Sample*, *Explore*, *Modify*, *Model*, *Assess* (SEMMA) consiste em cinco fases diferentes que foca nas tarefas de criação de um modelo. Por fim, a metologia *CRoss-Industry Standard Process for Data Mining* (CRISP-DM) possui seis etapas em que permite o entendimento do problema em questão e os processos de criação de um modelo (SHAFIQUE; QAISER, 2014).

3.2.1 METODOLOGIA CRISP-DM

Visto a necessidade do entendimento da área de pesquisa na qual esse trabalho está inserido e a proposta de desenvolvimento de um modelo inteligente de classificação, a metodologia CRISP-DM se torna mais viável a ser utilizada como referência para o desenvolvimento da solução proposta.

Assim sendo, a metologia CRISP-DM pode ser descrita por seis etapas conforme disposto na Figura 2.

Figura 2: Metodologia CRISP-DM.



Fonte: Adaptado de Shafique e Qaiser (2014)

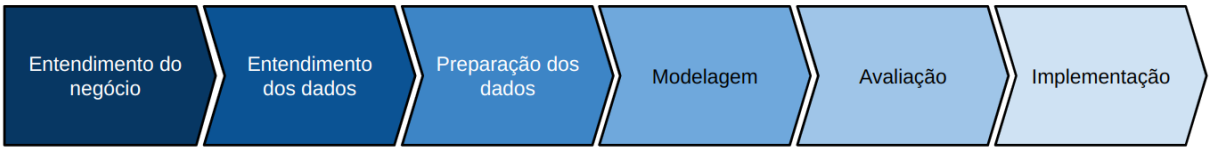
Segundo Schröer et al. (2021) cada etapa pode ser descrita como:

- *Business Understanding* (Entendimento do negócio): deve-se obter uma visão geral dos recursos disponíveis e necessários, além de determinar os objetivos e critérios de sucesso;
- *Data Understanding* (Entendimento dos dados): coletar dados de fontes de dados, explorá-los e descrevê-los a fim de verificar a qualidade dos dados, determinando os atributos e agrupamentos;
- *Data Preparation* (Preparação dos dados): realizar seleção de dados a partir de critérios de inclusão e exclusão definidos e aplicar tratamentos em dados de má qualidade;
- *Modeling* (Modelagem): selecionar técnicas de modelagem, construir casos de teste e selecionar os melhores modelos;

- *Evaluation* (Avaliação): verificar e interpretar os resultados em relação aos objetivos definidos;
- *Deployment* (Implantação): implantação do modelo desenvolvido, geralmente consiste em realizar monitoramentos e manutenções.

Aliado as etapas propostas pela metodologia CRISP-DM, criou-se um *pipeline* aplicada ao problema da classificação de subgrupos de leucemia, ilustrado na Figura 3.

Figura 3: Pipeline proposto.



Fonte: Autoria Própria (2023)

As etapas do *pipeline* proposto consistem em:

- Entendimento do negócio: identificação de subtipos de leucemia a serem classificadas;
- Entendimento dos dados: coleta dos dados (fontes de dados), análise exploratória dos dados;
- Preparação dos dados: tratamento dos dados (limpeza de dados, padronização, divisão de dados para treinamento e teste) e seleção de atributos (redução de dimensionalidade);
- Modelagem: treinamento utilizando diferentes abordagens e técnicas de classificação, construção de casos de teste e validação cruzada;
- Avaliação: comparação de desempenho dos diferentes modelos criados por métricas de avaliação;
- Implementação: criação de ferramenta para realização de inferências para dados desconhecidos e interpretabilidade dos modelos utilizando a técnica SHAP;

Nas seções subsequentes apresentam-se as técnicas a serem utilizadas pelo *pipeline*.

3.2.2 SELEÇÃO DE ATRIBUTOS

A seleção de atributos é uma das principais técnicas frequentemente usadas em pré-processamento de dados, a qual permite a redução do número de recursos segundo algum critério de importância, removendo dados irrelevantes, ruídos e redundantes, aumentando a qualidade dos dados e os modelos construídos podem ser mais compreensíveis (LIU; YU, 2005).

Em casos nos quais a medição de atributos é muito custosa, a seleção de atributos se torna uma técnica fundamental, selecionando um subconjunto representativo menor que o original e consequentemente, é produzido efeitos imediatos como melhorar o desempenho e os tempos de execução de predição do modelo, reduzir o sobreajuste do modelo e aumentar a generalização (MORAN; GORDON, 2019).

Existem vários algoritmos de seleção de recursos, a maioria utiliza técnicas de pesquisa com um índice de avaliação para encontrar um subconjunto de recursos apropriado. A técnica mais básica pesquisa todos os subconjuntos possíveis e seleciona o subconjunto que minimiza o erro do modelo. No entanto, esta busca exaustiva não é computacionalmente eficaz, particularmente em situações onde existem muitos recursos (CHANDRASHEKAR; SAHIN, 2014).

3.2.2.1 ABORDAGEM *FILTER*

As técnicas de filtro avaliam o poder discriminativo dos recursos com base apenas nas propriedades intrínsecas dos dados. Como regra geral, esses métodos estimam uma pontuação de relevância e um esquema de limite é usado para selecionar as características de melhor pontuação. As técnicas de filtro não são necessariamente usadas para construir preditores (LAZAR et al., 2012).

Nenhum algoritmo de classificação é usado durante o processo de seleção de recursos. Especificamente, o processo de seleção de recursos não é orientado pelo

desempenho de um método de classificação (SANTANA; CANUTO, 2014).

3.2.2.2 ABORDAGEM *WRAPPER*

Trata-se de um processo de seleção de recursos guiado pela precisão de um algoritmo de classificação. Um subconjunto de recursos é escolhido com base no desempenho de um método de classificação específico. Dois algoritmos de classificação diferentes levam à escolha de diferentes subconjuntos de recursos. A principal desvantagem dessa abordagem está relacionada à enorme complexidade computacional para avaliar os subconjuntos de recursos obtidos executando um algoritmo de classificação específico em um conjunto de dados para todos os subconjuntos (SANTANA; CANUTO, 2014).

Os algoritmos de *wrapper* selecionam um subconjunto de recursos relevantes com base em uma medição de desempenho de um método de aprendizagem. Pode-se esquematizar a metodologia do *wrapper* em três etapas: a definição da medida de desempenho que serve como critério de seleção de recursos e a estratégia de reamostragem para validação; o estabelecimento da estratégia de busca para a definição da ordem em que os subconjuntos de variáveis são avaliados e, o método de aprendizagem adotado. A medição de desempenho preditivo de um modelo de aprendizagem de classificação estabelecerá o subconjunto de recursos relevantes (GUYON; ELISSEEFF, 2003).

3.2.2.3 ABORDAGEM *EMBEDDED*

O terceiro grupo de abordagens de seleção de recursos são os métodos incorporados, que integram a seleção de recursos e o procedimento de aprendizagem em um único processo. Os métodos de regularização são uma técnica embarcada importante e executam a construção do modelo de aprendizagem e a seleção automática de recursos simultaneamente (LIU et al., 2018).

Também pode evitar o problema de demora em comparação com a abordagem *wrapper*. Com isso, os recursos selecionados são mais diretamente para melhorar o desempenho da classificação. Mas com a mudança dos dados de treinamento, o

modelo treinado gerará diferentes classificações de recursos, portanto, é difícil obter classificações de recursos relativamente estáveis. Isso não conduzirá a uma análise mais aprofundada dos dados e à generalização do algoritmo (GUO et al., 2019).

3.2.3 MODELAGEM

Trata-se da etapa da seleção de processos e aplicação de várias técnicas de aprendizado de máquina. São definidos diferentes parâmetros e modelos distintos para solução do problema em questão (SHAFIQUE; QAISER, 2014).

As técnicas de aprendizado de máquina são baseadas em algoritmos que possuem conjuntos de procedimentos matemáticos que descrevem as relações entre os recursos disponíveis nos dados. Embora os algoritmos funcionem de maneiras diferentes, há semelhanças na forma em que são desenvolvidos (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019).

Nas subseções seguintes, serão apresentadas algumas técnicas de aprendizado de máquina utilizadas para a realização de experimentos.

3.2.3.1 ÁRVORE DE DECISÃO

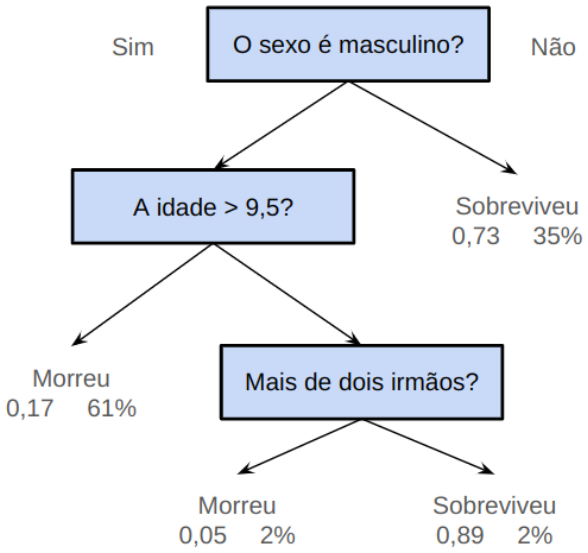
Uma árvore de decisão consiste em um modelo sequencial que combina regras lógicas que comparam um atributo numérico com um valor limite ou atributo nominal com um conjunto de valores possíveis (KOTSIANTIS, 2011). Tais regras lógicas formam uma árvore direcionada, com um nó inicial denominado de raiz. Todos os outros nós possuem arestas de entrada, sendo denominado de nó interno. Cada nó interno divide o espaço de instância em dois ou mais subespaços de acordo com uma função discreta dos valores dos atributos de entrada. Por último tem-se os nós terminais, em que consistem nos nós de decisão, onde representam uma classe ou uma porcentagem de chance do valor mais apropriado (ROKACH; MAIMON, 2005).

O algoritmo possui duas fases principais: crescimento e poda. A fase de crescimento corresponde a um particionamento recursivo dos dados de treinamento, resultando em uma árvore de decisão cujo cada nó terminal esteja associado a uma única classe. Já a fase de poda, visa generalizar a árvore criada por uma sub-árvore

que evita o sobreajuste excessivo dos dados de treinamento (KOTSIANTIS, 2011).

A cada iteração, o algoritmo considera uma partição do conjunto de treinamento. A seleção da divisão mais apropriada é feito com algumas medidas de divisão, onde cada nó subdivide ainda mais o conjunto de treinamento em subconjuntos menores, até que as condições de parada estejam satisfeitas (KOTSIANTIS, 2011).

Figura 4: Exemplo de Árvore de decisão [2]



Fonte: Adaptado de Skiena (2017)

3.2.3.2 FLORESTAS ALEATÓRIAS

O esquema de florestas aleatórias foi proposto por Breiman (2001) para a construção de um conjunto de predição com a utilização de árvores de decisão que crescem em subespaços de dados selecionados aleatoriamente. Na prática, a técnica encaixa várias subamostras do conjunto de dados original e utiliza a média para o controle de ajuste e melhoramento da precisão preditiva.

A técnica de florestas aleatórias consiste tipicamente em um conjunto de dezenas a milhares de classificadores baseados em árvores de decisão. O conjunto opera em paralelo e classifica por maioria dos votos, melhorando o desempenho da

[2] Exemplo de Árvore de decisão simples para o problema de prever a mortalidade no Titanic.

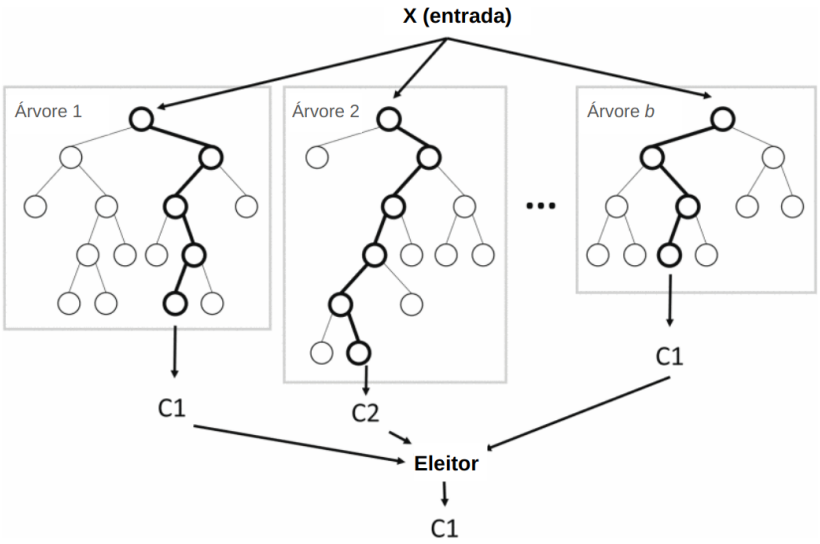
classificação em conjunto comparado a um único classificador, considerando que cada classificador tenha um desempenho melhor do que uma estimativa aleatória e em um conjunto onde possui múltiplos classificadores, considera-se que as classificações incorretas são superadas em número pelas classificações corretas em termos de probabilidade, resultando assim no agregado uma classificação correta (HATWELL et al., 2020).

O processo possui dois grandes estágios para a indução das árvores de decisão. O primeiro estágio consiste em cada árvore de decisão ser induzido em uma amostra uniforme com substituição (*bootstrap*) do conjunto de treinamento (HATWELL et al., 2020).

No segundo estágio, as divisões de candidatos são limitadas a uma subamostra aleatória de possíveis candidatos. Nessas condições, as árvores crescem totalmente, de modo que suas folhas sejam puras e cada folha cobre instâncias de treinamento de apenas uma classe (HATWELL et al., 2020).

Em estudos realizados por Biau (2010) destaca que esse procedimento é consistente e adaptativo a escassez, onde sua taxa de convergência é dependente apenas do número de características fortes, não sendo dependente da quantidade de variáveis de ruído presentes.

Figura 5: Exemplo da estrutura de uma floresta aleatória.



Fonte: Adaptado de Nakahara et al. (2017)

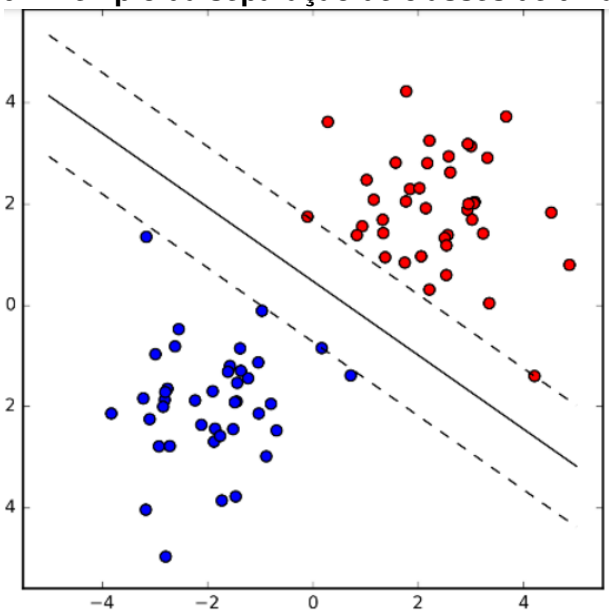
3.2.3.3 MÁQUINA DE VETORES DE SUPORTE MULTI-CLASSE

As Máquinas de Vetores de Suporte (do inglês Support Vector Machines) (SVMs) são uma forma importante de construir classificadores. Tal técnica consiste em buscar separadores lineares de margem máxima entre duas classes, visto que o separador de margem máxima deve ser o mais robusto entre duas classes (SKIENA, 2017).

O padrão SVM procura encontrar o hiperplano de separação ideal entre classes binárias seguindo o critério de margem maximizada (WANG; XUE, 2014). Dessa forma, tal técnica é projetada para resolução de problemas de classificação dicotômica. Porém, os problemas de classificação multi-classe é comumente resolvido por uma decomposição em vários problemas binários para os quais os SVM padrão pode ser utilizado (FRANC; HLAVAC,). Assim sendo, para problemas multi-classes, algumas abordagens são propostas como:

- *One-Versus-Rest*: a abordagem um contra o resto (também conhecido como *One-vs-All*) constrói *k* classificadores binários separados para a classificação da classe *k*. Um dos classificadores binários é treinado usando dados de uma determinada classe como exemplos positivos e o restante como exemplos de classes negativas. Para a previsão de uma classe, o modelo prevê uma probabilidade de associação, selecionando o rótulo determinado pelo classificador binário que fornece maior valor de saída (WANG; XUE, 2014).
- *One-Versus-One*: outra abordagem clássica para classificação multi-classe é a classificação um contra um ou decomposição em pares. Tal abordagem consiste em avaliar todos os classificadores em pares possíveis, portanto, induz $k(k-1)/2$ classificadores binários individuais. Assim sendo, para a realização da previsão de uma classe, os classificadores binários induzem um voto para uma determinada classe, sendo rotulado a classe que obtiver mais votos (WANG; XUE, 2014).

Figura 6: Exemplo da separação de classes de uma SVM [3]



Fonte: Skiena (2017)

3.2.3.4 ADAPTIVE BOOSTING (ADABOOST)

Consiste em uma técnica geral que melhora o desempenho de qualquer algoritmo de aprendizagem, sintetizando que o reforço pode ser usado para reduzir o erro em qualquer algoritmo de aprendizagem ”fraco”, gerando classificadores que precisam ser um pouco melhores que suposições aleatórias, alcançando precisão arbitrariamente alta (FREUND, 1995), (SCHAPIRE, 1990).

Dessa forma, é implementado um algoritmo de aprendizagem “fraco” e a cada nova iteração é produzida uma nova regra fraca, que após muitas repetições, gera uma combinação dessas regras em uma única regra de predição, obtendo uma precisão maior que qualquer uma das anteriores (SCHAPIRE, 2003).

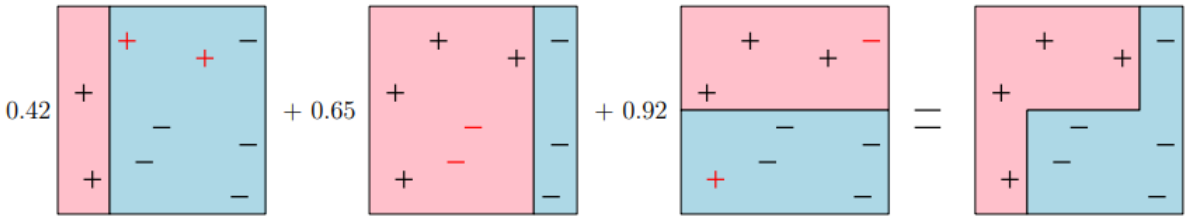
Para tal, a ideia da técnica consiste em pesar os exemplos conforme o quão difícil é acertá-los e recompensar os classificadores com base no peso dos exemplos que acertam. Para definir os pesos do classificador, é ajustado os pesos dos exemplos de treinamento, recompensando os classificadores que acertam os casos difíceis

[3] SVMs procuram separar as duas classes pela maior margem, criando um canal em torno da linha de separação.

(SKIENA, 2017).

No exemplo da Figura 7, é apresentado classificadores fracos (geralmente são árvores de decisão mais simples possíveis, somente com 1 nó), que codificam uma lógica simples para não ajustar demais os dados. Os pesos relativos atribuídos a cada uma dessas árvores decorrem do procedimento de treinamento, que tenta ajustar os resíduos (erros das rodadas anteriores) e aumenta os pesos das árvores que classificam os exemplos mais difíceis. Logo, o classificador será construído como a união de classificadores não lineares, usando características limiarizadas (SKIENA, 2017).

Figura 7: Exemplo do funcionamento do algoritmo AdaBoosting



Fonte: Skiena (2017)

3.2.3.5 CLASSIFICAÇÃO HIERÁRQUICA

Classificação hierárquica é um tipo de problema onde as classes envolvidas podem ser divididas em subclasses ou agrupadas em superclasses. Em métodos hierárquicos de classificação, o algoritmo de aprendizagem captura relacionamentos relevantes entre as classes, considerando a hierarquia presente no conjunto de dados de treinamento (FREITAS; CARVALHO, 2007).

Segundo Freitas e Carvalho (2007), quatro soluções podem ser utilizadas para tratar problemas de classificação hierárquica:

- Transformação de um problema de classificação hierárquica em um problema de classificação não hierárquica: trata-se da ideia de que um problema de classificação não hierárquica seja um caso particular de classificação hierárquica, no qual não existem subclasses e superclasses. Dessa forma, pode-se utilizar técnicas tradicionais, sem a necessidade de alterações;

- Predição hierárquica com algoritmos de classificação não hierárquicos: essa abordagem divide o problema de classificação hierárquica em um conjunto de problemas de classificação não hierárquica. A diferença entre a abordagem anterior consiste no fato dessa abordagem considerar os diferentes níveis da hierarquia simultaneamente.

- Classificação hierárquica local ou *Top-Down*: em tal abordagem, a hierarquia de classes é processada um nível de cada vez, produzindo um ou mais classificadores para cada nível da hierarquia, durante a fase de treinamento do algoritmo. O classificador raiz é treinado com todos os exemplos de treinamento, logo, nos próximos níveis da hierarquia, um classificador é treinado usando apenas exemplos pertencentes às classes preditas anteriormente na hierarquia.

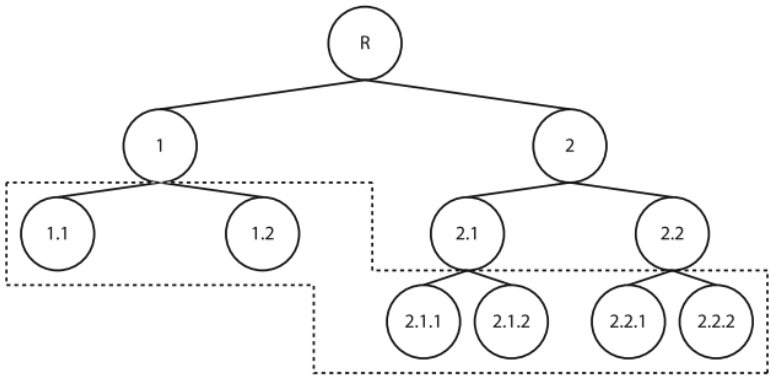
- Classificação hierárquica global ou *One-Shot*: a abordagem consiste em um modelo de classificação criado considerando a hierarquia de classes em sua totalidade, com uma única iteração do algoritmo de indução. Essa abordagem apresenta uma complexidade de implementação maior, mas evita problemas de propagação de erros da abordagem *Top-Down*.

Dessa forma, a seguir serão descritos algumas abordagens que permitem realizar as soluções supracitadas.

3.2.3.5.1 ABORDAGEM *FLAT* (CANÔNICA) A abordagem de classificação *flat* (canônica) consiste em uma forma simples de tratar problemas de classificação hierárquica. Fundamenta-se em ignorar completamente a hierarquia de classes, predizendo apenas as classes nos nós folhas. Sendo assim, a abordagem permite trabalhar com algoritmos tradicionais durante o treinamento e teste, fornecendo uma solução indireta para o problema de hierarquia, considerando que todas as classes ancestrais também são atribuídas implicitamente a essa instância (SILLA; FREITAS, 2010).

Logo, a Figura 8 demonstra um exemplo de problemática hierárquica, sendo aplicada a abordagem *flat*.

Figura 8: Exemplo de classificação *flat* [4]



Fonte: Silla e Freitas (2010)

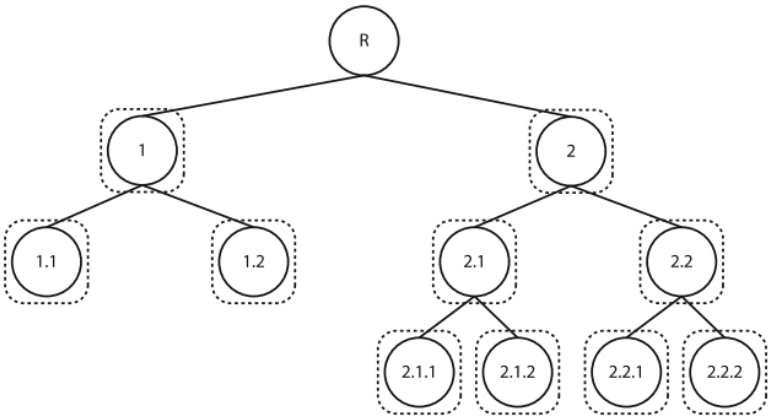
3.2.3.5.2 ABORDAGEM *LOCAL* Em contrapartida, a abordagem local permite a construção de classificadores independentes em cada nível da hierarquia, empregando apenas informações locais de classes e dados de treinamento. Isto posto, cada classificador é induzido nos exemplos anexados ao rótulo que ele representa e desconhece outros classificadores locais (PARMEZAN et al., 2022). Existem três maneiras diferentes de usar informações locais para construir classificadores e vinculá-los hierarquicamente:

3.2.3.5.2.1 CLASSIFICAÇÃO LOCAL POR NÓ (LCN) A metodologia LCN baseia-se no treinamento de um classificador binário para cada nó da hierarquia, exceto o nó raiz, selecionando exemplos positivos que representam a classe do nó atual e negativos para as demais classes (PARMEZAN et al., 2022).

A Figura 9 ilustra a aplicabilidade da abordagem LCN em um problema hierárquico e destaca os classificadores binários locais que devem ser considerados na estrutura.

[4] O retângulo tracejado representa um classificador multi-classe.

Figura 9: Exemplo de classificação hierárquica local por nó (LCN) [5]



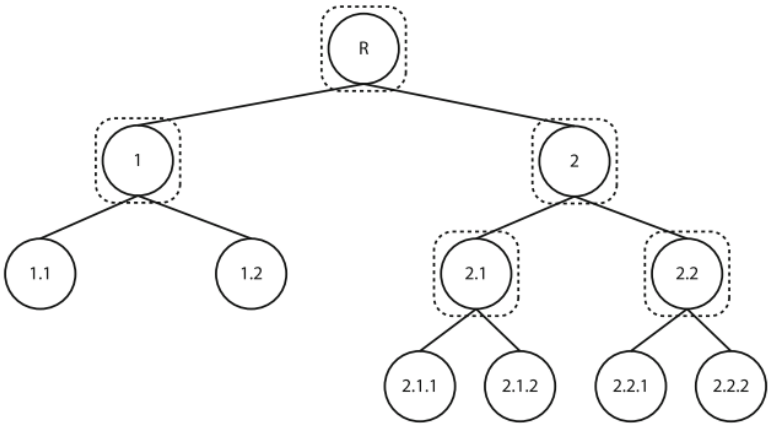
Fonte: Silla e Freitas (2010)

3.2.3.5.2.2 CLASSIFICAÇÃO LOCAL POR NÓ PAI (LCPN) A lógica do método LCPN constitui-se na geração de um classificador multi-classe para cada nó não folha da hierarquia, distinguindo entre suas classes filhas. Tal abordagem compõem-se de um conjunto de dados local com exemplos de classes filhas e seus descendentes. Cada subclasse, no entanto, deve ser generalizada para estar presente somente os rótulos referentes às classes filhas do nó analisado (PARMEZAN et al., 2022).

Na Figura 10 demonstra a lógica supracitada, destacando os classificadores multi-classes em cada nó pai, que consequentemente, possuem exemplos com as classes filhas.

[5] Cada retângulo tracejado representa um classificador binário.

Figura 10: Exemplo de classificação hierárquica local por nó pai (LCPN) [6]

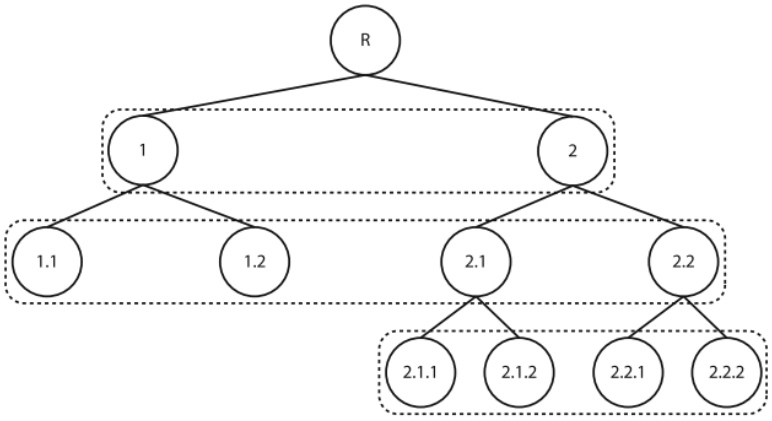


Fonte: Silla e Freitas (2010)

3.2.3.5.2.3 CLASSIFICAÇÃO LOCAL POR NÍVEL (LCL) Ademais, a abordagem LCL constrói um classificador multi-classe para cada nível da hierarquia, exceto o mais superficial (nível do nó raiz). Os conjuntos de dados locais de treinamento dos classificadores são determinados utilizando exemplos de classes para cada nível da hierarquia (PARMEZAN et al., 2022).

A Figura 11 exemplifica tal abordagem, detalhando os classificadores com suas respectivas classes em cada nível.

Figura 11: Exemplo de classificação hierárquica local por nível (LCL) [7]



Fonte: Silla e Freitas (2010)

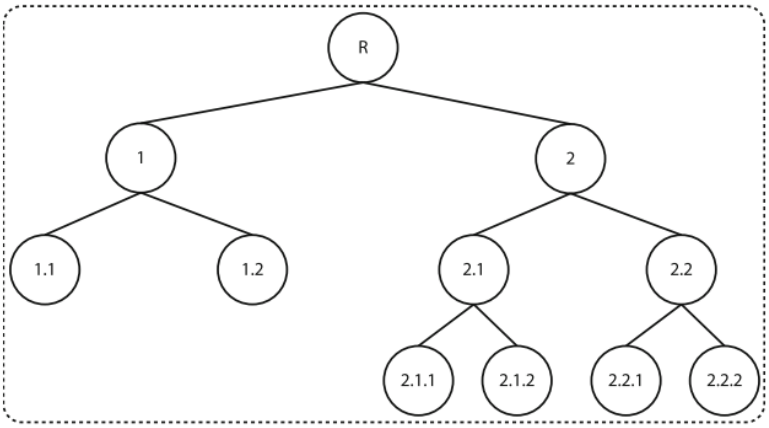
[6] Cada retângulo tracejado representa um classificador multi-classe.
[7] Cada retângulo tracejado representa um classificador multi-classe.

3.2.3.5.3 ABORDAGEM *GLOBAL* Por fim, na abordagem *global* (*Big-Bang*), um modelo de classificação é criado considerando toda a hierarquia de classes, apresentando uma maior complexidade de implementação. Logo, após o treinamento de um modelo de classificação utilizando essa metodologia, a predição é realizada em um único passo (FREITAS; CARVALHO, 2007).

Ao contrário das demais abordagens, nessa metodologia não se pode ser utilizada as técnicas de classificação canônicas em sua forma padrão, sendo necessário a realização de modificações no modo como é feita a indução do classificador, para considerar toda a hierarquia. Todavia, tal abordagem evita obter propagação de erros em níveis superiores para os níveis mais específicos da hierarquia (FREITAS; CARVALHO, 2007).

Um exemplo de aplicação da metodologia de classificação global em um problema hierárquico pode ser representado pela Figura 12.

Figura 12: Exemplo de classificação hierárquica global (*Big Bang*) [8]



Fonte: Silla e Freitas (2010)

3.2.4 AVALIAÇÃO

Um modelo de aprendizado de máquina deve possuir um bom desempenho de forma generalizada e confiável. Dessa forma, é necessário utilizar técnicas de validação que permitem metrificar tal desempenho.

[8] O retângulo tracejado representa um classificador global para toda a hierarquia de classes.

3.2.4.1 VALIDAÇÃO CRUZADA

Pretende-se que um modelo de *machine learning* seja generalizado, ou seja, que o seu desempenho e comportamento sejam os desejáveis com dados desconhecidos. Logo, a validação cruzada se torna uma prática muito interessante de ser utilizada para analisar esse propósito definido. Tal método consiste no treinamento e teste em diferentes subconjuntos de dados, identificando o desempenho do modelo em diferentes cenários (BROWNE, 2000).

3.2.4.1.1 K-FOLD A técnica de *k-fold* é uma das mais utilizadas no esquema de validação cruzada. Os dados são divididos aleatoriamente em *k* subconjuntos de tamanhos iguais. Para cada iteração, um desses subconjuntos é utilizado para o teste do modelo e o restante se tornam o conjunto de treinamento do mesmo. Tal processo é repetido *k* vezes, com cada subconjunto sendo o conjunto de teste designado uma vez. Por fim, é calculado a média dos *k* conjuntos de teste para fins de avaliação. Os valores mais comuns para *k* são 3, 5 e 10 (LIU, 2017).

O Quadro 6 ilustra a configuração para cinco dobras:

Quadro 6: Exemplo divisão de dados para a técnica *K-Fold*.

Iteração	Conjunto 1	Conjunto 2	Conjunto 3	Conjunto 4	Conjunto 5
1	Teste	Treinamento	Treinamento	Treinamento	Treinamento
2	Treinamento	Teste	Treinamento	Treinamento	Treinamento
3	Treinamento	Treinamento	Teste	Treinamento	Treinamento
4	Treinamento	Treinamento	Treinamento	Teste	Treinamento
5	Treinamento	Treinamento	Treinamento	Treinamento	Teste

Fonte: Adaptado de Liu (2017)

Tal técnica pode ser utilizada tanto para encontrar o melhor ajuste quanto como método de avaliação de desempenho e análise estatística. Também é comum o seu uso de forma aninhada, ou seja, uma combinação de validações cruzadas (LIU, 2017).

3.2.4.2 MÉTRICAS DE AVALIAÇÃO

O objetivo de um algoritmo de aprendizado de máquina é aprender com dados de treinamento para prever rótulos de classes de dados desconhecidos. Por esse motivo, os dados são divididos entre conjuntos de treino e teste. Logo, com o conjunto de teste, é realizado a validação do desempenho do modelo treinado. Nessa etapa, os dados de validação fornecem uma avaliação imparcial do modelo desenvolvido (THARWAT, 2020).

Seguindo essa linha de pensamento, as métricas ajudam a entender o desempenho de um classificador. Tais métricas servem para compreensão do desempenho de um modelo e podem ser utilizadas para avaliação e comparação entre diferentes classificadores, logo, um estudo deve apresentar várias métricas de avaliação (LEVER et al., 2016).

Existem várias maneiras de avaliar o desempenho de algoritmos de aprendizagem e os classificadores que eles produzem. Essas medidas de qualidade são construídas a partir de uma matriz de confusão que registra correta e incorretamente exemplos reconhecidos para cada classe (SOKOLOVA et al., 2006), onde:

- TP: representa a quantidade de verdadeiros positivos;
- FP: representa a quantidade de falso positivos;
- FN: representa a quantidade de falso negativos;
- TN: representa a quantidade de verdadeiros negativos.

Partindo desse princípio, várias métricas possuem o intuito de medir a eficiência de um modelo de aprendizado de máquina, nas quais são destacadas nas subseções a seguir.

3.2.4.2.1 ACURÁCIA A acurácia é uma métrica de avaliação que revela a precisão do modelo para todas as amostras (YIN et al., 2021), a sua fórmula é representada

pela Equação 1:

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

3.2.4.2.2 PRECISÃO A precisão representa a proporção de amostras positivas classificadas corretamente em relação ao número total de amostras preditas positivas (THARWAT, 2020), expressa pela Equação 2:

$$\frac{TP}{TP+FP} \tag{2}$$

3.2.4.2.3 REVOCAÇÃO A revocação mede quantos exemplos positivos são corretamente previstos como positivos (YIN et al., 2021), tal como mostrado na Equação 3:

$$\frac{TP}{TP+FN} \tag{3}$$

3.2.4.2.4 SENSIBILIDADE Trata-se da proporção de verdadeiros positivos corretamente identificados pelo teste (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019), conforme Equação 4:

$$\frac{TP}{TP+TN} \tag{4}$$

3.2.4.2.5 ESPECIFICIDADE Composto pela proporção de verdadeiros negativos corretamente identificados pelo teste (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019), conforme Equação 5:

$$\frac{TN}{FN+TN} \tag{5}$$

3.2.4.2.6 *F1-SCORE* O *F1-Score* é uma métrica que indica a relativa estabilidade do modelo para amostras positivas e negativas (YIN et al., 2021). Sua fórmula utiliza

os conceitos de precisão e revocação supracitados, notado pela Equação 6:

$$\frac{2 \cdot (precisao \cdot revocacao)}{precisao + revocacao} \tag{6}$$

3.2.4.2.7 CURVA ROC E AUC A curva *Receiver operating characteristic* (*ROC*) é um gráfico bidimensional que tem sido utilizado para avaliação de sistemas, como o aprendizado de máquina. O eixo *y* representa a taxa de verdadeiros positivos e o eixo *x* representa a taxa de falsos positivos. É utilizado para encontrar um equilíbrio entre benefícios estipulados pela taxa de verdadeiros positivos e custos definidos pela taxa de falsos positivos (THARWAT, 2020).

Realizar essa comparação muitas vezes não se torna uma tarefa fácil. Isso ocorre porque não há um valor escalar que represente o desempenho esperado. Portanto, a métrica *Area Under the Curve* (*AUC*) é utilizada para calcular a área sob a curva *ROC*, onde a pontuação é sempre limitada entre zero e um, com os melhores valores sendo próximos a um (THARWAT, 2020). A Equação 7 mostra como é realizada essa métrica:

$$\frac{1}{2} \cdot (revocacao + \frac{TN}{TN + FP}) \tag{7}$$

3.2.5 EXPLICABILIDADE E INTERPRETABILIDADE

A necessidade de ter modelos com alta acurácia e compreensíveis impulsiona o *eXplainable Artificial Intelligence* (XAI). Nesse paradigma é importante a criação de ferramentas preditivas que dão sugestões e percepções a um usuário humano para assim ser o responsável de realizar um processo de alto impacto (MARIOTTI et al., 2022).

Dessa forma, métodos que permitem a interpretabilidade de modelos de *machine learning* foram desenvolvidas, conforme o método apresentado a seguir.

3.2.5.1 SHAPLEY ADDITIVE EXPLANATIONS (SHAP)

Desenvolvido por Lundberg e Lee (2017), o método SHAP permite explicar previsões individuais baseado em valores de Shapley ótimos da teoria de jogos. Dessa forma, é possível aplicar tal método para interpretação de modelos de *machine learning* identificando a contribuição de cada recurso para a previsão.

O algoritmo parte da definição que os métodos de atribuição de recursos aditivos têm um modelo de explicação que é uma função linear de variáveis binárias, conforme a equação:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{8}$$

Sendo z' ε $\{0, 1\}^{M}$, M é o número de recursos de entrada simplificados e ϕ_i ε $\mathbb{R}$.

Essa definição implica que os métodos atribuem um efeito ϕ a cada recurso e a soma de todas as atribuições dos recursos se aproximam do resultado de saída do modelo original.

Para tal é desejado que o método corresponda há algumas propriedades como:

- Precisão local: ao aproximar o modelo original f(x) para uma entrada específica de x, a precisão local requer que o modelo de explicação corresponda a entrada original.
- Ausência: se a simplificação representa a presença de um recurso, logo a omissão da mesma requer que os recursos ausentes na entrada original não tenham impacto.
- Consistência: se um modelo mudar para que a contribuição de uma entrada simplificada aumente ou permaneça independente das outras entradas, a atribuição dessa entrada não deve diminuir.

Dessa forma, o método proposto por Lundberg e Lee (2017), propõe uma

medida unificada da importância de cada recurso utilizado por um modelo. O método consiste em obter os valores de Shapley em função de uma expectativa condicional do modelo original. O SHAP fornece a medida de importância de recurso aditivo exclusivo que adere às propriedades supracitadas e usa expectativas condicionais para definir entradas simplificadas.

4 EXPERIMENTOS

Para a realização de experimentos em problemas de subclasses de leucemia e validação do sistema e da metodologia proposta, realizaram-se alguns experimentos que permitiram validar o *pipeline*, analisar a eficácia dos modelos e obter ideias sobre a interpretabilidade dos modelos treinados.

Dessa forma, desenvolveu-se um *pipeline* que permite aplicar diferentes seleções de *features*, abordagens e técnicas de aprendizado de máquina, visando extrair a máxima precisão para projetos de detecção de multi-classes de leucemia utilizando dados de contagem de genes e com potencial de reduzir o número de *features* para melhor direção de análises futuras sobre os mesmos.

Assim sendo, todo o *pipeline* consiste em realizar todas as etapas propostas na Figura 3, detalhadas nas seções seguintes.

4.1 ENTENDIMENTO DO NEGÓCIO

Tal etapa consta em estudos sobre o problema, na qual foram discutidas nas seções anteriores, abordando conceitos teóricos sobre o tema, trabalhos correlatos, análise de metodologias, técnicas e abordagens presentes na área de ciência de dados com foco em aprendizado de máquina.

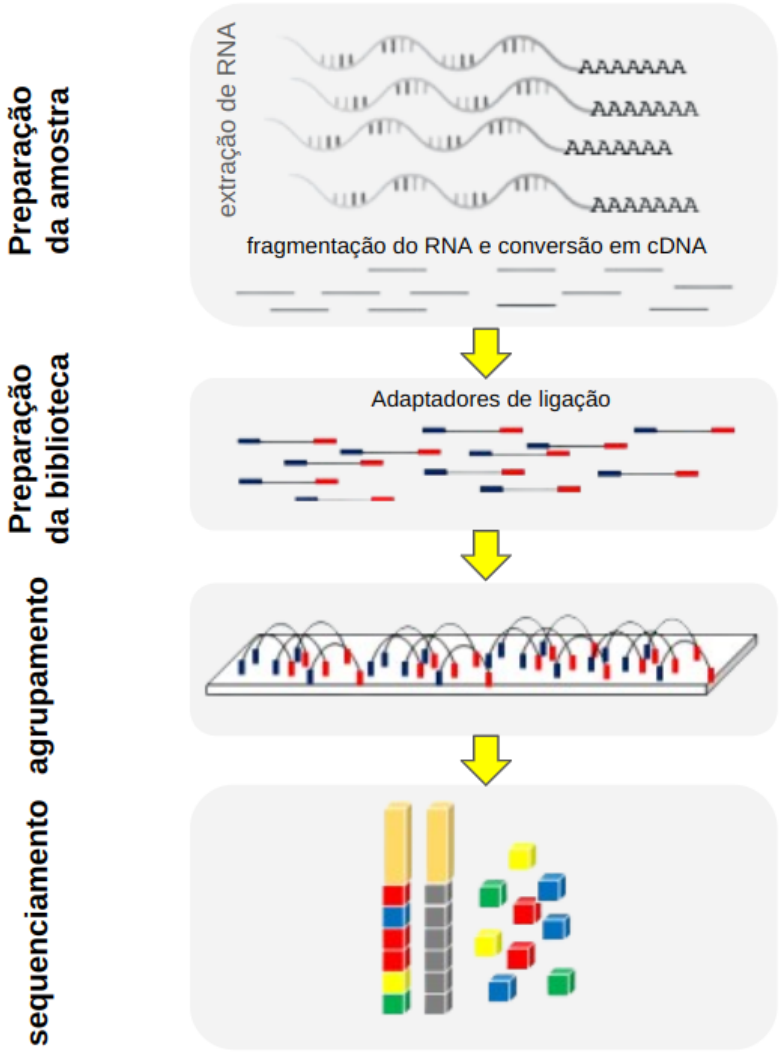
4.2 ENTENDIMENTO DOS DADOS

Dessa forma, pretende-se adquirir dados que permitam a classificação de multi-classes de leucemia, utilizando dados de contagem de genes.

Para a geração desse formato de conjunto de dados, primeiramente o RNA

deve ser extraído, purificado, quebrado em fragmentos curtos e convertido em DNA complementar (cDNA) por transcriptase reversa. Depois, os adaptadores são anexados e os fragmentos selecionados por tamanho, para assim ser realizado o sequenciamento, conforme expresso na Figura 13 (DINIZ; CANDURI, 2017).

Figura 13: Método de sequenciamento genético



Fonte: Adaptado de Diniz e Canduri (2017)

Em seguida, realiza-se o mapeamento de leitura (processo para alinhar as leituras em genomas de referência). Com esses alinhamentos são montados os transcritos e por fim realiza-se o processo de contagem e estimação de abundância dos genes/transcritos (DINIZ; CANDURI, 2017).

4.2.1 AQUISIÇÃO DE DADOS

Para validar o *pipeline*, entende-se que é necessário aplicá-lo a diferentes bases de dados de expressão gênica de leucemia. Portanto, foram selecionadas 6 bases de dados de diferentes repositórios: NCBI/GEO e CuMiDa (FELTES et al., 2019), já contendo os dados de contagem de genes e múltiplas classes de leucemia.

A descrição de cada banco de dados é mostrada na Tabela 1.

Tabela 1: Descrição das bases de dados

Projeto	Repositório	Amostras	*Features*	Classes
GSE87070	NCBI/Geo	654	54675	3
GSE13159	NCBI/Geo	2096	54675	18
GSE13164	NCBI/Geo	1152	1480	18
GSE9476	CuMiDa	64	22283	5
GSE71449	CuMiDa	45	52200	4
GSE28497	CuMiDa	281	22283	7

Fonte: Autoria Própria (2023)

4.2.2 ANÁLISE EXPLORATÓRIA DOS DADOS

Cada projeto dispõe de diferentes classes para serem classificadas com quantidades amostrais distintas. As classes dispostas de cada projeto e o número de amostras representadas respectivamente estão disponíveis no Apêndice A.

4.3 PREPARAÇÃO DOS DADOS

Essa etapa consiste em preparar os dados adquiridos para a utilização futura de técnicas de aprendizado de máquina em diferentes contextos de abordagens.

4.3.1 PRÉ-PROCESSAMENTO

Logo, identificou-se a integridade dos dados (através da limpeza das amostras com valores faltantes) e padronização dos tipos de dados e nomes das colunas para todas as bases de dados.

Como este experimento tem a finalidade de abordar diferentes metodologias de modelagem (canônica e hierárquica), é necessário estruturar os conjuntos de dados para cada abordagem.

4.3.1.1 MÉTODOS CANÔNICOS

A estrutura de dados segue a forma clássica de treinamento de modelos, onde são utilizadas colunas que identificam características (contagem de um gene específico) e uma coluna que identifica as classes às quais cada amostra pertence, conforme destacado na Figura 14.

Figura 14: Formato de conjunto de dados para abordagens canônicas (planas)

Sample	Features					Class
	X_1	X_2	X_3	...	X_m	
S_0	X_{10}	X_{20}	X_{30}	...	X_{m0}	Y_0
S_1	X_{11}	X_{21}	X_{31}	...	X_{m1}	Y_1
S_2	X_{12}	X_{22}	X_{32}	...	X_{m2}	Y_2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
S_n	X_{1n}	X_{2n}	X_{3n}	...	X_{mn}	Y_n

Fonte: Autoria Própria (2023)

4.3.1.2 MÉTODOS HIERÁRQUICOS

Ao analisar as classes de conjunto de dados, identificou-se a possibilidade de abordagens de modelagem hierárquica para os projetos GSE13159 e GSE13164 do repositório NCBI/GEO e GSE71449 do repositório CuMiDa, por apresentarem classes que possam ser organizadas em subgrupos que respeitam uma certa hierarquia biológica.

Portanto, é necessário tratar a estrutura de dados dos conjuntos para representar as amostras de forma hierárquica, conforme descrito na Figura 15, onde a coluna que representa a classe no conjunto de dados, é modificada e dividida em múltiplas colunas que representam cada nível da hierarquia. As amostras são decompostas em classes pertencentes a cada nível da hierarquia, onde caso a amostra não possua representação em determinado nível, este valor será imputado como nulo.

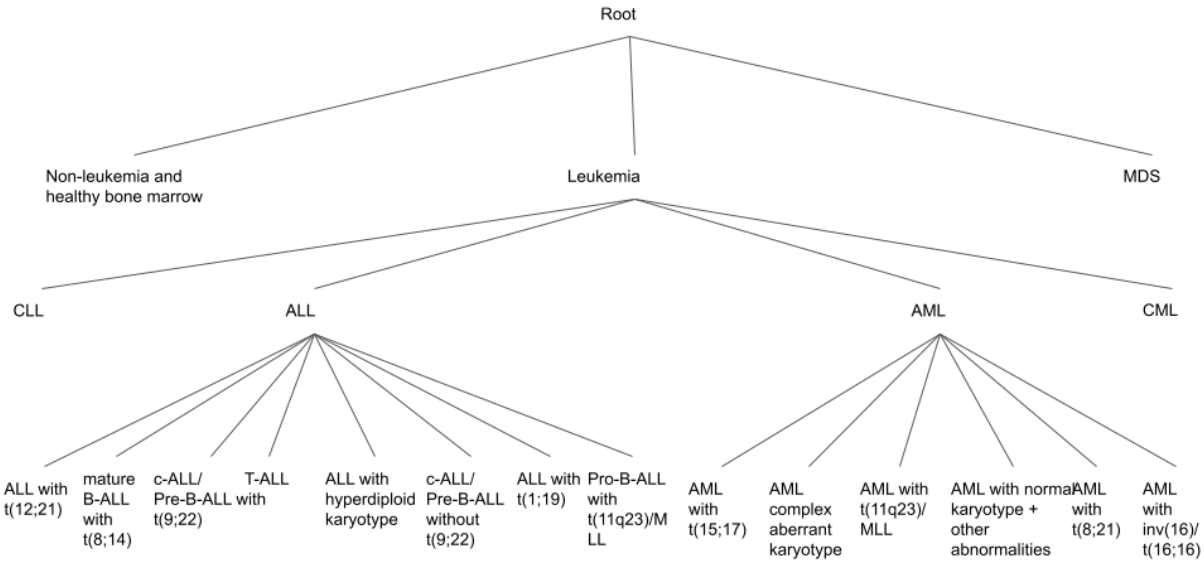
Figura 15: Formato de conjunto de dados para abordagens hierárquicas

Sample	Features					Class				
	X_1	X_2	X_3	...	X_m	Y_{L0}	Y_{L1}	Y_{L2}	...	Y_{Lz}
S_0	X_{10}	X_{20}	X_{30}	...	X_{m0}	Y_{L00}	Y_{L10}	Y_{L20}	...	Y_{Lz0}
S_1	X_{11}	X_{21}	X_{31}	...	X_{m1}	Y_{L01}	Y_{L11}	Y_{L21}	...	Y_{Lz1}
S_2	X_{12}	X_{22}	X_{32}	...	X_{m2}	Y_{L02}	Y_{L12}	Y_{L22}	...	Y_{Lz2}
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
S_n	X_{1n}	X_{2n}	X_{3n}	...	X_{mn}	Y_{L0n}	Y_{L1n}	Y_{L2n}	...	Y_{Lzn}

Fonte: Autoria Própria (2023)

Definida a estrutura de dados, foi desenhada a estrutura para cada projeto, representada nas Figuras 16 e 17.

Figura 16: Hierarquia representada para as classes dos conjuntos de dados dos projetos GSE13159 e GSE13164

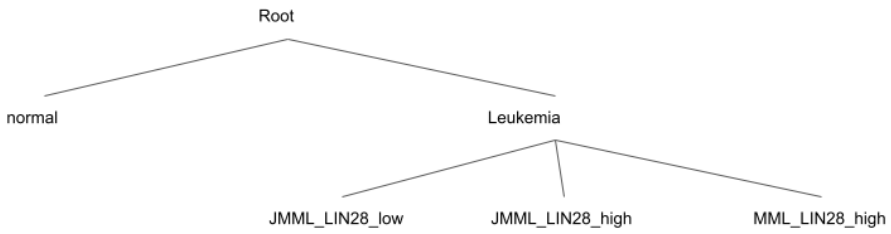


Fonte: Autoria Própria (2023)

A hierarquia presente na Figura 16 foi dividida em 4 níveis:

- Nível 0 (Raiz): Início da hierarquia;
- Nível 1 (Doença): Corresponde à categorização de determinado tipo de doença, como MDS (Síndrome Mielodisplásica), Leucemia ou Medula Óssea Saudável;
- Nível 2 (Tipo): Corresponde ao aprofundamento da doença (leucemia), como

Figura 17: Hierarquia representada para as classes dos conjuntos de dados do projeto GSE71449



Fonte: Autoria Própria (2023)

LLC (Leucemia Linfocítica Crônica), LLA (Leucemia Linfoblástica Aguda), LMA (Leucemia Mieloide Aguda) e LMC (Leucemia Mielóide Crônica).

- Nível 3 (Subtipo): Corresponde a um subtipo específico da doença que leva em consideração causas citogenéticas.

Em contraste, a hierarquia na Figura 17 foi dividida em 3 níveis:

- Nível 0 (Raiz): Início da hierarquia;
- Nível 1 (Doença): Corresponde à categorização de determinado tipo de doença, Leucemia ou saudável (normal);
- Nível 2 (Tipo): Corresponde a um tipo específico de leucemia;

Para os demais projetos não foram definidas estruturas hierárquicas, sendo utilizadas apenas técnicas de modelagem canônicas.

4.3.2 SELEÇÃO DE *FEATURES*

Dado o grande número de características presentes nos dados, pretende-se selecionar os atributos que produzam efeitos imediatos nas classificações, reduzindo o número de *features* a serem analisadas e mantendo um alto nível de precisão. Assim, foram definidas algumas estratégias de seleção, nas quais serão utilizadas de forma isolada e combinadas, permitindo avaliar a eficácia de sua utilização nas modelagens:

- Abordagem *Filter* baseada em correlação e variância: Esta estratégia consiste em analisar as características do conjunto de dados e formar grupos altamente correlacionados, e então para cada grupo selecionar uma característica que possui maior variância entre as amostras.
- Abordagem *Embedded*: Consiste em treinar um modelo de aprendizado de máquina e selecionar as *features* com o valor de importância maior que a média de todo o modelo treinado. Assim, o modelo é retreinado apenas com os recursos selecionados.

Para criar grupos correlacionados foi utilizado o coeficiente de correlação de Spearman, sendo definido o valor de 0,5 como limite para definição de correlação forte, conforme especificado por (COHEN, 1988).

4.4 MODELAGEM

Dada a diversidade de técnicas de aprendizado de máquina e possibilidades de redução de dimensionalidade de recursos, algumas técnicas foram selecionadas para serem abordadas e diferentes cenários de treinamento serão testados:

4.4.1 CENÁRIOS DE TREINAMENTO

No estágio de treinamento, combinações de métodos de aprendizado de máquinas e diferentes tipos de abordagens (canônicas e hierárquicas) são implementadas para os seguintes cenários de seleção de *features*:

- Todas as *features*: o modelo é treinado usando todas as *features* do conjunto de dados;
- *Features* selecionadas baseadas em correlação/variância: Os modelos são treinados usando recursos que não estão correlacionados entre si;
- *Features* selecionadas baseadas nas importâncias para os modelos: Baseia-se no retreinamento do modelo com base nos recursos que são mais importantes para o modelo (valor de importância acima da média);
- *Features* selecionadas com base em correlação/ variância e importância para os modelos: Consiste em treinar modelos usando recursos selecionados que não estão correlacionados entre si e retreinar o modelo adequadamente com base nos recursos mais importantes.

4.4.2 DEFINIÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA

Foram definidas possíveis técnicas de aprendizado de máquina a serem adotadas para modelos de treinamento baseados em trabalhos relacionados e a utilização de diferentes escopos de aprendizado de máquina.

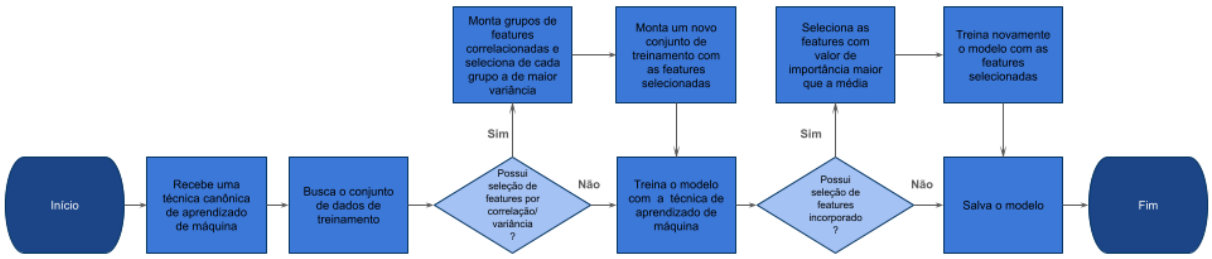
Ressalta-se que tais metodologias foram escolhidas por possuírem uma forma de encontrar a importância das características para o modelo (seja pela importância de Gini para modelos baseados em árvores ou por coeficientes para modelos lineares), uma vez que tais informações serão utilizadas para a construção de modelos de seleção de *features* com a abordagem *embedded*. Portanto, as técnicas definidas foram: Decision Tree, Random Forest, SVM e AdaBoosting.

4.4.2.1 *PIPELINE* DE ABORDAGEM CANÔNICA

Em abordagens canônicas, o *pipeline* de treinamento consiste em definir uma determinada técnica de aprendizado de máquina, buscar os dados de treinamento (originais ou pré-processados com seleção de *features* baseado em correlação/variância) e treinar o modelo. Caso haja a seleção de *features embedded*,

o modelo é retreinado com as *features* com importância para o modelo maior que a média. Tais etapas são ilustradas na Figura 18.

Figura 18: *Pipeline* de treinamento com a metodologia canônica (*flat*)



Fonte: Autoria Própria (2023)

4.4.2.2 *PIPELINE* DE ABORDAGEM HIERÁRQUICA

Para as abordagens hierárquicas aplicou-se o método de Classificador Local por Nó Pai (LCPN), gerando um classificador multi-classe para cada nó não folha da hierarquia, sendo definido a mesma técnica de aprendizado de máquina para todos os nós classificadores em uma hierarquia.

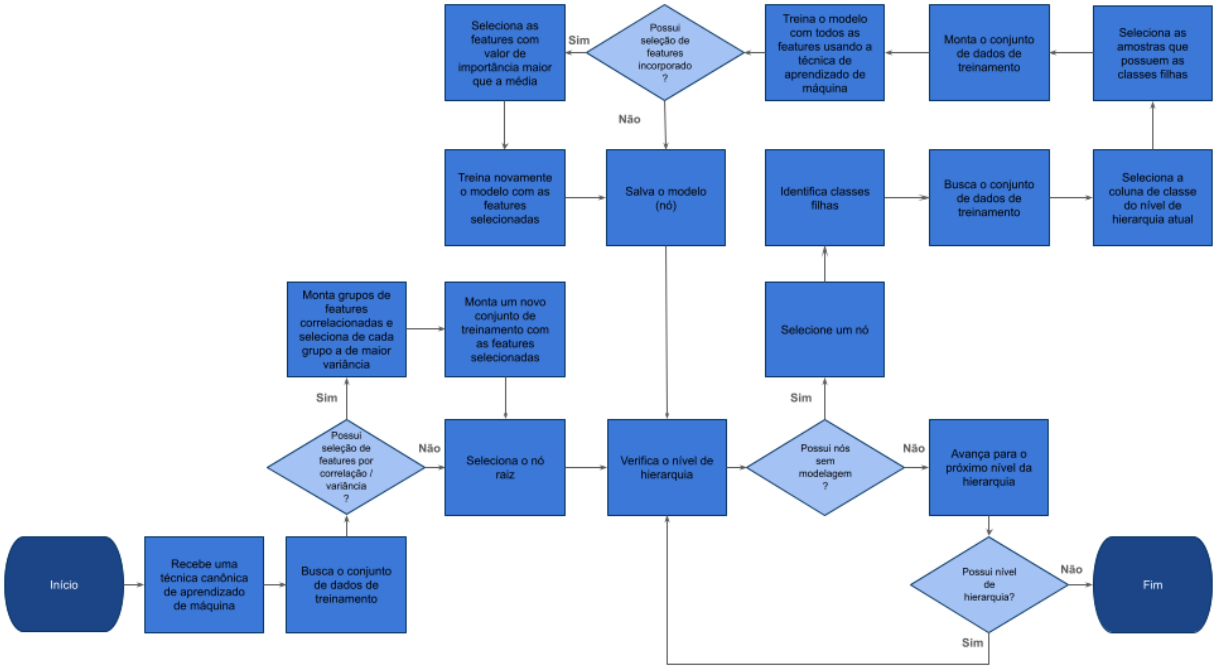
Essa implementação foi escolhida devido tais aspectos:

- A abordagem é intuitiva;
- Permite aplicação de qualquer algoritmo canônico;
- Preserva informações sobre a hierarquia de dados;
- Menor complexidade de implementação comparado a uma abordagem *Big-Bang*;

- Não apresentar problemas de inconsistências entre os níveis de hierarquia, que podem ocorrer na abordagem de classificação local por nível (LCL);
- Apresentar uma quantidade menor de treinamento de modelos comparado a abordagem de classificação local por nó (LCN);

Sendo assim, o *pipeline* desenvolvido consiste também em definir uma técnica de aprendizado de máquina canônica, selecionar os dados de treinamento (pré-processados por correlação/variância ou não) e assim montar uma hierarquia de treinamento de modelos conforme a hierarquia definida para o projeto. Na hierarquia, em cada nó não folha é treinado um modelo de aprendizado de máquina utilizando como classes preditivas os nós filhos. Cada nó pode realizar retreinamento ou não com as *features* mais importantes que a média do modelo do nó corrente. Toda arquitetura desenvolvida é esclarecida através da Figura 19.

Figura 19: *Pipeline* de treinamento com a metodologia canônica



Fonte: Autoria Própria (2023)

Na identificação do número de *features* utilizadas pela abordagem, são contadas as *features* distintas entre cada modelo treinado em cada nó da hierarquia.

Informações sobre as configurações usadas para cada técnica de aprendizado de máquina podem ser visualizadas no Apêndice B.

4.4.3 VALIDAÇÃO CRUZADA

Para a realização da validação cruzada, a técnica *k-fold* é aplicada de forma estratificada para manter as proporções das classes em cada subconjunto. O valor 3 foi escolhido para o termo *k* (mesmo valor abordado no trabalho relacionado ao projeto CuMiDa (FELTES et al., 2019)).

Para cada subconjunto foram calculadas as métricas de acurácia, precisão, revocação e F1-score.

Como os conjuntos de dados possuem multi-classes, precisa-se de uma forma de agregar as métricas para cada resultado obtido em cada classe. Assim, é feita uma média ponderada para as métricas de precisão, revocação e F1-score, levando em consideração o número de amostras presentes para cada classe estimada, evidenciado pela Equação 9, onde *p* é o número de amostras e *x* é o valor obtido para a métrica em uma determinada classe.

$$\bar{x} = \frac{\sum_{i=1}^{n}(x_i \cdot p_i)}{\sum_{i=1}^{n} p_i} \tag{9}$$

4.5 AVALIAÇÃO

Por fim, após a realização dos experimentos utilizando todas as combinações de abordagens, seleções de recursos e técnicas de aprendizado de máquina, são obtidos os valores médios de cada métrica de validação cruzada.

No total foram treinados 144 modelos diferentes, onde os resultados de desempenho podem ser visualizados no Apêndice C.

4.6 IMPLEMENTAÇÃO

Após a realização dos experimentos e homologação dos resultados, criou-se uma ferramenta que possibilita realizar inferências para dados desconhecidos e interpretabilidade dos modelos. A ferramenta será abordada no capítulo 6.

5 RESULTADOS E DISCUSSÕES

Após a aplicação dos experimentos realizados, foi-se analisado o seu desempenho em cada projeto e depois comparado com os resultados da literatura.

5.1 RESULTADOS DOS EXPERIMENTOS

Para encontrar a melhor abordagem para cada projeto com base nos resultados das métricas realizadas no experimento, foram utilizados os seguintes critérios de comparação:

- foram identificados os modelos que obtiveram as melhores métricas de desempenho (acurácia, precisão, revocação e F1-score), seguindo esta respectiva ordem para critérios de desempate.
- se os modelos obtiverem os mesmos resultados de métricas de desempenho, o modelo que tiver o menor número de *features* selecionados será selecionado (maior otimização de *features*).

Portanto, a Tabela 2 apresenta os modelos (técnica [SVM - Support Vector Machine, RF - Random Forest e AB - AdaBoosting], abordagem [C - Canônico e H - Hierárquico] e técnica de seleção de características) com os melhores resultados selecionados, ou seja, a métrica utilizada é o valor máximo.

Conforme os resultados obtidos, pode-se identificar que a metodologia utilizada propôs a criação de modelos de aprendizado de máquina que apresentam desempenho promissor de classificação multi-classe para projetos que utilizam expressão gênica de leucemia, pois resultados de acurácia são encontrados entre 0,835 a 1.

Tabela 2: Melhores resultados por conjunto de dados

Projeto (GSE)	Seleção Feature	Técnica (Abordagem)	ACC	PRE	REC	F1	Redução Features
87070	Embedded	SVM (C)	0,991	0,991	0,991	0,991	66,31%
13159	Ambas	SVM (C)	0,877	0,881	0,877	0,877	91,70%
13164	Embedded	RF (H)	0,835	0,838	0,835	0,823	41,42%
9476	Ambas	SVM (C)	1,000	1,000	1,000	1,000	98,32%
71449	Embedded	AB (H)	0,889	0,877	0,889	0,877	99,99%
28497	Embedded	SVM (C)	0,907	0,900	0,907	0,899	66,82%

Fonte: Autoria Própria (2023)

Vale destacar também que em termos de tratamento de características, em todos os projetos que apresentam os melhores resultados possuem abordagens que ao menos realizam uma forma de tratamento de seleção de características, tanto utilizando o método de seleção de *features* incorporadas, como também utilizando a estratégia de filtro combinada com correlação/variância.

Ressalta-se que a utilização de tais metodologias, além de implicar em melhores resultados, também apresenta otimização do número de funcionalidades utilizadas pelos modelos, destacando-se a redução de funcionalidades entre 41,42% e chegando até 99,99%. Além dos ganhos na redução do processamento computacional dos modelos para inferências futuras, esta abordagem determina um escopo menor de *features* para possíveis análises e estudos sobre a interpretabilidade de cada uma para a classificação de uma determinada classe.

Levando em consideração que a utilização da abordagem hierárquica foi possível de ser realizada em 3 projetos diferentes (GSE13159, GSE13164 e GSE71449), e apresentou os melhores resultados em 2 deles (GSE13164 e GSE71449), pode-se considerar que a utilização de tal abordagem é promissora em questões onde se é possível estimar uma hierarquia entre classes. Vale ressaltar também que a utilização desta abordagem apresentou resultados significativos aliada ao uso de técnicas de aprendizagem em conjunto (Random Forest e Adaboosting).

Em contrapartida, a técnica SVM aplicada de forma canônica apresentou os melhores resultados na maioria dos experimentos dos projetos estudados, destacando-se sua utilização para projetos de classificação multi-classe de leucemia, que também foram realizados em trabalhos relacionados.

5.2 COMPARAÇÃO COM A LITERATURA

Para os projetos extraídos da base de dados CuMiDa, têm-se os resultados desenvolvidos pelo próprio projeto (FELTES et al., 2019), abordando diferentes técnicas de aprendizado de máquina. Logo, a Tabela 3 faz o paralelo entre os resultados obtidos pelo projeto Cumida e compara os resultados com o melhor modelo desenvolvido neste trabalho.

Tabela 3: Comparação dos resultados entre o melhor modelo desenvolvido com os modelos desenvolvidos pelo projeto CuMiDa

Projeto (GSE)	Técnica	Abordagem	Autor	ACC	PRE	REC	F1
9476	ZERROR	Canônico	CuMiDa	0.406	?	0.406	?
	SVM	Canônico	CuMiDa	**0.984**	0.986	**0.984**	**0.985**
	MLP	Canônico	CuMiDa	0.938	0.950	0.938	0.938
	DT	Canônico	CuMiDa	0.891	0.909	0.891	0.885
	NB	Canônico	CuMiDa	0.891	0.914	0.891	0.889
	RF	Canônico	CuMiDa	**0.984**	0.985	**0.984**	0.984
	KNN	Canônico	CuMiDa	0.891	0.906	0.891	0.890
	SVM	**Canônico**	**Próprio**	**0.984**	**0.988**	**0.984**	**0.985**
71449	ZERROR	Canônico	CuMiDa	0.444	?	0.444	?
	SVM	Canônico	CuMiDa	0.578	?	0.578	?
	MLP	Canônico	CuMiDa	0.422	?	0.422	?
	DT	Canônico	CuMiDa	0.711	?	0.711	?
	NB	Canônico	CuMiDa	0.489	?	0.489	?
	RF	Canônico	CuMiDa	0.378	?	0.378	?
	KNN	Canônico	CuMiDa	0.378	0.473	0.378	0.416
	Adaboosting	**Hierárquico**	**Próprio**	**0.889**	**0.877**	**0.889**	**0.877**
28497	ZERROR	Canônico	CuMiDa	0.263	?	0.263	?
	SVM	Canônico	CuMiDa	0.879	0.881	0.879	0.868
	MLP	Canônico	CumiDa	0.719	0.743	0.719	0.682
	DT	Canônico	CuMiDa	0.733	0.739	0.733	0.734
	NB	Canônico	CuMiDa	0.776	0.771	0.776	0.769
	RF	Canônico	CuMiDa	0.794	?	0.794	?
	KNN	Canônico	CuMiDa	0.701	0.711	0.701	0.702
	SVM	**Canônico**	**Próprio**	**0.904**	**0.894**	**0.904**	**0.894**

Fonte: Autoria Própria (2023)

Pode-se notar que o melhor modelo desenvolvido para cada projeto com base na metodologia proposta apresenta os melhores resultados comparado a todas as abordagens realizadas pelo projeto CuMiDa. Entende-se que para o projeto GSE9476, os valores de acurácia, revocação e F1-score possuem o mesmo valor com base

na técnica SVM, porém o modelo desenvolvido possui o valor de precisão mais preciso. Para os demais projetos (GSE74449 e GSE28497), o modelo apresenta melhor desempenho em todas as métricas.

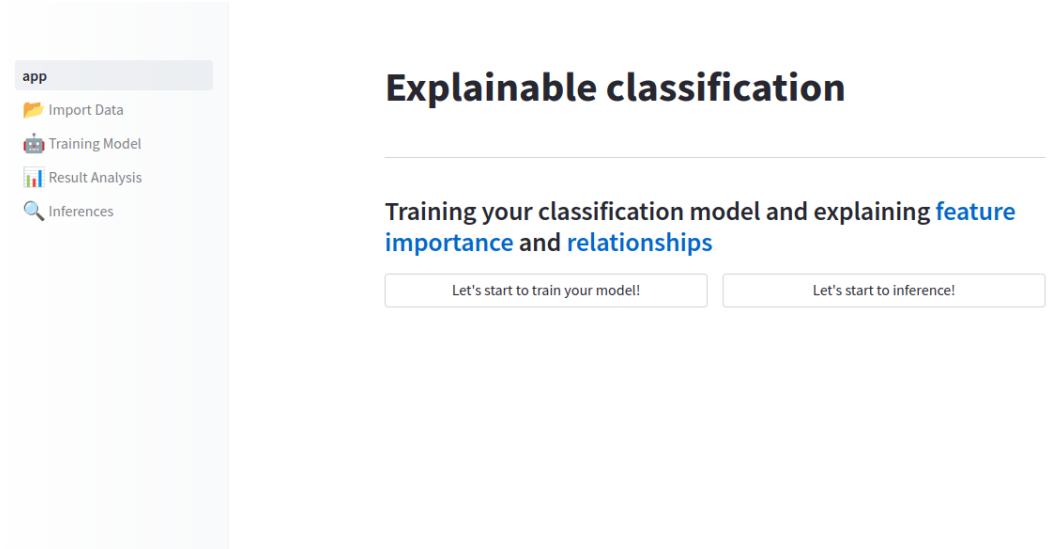
Vale ressaltar que os modelos possuem métricas melhores do que modelos baseados em diversas técnicas diferentes não abordadas neste trabalho, como redes neurais, modelos bayesianos e algoritmos de regra zero (FELTES et al., 2019).

6 SISTEMA DE TREINAMENTO E INTERPRETABILIDADE DE MODELOS CLASSIFICADORES

Este trabalho tem por objetivo a criação de uma ferramenta para aplicação de aprendizado de máquina interpretável para identificação de subtipos de leucemia. Conforme os resultados promissores obtidos nos experimentos realizados, criou-se uma aplicação que permite a realização do treinamento de um modelo de *machine learning* conforme as abordagens exploradas anteriormente, além de obter opções de visualização que permitem melhor interpretabilidade das decisões do modelo.

Na Figura 20, é apresentada a tela inicial da aplicação que está implantada *online* [9]. A ferramenta dispõe de 4 funcionalidades nas quais serão destacadas nas seções subsequentes.

Figura 20: Tela inicial da aplicação [10]



Fonte: Autoria Própria (2023)

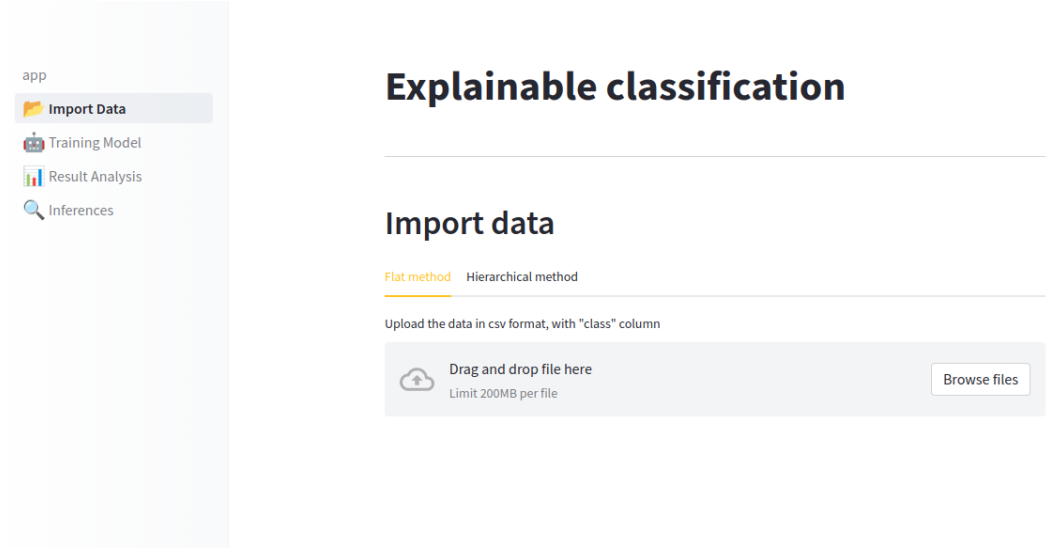
[9] https://huggingface.co/spaces/alvaropq/explainable_classification.

[10] Na tela é mostrada as etapas para o treinamento e análise de um modelo (*Import Data*, *Training Model*, *Result Analysis*) ou realizar uma inferência caso já tenha um modelo treinado (*Inferences*).

6.1 IMPORTAÇÃO DOS DADOS

Essa etapa consiste em realizar a importação dos dados que serão utilizados para o treinamento do modelo. Tal etapa necessita que o usuário padronize os dados conforme os requisitos da aplicação, logo, toda preparação de dados deve ser realizada fora da interface. Na Figura 21, é apresentada a tela de importação de dados.

Figura 21: Tela de importação de dados da aplicação [11]



Fonte: Autoria Própria (2023)

Essa etapa possui duas opções, a *Flat method* e *Hierarchical method*, onde o usuário pode escolher utilizar uma metodologia canônica (*flat*) ou hierárquica.

6.1.1 FLAT METHOD

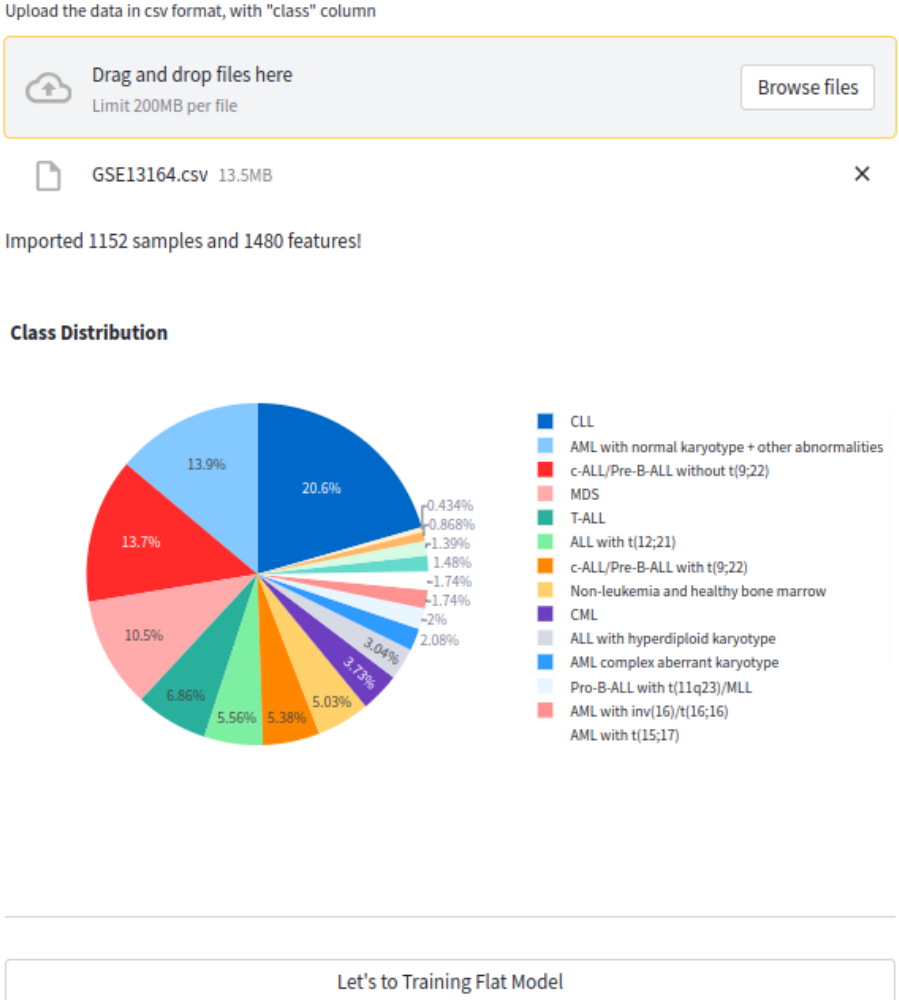
Essa opção permite receber dados para a criação de modelos no estilo canônico, onde é necessário importar os dados em arquivo de extensão *csv*, contendo as colunas com as *features* e uma coluna com as classes (com nome ”*class*”), conforme o exemplo genérico da Figura 14 apresentada anteriormente.

Ao importar os dados, é apresentado a quantidade de amostras e *features* importadas. Logo mais, um gráfico é gerado mostrando as distribuições de amostras

[11] Pode-se importar dados para metodologias de modelagem canônica (*flat*) ou hierárquica.

por classe, conforme o exemplo na Figura 22.

Figura 22: Tela com dados importados da opção *Flat Method* [12]



Fonte: Autoria Própria (2023)

6.1.2 HIERARCHICAL METHOD

Pode-se modelar um problema de classificação em formato hierárquico, onde os valores das amostras passam por um *pipeline* de modelos que determinam um fluxo na qual será definido sua classe final.

Dessa forma, essa opção permite receber dados para a criação de modelos em tal estilo, onde primeiramente é necessário importar um arquivo em formato de extensão *json*, apresentando o desenho da hierarquia dos dados. O arquivo consiste

[12] O exemplo consiste em dados de subclasses de leucemia do projeto GSE13164. Ao mostrar as informações dos dados, pode-se avançar para a próxima etapa de treinamento do modelo.

em expressar os níveis de hierarquia, obtendo para cada subnível, relações existentes com os valores do nível superior. A Figura 23 apresenta um exemplo de importação do arquivo de configuração de hierarquia, baseado na hierarquia definida pela Figura 16 apresentada anteriormente.

Figura 23: Tela com o arquivo de declaração de hierarquia em *Hierarchical Method* [13]

Fonte: Autoria Própria (2023)

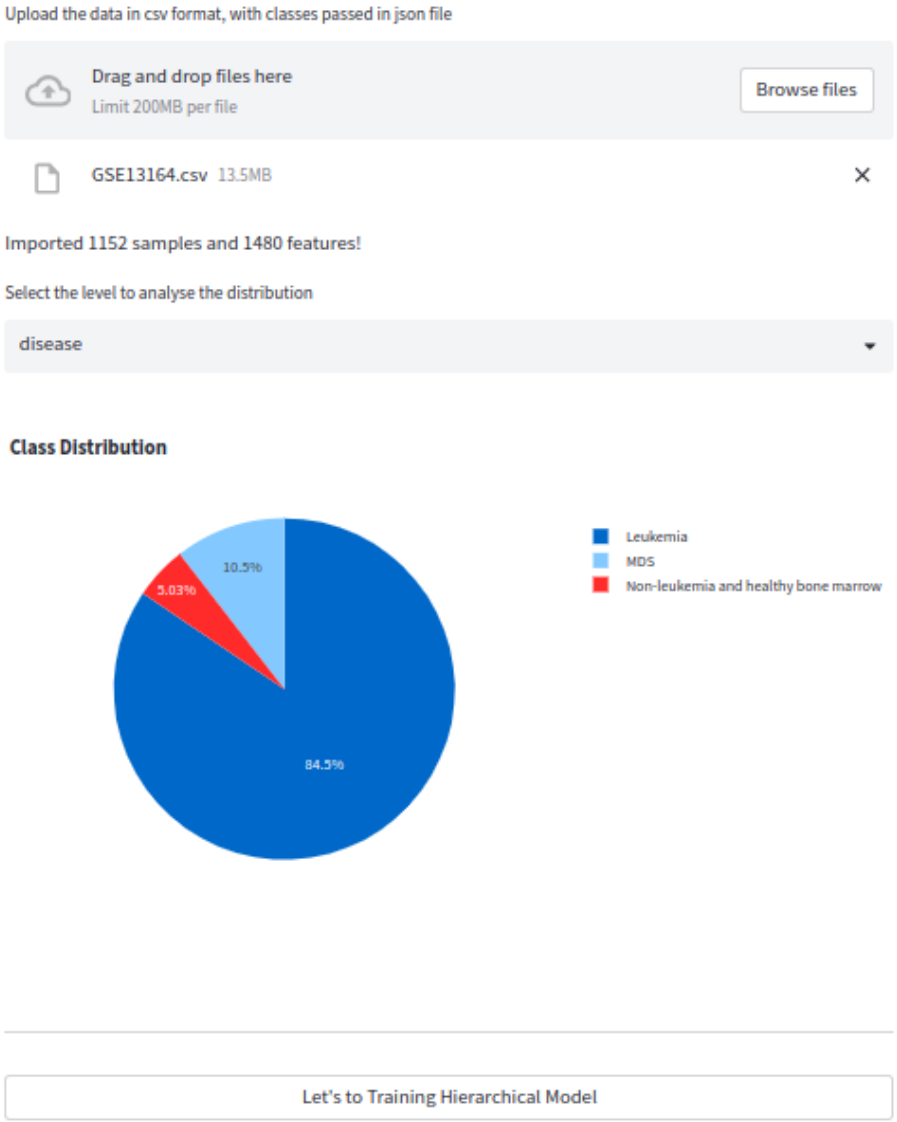
Após a importação do formato da hierarquia, deve-se importar os dados em arquivo de extensão *csv*, contendo as colunas com as *features* e as colunas com as classes de cada nível da hierarquia (cada nível da hierarquia deve ser uma

[13] No exemplo têm-se os níveis de hierarquia ”*disease*”, ”*leukemia*”e ”*subtypes*”. Para o nível ”*disease*”, têm-se como a raiz da hierarquia, por isso denota-se o valor ”*root*”, em que se possui as possibilidades de saudável, leucemia e MDS. Para o nível ”*leukemia*”, têm-se as relações conforme os valores do nível superior, ou seja, para o valor *leukemia*, há as possibilidades de CLL, ALL, AML e CML. Por fim, no nível ”*subtypes*”, cada tipo de leucemia possui suas possibilidades de subtipos.

coluna), conforme o exemplo expresso de forma genérica apresentado na Figura 15 anteriormente.

Ao importar os dados, é apresentado a quantidade de amostras e *features* importadas. Logo mais, um gráfico é gerado mostrando as distribuições de amostras por classe para cada nível da hierarquia, conforme o exemplo na Figura 24.

Figura 24: Tela com dados importados da opção *Hierarchical Method* [14]



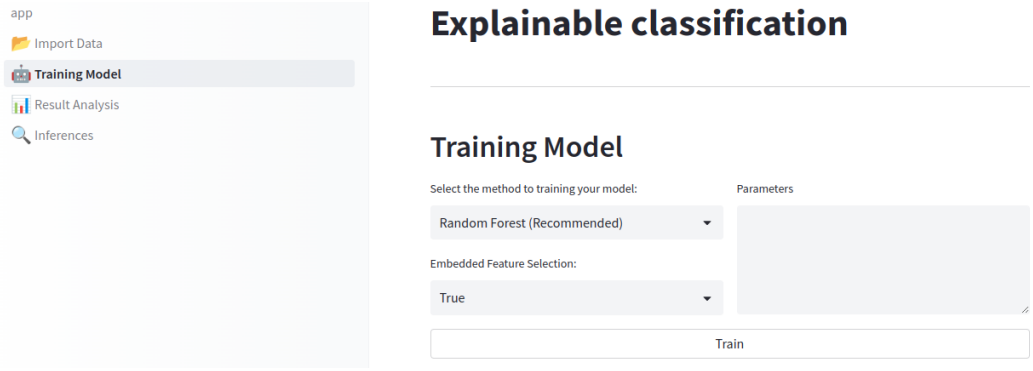
Fonte: Autoria Própria (2023)

[14] O exemplo consiste em dados de classes do nível "*disease*"da hierarquia. A ferramenta possui a possibilidade de selecionar o nível da hierarquia e verificar as distribuições das classes para o mesmo. Ao mostrar as informações dos dados, pode-se avançar para a próxima etapa de treinamento do modelo.

6.2 TREINAMENTO DO MODELO

Após a importação dos dados, tanto para os métodos canônicos e hierárquicos, pode-se escolher uma opção de técnicas de *machine learning* baseado em bibliotecas do Scikit-Learn. Também possui a opção de atribuir valores para hiper-parâmetros utilizando o formato de dicionário em Python. A Figura 25, apresenta um exemplo da interface com os componentes citados.

Figura 25: Tela inicial de treinamento de modelos [15]



Fonte: Autoria Própria (2023)

Ao clicar para treinar, o *pipeline* de treinamento será executado, abordando a arquitetura definida na Figura 18 para as abordagens canônicas e a arquitetura da Figura 19 para as abordagens hierárquicas.

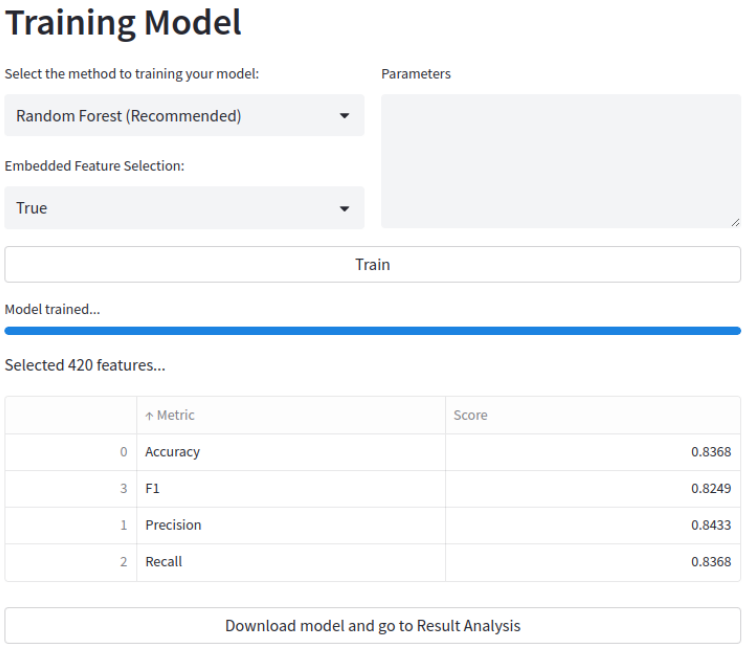
Nessa etapa de treinamento, é mostrado para o usuário uma tabela de métricas (Acurácia, precisão, revocação e F1-score), onde se é aplicado a metodologia de validação cruzada de forma estratificada, com o valor de *k* igual a 3. Os resultados das métricas são resultados médios considerando todas as iterações, conforme ilustrado no exemplo da Figura 26.

6.3 ANÁLISE DE RESULTADOS

Nessa etapa, a interface permite a realização de análises que consistem em explorar as decisões do modelo na tentativa de explicá-lo. Para tal, é utilizado a

[15] Possui opções de escolha de técnicas de *machine learning* e definição de hiper-parâmetros. Caso não seja definido hiper-parâmetros, as configurações padrões da biblioteca *scikit-learn* serão consideradas. Ao clicar no botão "*train*", o *pipeline* de treinamento será executado.

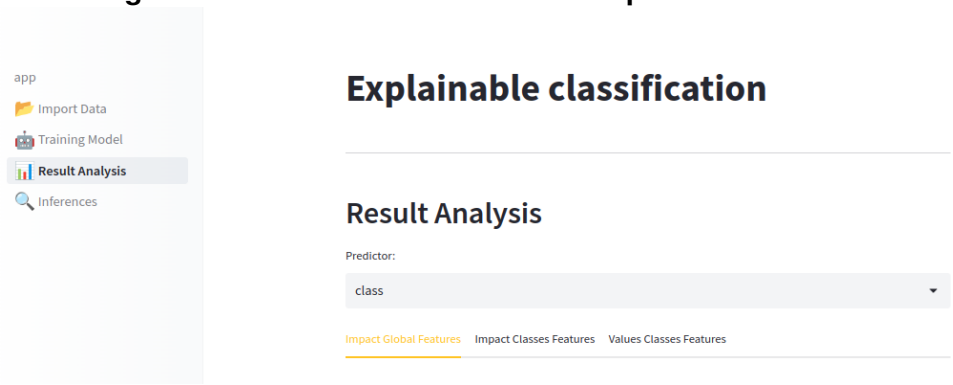
Figura 26: Tela após treinamento do modelo [16]



Fonte: Autoria Própria (2023)

técnica SHAP, identificando como as *features* contribuem para a decisão do modelo. É possível realizar 3 formas de análises: o impacto Global das *Features* (*Impact Global Features*), o impacto das *features* por classe (*Impact Classes Features*) e a relação do valor das *features* com uma classe (*Values Classes Features*). Na Figura 27 é destacada essa etapa na interface.

Figura 27: Tela inicial de análise de explicabilidade do modelo [17]



Fonte: Autoria Própria (2023)

[16] É exibido uma tabela com as métricas médias da validação cruzada. Têm-se a opção de treinar novamente ou realizar o download do modelo e seguir com a análise de explicabilidade do modelo. O modelo é salvo em formato *pickle*.

[17] Possuem 3 formas de análises, onde pode-se escolher o modelo a ser analisado (tal recurso é utilizado em questões de metodologia hierárquica, onde cada nó, possui um modelo diferente).

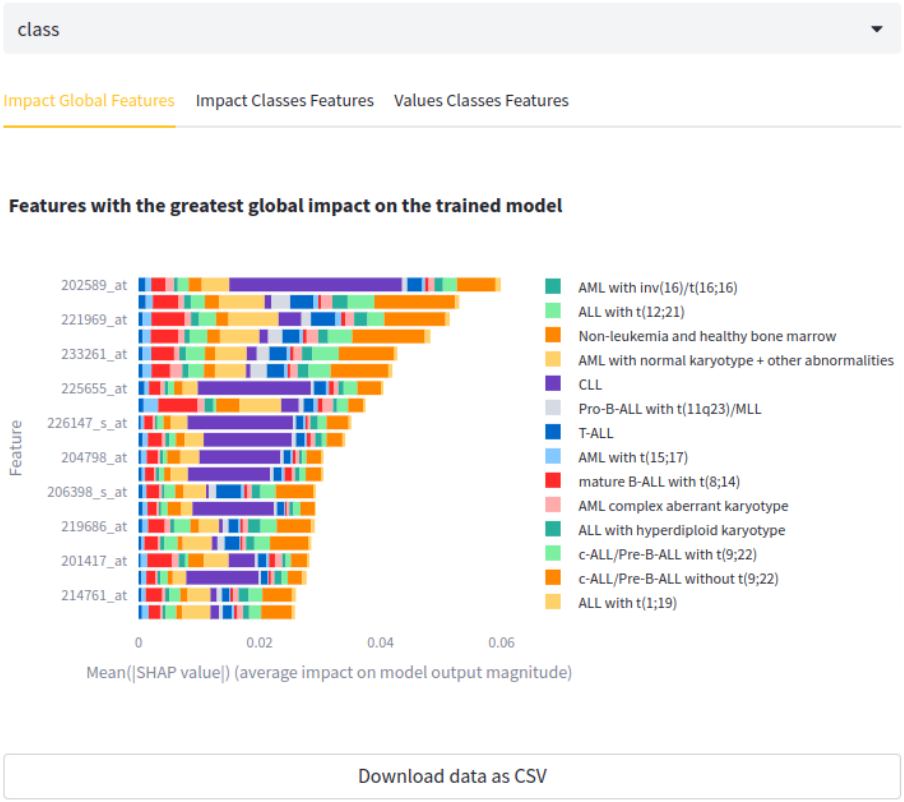
6.3.1 IMPACTO GLOBAL DAS *FEATURES*

Essa forma de análise permite identificar os impactos globais de cada *feature* para as decisões do modelo treinado, conforme os dados utilizados como base para o treinamento.

Dessa forma, é apresentado um gráfico que evidencia as 20 *features* que mais possuem contribuições nas decisões do modelo. Essas contribuições são calculadas através dos valores SHAP médios absolutos (visto que tal técnica permite identificar uma probabilidade de ganho ou perda para a decisão do modelo para uma determinada classe) para todas as classes.

A Figura 28 ilustra um exemplo da interface para análise global de impacto nas decisões do modelo.

Figura 28: Tela de explicabilidade de impacto global no modelo [18]



Fonte: Autoria Própria (2023)

[18] As *features* podem ter contribuições diferentes para a decisão de uma determinada classe no modelo, expressadas conforme os diferentes tamanhos da barra para cada classe. Cada classe possui uma cor diferente. O resultado total com todos os valores SHAP das *features* podem ser baixadas em formato de arquivo *csv*.

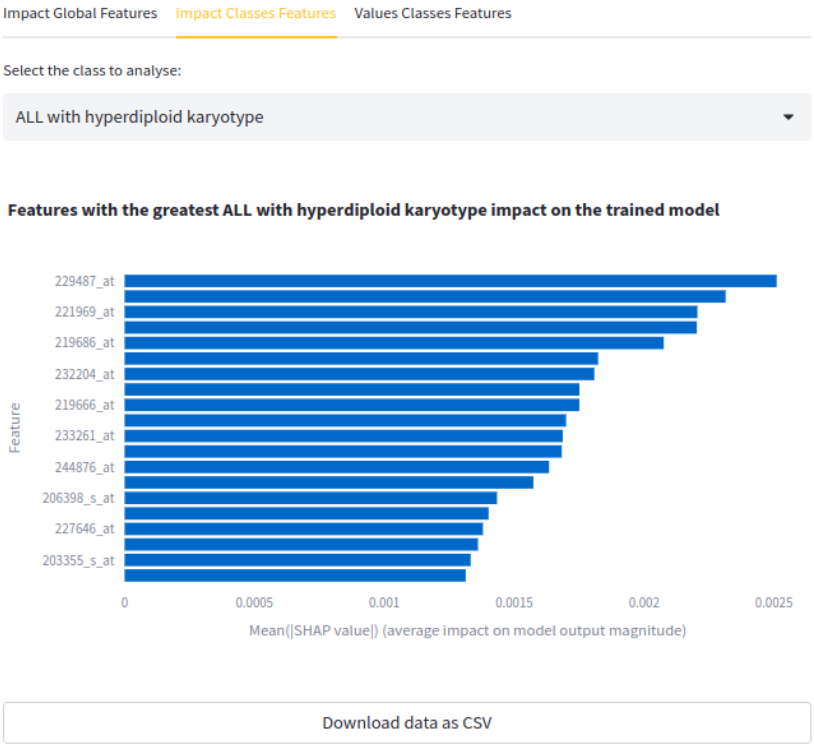
6.3.2 IMPACTO DAS *FEATURES* POR CLASSE

Conforme expresso na Figura 28 sobre o impacto global das *features*, cada *feature* pode ter contribuições diferentes para a decisão de uma determinada classe no modelo.

Assim sendo, essa forma de análise permite identificar para uma classe específica as contribuições de cada *feature*. Também é apresentado um gráfico que evidencia as 20 *features* que mais possuem contribuições nas decisões do modelo. Essas contribuições são calculadas através dos valores SHAP médios absolutos para a classe escolhida.

A Figura 29 ilustra um exemplo da interface para análise de impacto por classe nas decisões do modelo.

Figura 29: Tela de explicabilidade de impacto das *features* por classe no modelo [19]



Fonte: Autoria Própria (2023)

[19] Possui uma opção de seleção da classe desejada, obtendo um gráfico correspondente as *features* de maiores contribuições para a classe especificada. O resultado total com todos os valores SHAP das *features* podem ser baixadas em formato de arquivo *csv*.

6.3.3 RELAÇÃO DO VALOR DAS *FEATURES* COM UMA CLASSE

Por fim, a análise de explicabilidade do modelo através da relação de valor das *features* com o impacto na decisão de uma determinada classe, consiste em analisar os valores de cada *feature* de cada amostra utilizada para treinamento.

Isso posto, têm-se um gráfico de enxame (*bee swarm plot*) que consiste em evidenciar para cada amostra e cada *feature* a contribuição em valor SHAP para a determinação de uma classe específica (quando se obtêm valores negativos, indica-se que o valor real daquela amostra e *feature* contribuiu de forma negativa para que o modelo determine a amostra como a classe específica, caso contrário, a contribuição é positiva para ela). Além disso, para aquela determinada amostra e *feature* também é identificada se o valor real consiste em um valor alto ou baixo em relação as demais amostras daquela *feature*, obtendo cores mais próximas do vermelho em casos de valores considerados altos e próximas do tom de azul para valores considerados baixos.

A Figura 30 ilustra um exemplo da interface para análise de relação dos valores das *features* para decisão de uma classe pelo modelo.

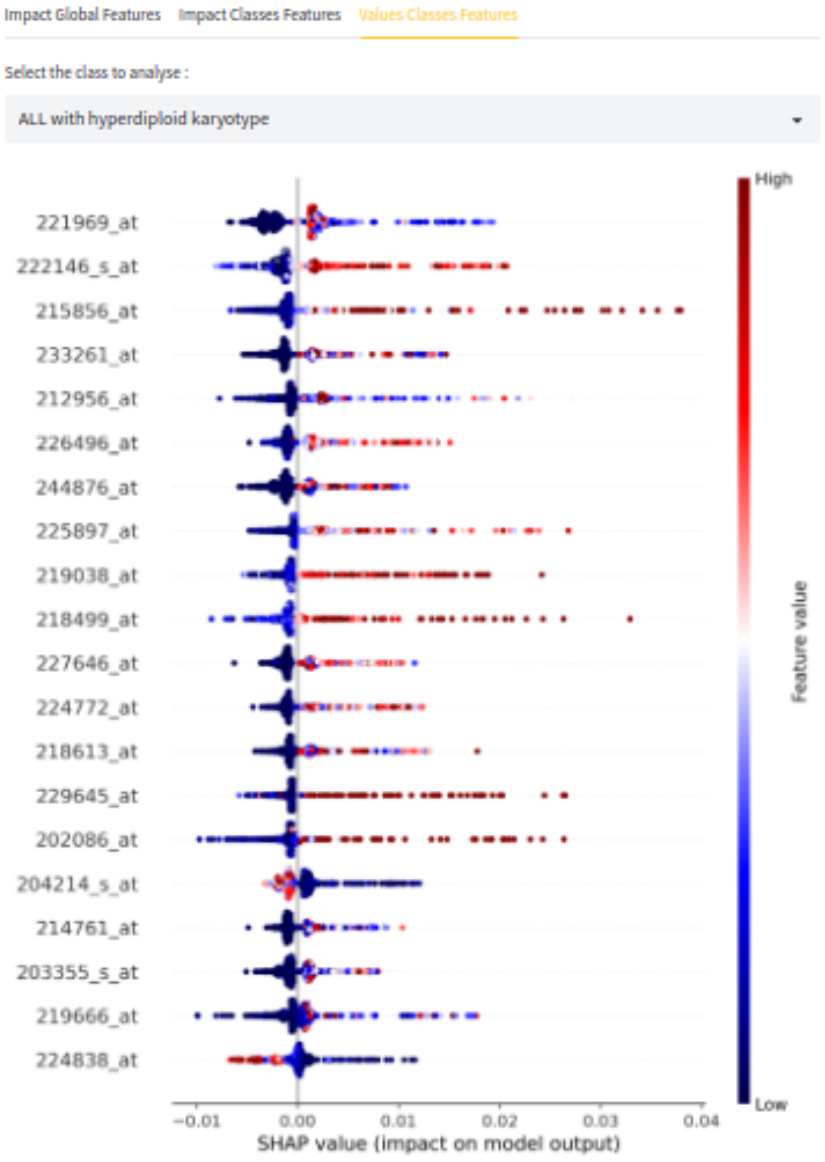
6.4 INFERÊNCIAS

Enfim, a etapa de inferências consiste em importar os modelos treinados utilizando a ferramenta e gerar previsões a partir de novos dados com classes desconhecidas e assim obter melhores visualizações da probabilidade e os motivos do modelo decidir que uma determinada amostra pertence a uma certa classe. Na Figura 31 é apresentada essa aplicação e as subseções seguintes irão aprofundar sobre as funcionalidades presentes nela.

6.4.1 PREDIÇÕES

Tal funcionalidade permite identificar as previsões do modelo para cada amostra importada. É criada uma tabela com índices numéricos identificando cada amostra e uma coluna com as classes previstas pelo modelo importado. Na Figura 32

Figura 30: Tela de explicabilidade de relação do valor das *features* com uma classe no modelo [20]



Fonte: Autoria Própria (2023)

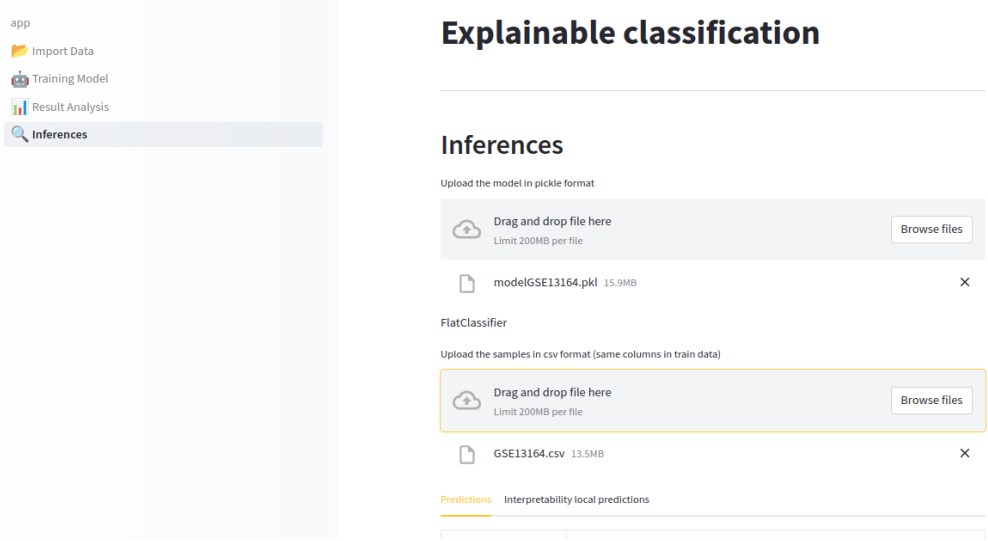
é apresentada um exemplo de uso.

6.4.2 INTERPRETABILIDADE LOCAL DAS PREDIÇÕES

Gera meios visuais e analíticos que permitem a interpretabilidade dos motivos da decisão do modelo em classificar uma amostra para uma determinada classe.

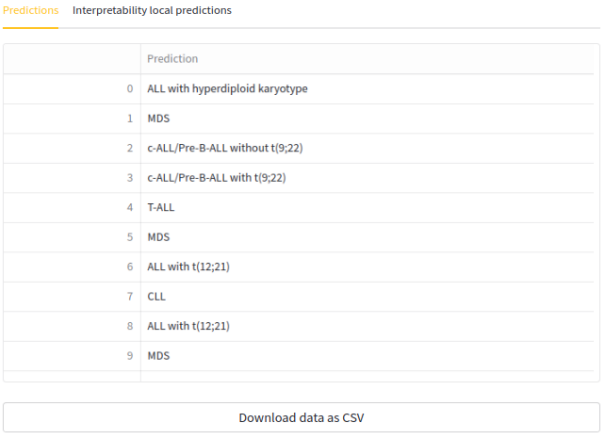
[20] Possui uma opção de seleção da classe desejada, obtendo um gráfico de enxame destacando como cada amostra e *feature* estão contribuindo para que a decisão do modelo seja a classe especificada. A ordem de apresentação é baseada nas 20 *features*com maiores contribuições médias absolutas para a classe selecionada (no exemplo selecionou-se: ”ALL with hyperdiploid karyotype”).

Figura 31: Tela inicial de inferências [21]



Fonte: Autoria Própria (2023)

Figura 32: Tela de predições realizadas pelo modelo [22]



Fonte: Autoria Própria (2023)

Para obter tal informação, também são calculados os valores SHAP do modelo sobre os dados importados para inferência, obtendo as contribuições das *features* para a definição da classe.

Logo, a interface permite escolher um modelo (quando o método é canônico

[21] A tela consiste em importar o modelo treinado ao longo das etapas anteriores. O sistema já identifica se o modelo treinado foi realizado pelo método canônico ou hierárquico. Após a importação do modelo, é necessário importar os dados com as amostras que serão previstas pelo modelo, sendo fundamental as colunas com as *features* utilizadas para o treinamento do modelo. Por fim duas opções são habilitadas: Predições e interpretabilidade local das predições.

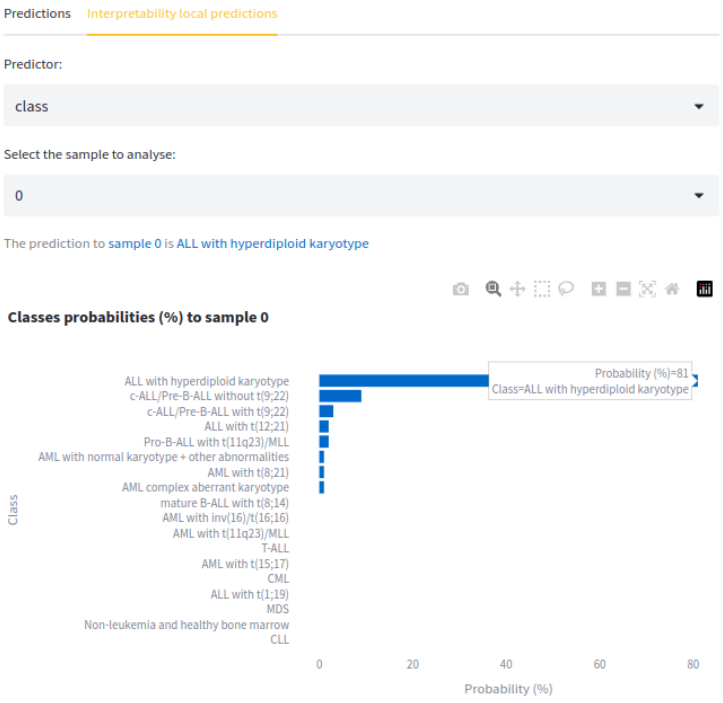
[22] Destaca-se a tabela com as predições do modelo e a possibilidade de baixar os resultados em formato *csv*.

(*flat*), a opção já aparece selecionada com o valor padrão ”*class*”, caso seja hierárquico, é apresentado a opção de todos os modelos criados nos nós da hierarquia) e uma amostra e assim obter visualizações gráficas sobre a probabilidade de escolha do modelo e as contribuições das *features* para cada classe.

Assim, é possível identificar a probabilidade do modelo em determinar uma classe (dada todas as opções que o modelo obteve no momento do treinamento). O cálculo da probabilidade consiste em somar o valor de viés que modelo possui para a determinação de uma classe (valor determinado através do treinamento do modelo, relacionado com o balanceamento ou desbalanceamento das classes) e o somatório das contribuições dos valores SHAP das *features* para cada classe.

A Figura 33 apresenta um gráfico de barras que permite a visualização das probabilidades de decisão para cada classe por meio de uma amostra escolhida.

Figura 33: Tela de interpretabilidade do modelo com probabilidades de decisão de classes [23]



Fonte: Autoria Própria (2023)

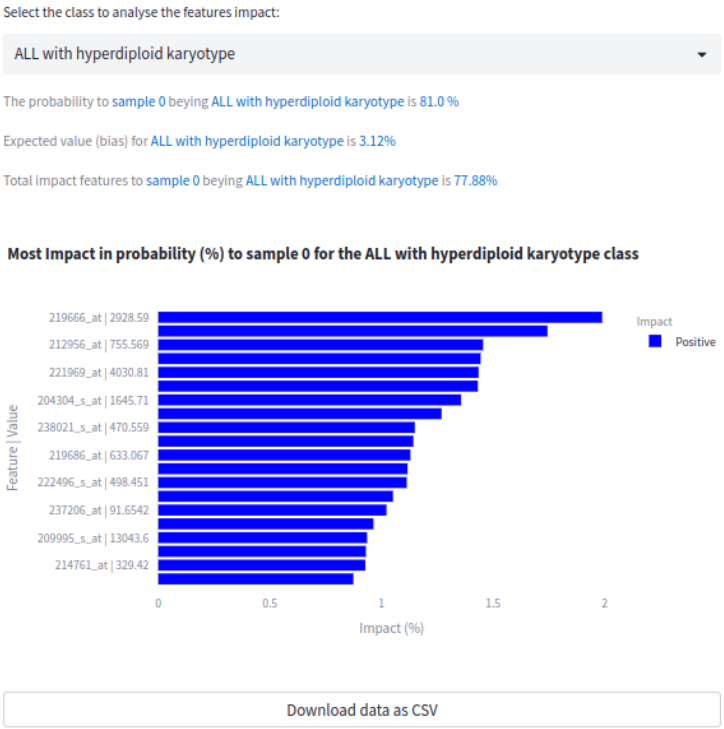
[23] Escolhido o modelo e a amostra a ser analisada, é calculado através dos valores SHAP a probabilidade de decisão do modelo para todas as classes passadas em treinamento. Dá-se que a decisão final do modelo seja a classe de maior probabilidade, porém é possível identificar se a decisão do modelo possui valores altos de probabilidade, ou se o modelo possui certas indecisões entre as opções.

Visto a probabilidade de decisão para cada classe do modelo, pode-se também identificar como os valores das *features* contribuíram para se obter o valor final de probabilidade e por consequência obter uma decisão.

Logo, dado um modelo e uma amostra escolhida, a interface disponibiliza a informação do viés atrelado pelo modelo para a classe, os valores totais de contribuição das *features* e a probabilidade final do modelo para a classe naquela amostra. Também se é disponibilizado um gráfico que permite mostrar os valores das *features* de maior impacto (tanto positivo quanto negativo) para a formulação da probabilidade de decisão do modelo para a classe específica.

A Figura 34 mostra um exemplo escolhendo uma classe que possui maior probabilidade de decisão do modelo.

Figura 34: Tela de interpretabilidade do modelo com a contribuição das *features* para a decisão do modelo da amostra ser da classe "ALL with hyperdiploid karyotype"[24]



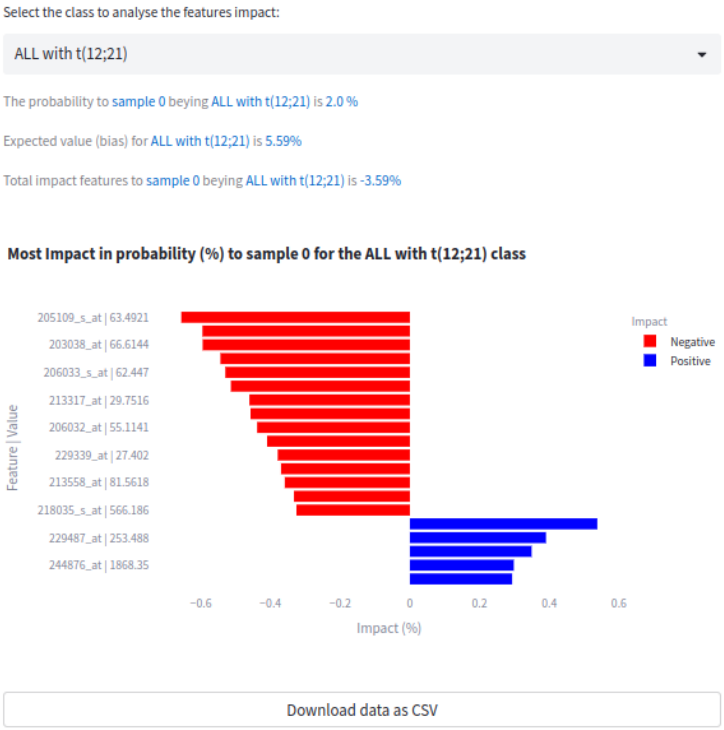
Fonte: Autoria Própria (2023)

Em contrapartida a Figura 35 apresenta uma classe que não possui a maior

[24] As *features* apresentadas no gráfico são as que possuem maior impacto (tanto positivo quanto negativo). Os valores totais podem ser baixados em formato *csv*. O exemplo apresenta as contribuições dos valores das *features* para a classe "ALL with hyperdiploid karyotype"dada uma amostra específica. Percebe-se que os valores das *features* de maior impacto apresentam contribuições positivas para a decisão do modelo.

probabilidade na decisão do modelo.

Figura 35: Tela de interpretabilidade do modelo com a contribuição das features para a decisão do modelo da amostra ser da classe *ALL with t(12;21)* [25]



Fonte: Autoria Própria (2023)

[25] O exemplo apresenta as contribuições dos valores das *features* para a classe ”ALL with t(12;21)”dada uma amostra específica. Percebe-se que em sua maioria os valores das *features* de maior impacto apresentam contribuições negativas para a decisão do modelo.

7 CONCLUSÕES

Este trabalho pretendeu criar uma ferramenta para aplicação de modelos de aprendizado de máquina utilizando genes diferencialmente expressos de pacientes que possuem leucemia de forma interpretável para classificar subtipos da doença. Para tal, desenvolveram-se experimentos que permitiram homologar um *pipeline* de predição e adaptar as metodologias utilizadas para aplicação criada.

Com isso, pode-se concluir que a metodologia proposta apresenta resultados promissores, destacando-se os resultados de métricas superiores à linha de base apresentada em trabalhos relacionados. Além disso, têm-se que a aplicação criada permite a realização da metodologia proposta de forma intuitiva e possui potencial de expandir a possibilidade de análises e interpretações sobre os modelos criados.

7.1 PRINCIPAIS CONTRIBUIÇÕES

O trabalho permitiu contribuir com a metodologia de desenvolvimento de um *pipeline* que pretende testar diferentes abordagens, técnicas de aprendizado de máquina e seleção de *features* para obter o máximo de métricas de desempenho de precisão nos projetos selecionados.

Vale destacar que este trabalho proporciona identificar a importância de realizar o processamento de características, pois esse processo pode resultar em melhores resultados de classificação, além de fornecer um escopo menor de características para análises e estudos futuros.

Pode-se citar também a utilização de abordagens hierárquicas em problemas que possuem tal nível de informação, pois tal utilização permite a criação de modelos com alto valor preditivo.

7.2 PRINCIPAIS BENEFÍCIOS DA FERRAMENTA

A ferramenta proporciona o uso simplificado para criação de modelos de aprendizado de máquina, visando ser uma aplicação de grande utilidade para pesquisadores de diferentes áreas, diminuindo a curva de aprendizagem para conceitos de inteligência artificial.

A questão da interpretabilidade dos modelos se torna uma das principais contribuições da ferramenta, permitindo a exploração de decisões de modelos de aprendizado de máquina e assim obter possíveis compreensões sobre determinados motivos de classificações de multi-classes.

Também proporciona a replicabilidade de experimentos de forma rápida, visto que grande parte do *pipeline* desenvolvido foi implementado visando usabilidade e praticidade.

7.3 PRINCIPAIS LIMITAÇÕES E DIFICULDADES

As etapas de preparação de dados não foram incluídas no sistema de desenvolvimento atual, devido a essa etapa conter um valor elevado de processamento e diferentes possibilidades de abordagens. Dessa forma, reitera-se que o aplicativo atual seja uma ferramenta de treinamento e análise de abordagens de aprendizado de máquina, deixando a parte de preparação de dados (fundamental para todo o processo) ao usuário final.

7.4 PUBLICAÇÕES

Durante o desenvolvimento dessa pesquisa, foi submetido um artigo intitulado de *Multiple machine learning algorithms for Leukemia multiclass task and gene expression data* para o periódico IEEE/ACM Transactions on Computational Biology and Bioinformatics, abordando sobre os experimentos realizados utilizando projetos de contagem de genes de multi-classes de leucemia.

7.5 TRABALHOS FUTUROS

Para trabalhos futuros serão exploradas novas abordagens para a estrutura hierárquica, considerando diferentes combinações de técnicas de aprendizado de máquina entre os nós da hierarquia e o uso de eliminação recursiva de características para a seleção de características em cada nó, buscando encontrar as melhores abordagens que permitem obter valores de alta precisão com a utilização de menores quantidades de características, buscando reduzir o escopo de características a serem analisadas para a interpretabilidade dos modelos de classificação.

8 CONSIDERAÇÕES FINAIS

O projeto apresentado obteve orientação de cientistas e pesquisadores do Instituto de Pesquisa Pelé Pequeno Príncipe. O instituto possui cientistas que estão à frente de dezenas de projetos inovadores, com importantes publicações científicas que buscam alternativas para cura e melhoria de vida de crianças e adolescentes portadores de doenças graves.

Também obteve-se parceria com demais pesquisadores da Universidade Tecnológica Federal do Paraná (UTFPR) que realizam projetos voltados a bioinformática pelo programa de mestrado PPGBIOINFO e iniciação científica no curso de Engenharia de Computação.

REFERÊNCIAS

ALLEGRA, A.; TONACCI, A.; SCIACCOTTA, R.; GENOVESE, S.; MUSOLINO, C.; PIOGGIA, G.; GANGEMI, S. Machine learning and deep learning applications in multiple myeloma diagnosis, prognosis, and treatment selection. **Cancers**, MDPI AG, v. 14, n. 3, p. 606, jan. 2022. Disponível em: <https://doi.org/10.3390/cancers14030606>.

ALSALEM, M.; ZAIDAN, A.; ZAIDAN, B.; HASHIM, M.; MADHLOOM, H.; AZEEZ, N.; ALSYISUF, S. A review of the automated detection and classification of acute leukaemia: Coherent taxonomy, datasets, validation and performance measurements, motivation, open challenges and recommendations. **Computer Methods and Programs in Biomedicine**, Elsevier BV, v. 158, p. 93–112, maio 2018. Disponível em: <https://doi.org/10.1016/j.cmpb.2018.02.005>.

ALSALEM, M. A.; ZAIDAN, A. A.; ZAIDAN, B. B.; HASHIM, M.; ALBAHRI, O. S.; ALBAHRI, A. S.; HADI, A.; MOHAMMED, K. I. Systematic review of an automated multiclass detection and classification system for acute leukaemia in terms of evaluation and benchmarking, open challenges, issues and methodological aspects. **Journal of Medical Systems**, Springer Science and Business Media LLC, v. 42, n. 11, set. 2018. Disponível em: <https://doi.org/10.1007/s10916-018-1064-9>.

ARBER, D. A.; ORAZI, A.; HASSERJIAN, R.; THIELE, J.; BOROWITZ, M. J.; BEAU, M. M. L.; BLOOMFIELD, C. D.; CAZZOLA, M.; VARDIMAN, J. W. The 2016 revision to the world health organization classification of myeloid neoplasms and acute leukemia. **Blood**, American Society of Hematology, v. 127, n. 20, p. 2391–2405, maio 2016. Disponível em: <https://doi.org/10.1182/blood-2016-03-643544>.

BIAU, G. Analysis of a random forests model. **Journal of Machine Learning Research**, v. 13, 05 2010.

BOSCH, F.; DALLA-FAVERA, R. Chronic lymphocytic leukaemia: from genetics to treatment. **Nature Reviews Clinical Oncology**, Springer Science and Business Media LLC, v. 16, n. 11, p. 684–701, jul 2019. Disponível em: <https://doi.org/10.1038%2Fs41571-019-0239-8>.

BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>.

BROWNE, M. W. Cross-validation methods. **Journal of Mathematical Psychology**, Elsevier BV, v. 44, n. 1, p. 108–132, mar. 2000. Disponível em: <https://doi.org/10.1006/jmps.1999.1279>.

CASTILLO, D.; GALVEZ, J. M.; HERRERA, L. J.; ROJAS, F.; VALENZUELA, O.; CABA, O.; PRADOS, J.; ROJAS, I. Leukemia multiclass assessment and classification

from microarray and RNA-seq technologies integration at gene expression level. **PLOS ONE**, Public Library of Science (PLoS), v. 14, n. 2, p. e0212127, fev. 2019. Disponível em: <https://doi.org/10.1371/journal.pone.0212127>.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Computers & Electrical Engineering**, v. 40, n. 1, p. 16 – 28, 2014. ISSN 0045-7906. 40th-year commemorative issue. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0045790613003066>.

COHEN, J. **Statistical Power Analysis for the Behavioral Sciences**. [S.l.]: Lawrence Erlbaum Associates, 1988.

DINIZ, W.; CANDURI, F. Review-article bioinformatics: an overview and its applications. **Genetics and Molecular Research**, Genetics and Molecular Research, v. 16, n. 1, 2017. ISSN 1676-5680. Disponível em: <http://dx.doi.org/10.4238/gmr16019645>.

EDEN, R. E.; COVIELLO, J. M. Chronic myelogenous leukemia. In: **StatPearls [Internet]**. StatPearls Publishing, 2022. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK531459/>.

FAUZI, I. R.; RUSTAM, Z.; WIBOWO, A. Multiclass classification of leukemia cancer data using fuzzy support vector machine (FSVM) with feature selection using principal component analysis (PCA). **Journal of Physics: Conference Series**, IOP Publishing, v. 1725, n. 1, p. 012012, jan. 2021. Disponível em: <https://doi.org/10.1088/1742-6596/1725/1/012012>.

FELTES, B. C.; CHANDELIER, E. B.; GRISCI, B. I.; DORN, M. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. **Journal of Computational Biology**, v. 26, n. 4, p. 376–386, 2019. PMID: 30789283. Disponível em: <https://doi.org/10.1089/cmb.2018.0238>.

FRANC, V.; HLAVAC, V. Multi-class support vector machine. In: **Object recognition supported by user interaction for service robots**. IEEE Comput. Soc. Disponível em: <https://doi.org/10.1109/icpr.2002.1048282>.

FREITAS, A.; CARVALHO, A. A tutorial on hierarchical classification with applications in bioinformatics. In: **Research and Trends in Data Mining Technologies and Applications**. IGI Global, 2007. p. 175–208. Disponível em: <https://doi.org/10.4018/978-1-59904-271-8.ch007>.

FREUND, Y. Boosting a weak learning algorithm by majority. **Information and Computation**, Elsevier BV, v. 121, n. 2, p. 256–285, set. 1995. ISSN 0890-5401. Disponível em: <http://dx.doi.org/10.1006/inco.1995.1136>.

GUO, Y.; CHUNG, F.-L.; LI, G.; ZHANG, L. Multi-label bioinformatics data classification with ensemble embedded feature selection. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, p. 103863–103875, 2019. Disponível em: <https://doi.org/10.1109/access.2019.2931035>.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 1157–1182, mar 2003. ISSN 1532-4435. Disponível em: <https://dl.acm.org/doi/pdf/10.5555/944919.944968>.

HAMERSCHLAK, N. Leucemia: fatores prognósticos e genética. **Jornal de Pediatria**, FapUNIFESP (SciELO), v. 84, n. 4, p. S52–S57, aug 2008. Disponível em: <https://doi.org/10.1590%2Fs0021-75572008000500008>.

HAMIDAH; RUSTAM, Z.; UTAMA, S.; SISWANTINING, T. Multiclass classification of acute lymphoblastic leukemia microarrays data using support vector machine algorithms. **Journal of Physics: Conference Series**, IOP Publishing, v. 1490, n. 1, p. 012027, mar. 2020. Disponível em: <https://doi.org/10.1088/1742-6596/1490/1/012027>.

HATWELL, J.; GABER, M. M.; AZAD, R. M. A. CHIRPS: Explaining random forest classification. **Artificial Intelligence Review**, Springer Science and Business Media LLC, v. 53, n. 8, p. 5747–5788, jun. 2020. Disponível em: <https://doi.org/10.1007/s10462-020-09833-6>.

HUTTER, J. J. Childhood leukemia. **Pediatrics In Review**, American Academy of Pediatrics (AAP), v. 31, n. 6, p. 234–241, jun 2010. Disponível em: <https://doi.org/10.1542%2Fpir.31.6.234>.

INABA, H.; PUI, C.-H. Advances in the diagnosis and treatment of pediatric acute lymphoblastic leukemia. **Journal of Clinical Medicine**, MDPI AG, v. 10, n. 9, p. 1926, abr. 2021. Disponível em: <https://doi.org/10.3390/jcm10091926>.

JULIUSSON, G.; HOUGH, R. Leukemia. In: **Progress in Tumor Research**. S. Karger AG, 2016. p. 87–100. Disponível em: <https://doi.org/10.1159%2F000447076>.

KADIYALA, A.; KUMAR, A. Applications of python to evaluate environmental data science problems. **Environmental Progress & Sustainable Energy**, Wiley, v. 36, n. 6, p. 1580–1586, out. 2017. Disponível em: <https://doi.org/10.1002/ep.12786>.

KESAVARAJ, G.; SUKUMARAN, S. A study on classification techniques in data mining. In: **2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)**. IEEE, 2013. Disponível em: <https://doi.org/10.1109/icccnt.2013.6726842>.

KHORASANI, M.; ABDOU, M.; FERNÁNDEZ, J. H. Streamlit basics. In: **Web Application Development with Streamlit**. Apress, 2022. p. 31–62. Disponível em: <https://doi.org/10.1007/978-1-4842-8111-6_2>.

KOTSIANTIS, S. B. Decision trees: a recent overview. **Artificial Intelligence Review**, Springer Science and Business Media LLC, v. 39, n. 4, p. 261–283, jun. 2011. Disponível em: <https://doi.org/10.1007/s10462-011-9272-4>.

LAZAR, C.; TAMINAU, J.; MEGANCK, S.; STEENHOFF, D.; COLETTA, A.; MOLTER, C.; SCHAETZEN, V. de; DUQUE, R.; BERSINI, H.; NOWE, A. A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, Institute of Electrical

and Electronics Engineers (IEEE), v. 9, n. 4, p. 1106–1119, jul. 2012. Disponível em: <https://doi.org/10.1109/tcbb.2012.33>.

LEVER, J.; KRZYWINSKI, M.; ALTMAN, N. Classification evaluation. **Nature Methods**, Springer Science and Business Media LLC, v. 13, n. 8, p. 603–604, jul. 2016. Disponível em: <https://doi.org/10.1038/nmeth.3945>.

LIU, H.; YU, L. Yu, l.: Toward integrating feature selection algorithm for classification and clustering. ieee transaction on knowledge and data engineering 17(4), 491-502. **IEEE Transactions on Knowledge and Data Engineering - TKDE**, v. 17, p. 491–502, 04 2005.

LIU, X.-Y.; LIANG, Y.; WANG, S.; YANG, Z.-Y.; YE, H.-S. A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 22863–22874, 2018. Disponível em: <https://doi.org/10.1109/access.2018.2818682>.

LIU, Y. **Python machine learning by example**. Birmingham, England: Packt Publishing, 2017.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>.

MARIOTTI, E.; ALONSO-MORAL, J. M.; GATT, A. Measuring model understandability by means of shapley additive explanations. In: **2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.: s.n.], 2022. p. 1–8.

MORAN, M.; GORDON, G. Curious feature selection. **Information Sciences**, v. 485, p. 42 – 54, 2019. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025519301100>.

MUKKAMALLA, S. K. R.; TANEJA, A.; MALIPEDDI, D.; MASTER, S. R. Chronic lymphocytic leukemia. In: **StatPearls [Internet]**. StatPearls Publishing, 2022. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK470433/>.

NAKAHARA, H.; JINGUJI, A.; SATO, S.; SASAO, T. A random forest using a multi-valued decision diagram on an fpga. In: **2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)**. [S.l.: s.n.], 2017. p. 266–271.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: **Machine Learning in Radiation Oncology**. Springer International Publishing, 2015. p. 3–11. Disponível em: <https://doi.org/10.1007/978-3-319-18305-3_1>.

PARMEZAN, A. R.; SOUZA, V. M.; SETH, A.; ŽLIOBAITĖ, I.; BATISTA, G. E. Hierarchical classification of pollinating flying insects under changing environments. **Ecological Informatics**, Elsevier BV, v. 70, p. 101751, set. 2022. Disponível em: <https://doi.org/10.1016/j.ecoinf.2022.101751>.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Disponível em: <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>.

PELCOVITS, A.; NIROULA, R. Acute myeloid leukemia: A review. **R. I. Med. J. (2013)**, v. 103, n. 3, p. 38–40, abr. 2020.

PUCKETT, Y.; CHAN, O. Acute lymphocytic leukemia. In: **StatPearls [Internet]**. StatPearls Publishing, 2022. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK459149/>.

ROKACH, L.; MAIMON, O. Decision trees. In: **Data Mining and Knowledge Discovery Handbook**. Springer-Verlag, 2005. p. 165–192. Disponível em: <https://doi.org/10.1007/0-387-25465-x_9>.

SALAH, H. T.; MUHSEN, I. N.; SALAMA, M. E.; OWAIDAH, T.; HASHMI, S. K. Machine learning applications in the diagnosis of leukemia: Current trends and future directions. **International Journal of Laboratory Hematology**, Wiley, v. 41, n. 6, p. 717–725, set. 2019. Disponível em: <https://doi.org/10.1111/ijlh.13089>.

SANTANA, L. E. A. dos S.; CANUTO, A. M. de P. Filter-based optimization techniques for selection of feature subsets in ensemble systems. **Expert Systems with Applications**, Elsevier BV, v. 41, n. 4, p. 1622–1631, mar. 2014. Disponível em: <https://doi.org/10.1016/j.eswa.2013.08.059>.

SARDER, M. A.; MANIRUZZAMAN, M.; AHAMMED, B. Feature selection and classification of leukemia cancer using machine learning techniques. **Machine Learning Research**, Science Publishing Group, v. 5, n. 2, p. 18, 2020. Disponível em: <https://doi.org/10.11648/j.mlr.20200502.11>.

SCARFÒ, L.; FERRERI, A. J. M.; GHIA, P. Chronic lymphocytic leukaemia. **Critical Reviews in Oncology/Hematology**, Elsevier BV, v. 104, p. 169–182, aug 2016. Disponível em: <https://doi.org/10.1016%2Fj.critrevonc.2016.06.003>.

SCHAPIRE, R. E. The strength of weak learnability. **Machine Learning**, Springer Science and Business Media LLC, v. 5, n. 2, p. 197–227, jun. 1990. ISSN 1573-0565. Disponível em: <http://dx.doi.org/10.1007/BF00116037>.

SCHAPIRE, R. E. The boosting approach to machine learning: An overview. In: ____. **Lecture Notes in Statistics**. Springer New York, 2003. p. 149–171. ISBN 9780387215792. Disponível em: <http://dx.doi.org/10.1007/978-0-387-21579-2_9>.

SCHMIDT, B.; BROWN, L. M.; RYLAND, G. L.; LONSDALE, A.; KOSASIH, H. J.; LUDLOW, L. E.; MAJEWSKI, I. J.; BLOMBERY, P.; EKERT, P. G.; DAVIDSON, N. M.; OSHLACK, A. ALLSorts: an RNA-seq subtype classifier for b-cell acute lymphoblastic leukemia. **Blood Advances**, American Society of Hematology, v. 6, n. 14, p. 4093–4097, jul. 2022. Disponível em: <https://doi.org/10.1182/bloodadvances.2021005894>.

SCHRöER, C.; KRUSE, F.; GÓMEZ, J. M. A systematic literature review on applying CRISP-DM process model. **Procedia Computer Science**, Elsevier BV, v. 181, p. 526–534, 2021. Disponível em: <https://doi.org/10.1016/j.procs.2021.01.199>.

SESSIONS, J. Chronic myeloid leukemia in 2007. **Journal of Managed Care Pharmacy**, Academy of Managed Care Pharmacy, v. 13, n. 8 Supp A, p. 4–7, oct 2007. Disponível em: <https://doi.org/10.18553%2Fjmcp.2007.13.s8-a.4>.

SHAFIQUE, U.; QAISER, H. A comparative study of data mining process models (kdd, crisp-dm and semma). **International Journal of Innovation and Scientific Research**, v. 12, n. 1, p. 217–222, 2014.

SIDEY-GIBBONS, J. A. M.; SIDEY-GIBBONS, C. J. Machine learning in medicine: a practical introduction. **BMC Medical Research Methodology**, Springer Science and Business Media LLC, v. 19, n. 1, mar. 2019. Disponível em: <https://doi.org/10.1186/s12874-019-0681-4>.

SILLA, C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. **Data Mining and Knowledge Discovery**, Springer Science and Business Media LLC, v. 22, n. 1-2, p. 31–72, abr. 2010. Disponível em: <https://doi.org/10.1007/s10618-010-0175-9>.

SKIENA, S. S. **The Data Science Design Manual**. 1. ed. Cham, Switzerland: Springer International Publishing, 2017. (Texts in computer science).

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation. In: **Lecture Notes in Computer Science**. Springer Berlin Heidelberg, 2006. p. 1015–1021. Disponível em: <https://doi.org/10.1007/11941439_114>.

TAYLOR, C. C.; SPIEGELHALTER, D. J.; MICHIE, D. **Machine Learning, Neural and Statistical Classification**. Harlow, England: Ellis Horwood Ltd, Publisher, 1994. (Ellis Horwood Workshops S.).

TERWILLIGER, T.; ABDUL-HAY, M. Acute lymphoblastic leukemia: a comprehensive review and 2017 update. **Blood Cancer J.**, v. 7, n. 6, p. e577, jun. 2017.

THARWAT, A. Classification assessment methods. **Applied Computing and Informatics**, Emerald, v. 17, n. 1, p. 168–192, jul. 2020. Disponível em: <https://doi.org/10.1016/j.aci.2018.08.003>.

THEOBALD, O. **Machine learning for absolute beginners**. [S.l.]: Independently Published, 2018. (Machine Learning for Beginners).

VAKITI, A.; MEWAWALLA, P. Acute myeloid leukemia. In: **StatPearls [Internet]**. StatPearls Publishing, 2021. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK507875/>.

VANITHA, C. D. A.; DEVARAJ, D.; VENKATESULU, M. Multiclass cancer diagnosis in microarray gene expression profile using mutual information and support vector machine. **Intelligent Data Analysis**, IOS Press, v. 20, n. 6, p. 1425–1439, nov. 2016. Disponível em: <https://doi.org/10.3233/ida-150203>.

VASIGHIZAKER, A.; SHARMA, A.; DEHZANGI, A. A novel one-class classification approach to accurately predict disease-gene association in acute myeloid leukemia cancer. **PLOS ONE**, Public Library of Science (PLoS), v. 14, n. 12, p. e0226115, dez. 2019. Disponível em: <https://doi.org/10.1371/journal.pone.0226115>.

WANG, Z.; XUE, X. Multi-class support vector machine. In: **Support Vector Machines Applications**. Springer International Publishing, 2014. p. 23–48. Disponível em: <https://doi.org/10.1007/978-3-319-02300-7_2>.

YIN, S.; TIAN, X.; ZHANG, J.; SUN, P.; LI, G. PCirc: random forest-based plant circRNA identification software. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 22, n. 1, jan. 2021. Disponível em: <https://doi.org/10.1186/s12859-020-03944-1>.

APÊNDICE A – DISTRIBUIÇÃO DE CLASSES

Tabela 4: Distribuição de classes do projeto GSE87070

Classe	Número de amostras	%
ETV6-RUNX1	172	26,30%
Other BCP	402	61,47%
T-ALL	80	12,23%

Fonte: Autoria Própria (2023)

Tabela 5: Distribuição de classes do projeto GSE13159

Classe	Número de amostras	%
ALL with hyperdiploid karyotype	40	1,91%
ALL with t(12;21)	58	2,77%
ALL with t(1;19)	36	1,72%
AML complex aberrant karyotype	48	2,29%
AML with inv(16)/t(16;16)	28	1,34%
AML with normal karyotype + other abnormalities	351	16,75%
AML with t(11q23)/MLL	38	1,81%
AML with t(15;17)	37	1,77%
AML with t(8;21)	40	1,91%
CLL	448	21,37%
CML	76	3,63%
MDS	206	9,83%
Non-leukemia and healthy bone marrow	74	3,53%
Pro-B-ALL with t(11q23)/MLL	70	3,34%
T-ALL	174	8,30%
c-ALL/Pre-B-ALL with t(9;22)	122	5,82%
c-ALL/Pre-B-ALL without t(9;22)	237	11,31%
mature B-ALL with t(8;14)	13	0,62%

Fonte: Autoria Própria (2023)

Tabela 6: Distribuição de classes do projeto GSE13164

Classe	Número de amostras	%
ALL with hyperdiploid karyotype	35	3,04%
ALL with t(12;21)	64	5,56%
ALL with t(1;19)	10	0,87%
AML complex aberrant karyotype	24	2,08%
AML with inv(16)/t(16;16)	20	1,74%
AML with normal karyotype + other abnormalities	160	13,89%
AML with t(11q23)/MLL	17	1,48%
AML with t(15;17)	20	1,74%
AML with t(8;21)	16	1,39%
CLL	237	20,57%
CML	43	3,73%
MDS	121	10,50%
Non-leukemia and healthy bone marrow	58	5,03%
Pro-B-ALL with t(11q23)/MLL	23	2,00%
T-ALL	79	6,86%
c-ALL/Pre-B-ALL with t(9;22)	62	5,38%
c-ALL/Pre-B-ALL without t(9;22)	158	13,72%
mature B-ALL with t(8;14)	5	0,43%

Fonte: Autoria Própria (2023)

Tabela 7: Distribuição de classes do projeto GSE9476

Classe	Número de amostras	%
AML	26	40,63%
Bone_Marrow	10	15,63%
Bone_Marrow_CD34	8	12,50%
PB	10	15,63%
PBSC_CD34	10	15,63%

Fonte: Autoria Própria (2023)

Tabela 8: Distribuição de classes do projeto GSE71449

Classe	Número de amostras	%
JMML_LIN28_high	20	44,44%
JMML_LIN28_low	18	40,00%
MML_LIN28_high	1	2,22%
normal	6	13,33%

Fonte: Autoria Própria (2023)

Tabela 9: Distribuição de classes do projeto GSE28497

Classe	Número de amostras	%
B-CELL_ALL	74	26,33%
B-CELL_ALL_ETV6-RUNX1	53	18,86%
B-CELL_ALL_HYPERDIP	51	18,15%
B-CELL_ALL_HYPO	18	6,41%
B-CELL_ALL_MLL	17	6,05%
B-CELL_ALL_T-ALL	46	16,37%
B-CELL_ALL_TCF3-PBX1	22	7,83%

Fonte: Autoria Própria (2023)

APÊNDICE B – CONFIGURAÇÕES DE HIPER-PARÂMETROS DAS TÉCNICAS DE APRENDIZADO DE MÁQUINA

B.1 RANDOM FOREST

•Pacote: sklearn.ensemble.RandomForestClassifier

•Parametros: n_estimators= 1000, n_estimators= 1000, criterion= 'gini', max_depth= 50, max_depth= 50, min_samples_split= 2, min_samples_leaf= 1, min_samples_leaf= 1, min_weight_fraction_leaf= 0.0, max_features= 'auto', max_leaf_nodes= None, min_impurity_decrease= 0.0, bootstrap= True, oob_score= False, n_jobs= None, warm_start= False, class_weight= None, ccp_alpha= 0.0, max_samples= None;

B.2 SVM

•Pacote: sklearn.svm.SVC

•Parâmetros: C= 1.0, kernel= 'linear', degree= 3, gamma= 'scale', coef0= 0.0, shrinking= True, probability= True, tol= 0.001, cache_size= 200, class_weight= None, max_iter= -1, decision_function_shape= 'ovr', break_ties= False

B.3 DECISION TREE

•Pacote: sklearn.tree.DecisionTreeClassifier

•Parâmetros: criterion= 'gini', splitter= 'best', max_depth= None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features= None, max_leaf_nodes= None, min_impurity_decrease= 0.0, class_weight= None, ccp_alpha= 0.0

B.4 ADABOOSTING

- Pacote: sklearn.ensemble.AdaBoostClassifier
- Parâmetros: estimator= DecisionTreeClassifier(), n_estimators= 100, learning_rate= 1.0, algorithm= 'SAMME.R'

APÊNDICE C – RESULTADO DOS EXPERIMENTOS

Tabela 10: Resultados do experimento realizado no projeto GSE87070

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,977	0,978	0,977	0,977	**54675**
SVM (C)	Todas as features	**0,989**	**0,990**	**0,989**	**0,989**	**54675**
DT (C)	Todas as features	0,965	0,965	0,965	0,965	**54675**
AB (C)	Todas as features	0,951	0,951	0,951	0,951	**54675**
RF (C)	Correlação/variância	0,971	0,971	0,971	0,971	**28887**
SVM (C)	Correlação/variância	**0,989**	**0,990**	**0,989**	**0,989**	**28887**
DT (C)	Correlação/variância	0,916	0,916	0,916	0,915	**28887**
AB (C)	Correlação/variância	0,905	0,905	0,905	0,905	**28887**
RF (C)	Embedded	0,982	0,983	0,982	0,982	4279
SVM (C)	Embedded	**0,991**	**0,991**	**0,991**	**0,991**	18422
DT (C)	Embedded	0,960	0,960	0,960	0,960	**8**
AB (C)	Embedded	0,951	0,951	0,951	0,951	**8**
RF (C)	Ambos	0,979	0,979	0,979	0,979	3172
SVM (C)	Ambos	**0,988**	**0,988**	**0,988**	**0,988**	9099
DT (C)	Ambos	**0,988**	**0,988**	**0,988**	**0,988**	**17**
AB (C)	Ambos	0,907	0,906	0,907	0,906	**17**

Fonte: Autoria Própria (2023)

Tabela 11: Resultados do experimento realizado no projeto GSE13159

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,826	0,820	0,826	0,799	**54675**
SVM (C)	Todas as features	0,873	0,876	0,873	0,872	**54675**
DT (C)	Todas as features	0,768	0,770	0,768	0,766	**54675**
AB (C)	Todas as features	0,765	0,768	0,765	0,763	**54675**
RF (H)	Todas as features	0,816	0,834	0,816	0,793	**54675**
SVM (H)	Todas as features	**0,875**	**0,877**	**0,875**	**0,874**	**54675**
DT (H)	Todas as features	0,759	0,762	0,759	0,756	**54675**
AB (H)	Todas as features	0,760	0,762	0,760	0,757	**54675**
RF (C)	Correlação/variância	0,788	0,755	0,788	0,745	**31728**
SVM (C)	Correlação/variância	**0,876**	**0,879**	**0,876**	**0,876**	**31728**
DT (C)	Correlação/variância	0,717	0,724	0,717	0,718	**31728**
AB (C)	Correlação/variância	0,721	0,724	0,721	0,721	**31728**
RF (H)	Correlação/variância	0,767	0,741	0,767	0,728	**31728**
SVM (H)	Correlação/variância	0,872	0,875	0,872	0,872	**31728**
DT (H)	Correlação/variância	0,704	0,708	0,704	0,704	**31728**
AB (H)	Correlação/variância	0,710	0,717	0,710	0,710	**31728**
RF (C)	Embedded	0,858	0,859	0,858	0,843	8105
SVM (C)	Embedded	0,872	0,875	0,872	0,871	12534
DT (C)	Embedded	0,762	0,768	0,762	0,762	154
AB (C)	Embedded	0,765	0,771	0,765	0,765	154
RF (H)	Embedded	0,848	0,858	0,848	0,835	24981
SVM (H)	Embedded	**0,874**	**0,876**	**0,874**	**0,873**	17066
DT (H)	Embedded	0,757	0,760	0,757	0,756	**146**
AB (H)	Embedded	0,457	0,369	0,457	0,380	**146**
RF (C)	Ambos	0,833	0,820	0,833	0,806	4740
SVM (C)	Ambos	**0,877**	**0,881**	**0,877**	**0,877**	4537
DT (C)	Ambos	0,723	0,727	0,723	0,723	180
AB (C)	Ambos	0,723	0,726	0,723	0,722	185
RF (H)	Ambos	0,810	0,795	0,810	0,780	14481
SVM (H)	Ambos	0,872	0,875	0,872	0,871	9205
DT (H)	Ambos	0,716	0,722	0,716	0,715	178
AB (H)	Ambos	0,715	0,725	0,715	0,716	**173**

Fonte: Autoria Própria (2023)

Tabela 12: Resultados do experimento realizado no projeto GSE13164

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,828	0,823	0,828	0,811	**1480**
SVM (C)	Todas as features	**0,832**	**0,836**	**0,832**	**0,831**	**1480**
DT (C)	Todas as features	0,717	0,721	0,717	0,715	**1480**
AB (C)	Todas as features	0,727	0,725	0,727	0,722	**1480**
RF (H)	Todas as features	0,808	0,816	0,808	0,790	**1480**
SVM (H)	Todas as features	0,816	0,825	0,816	0,818	**1480**
DT (H)	Todas as features	0,699	0,713	0,699	0,701	**1480**
AB (H)	Todas as features	0,699	0,708	0,699	0,700	**1480**
RF (C)	Correlação/variância	**0,826**	**0,827**	**0,826**	**0,807**	**656**
SVM (C)	Correlação/variância	0,799	0,807	0,799	0,800	**656**
DT (C)	Correlação/variância	0,699	0,704	0,699	0,698	**656**
AB (C)	Correlação/variância	0,700	0,704	0,700	0,699	**656**
RF (H)	Correlação/variância	0,786	0,816	0,786	0,762	**656**
SVM (H)	Correlação/variância	0,787	0,796	0,787	0,788	**656**
DT (H)	Correlação/variância	0,687	0,697	0,687	0,688	**656**
AB (H)	Correlação/variância	0,689	0,700	0,689	0,690	**656**
RF (C)	Embedded	0,833	0,825	0,833	0,819	453
SVM (C)	Embedded	0,826	0,831	0,826	**0,826**	455
DT (C)	Embedded	0,712	0,717	0,712	0,711	123
AB (C)	Embedded	0,728	0,730	0,728	0,725	124
RF (H)	Embedded	**0,835**	**0,838**	**0,835**	0,823	867
SVM (H)	Embedded	0,815	0,824	0,815	0,817	710
DT (H)	Embedded	0,697	0,705	0,697	0,697	123
AB (H)	Embedded	0,692	0,701	0,692	0,692	**120**
RF (C)	Ambos	**0,827**	**0,831**	**0,827**	**0,812**	186
SVM (C)	Ambos	0,795	0,803	0,795	0,796	179
DT (C)	Ambos	0,681	0,689	0,681	0,681	91
AB (C)	Ambos	0,705	0,712	0,705	0,705	**84**
RF (H)	Ambos	0,815	0,827	0,815	0,803	367
SVM (H)	Ambos	0,788	0,798	0,788	0,790	317
DT (H)	Ambos	0,695	0,705	0,695	0,696	119
AB (H)	Ambos	0,687	0,699	0,687	0,688	120

Fonte: Autoria Própria (2023)

Tabela 13: Resultados do experimento realizado no projeto GSE9476

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,968	0,974	0,968	0,968	**22283**
SVM (C)	Todas as features	**0,984**	**0,988**	**0,984**	**0,985**	**22283**
DT (C)	Todas as features	0,749	0,822	0,749	0,755	**22283**
AB (C)	Todas as features	0,701	0,775	0,701	0,703	**22283**
RF (C)	Correlação/variância	0,969	0,972	0,969	0,967	**1447**
SVM (C)	Correlação/variância	**1,000**	**1,000**	**1,000**	**1,000**	**1447**
DT (C)	Correlação/variância	0,749	0,792	0,749	0,753	**1447**
AB (C)	Correlação/variância	0,766	0,801	0,766	0,764	**1447**
RF (C)	Embedded	0,984	0,988	0,984	0,985	3159
SVM (C)	Embedded	**1,000**	**1,000**	**1,000**	**1,000**	7578
DT (C)	Embedded	0,779	0,832	0,779	0,786	**4**
AB (C)	Embedded	0,670	0,747	0,670	0,682	**4**
RF (C)	Ambas	0,984	0,988	0,984	0,985	291
SVM (C)	Ambas	**1,000**	**1,000**	**1,000**	**1,000**	375
DT (C)	Ambas	0,733	0,781	0,733	0,722	**5**
AB (C)	Ambas	0,766	0,806	0,766	0,761	**5**

Fonte: Autoria Própria (2023)

Tabela 14: Resultados do experimento realizado no projeto GSE71449

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,489	0,418	0,489	0,447	**52200**
SVM (C)	Todas as features	0,689	0,699	0,689	0,677	**52200**
DT (C)	Todas as features	0,867	**0,877**	0,867	0,861	**52200**
AB (C)	Todas as features	0,800	0,822	0,800	0,801	**52200**
RF (H)	Todas as features	0,489	0,418	0,489	0,439	**52200**
SVM (H)	Todas as features	0,711	0,727	0,711	0,701	**52200**
DT (H)	Todas as features	0,822	0,862	0,822	0,833	**52200**
AB (H)	Todas as features	**0,889**	**0,877**	**0,889**	**0,877**	**52200**
RF (C)	Correlação/variância	0,511	0,411	0,511	0,452	**1325**
SVM (C)	Correlação/variância	0,622	0,581	0,622	0,580	**1325**
DT (C)	Correlação/variância	0,400	0,397	0,400	0,391	**1325**
AB (C)	Correlação/variância	0,356	0,386	0,356	0,354	**1325**
RF (H)	Correlação/variância	0,578	0,450	0,578	0,493	**1325**
SVM (H)	Correlação/variância	**0,644**	**0,611**	**0,644**	**0,607**	**1325**
DT (H)	Correlação/variância	0,467	0,467	0,467	0,459	**1325**
AB (H)	Correlação/variância	0,467	0,545	0,467	0,462	**1325**
RF (C)	Embedded	0,644	0,625	0,644	0,609	4760
SVM (C)	Embedded	0,733	0,744	0,733	0,719	18875
DT (C)	Embedded	0,733	0,731	0,733	0,728	**3**
AB (C)	Embedded	0,800	0,822	0,800	0,801	**3**
RF (H)	Embedded	0,622	0,524	0,622	0,558	4872
SVM (H)	Embedded	0,733	0,756	0,733	0,719	25273
DT (H)	Embedded	0,822	0,876	0,822	0,822	4
AB (H)	Embedded	**0,889**	**0,877**	**0,889**	**0,877**	4
RF (C)	Ambos	0,511	0,428	0,511	0,455	416
SVM (C)	Ambos	0,600	0,557	0,600	0,559	408
DT (C)	Ambos	0,422	0,436	0,422	0,421	**6**
AB (C)	Ambos	0,356	0,386	0,356	0,354	**6**
RF (H)	Ambos	0,511	0,411	0,511	0,448	624
SVM (H)	Ambos	**0,622**	**0,587**	**0,622**	**0,585**	540
DT (H)	Ambos	0,533	0,535	0,533	0,518	7
AB (H)	Ambos	0,467	0,534	0,467	0,460	**6**

Fonte: Autoria Própria (2023)

Tabela 15: Resultados do experimento realizado no projeto GSE28497

Técnica	Seleção de Features	Acc	Prec	Rec	F1	Features
RF (C)	Todas as features	0,844	0,802	0,844	0,815	**22283**
SVM (C)	Todas as features	**0,904**	**0,894**	**0,904**	**0,894**	**22283**
DT (C)	Todas as features	0,719	0,718	0,719	0,714	**22283**
AB (C)	Todas as features	0,716	0,706	0,716	0,706	**22283**
RF (C)	Correlação/variância	0,851	0,814	0,851	0,824	**9250**
SVM (C)	Correlação/variância	**0,897**	**0,883**	**0,897**	**0,884**	**9250**
DT (C)	Correlação/variância	0,719	0,731	0,719	0,714	**9250**
AB (C)	Correlação/variância	0,708	0,716	0,708	0,703	**9250**
RF (C)	Embedded	0,851	0,803	0,851	0,823	4110
SVM (C)	Embedded	**0,907**	**0,900**	**0,907**	**0,899**	7393
DT (C)	Embedded	0,726	0,721	0,726	0,717	**26**
AB (C)	Embedded	0,712	0,705	0,712	0,704	**26**
RF (C)	Ambas	0,854	0,811	0,854	0,827	1637
SVM (C)	Ambas	**0,886**	**0,872**	**0,886**	**0,873**	2873
DT (C)	Ambas	0,701	0,700	0,701	0,696	**27**
AB (C)	Ambas	0,712	0,719	0,712	0,706	**27**

Fonte: Autoria Própria (2023)

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PPGI
programa de pós-graduação
em informática

@ppgicp

www ppgi-cp@utfpr.edu.br
portal.utfpr.edu.br/cp/ppgi