

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Física

Miguel Felipe de Abreu Rocha e Senra

**PREDIÇÃO DA ENERGIA DE LIGAÇÃO DO PÓSITRON
COM MOLÉCULAS:**
**uma análise exploratória utilizando
algoritmos de machine learning**

Belo Horizonte
2024

Miguel Felipe de Abreu Rocha e Senra

**PREDIÇÃO DA ENERGIA DE LIGAÇÃO DE PÓSITRON
COM MOLÉCULAS: uma análise exploratória utilizando
algoritmos de machine learning**

Dissertação apresentada ao Programa de Pós-Graduação em Física do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para obtenção do título de Mestre em Ciências.

Orientador: José Rachid Mohallem

Coorientador: Paulo Henrique Ribeiro Amaral

Belo Horizonte
2024

Dados Internacionais de Catalogação na Publicação (CIP)

S478p Senra, Miguel Felipe de Abreu Rocha e.
Predição da energia de ligação do pósitron com moléculas: uma análise exploratória utilizando algoritmos de machine learning / Miguel Felipe de Abreu Rocha e Senra. – 2024.
96 f. : il.

Orientador: José Rachid Mohallem.
Coorientador: Paulo Henrique Ribeiro Amaral.
Dissertação (mestrado) – Universidade Federal de Minas Gerais,
Departamento de Física.
Bibliografia: f. 92-96.

1. Pósitrons. 2. Energia de ligação. 3. Aprendizado do computador. I. Título.
II. Mohallem, José Rachid. III. Amaral, Paulo Henrique Ribeiro. IV.
Universidade Federal de Minas Gerais, Departamento de Física.

CDU – 539.124.6 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

FOLHA DE APROVAÇÃO

A presente dissertação, intitulada “**Predição da Energia de Ligação de Pósitron com Moléculas: Uma Análise Exploratória Utilizando Algoritmos de Machine Learning**”, de autoria de **MIGUEL FELIPE DE ABREU ROCHA E SENRA**, submetida à Comissão Examinadora, abaixo-assinada, foi aprovada para obtenção do grau de **MESTRE EM FÍSICA** em dezenove de agosto de 2024.

Prof. José Rachid Mohallem
Orientador do aluno
Departamento de Física/UFMG

Dr. Paulo Henrique Ribeiro Amaral
Coorientador do aluno
Departamento de Física/UFMG

Prof. Maurício Veloso Brant Pinheiro
Departamento de Física/UFMG

Prof. Omar Paranaíba Vilela Neto
Departamento de Ciência da Computação/UFMG

Prof. Juan Carlos González Pérez
Departamento de Física/UFMG



Documento assinado eletronicamente por **Juan Carlos Gonzalez Perez, Professor do Magistério Superior**, em 21/08/2024, às 09:08, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Mauricio Veloso Brant Pinheiro, Professor do Magistério Superior**, em 21/08/2024, às 19:50, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Jose Rachid Mohallem, Professor do Magistério Superior**, em 23/08/2024, às 13:35, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **Omar Paranaiba Vilela Neto, Professor do**



Magistério Superior, em 23/08/2024, às 15:08, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufmg.br/sei/controlador_externo.php?](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **3475129** e o código CRC **321436E2**.

Referência: Processo nº 23072.246537/2024-51

SEI nº 3475129

*Dedico este trabalho aos que me permitiram
perguntar e descobrir,
tal qual se permite às crianças que perguntam
o porquê do céu ser azul.*

Agradecimentos

Essa foi, sem dúvida, uma das experiências mais desafiadoras que enfrentei em minha vida. Iniciando ainda durante a pandemia do COVID-19, passei por inúmeras dificuldades: espirituais, intelectuais e emocionais. Se hoje, mantenho cada uma dessas esferas em vigor, é aqui que demonstro minha gratidão aos envolvidos.

Aos meus amigos, que sempre me apoiaram em discussões, aconselhamentos, momentos de descontração e crises: boa parte do orgulho que sinto por quem sou e pelo caminho que trilhei é graças a vocês, a família que escolhi.

Aos meus pais, devo-lhes a liberdade de ser insistente em minhas perguntas e curiosidades sobre a vida, além da confiança de saber que, em tempos de incerteza e falta de clareza, eles continuam sendo meus pilares.

À minha parceira de vida, obrigado por toda paciência, apoio e por sempre ter sido minha melhor amiga. Sem seu suporte, eu teria perdido minha sanidade e boa-aventurança.

Aos integrantes do LAtME e principalmente aos meus orientadores, José Rachid Mohallem e Paulo Henrique Ribeiro Amaral, meu sincero agradecimento por todo auxílio, persistência, instrução e ensinamento. Tenham certeza de que esses conhecimentos e trocas não serão esquecidos, e meu desenvolvimento profissional se deve inteiramente à fagulha que vocês me ajudaram a acender.

À UFMG e ICEX, meu profundo agradecimento por todo auxílio e orientação no âmbito institucional, além do suporte estrutural que tornou possível a realização deste trabalho. Por fim, é imprescindível reconhecer a FAPEMIG pela bolsa de pesquisa que possibilitou minha dedicação integral ao aprendizado e ao trabalho necessário para o desenvolvimento desta dissertação.

Você deve aproveitar ao máximo os desvios, pois é onde encontrará as coisas mais importantes do que aquilo que realmente deseja.
(Yoshihiro Togashi)

Resumo

O objetivo desta dissertação é empregar técnicas de aprendizado de máquina para investigar a importância de propriedades moleculares específicas na determinação da energia de ligação do pósitron com moléculas. Foram utilizados dados experimentais de energias de ligação de pósitrons (PBEs) e propriedades moleculares como polarizabilidade, momento de dipolo e número de ligações π . Cinco algoritmos de aprendizado de máquina foram explorados: máquinas de vetores de suporte para regressão (SVR), regressão polinomial com regularização *ridge*, XGBoost, perceptron de multicamadas (RNA) e um modelo *ensemble*.

Os dados foram divididos em três grupos: moléculas apolares, polares e uma mistura dos dois tipos. Diferentes conjuntos de propriedades moleculares foram testados para cada grupo. A avaliação e ajuste dos algoritmos foram realizados utilizando validação cruzada aninhada e otimização bayesiana, respectivamente, com a métrica de referência sendo o erro percentual absoluto médio. Além disso foi utilizado a abordagem SHAP para interpretabilidade dos modelos utilizados.

Os resultados sugerem que a polarizabilidade anisotrópica é um descritor importante para moléculas polares, bem como as ligações π para as moléculas apolares, refletindo a influência da estrutura e distribuição espacial da nuvem eletrônica no cálculo energia de ligação do pósitron. Para as moléculas polares, observou-se também a melhoria do desempenho dos algoritmos quando utiliza-se o potencial de ionização como atributo. Este estudo demonstra que técnicas de aprendizado de máquina podem fornecer pistas valiosas sobre os mecanismos físicos envolvidos na ligação do pósitron com moléculas, complementando abordagens teóricas e experimentais.

No entanto, limitações como a escassez de dados experimentais precisam ser superadas em pesquisas futuras. Trabalhos futuros devem focar na aquisição de mais dados, exploração de outros modelos específicos para poucas amostras e técnicas avançadas de interpretabilidade de modelos.

Palavras-chave: Pósitron, Energia de Ligação, Aprendizado de Máquina, Propriedades Moleculares, Polarizabilidade Anisotrópica, Ligações π .

Abstract

The goal of this dissertation is to employ machine learning techniques to investigate the importance of specific molecular properties in determining the binding energy of the positron with molecules. Experimental data on positron binding energies (PBEs) and molecular properties such as polarizability, dipole moment, and the number of π bonds were used. Five machine learning algorithms were explored: support vector machines for regression (SVR), polynomial regression with ridge regularization, XGBoost, multilayer perceptron, and an ensemble model.

The data were divided into three groups: nonpolar molecules, polar molecules, and a mix of both types. Different sets of molecular properties were tested for each group. The evaluation and tuning of the algorithms were performed using nested cross-validation and Bayesian optimization, respectively, with the benchmark metric being the mean absolute percentage error. Additionally, the SHAP approach was used for interpretability of the models used.

The results suggest that anisotropic polarizability is an important descriptor for polar molecules, as well as π bonds for nonpolar molecules, reflecting the influence of structure and spatial distribution of the electron cloud in calculating positron binding energy. For polar molecules, it was also observed that the performance of algorithms improves when ionization potential is used as an attribute. This study demonstrates that machine learning techniques can provide valuable insights about the physical mechanisms involved in positron binding with molecules, complementing theoretical and experimental approaches.

However, limitations such as the scarcity of experimental data need to be overcome in future research. Future work should focus on acquiring more data, exploring other models specific to few samples, and advanced model interpretability techniques.

Keywords: Positron, Binding Energy, Machine Learning, Molecular Properties, Anisotropic Polarizability, π bonds.

Lista de abreviaturas e siglas

aug-cc-pVDZ	<i>Augmented Correlation-consistent Valence Double Zeta</i>
B3LYP	<i>Becke 3-parameter Lee–Yang–Parr</i>
CART	<i>Classification and Regression Trees</i>
D	Debye
eV	Elétron-volt
IA	Inteligência Artificial
IQR	Desvio Interquartil
LASSO	<i>Least Absolute Shrinkage and Selection Operator</i>
MAPE	<i>Mean Absolute Percentage Error</i>
meV	Milielétron-volt
ML	<i>Machine Learning</i>
MLP	<i>Multi Layer Perceptron</i>
PBE	<i>Positron Binding Energy</i>
Ps	Positrônio
ReLU	<i>Rectified Linear Unit</i>
RNA	Rede Neural Artificial
SHAP	<i>Shapley Additive Explanations</i>
SVM	<i>Support Vector Machine</i>
SVR	<i>Support Vector Regressor</i>
VFR	<i>Vibrational Feshbach Resonance</i>
XGBoost	<i>Extreme Gradient Boost</i>

Lista de símbolos

E_{lt}	Energia do limite de formação do Ps
I_p	Potencial de ionização
E_{Ps}	Energia do estado fundamental de formação do Ps
λ_D	Taxa de Dirac
r_o	Raio clássico do elétron
c	Velocidade da luz no vácuo
n_e	Densidade de elétrons
n	Densidade de átomos ou moléculas
Z	Número de elétrons
Z_{eff}	Número efetivo de elétrons
λ	Taxa de aniquilação
ϵ_ν	Energia de ressonância vibracional associada ao modo ν
ω_ν	Energia do modo vibracional ν
ϵ_b	Energia de ligação do pósitron
ϵ	Energia de incidência do pósitron
α	Polarizabilidade elétrica isotrópica
N_π	Número de ligações π da molécula
μ	Momento de dipolo elétrico
$\overleftrightarrow{\alpha}$	Tensor de polarizabilidade anisotrópica
$\text{\AA}^3$	Angstrom cúbico
e	Elétron
r	Posição da carga
$\rho(r)$	Densidade de carga em r

q	Carga elétrica
μ_{ind}	Momento de dipolo elétrico induzido
E	Campo elétrico
μ_x	Componente x do momento de dipolo elétrico induzido
α_{ij}	Componente ij do tensor de polarizabilidade elétrica anisotrópica
α'_{ij}	Componente ij do tensor de polarizabilidade elétrica anisotrópica diagonalizado
ε_0	Constante de permissividade elétrica no vácuo
β	Estatística a ser mensurada
$\hat{\beta}$	Estatística prevista a partir de amostragens
ϵ	Erro irreduzível
x_i	Variável explicativa da i -ésima amostra
y_i	Valor da variável alvo da i -ésima amostra
$\hat{y}_i$	Valor previsto da variável alvo da i -ésima amostra
S	Função de erro total
a	Coefficiente linear
b_i	Coefficiente da variável x da i -ésima amostra
J	Função de custo
W	Função de regularização
η	Taxa de aprendizagem
h_i	Estimador dos erros da rodada i
ξ e ξ^*	Variáveis de folga
φ_i	Valor Shapley da i -ésima propriedade
v	Valor de performance
C	Coalizão
θ	Média aritmética de um conjunto de valores
σ	Desvio padrão de um conjunto de valores

Sumário

1	INTRODUÇÃO	15
1.1	Estados ligados de pósitron com átomos e moléculas	17
1.2	Ressonância Vibracional de Feshbach (VFR)	17
2	PROPRIEDADES E POTENCIAIS MOLECULARES	25
2.1	Potencial de Ionização	25
2.2	Momento de Dipolo e Polarizabilidade Elétrica	26
2.2.1	Polarizabilidade Elétrica	27
2.3	Relação entre propriedades moleculares e ligação do pósitron a moléculas	28
3	APRENDIZADO DE MÁQUINA	31
3.1	Avaliação de Performance de Modelos	32
3.1.1	Viés e Variância	33
3.1.2	Dilema Viés e Variância	34
3.1.3	Métodos de Avaliação	37
3.1.3.1	Método Holdout	37
3.1.3.2	Método de Validação Cruzada	38
3.1.3.3	Método de Validação Cruzada Aninhada	40
3.2	Algoritmos de Aprendizado de Máquina	44
3.2.1	Regressão Linear e Polinomial	44
3.2.2	Regularizações L1 e L2	45
3.2.3	Árvores de Decisão	46
3.2.3.1	XGBoost	49
3.2.4	Máquina de Vetores de Suporte	50
3.2.5	Redes Neurais Artificiais	54
3.2.5.1	Estrutura de uma RNA	55
3.3	Importância de Atributos via SHAP	58
3.3.1	Valores SHAPLEY e SHAP	58
3.4	Escalonamento de Atributos	59
3.4.1	Padronização	60
3.4.2	Normalização	61
3.4.3	Escalonamento Robusto	61
3.5	Otimização de hiperparâmetros	62
3.5.1	Otimização Bayesiana	62

4	METODOLOGIA	67
5	RESULTADOS	75
5.1	Moléculas Apolares	75
5.2	Moléculas Polares	77
5.3	Moléculas Mistas	81
5.4	Comparação dos resultados com literatura	86
6	CONCLUSÕES	90
	REFERÊNCIAS	92

1 Introdução

O pósitron, também conhecido como anti-elétron, foi previsto teoricamente em 1928 por Paul Dirac ao obter níveis de energia negativos em sua teoria quântica relativística para o elétron [1]. Em agosto de 1932, Carl Anderson realizou um experimento com raios cósmicos no qual observou uma trajetória oposta à esperada para um elétron e possuindo uma curvatura de tamanho pelo menos dez vezes maior do que seria esperado para um próton, como mostrado na Figura 1, que ele identificou como sendo a trajetória do anti-elétron. Anderson recebeu o Nobel em 1936 pela descoberta experimental do pósitron. [2]

A descoberta do pósitron teve um impacto profundo na física teórica. Primeiramente, ao fornecer evidências concretas para a existência da antimatéria, a descoberta confirmou a teoria proposta por Dirac, revolucionando nossa compreensão dos componentes fundamentais do universo.

Essa descoberta abriu uma nova área de pesquisa dedicada à antimatéria, impulsionando inúmeros experimentos para estudar as propriedades dessas partículas e investigar as simetrias e leis fundamentais da física.

Ademais, a existência do pósitron desafiou as teorias predominantes da época e

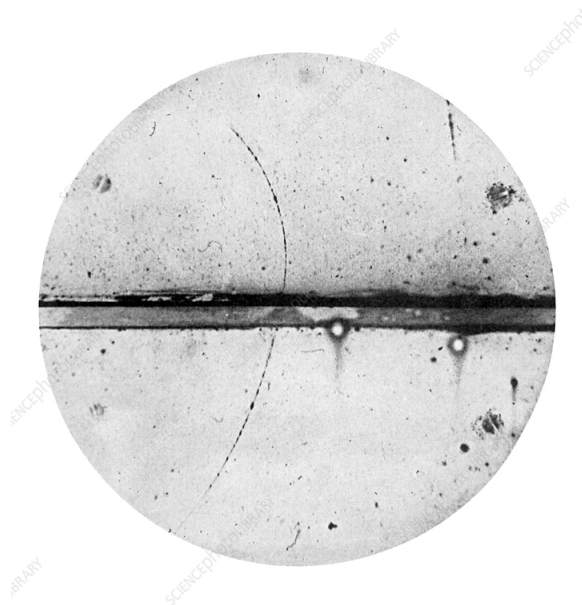


Figura 1 – Imagem da câmara de nuvem de Anderson de 1932 mostrando pela primeira vez a existência do anti-elétron. A partícula entra por baixo, atinge a placa de chumbo no meio e perde energia como pode ser visto pela curvatura maior da parte superior da pista. [2]

estimulou o desenvolvimento da teoria quântica de campos. A equação de Dirac, responsável por prever a existência do pósitron, serviu como base para a formulação das teorias quânticas de campos, as quais descrevem o comportamento das partículas e suas interações.

Dentre as inúmeras contribuições significativas da descoberta do pósitron, está a sua aplicação em técnicas de imagem médica, como a tomografia por emissão de pósitrons (PET). Devido às propriedades únicas do pósitron, incluindo sua capacidade de aniquilar-se com um elétron, tornou-se possível o diagnóstico e monitoramento de diversas doenças, tais como câncer, doenças cardiovasculares e distúrbios neurológicos [3].

Com base em suas diversas aplicações e possibilidades, torna-se importante a exploração e o estudo dos mecanismos de interação do pósitron com a matéria, isto é, átomos, moléculas e sólidos. Esse campo de pesquisa oferece a oportunidade de um maior entendimento dos fenômenos envolvidos e pode fornecer pistas importantes para o desenvolvimento de novas tecnologias e aplicações.

O foco do presente trabalho de pesquisa reside nas energias de ligação do pósitron com átomos e moléculas, bem como na relação e explicação dessas energias através de propriedades atômicas e moleculares. Atualmente, a literatura apresenta um tratamento *ab initio* robusto para a quantificação dessa energia em átomos, no entanto, existem escassos experimentos que permitem a verificação desses fenômenos.

No contexto molecular, observa-se uma abundância de dados experimentais, mas uma carência de teorias que expliquem satisfatoriamente o fenômeno. Tanto para moléculas quanto para átomos, enfrentam-se desafios significativos na experimentação e no desenvolvimento teórico, os quais serão detalhados posteriormente. Como alternativa, especialmente para moléculas, recorre-se a métodos empíricos e semi-empíricos. Existem estudos que buscam compreender e estabelecer relações físicas e matemáticas a partir dos dados experimentais.

Este trabalho tem como objetivo empregar técnicas de aprendizado de máquina para investigar como certos algoritmos percebem a importância de propriedades moleculares específicas, buscando corroborar as afirmações da literatura ou introduzir novas perspectivas e ideias que possam avançar a pesquisa sobre a interação entre moléculas e pósitrons.

As técnicas oriundas do universo de aprendizado de máquina têm sido aplicadas com sucesso em várias áreas da ciência, incluindo a física, atuando como alternativas ou complementos a experimentos, simulações ou descrições analíticas. Essas abordagens são notáveis por proporcionarem redução nos custos computacionais e por oferecerem suporte na formulação de novas teorias e na caracterização de fenômenos.

Embora a maioria dos algoritmos não forneça formas analíticas conclusivas, é possível propor e discernir quais variáveis são preponderantes para o bom desempenho dos modelos, possibilitando a criação de argumentos físicos. Este estudo faz o uso desses

algoritmos de forma exploratória, investigando, por exemplo, se a inclusão de polarizabilidades anisotrópicas como descritor molecular pode ser proveitosa — um aspecto que receberá uma explicação mais detalhada nos capítulos subsequentes.

1.1 Estados ligados de pósitron com átomos e moléculas

Átomos e moléculas neutros podem formar ligações com pósitrons, apesar de não possuírem carga líquida. Isso ocorre devido a características moleculares específicas, como o momento de dipolo e a polarizabilidade, além de fenômenos como a Ressonância Vibracional de Feshbach. Cálculos *ab initio* dentro da teoria de muitos corpos revelaram a possibilidade de estados ligados de pósitrons com diversos átomos [4].

Embora átomos representem estruturas mais simples, a observação experimental de ligações com pósitrons é impraticável, dada a complexidade em formar estados ligados em colisões de dois corpos [5]. Em contraste, com moléculas, é possível medir a energia de ligação do pósitron indiretamente, utilizando-se de modos vibracionais infravermelho e análise das taxas de aniquilação. Atualmente, o grupo de Cassidy *et al* [6] realiza cálculos *ab initio* das energias de ligação em moléculas, empregando a teoria de muitos corpos, embora com eficiência limitada.

1.2 Ressonância Vibracional de Feshbach (VFR)

Estados ressonantes são estados instáveis de sistemas quânticos. Ressonância de Feshbach é um estado imerso no contínuo de um certo grau de liberdade do sistema. Por exemplo, o estado $2s^2$ do átomo de Hélio tem energia maior do que o $1s^1\infty s^1$, logo pode decair para esse último, daí a instabilidade.

Um pósitron de baixa energia pode excitar um estado vibracional de uma molécula e, perdendo sua energia, é capturado por esta. É o caso dos VFRs, muito estudados experimentalmente pelo grupo de Surko *et al* [7–10].

Ao observar a interação do pósitron com moléculas, destaca-se eventualmente uma peculiaridade: a elevada taxa de aniquilação. Inicialmente, essa taxa não estava em consonância com a teoria de colisões. Essa significativa taxa de aniquilação encontra explicação no acoplamento do pósitron às moléculas por meio de ressonância com os modos vibracionais moleculares. [11].

Pósitrons de fontes naturais e convencionais, como radioisótopos, possuem altas energias. Se a energia de incidência do pósitron (ϵ) em uma molécula for maior que a energia do limite de formação do Ps (Positrônio) E_{lt} , isto é, $E_{lt} = I_p - E_{Ps}$, onde I_p é a energia de ionização do alvo e $E_{Ps} = 6.8$ eV é a energia do estado fundamental do átomo Ps, então o processo de aniquilação é através da formação de Ps. Quando não há formação

de Ps, ou seja, $0 < \epsilon < E_{lt}$, a aniquilação ocorre como um resultado da superposição das densidades de elétrons e do pósitron. A taxa de aniquilação λ_D , é a taxa de Dirac, onde para uma aniquilação resultante em dois quanta de raios gama é [1]:

$$\lambda_D = \pi r_o^2 c n_e, \quad (1.1)$$

onde r_o é o raio clássico do elétron, c a velocidade da luz e n_e a densidade de elétrons. É conveniente também normalizar as taxas de aniquilação em gases para a taxa de Dirac, resultando em seu correspondente adimensional o “número efetivo de elétrons”, denominado Z efetivo:

$$Z_{eff} = \frac{\lambda}{\pi r_o^2 c n}, \quad (1.2)$$

onde, de acordo com [12], n representa a densidade de átomos ou moléculas e λ a taxa de aniquilação. Em colisões onde as correlações elétron-pósitron são desconsideradas, espera-se que $\lambda \approx \lambda_D$, tornando Z_{eff} comparável a $Z = n_e/n$, o número de elétrons por alvo. Conforme a teoria de Z_{eff} , não são esperadas altas taxas de aniquilação. Na Figura 2, é feita uma comparação entre o valor esperado pela teoria para as taxas de aniquilação e o que é observado experimentalmente, verifica-se o surgimento de picos onde não deveria haver, esses picos são pontos chaves para o cálculo das energias de ligação do pósitron com moléculas.

Uma condição fundamental para a ocorrência das ressonâncias vibracionais é que deve existir uma ligação entre a partícula e a molécula. Isso possibilita calcular a energia de ligação do pósitron com a molécula em questão. A energia da ressonância vibracional associada ao modo ν , ϵ_ν , é determinada pela conservação de energia, sendo expressa por:

$$\epsilon_\nu = \omega_\nu - \epsilon_b. \quad (1.3)$$

Nesse contexto, ϵ_b representa a energia de ligação do pósitron, e ω_ν corresponde à energia do modo vibracional. Consequentemente, a energia de ligação do pósitron pode ser calculada utilizando a fórmula $\epsilon_b = \omega_\nu - \epsilon_\nu$. A premissa é que os pósitrons exibem uma taxa de aniquilação mais elevada, $Z_{eff}(\epsilon)$, nos picos do espectro infravermelho, ao descontar a energia de ligação do pósitron. Uma ilustração desse fenômeno pode ser observado na Figura 3, onde na parte (b) há diferença na localização dos picos das taxas de aniquilação e modos vibracionais.

Dessa maneira, Danielson *et al.* [17] conseguiram medir de forma indireta a energia de ligação do pósitron com várias moléculas, conforme apresentado na Tabela 1. Esta tabela também fornece valores para algumas propriedades moleculares.

É importante observar que, com base nos dados experimentais, é possível ajustar facilmente uma equação linear válida para os alcanos:

$$\epsilon_b = 12.4\alpha - 69.44 \text{ (meV)}, \quad (1.4)$$

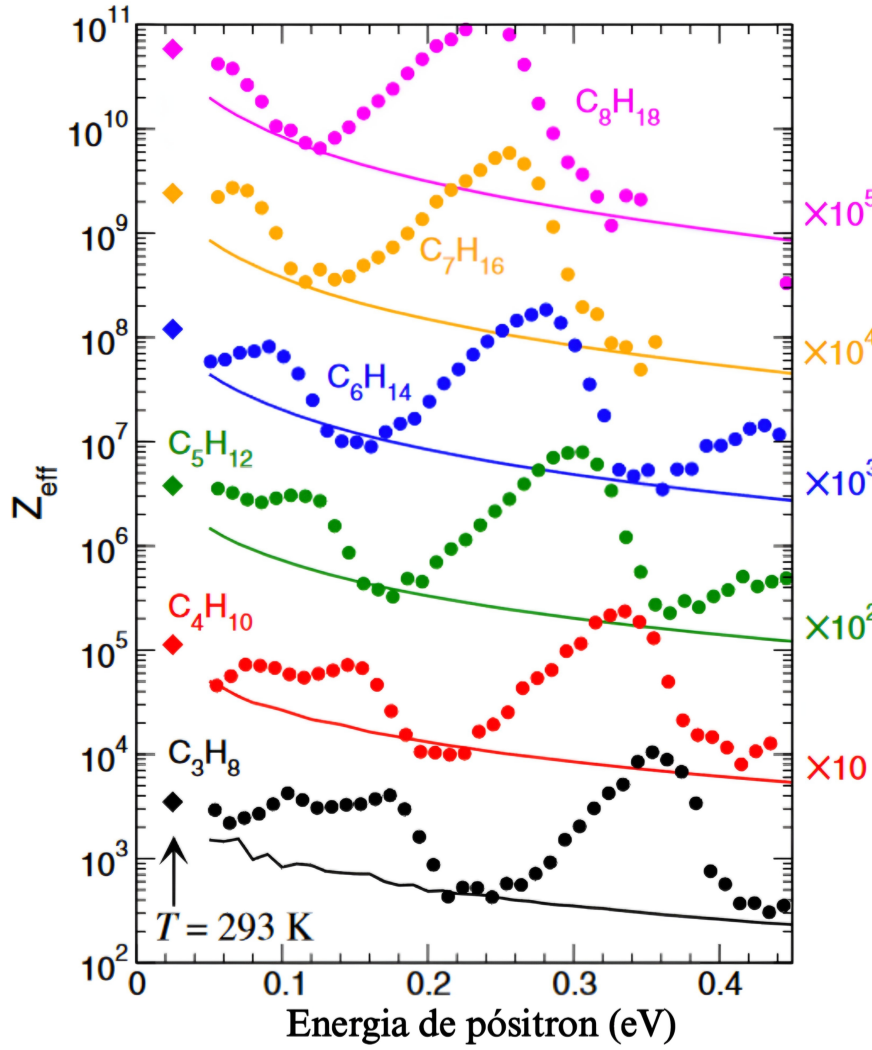


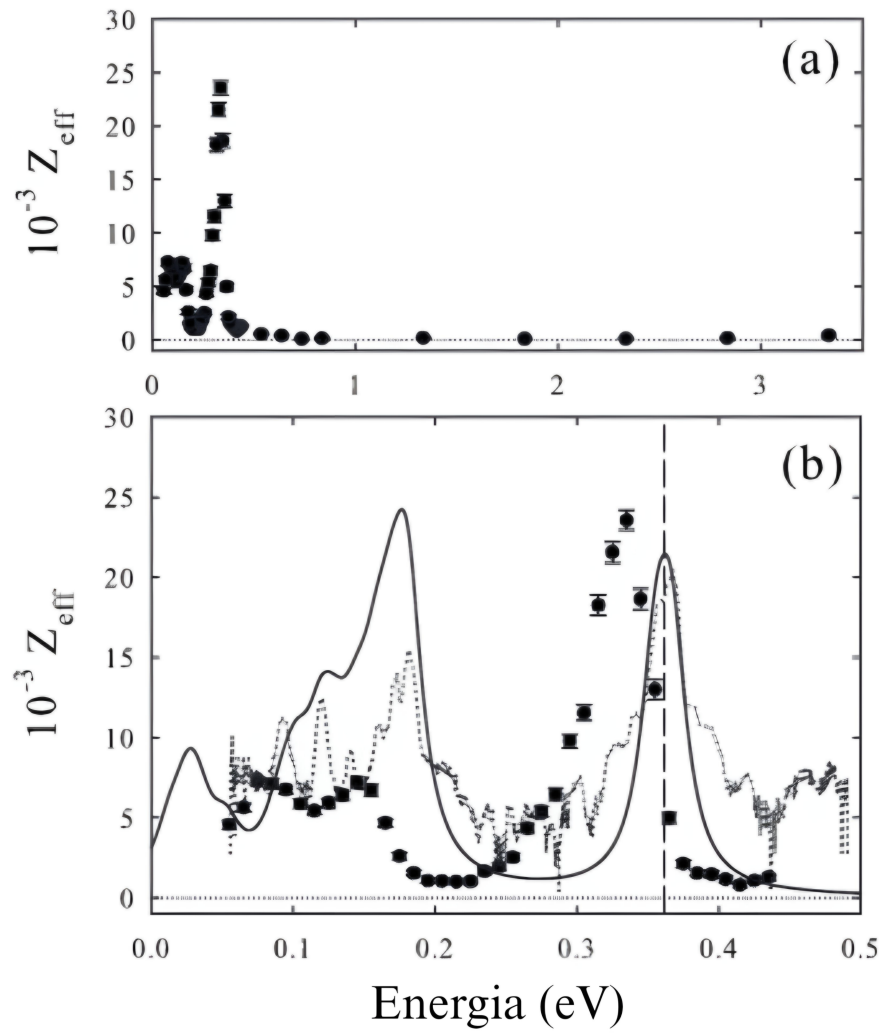
Figura 2 – Z_{eff} calculado em função da energia de pósitrons [13] (curvas sólidas). Círculos sólidos mostram Z_{eff} medido em função da energia do pósitron [14,15]. Para os alcanos maiores que o butano, Z_{eff} foi multiplicado por potências de 10 para maior clareza. [13]

onde α é a polarizabilidade da molécula alvo. Além disso, para determinado conjunto de moléculas, Danielson *et al* [17] perceberam que o acréscimo de um termo contendo as ligações π das moléculas aromáticas torna o ajuste linear ainda mais preciso, sendo a equação dessa reta:

$$\epsilon_b = 12.4(\alpha + 1.6\mu + 2.4N_\pi - 5.6) \text{ (meV)}. \quad (1.5)$$

Na equação 1.5, μ representa o momento de dipolo, incluindo portanto conjuntos de moléculas polares, enquanto N_π denota o número de ligações π presentes na molécula. As linearizações correspondentes estão ilustradas nas Figuras 4 e 5, utilizando as equações 1.4 e 1.5, respectivamente.

Ao observar a Figura 4, nota-se que o ajuste para os alcanos é mais eficaz em com-



v

Figura 3 – Parâmetro de aniquilação $Z_{\text{eff}}(\varepsilon)$ normalizado em função da energia dos pósitrons incidentes para o butano C_4H_{10} ($\bullet$): (a) até o limite de formação de Ps, $E_{lt} = 3.8$ eV; (b) na região das vibrações moleculares, onde os pontos são referentes ao espectro de absorção do infravermelho, a linha é a densidade do modo vibracional e a linha pontilhada é a média da energia dos estiramentos C-H. [16]

paração com outros tipos de moléculas. Embora seja possível criar retas com linearizações específicas para alguns grupos de moléculas, ainda assim podem ocorrer interseções e pontos fora da curva. É importante observar também que o acréscimo das ligações π na Figura 5 resulta em um ajuste mais preciso.

Esses modelos lineares representam as primeiras equações empíricas simples desenvolvidas para quantificar a energia de ligação do pósitron com outras moléculas, com base em propriedades moleculares. Amaral e Mohallem [18], por exemplo, procuraram estabelecer uma convergência entre teoria e valores experimentais ao trabalhar com essas equações empíricas.

A ligação de pósitrons com moléculas é dirigida preponderantemente por seus potenciais atômicos, através de α e μ (que representam a polarizabilidade elétrica e momento de dipolo permanente respectivamente). Para $\mu > 2D$ (Debye), essa propriedade passa a predominar e as energias de ligação crescem bastante [5]. No entanto, outras propriedades surgem como relevantes para determinar a intensidade da ligação. Os trabalhos experimentais, como vimos, sugerem um papel relevante para o número de ligações π (N_π), em moléculas cíclicas. Amaral e Mohallem [18], sugerem que o potencial de ionização, I_p , esteja relacionado com a formação do Ps virtual de modo que o potencial total seja sempre repulsivo em curtas distâncias, permitindo a existência de estados ligados, sugerindo a consideração das quatro propriedades.

Também, como para moléculas polares os resultados são pobres quando se utiliza a polarizabilidade isotrópica α , consideramos que a anisotropia que gera o dipolo μ também deveria ter efeito sobre a polarizabilidade, e introduzimos a polarizabilidade anisotrópica $\vec{\alpha}$, um tensor cujas componentes representam a influência das componentes de um campo elétrico sobre todos os eixos principais de uma molécula. Essas propriedades moleculares serão melhor discutidas no capítulo 2.

O objetivo deste trabalho é explorar as possibilidades oferecidas pelos modelos de aprendizagem de máquina na identificação de propriedades moleculares ou combinações delas que possam contribuir para uma descrição mais precisa, tanto numérica quanto fisicamente do fenômeno da ligação do pósitron a moléculas apolares e polares. Diferentemente dos ajustes mostrados nas Figuras 4 e 5, no contexto deste trabalho não se busca considerar grupos ou famílias de moléculas, tampouco a aromaticidade, a fim de garantir ajustes mais gerais.

Tabela 1 – Valores experimentais para a energia de ligação, ϵ_b (meV)

Nome Molecular	Fórmula	ϵ_b (meV)	Equação 1.5 (meV)	Z	α ($\text{\AA}^3$)	μ (D)	I_p (eV)
Alcanos							
Etano	C_2H_6	≥ 0	-14	18	4.4	0	11.5
Propano	C_3H_8	10	10	26	6.3	0.1	11.1
Butano	C_4H_{10}	35	31	34	8.1	0	10.6
Pentano	C_5H_{12}	60	54	42	10.0	0	10.4
Hexano	C_6H_{14}	80	77	50	11.8	0	10.2
Heptano	C_7H_{16}	105	100	58	13.7	0	9.9
Octano	C_8H_{18}	115	123	66	15.5	0	10.0
Nonano	C_9H_{20}	145	146	74	17.4	0	10.0
Dodecano	$\text{C}_{12}\text{H}_{26}$	220	214	98	22.9	0	9.9
Tetradecano	$\text{C}_{14}\text{H}_{30}$	260	261	114	26.6	0	-
Hexadecano	$\text{C}_{16}\text{H}_{34}$	310	306	130	30.3	0	9.9
Alcanos Deuterados							
Butano-d10	C_4D_{10}	35	31	34	8.1	0	-
Nonano-d20	C_9D_{20}	145	146	74	17.4	0	-
Moléculas relacionadas aos alcanos							
Acetileno	C_2H_2	≥ 0	-28	14	3.3	0	11.4
Etileno	C_2H_4	20	-17	16	4.2	0	10.5
Isopentano	C_5H_{12}	60	57	42	10.0	0.1	10.3
Ciclopropano	C_3H_6	10	1	24	5.7	0	9.9
Ciclohexano	C_6H_{12}	80	68	48	11.1	0	9.9
Aromáticos termo (N_π incluído na equação (5))							
Benzeno	C_6H_6	150	149	42	10.4	0	9.3
Benzeno-d6	C_6D_6	150	149	42	10.4	0	9.3
Naftaleno	C_{10}H_8	300	296	68	16.6	0	8.2
Álcoois							
Metanol	CH_3OH	2	5	18	3.3	1.7	10.9
Etanol	$\text{C}_2\text{H}_5\text{OH}$	45	27	26	5.1	1.7	10.5
Hidrocarbonetos parcialmente halogenados							
Fluoreto de Metila	CH_3F	≥ 0	-3	18	2.4	1.85	12.9
Cloreto de Metila	CH_3Cl	25	23	26	4.4	1.9	11.2
Brometo de Metila	CH_3Br	40	35	44	5.6	1.8	10.5
1-fluoropropano	$\text{C}_3\text{H}_7\text{F}$	35	45	34	6.0	2.0	11.3
2,2-difluoropropano	$\text{C}_3\text{H}_6\text{F}_2$	25	51	42	5.9	2.4	11.4
1-fluorobutano	$\text{C}_4\text{H}_9\text{F}$	70	(27)	42	7.8	-	-
1-fluorohexano	$\text{C}_6\text{H}_{13}\text{F}$	80	(73)	58	11.5	-	-
1-fluorononano	$\text{C}_9\text{H}_{19}\text{F}$	145	(141)	82	17.0	—	—
1-clorohexano	$\text{C}_6\text{H}_{13}\text{Cl}$	175	138	66	13.6	2.0	10.3

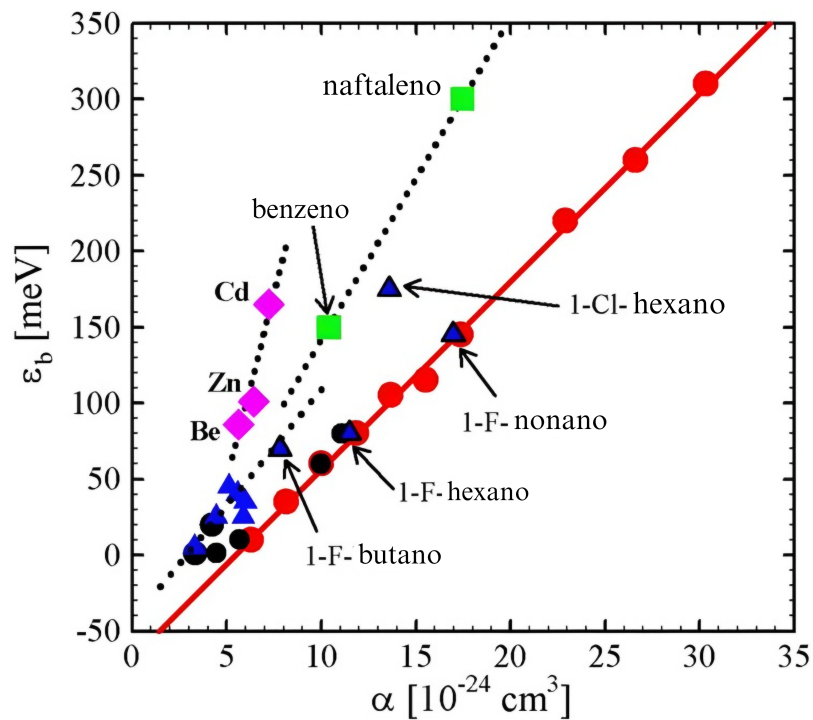


Figura 4 – Energias de ligação de pósitrons medidas, ϵ_b , em função da polarizabilidade molecular, α . Moléculas de alcanos utilizadas em ajuste (círculos); moléculas relacionadas a alcanos (círculos escuros); moléculas com momentos dipolares permanentes (triângulos); e aromáticos com ligações π (quadrados). A linha sólida é a que melhor se ajusta a alcanos (vide equação 1.4). As linhas pontilhadas são guias para mostrar a linearidade de ϵ_b versus α para as diferentes séries de moléculas. Além disso, são mostradas para comparação as energias de ligação calculadas para vários átomos de metal (diamante rosa) [19]. [17]

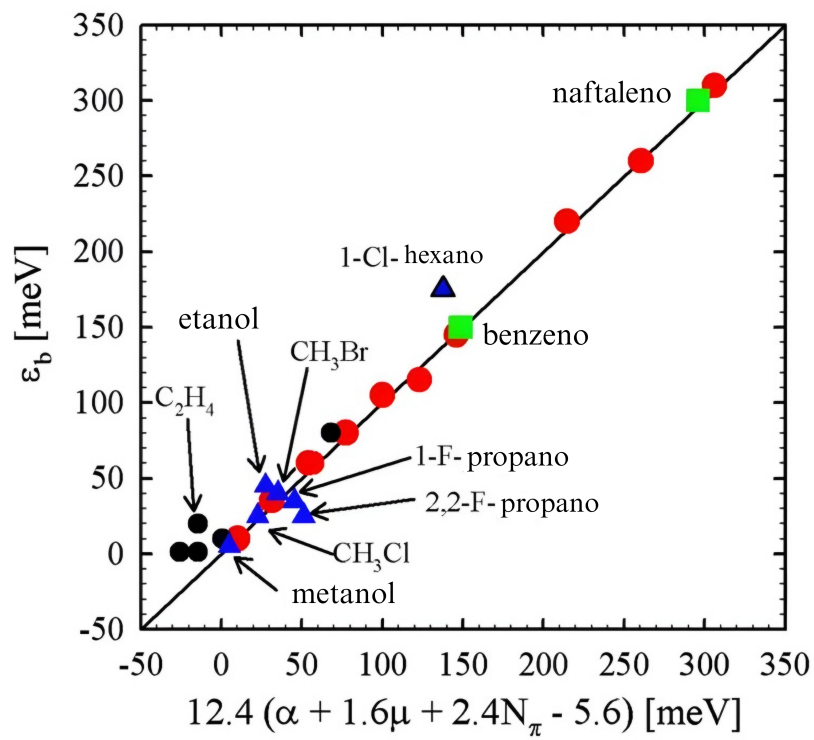


Figura 5 – Energia de ligação descrita pela equação 1.5, usando a polarizabilidade α , momento de dipolo, μ , e o número de ligações π , N_π , para moléculas aromáticas. A notação dos símbolos é a mesma da Figura 4. [17]

2 Propriedades e Potenciais Moleculares

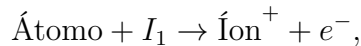
Dada a complexidade dos cálculos *ab initio* para moléculas poliatômicas, especialmente devido à relevância da correlação entre pósitrons e elétrons, é de grande interesse descobrir relações empíricas entre a energia de ligação do pósitron com propriedades moleculares.

Este capítulo tem como objetivo explorar a interação do pósitron com moléculas por meio de potenciais. Realizaremos uma revisão conceitual de algumas propriedades moleculares, como momento de dipolo, interações com cargas, ligações π , polarizabilidades e potenciais de ionização, com a intenção de facilitar a compreensão e contextualização do fenômeno da ligação do pósitron com moléculas polares e apolares.

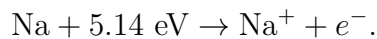
A análise dessas propriedades ao longo deste capítulo não se limita a descrevê-las, mas também a elucidar sua influência na formação e estabilidade das ligações pósitron-molécula.

2.1 Potencial de Ionização

O potencial de ionização de um átomo ou molécula é a energia requerida para a remoção de um elétron daquele sistema. Uma forma de representar esse fenômeno é pela seguinte esquematização:



onde I_1 é o primeiro potencial de ionização. Para o átomo de sódio, por exemplo, temos,



Ou seja, é necessário fornecer uma energia de 5.14 eV para um átomo de sódio para ionizá-lo. Uma consequência da retirada de elétrons subsequentes de um átomo ou molécula neutro é uma contribuição das cargas positivas cada vez maior e portanto, torna-se cada vez mais difícil a retirada de elétrons subsequentes, pelo aumento da atração das cargas positivas e negativas. Isto é, $I_1 < I_2 < I_3 < \dots < I_N$, onde I_N é o potencial de ionização para o n -ésimo elétron de um átomo ou molécula de n elétrons.

Somente pela consideração dos potenciais clássicos envolvendo dipolos permanente ou induzido, não é possível argumentar pela existência de estados ligados pósitron-molécula, tendo em vista o comportamento desses potenciais em curto alcance. Gribakin *et al* [20] introduzem ad hoc uma barreira em curtas distâncias e a justificam pela formação do Ps virtual. Este seria uma componente da função de onda do complexo contendo o par elétron-pósitron. Essa ideia foi utilizada com sucesso em alguns artigos [21–23].

Dzuba *et al* [4] mostraram que quando o potencial de ionização é superior à energia do estado fundamental do positronium é mais provável que a interação resulte em um complexo pósitron + átomo. Em contrapartida, quando a energia do pósitron incidente é menor que o potencial de ionização, a formação predominante tende a ser Ps + Íon Positivo.

No presente trabalho, além de [18], consideramos que a formação do Ps virtual modifica o potencial total em curtas distâncias de modo a tornar viáveis os estados ligados. Para o uso em IA é preciso obter uma propriedade molecular relacionada à formação de Ps, que consideramos ser o potencial de ionização, I_p .

2.2 Momento de Dipolo e Polarizabilidade Elétrica

Conceitualmente, o momento de dipolo é uma medida da polaridade de uma molécula ou conjunto de cargas. Uma molécula ou conjunto de cargas com momento de dipolo nulo é definido como apolar, pois não apresenta um centro de carga positivo e um centro de carga negativo. Por outro lado, uma molécula ou conjunto de cargas com momento de dipolo não nulo é definido como polar, pois apresenta um centro de carga positivo e um centro de carga negativo. Matematicamente, o momento de dipolo é definido como:

$$\mu \equiv \int r \rho(r) dV, \quad (2.1)$$

onde r é a posição da carga e $\rho(r)$ é a densidade de carga em r e a equação 2.1 é integrada em todo volume. O momento de dipolo é determinado pela geometria (tamanho, forma e densidade) da distribuição de carga. Para n cargas pontuais, o momento de dipolo assume a forma:

$$\mu = \sum_{i=1}^n q_i r_i. \quad (2.2)$$

Para uma molécula, $\mu = 2.1 + 2.2$, onde $\rho(r)$ é a densidade eletrônica. Quando uma molécula polar é submetida a um campo elétrico externo, ela experimenta um torque que tende a alinhar o momento de dipolo com o campo. Esse torque é proporcional ao momento de dipolo da molécula e à intensidade do campo elétrico. Além disso, campos podem induzir um momento de dipolo em moléculas apolares próximas. Esse fenômeno é chamado de polarização induzida.

Para ajustar o quão intenso será o dipolo produzido por uma molécula ao ser exposto a um campo elétrico, introduz-se a propriedade de polarizabilidade elétrica, que indica o quão suscetível é um átomo ou molécula de formar um dipolo induzido. Ambas propriedades moleculares são amplamente estudadas a fim de se detectar suas influências

diretas e indiretas no fenômeno de ligação do pósitron com moléculas - lembrando que aqui o cálculo *ab initio* é complexo e pouco confiável.

2.2.1 Polarizabilidade Elétrica

Considere a interação de uma carga e uma molécula. A carga induzirá um momento de dipolo nesta molécula dado por

$$\boldsymbol{\mu}_{ind} = \alpha \mathbf{E}, \quad (2.3)$$

onde α é a polarizabilidade isotrópica da molécula. Em geral, o vetor $\boldsymbol{\mu}_{ind}$ de 2.3 pode ser dividido em suas componentes. Suponha que queira-se analisar o momento de dipolo induzido na direção x gerado por um campo elétrico aplicado no eixo z , teria-se que

$$\mu_x = \alpha_{xz} \mathbf{E}, \quad (2.4)$$

havendo dessa forma um momento de dipolo específico para cada uma de suas componentes, dependendo da direção em que um campo elétrico foi aplicado. Para o caso no qual o campo elétrico é uniforme, a polarizabilidade isotrópica α se trata da média aritmética das componentes dos eixos principais do tensor de polarizabilidade elétrica, ou seja, dado um tensor de polarizabilidade elétrica como

$$\overset{\leftrightarrow}{\alpha} = \begin{pmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{pmatrix} \quad (2.5)$$

e

$$\overset{\leftrightarrow'}{\alpha} = \begin{pmatrix} \alpha'_{xx} & 0 & 0 \\ 0 & \alpha'_{yy} & 0 \\ 0 & 0 & \alpha'_{zz} \end{pmatrix}, \quad (2.6)$$

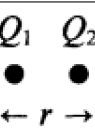
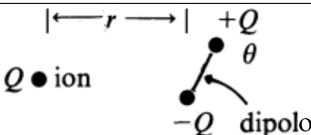
onde $\overset{\leftrightarrow'}{\alpha}$ é o tensor de polarizabilidade $\overset{\leftrightarrow}{\alpha}$ diagonalizado, representado por seus eixos principais (sendo suas intensidades α'_{xx} , α'_{yy} , α'_{zz}) - componentes da polarizabilidade elétrica anisotrópica -, a polarizabilidade elétrica isotrópica é escrita como

$$\alpha = \frac{1}{3}(\alpha'_{xx} + \alpha'_{yy} + \alpha'_{zz}). \quad (2.7)$$

A polarizabilidade média, α , é utilizada, por exemplo, quando uma molécula está girando em um fluido e apresenta todas as orientações para o campo aplicado.

Ao longo desse trabalho será utilizada a mesma notação das componentes da diagonal principal do tensor de polarizabilidade elétrica para as componentes da diagonal

Tabela 2 – Fórmulas de Potenciais de Interação

Tipo de interação	Notação	Energia de interação
Carga-carga		$V(r) = Q_1 Q_2 / 4\pi\epsilon_0 r$
Carga-dipolo		$V(r, \theta) = -\mu Q \cos \theta / 4\pi\epsilon_0 r^2$
Carga-dipolo-induzido	O mesmo da figura acima	$V(r) = -\alpha Q^2 / 32\pi^2 \epsilon_0^2 r^4$

principal do tensor diagonalizado, ou seja, α'_{ii} será escrito como α_{ii} , onde $i = x, y, z$, para que a notação não fique carregada.

É fundamental destacar a interação entre dipolos e partículas carregadas no contexto do momento de dipolo e polarizabilidade. Em situações onde um pósitron interage a distâncias consideráveis com moléculas — definidas pela ausência de influência significativa da superposição das nuvens eletrônicas —, tanto o potencial elétrico clássico quanto abordagens quânticas fornecem pistas valiosas sobre a dinâmica da interação.

Conforme apresentado na Tabela 2, as interações de longa distância mais relevantes para o estudo de momentos de dipolo e cargas são sumarizadas. Tais potenciais são frequentemente empregados na busca por descrições empíricas ou semi empíricas do fenômeno de ligação entre pósitrons e moléculas.

Ao considerar interações de curta distância, é imprescindível atentar para aspectos químicos e quânticos específicos, incluindo interações de correlação, potencial de ionização e os diversos tipos de ligações moleculares, como ligações π por exemplo. Já para interações de longa distância, a polarizabilidade elétrica e o momento de dipolo emergem como propriedades chave.

2.3 Relação entre propriedades moleculares e ligação do pósitron a moléculas

A análise das propriedades moleculares relacionadas às interações com pósitrons revela uma série de padrões e correlações significativas. Essas observações são particularmente evidentes em estudos empíricos ou semi-empíricos, que constituem o foco principal deste trabalho.

No contexto atômico, a ocorrência de um estado ligado com o pósitron é dependente principalmente da formação de um Ps virtual - o que costuma definir se irá ocorrer ou não essa formação virtual é a propriedade molecular potencial de ionização (I_p) - e da

polarizabilidade elétrica (α), conforme descrito por Mitroy *et al.* [19]

Embora essencial para avaliar a capacidade de ligação com pósitrons, o I_p raramente é empregado como descritor nas energias de ligação do pósitron (PBEs - *positron binding energies*), com o estudo de Amaral e Mohallem [18] sendo uma notável exceção a essa tendência. Supõe-se também que as PBEs correlacionam-se mais fortemente com as variações de I_p do que com seus valores absolutos. Assim, para moléculas com propriedades e estruturas semelhantes, as diferenças nas PBEs podem ser atribuídas às variações no I_p . [24]

A introdução de ligações π tende a elevar as PBEs e a reduzir o I_p . Entende-se que um incremento nas ligações π induz um efeito duplo, potencializando as PBEs por meio de alterações estruturais e pela diminuição do I_p . Danielson *et al.* [24] ressaltam a importância das ligações π em moléculas aromáticas, propondo que a deslocalização dos elétrons π pode vir a facilitar a interação com pósitrons.

Quanto ao momento de dipolo permanente (μ), maiores valores tendem a estar associados a PBEs elevadas, devido ao potencial atrativo que o momento de dipolo gera. Contudo, a relação preditiva não é simples. Momentos de dipolo superiores a 1 Debye sugerem uma influência significativa nas PBEs, o que ocorre pelo fato de haver separação mais definida entre os polos positivo e negativo e uma menor susceptibilidade a alterações por campos elétricos externos, o que também acaba diminuindo a relevância da polarizabilidade elétrica, conforme discutido por Danielson *et al.* [17].

Supõe-se que o pósitron possua localizações de preferência em moléculas com momentos de dipolo bem definidos ou com certos átomos em sua composição, como demonstrado em estudos com nitrilas e moléculas contendo cloro. [23, 25, 26]

A polarizabilidade elétrica também desempenha um papel fundamental. As PBEs apresentam uma dependência não linear com a polarizabilidade elétrica, exceto em alcanos, onde a relação é linear. Em moléculas com baixo valor de μ , a polarizabilidade elétrica torna-se mais influente na variação das PBEs devido à maior flexibilidade na configuração eletrônica sob influência de campos elétricos.

Por exemplo, moléculas com grupos hidroxila no final de cadeias carbônicas exibem uma relação linear semelhante à dos alcanos, possivelmente devido à constância dos valores de μ , sendo a polarizabilidade elétrica praticamente não relevante nesta situação - dado que os momentos de dipolo são maiores do que 1.2 Debyes - [27]. Embora fatores como a intensidade dos momentos de dipolos, a estrutura molecular e a geometria das moléculas sejam cruciais na determinação das PBEs, ainda não há consenso sobre como essas características moleculares podem ser simplificadas em modelos matemáticos.

Tendo o panorama explicitado acima, a principal motivação do presente estudo é compreender como algoritmos de aprendizado de máquina podem elucidar a importância

de propriedades moleculares específicas, como a polarizabilidade anisotrópica e as ligações π , que refletem a singularidade estrutural e eletrônica das moléculas, bem como suas interações com campos elétricos.

3 Aprendizado de Máquina

O Aprendizado de Máquina (*Machine Learning*), um subcampo da Inteligência Artificial, visa emular o comportamento do aprendizado humano na solução de problemas através de computadores. Essa disciplina emprega métodos matemáticos, estatísticos e computacionais para desenvolver modelos que resolvam problemas específicos. Adaptando estratégias de outras áreas, esses modelos capitalizam a eficiência e a rapidez dos cálculos das máquinas modernas para identificar rapidamente parâmetros otimizados.

Os modelos de aprendizagem de máquina têm sido amplamente utilizados nas últimas décadas, desde tarefas simples, como regressão, até tarefas de visão computacional, em diversos setores, como ciências, mercado, medicina, tecnologia, lazer, entre outros.

É importante destacar que, ao nos referirmos a um modelo, estamos tratando de uma relação final encontrada entre as variáveis dependentes e independentes pelo algoritmo de aprendizado de máquina. Por exemplo, no caso de uma regressão linear simples, o algoritmo se refere ao processo de minimização para obter os coeficientes, enquanto o modelo se refere à reta criada a partir desses coeficientes

Normalmente, o processo funciona da seguinte forma: um algoritmo de aprendizagem é fornecido com um conjunto de dados, é treinado e o modelo resultante é reutilizado para prever informações sobre dados fornecidos posteriormente que possam ser de interesse. Quando dizemos que um modelo está sendo treinado, isso implica que determinados parâmetros estão sendo ajustados de acordo com o problema e a arquitetura do modelo a fim de minimizarmos uma função de custo ou perda. Essas funções normalmente relacionam as entradas das nossas amostras com uma saída que indica o quão bom é o desempenho daquele algoritmo. Um exemplo de função de custo é a soma dos resíduos, que corresponde à diferença entre o que um modelo de aprendizagem prevê e o valor que deveria ter sido previsto.

Em geral podemos classificar os algoritmos de aprendizagem de máquina em dois grandes grupos: *supervisionados* e *não supervisionados*. Os problemas que utilizam algoritmos de aprendizagem supervisionada trabalham com uma variável alvo que é conhecida de antemão para cada amostra da base de dados, ou seja, o algoritmo tenta procurar relações entre as variáveis independentes e a variável alvo que é fornecida, dado a dado.

Problemas famosos de algoritmos supervisionados são principalmente *regressões*, onde tentamos prever o valor real de uma variável, como a propriedade de alguma molécula, ou *classificações*, onde tentamos prever uma classe, como por exemplo a que tipo de grupo determinado dado pertence.

Os problemas de aprendizado não supervisionado referem-se a algoritmos que buscam identificar padrões em dados sem a necessidade de instruções explícitas do usuário em relação às variáveis alvo. Um exemplo prático seria um conjunto de dados contendo informações sobre várias propriedades de espécies de plantas. Nesse cenário, um algoritmo de aprendizado não supervisionado poderia ser empregado para agrupar essas espécies em cinco (ou qualquer outra quantidade de interesse) categorias distintas, sem que fosse especificado previamente como essa separação deveria ser realizada.

Nos referiremos à uma amostra contendo os elementos (x_i, y_i) , onde x_i se refere às variáveis independentes, e y_i se refere à variável dependente. Diremos também que $\hat{y}_i = \hat{f}(x_i)$, sendo $\hat{f}(x_i)$ a previsão do modelo no ponto x_i .

Nas próximas seções, serão abordados conceitos e modelos de aprendizagem de máquina utilizados neste trabalho. Antes de prosseguir, é importante destacar que não será feita distinção entre os termos “variáveis independentes”, “variáveis explicativas” ou “variáveis previsoras”. Todos esses termos se referem às variáveis que serão utilizadas para prever ou calcular uma variável alvo. Portanto, ao longo deste trabalho, essas expressões serão utilizadas de forma intercambiável. O mesmo se aplica a “variável dependente” e “variável alvo”.

3.1 Avaliação de Performance de Modelos

Na elaboração de um modelo de aprendizado de máquina supervisionado, o procedimento adota uma sequência estruturada. Inicialmente, um conjunto de dados é apresentado ao algoritmo, definindo-se as variáveis independentes e a variável dependente a serem previstas. O algoritmo é então treinado com esses dados para gerar o modelo mais eficaz. Finalmente, realiza-se uma avaliação para verificar a precisão do modelo na previsão da variável dependente.

No entanto, o objetivo final é assegurar que o algoritmo tenha um bom desempenho em dados ainda não observados, uma vez que é comum que os modelos sejam utilizados para previsões futuras. Isso nos leva à questão de como podemos estimar o desempenho do modelo em dados ausentes no conjunto de treinamento, considerando que o modelo está sendo treinado com todos os dados disponíveis.

Embora os modelos sejam criados a partir de um conjunto de dados amostrais, ainda há um erro nas previsões do conjunto de treinamento em comparação com os valores que deveriam ser alcançados. Uma ideia seria apresentar o desempenho com base nesses erros de treinamento, mas isso pode ser prejudicialmente otimista, pois o algoritmo pode decorar os valores alvos das instâncias e ter um erro muito próximo de zero. Portanto, é essencial ter um conjunto separado de dados para testar a performance do algoritmo, conhecido como conjunto de dados de teste.

Antes de continuar a discussão sobre métodos de avaliação de performance de algoritmos, é importante introduzir dois novos conceitos: *viés* e *variância*.

3.1.1 Viés e Variância

Suponhamos que desejemos determinar a idade média da população brasileira. Para isso, realizamos duas amostragens, X e Y , com indivíduos aleatórios. Os valores obtidos foram $X = [20, 24, 28, 32]$ e $Y = [20, 40, 60, 80]$ em anos. É evidente que a médias aritméticas são 26 e 50 anos, para X e Y respectivamente.

Suponhamos também que a idade média real da população brasileira é de 40 anos de idade. Por essas suposições, pode-se observar que o valor médio de Y se aproxima mais do real do que o valor médio de X . Com isso, dizemos que a estimativa de idade gerada por Y possui menos viés do que X . Por outro lado, percebe-se que Y tem os valores individuais mais distantes de sua própria média, dizemos então, que Y possui uma maior variância.

O viés é uma medida que avalia o quão distante uma estatística calculada a partir de amostras está do valor verdadeiro. Essa distância pode variar devido a suposições ou considerações subjacentes, que podem ou não ser explicitadas durante a amostragem. Em outras palavras, o viés permite quantificar a diferença entre o resultado obtido e o valor real, que pode ser influenciado por fatores não evidentes na coleta dos dados.

Já a variância é uma medida de dispersão, ou seja, o quanto os dados coletados estão dispersos em relação ao seu valor esperado. A variância pode ser influenciada pelo tamanho da amostragem ou por sua heterogeneidade, quanto maior a amostragem maior a tendência de diminuir a variância e quanto mais heterogeneidade na natureza dos dados, maior a tendência de uma maior variância. Formalmente o viés e variância são definidos como:

$$Viés = E[\hat{\beta}] - \beta, \quad (3.1)$$

$$Variância = E[(\hat{\beta} - E[\hat{\beta}])^2], \quad (3.2)$$

onde $\hat{\beta}$ é a estatística prevista a partir das amostragens e β é a estatística real. O viés e variância estatística em algoritmos de aprendizagem utiliza o termo $E[\hat{\beta}]$ para se referir ao cálculo de valor esperado para diferentes conjuntos de dados de treino da estimativa $\hat{\beta}$ para o parâmetro β [28].

Uma ilustração do que viés e variância representam é dada pela Figura 6, onde o alvo vermelho seria a representação daquilo que se pretende medir, seja uma estatística ou um modelo, como no caso de aprendizado de máquinas.

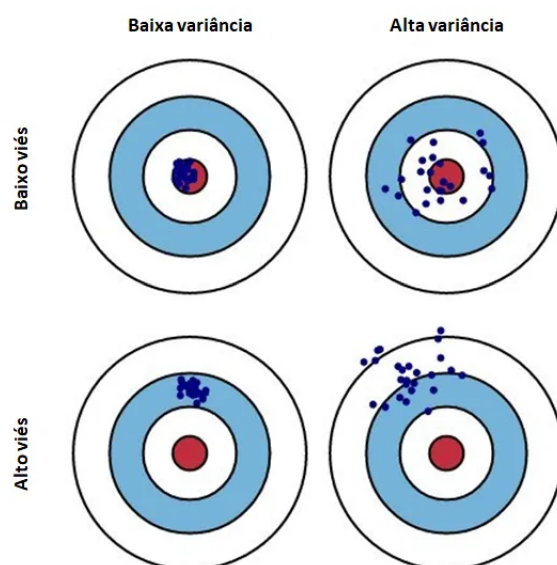


Figura 6 – Imagem ilustrativa de viés e variância. Fonte: <http://scott.fortmann-roe.com/docs/BiasVariance.html>

3.1.2 Dilema Viés e Variância

No exemplo das idades acima, não coincidentemente o conjunto de menor viés apresenta uma maior variância. Ambos os valores protagonizam um dilema popularmente conhecido como *Bias-Variance Tradeoff* (Dilema Viés-Variância) em aprendizado de máquina. Normalmente, ao tentar estimar o erro esperado de um algoritmo de aprendizado de máquina para dados não vistos, deve-se tomar cuidado e ter atenção à esse dilema, dado que ele trás implicações importantes na leitura do erro obtido a partir das amostras.

O erro no cálculo de uma estatística pode ser decomposto em uma combinação de três fatores [29]: viés, variância e um erro irreduzível que, como indicado pelo nome, não pode ser retirado, pois sua natureza reside nas imperfeições dos dados da amostragem:

$$Erro = Viés^2 + Variância + \epsilon, \quad (3.3)$$

onde ϵ é o erro irreduzível. É interessante perceber que a minimização do erro deverá buscar, por consequência a minimização das contribuições do viés e variância simultaneamente.

Quando se refere ao viés e variância em modelos de aprendizagem de máquina, é uma consequência indireta do viés e variância em aprendizagem estatística: Ao possuir um maior viés, o modelo em geral está fazendo suposições errôneas sobre as naturezas dos dados e portanto, errando em média mais do que um modelo que faz suposições mais acertadas. Costumadamente modelos mais simples como regressões lineares apresentarão um maior viés.

Ao possuir uma maior variância, o algoritmo é mais sensível ao conjunto de treino

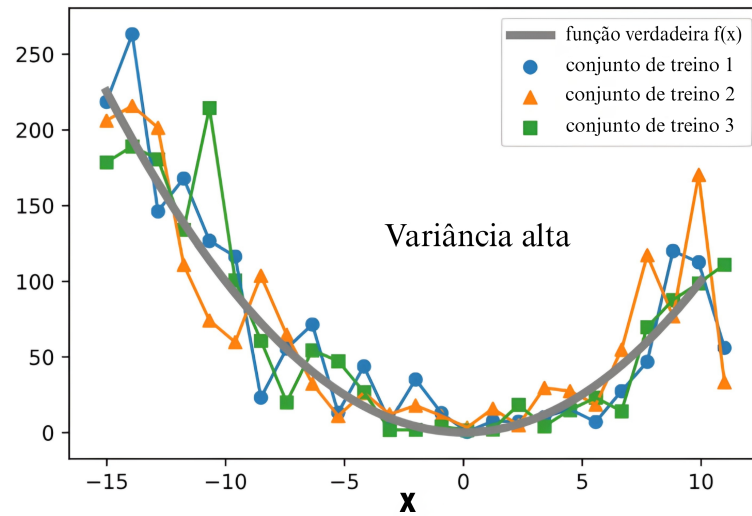


Figura 7 – 4 curvas são apresentadas, a cinza seria a curva apropriada para a natureza do problema e as outras 3 são as curvas geradas a partir de conjuntos de dado de treinamento distintos para o mesmo algoritmo. Fonte: https://github.com/rasbt/stat479-machine-learning-fs18/blob/master/08_eval-intro/08_eval-intro_slides.pdf

e portanto, os valores previstos para um mesmo ponto x_o a partir de dados de treino diferentes se diferenciarão mais. Ao modificar-se o conjunto de dados de treino não deveria haver muita diferença nos valores obtidos para a função calculada $\hat{f}(x_o)$ e em geral modelos mais complexos apresentarão uma maior variância [30].

A Figura 7 ilustra modelos (curvas) gerados por um algoritmo em diferentes conjuntos de treinamento. Observam-se variações significativas nos valores de $\hat{f}(x_o)$ para um x_o fixo. Isso sugere que os três modelos, embora gerados pelo mesmo algoritmo, sofrem de sobreajuste, evidenciado pela alta variância nas previsões. Esta figura exemplifica o cálculo de variância no contexto de aprendizado de máquina.

Na Figura 8, em que temos outro conjunto de dados e diferentes curvas geradas, a curva verde se ajusta quase perfeitamente a todos os pontos. No entanto, é importante notar que essa performance superior ocorre apenas no conjunto de treinamento. A curva verdadeira, que é o ajuste ideal e mais sensato, seria mais adequada para prever dados futuros, mesmo que os dados de treinamento não sejam perfeitamente descritos por ela.

Por outro lado, a curva em amarelo (uma regressão linear) apresenta uma certa rigidez e devido a isso não consegue captar as nuances da base de dados. Dizemos que a curva verde, por estar aprendendo excessivamente com os dados fornecidos, está sofrendo um sobreajuste ou ajuste excessivo (*overfitting*) e a curva amarela um subajuste (*underfitting*).

Além disso, devido à grande flexibilidade da curva verde em relação aos dados de treinamento, é possível inferir que ela apresentará maior variância. Essa alta sensibilidade

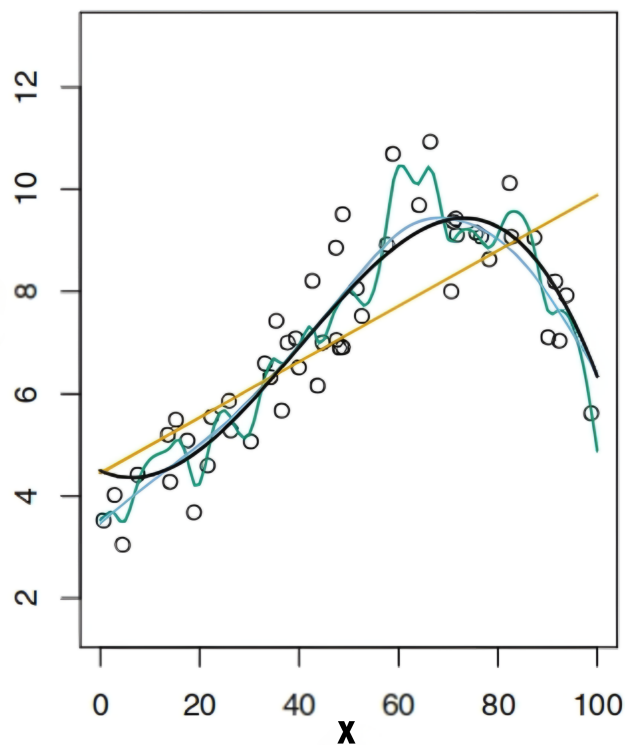


Figura 8 – Dados simulados representados por círculos pretos. As curvas em amarelo e verde representam diferentes $\hat{f}$, que são uma tentativa de estimar a curva em preto f . [30]

da curva verde resulta em um baixo viés, uma vez que ela está acertando as previsões de forma precisa. Por outro lado, a curva laranja exibe baixa variância, pois qualquer modificação no conjunto de treinamento causará apenas pequenas alterações na inclinação da reta. No entanto, essa mesma curva laranja possui alto viés, pois não consegue capturar completamente a natureza do ajuste aos dados.

O conhecimento desse dilema é de extrema importância, uma vez que, ao treinar algum algoritmo de aprendizagem de máquina, ele tentará se aproximar do modelo que melhor descreve aquela população de dados, com apenas informações da amostra [30]. Devemos, além disso, estar atentos a efeitos de *overfitting* e *underfitting*.

Quando nos referimos ao dilema viés-variância para um modelo de aprendizagem de máquina, estamos na busca de uma perfeita complexidade do algoritmo para determinado problema, no qual, se for muito complexo poderá apresentar *overfitting* e se for pouco complexo poderá apresentar *underfitting*. Além disso, vale ressaltar que a consequência última de ambas situações é um alto erro de teste, ou seja, um baixo poder de generalização do algoritmo para dados não presentes na base de dados de treinamento.

Em linhas gerais, modelos que apresentam *overfitting* possuem baixo erro de treino e um alto erro de teste (que pode ser um sintoma de uma alta variância do modelo).

Enquanto que para *underfitting* é comum alto erro de treino e teste (que pode ser um sintoma de um alto viés).

3.1.3 Métodos de Avaliação

Voltando aos métodos de avaliação, é importante definir a métrica que será analisada neste trabalho para informar o desempenho dos algoritmos. Nesse sentido, será utilizado o erro absoluto percentual médio (MAPE - *Mean Absolute Percentage Error*), que é definido por:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i}, \quad (3.4)$$

onde N representa a quantidade de amostras que serão utilizadas para calcular o erro, lembrando que essas amostras compõem o conjunto de teste e não o conjunto de treino. Além disso, y_i é o valor real da variável alvo para a amostra i e $\hat{y}_i$ é o valor previsto pelo modelo da mesma amostra.

Foi estabelecido que é necessário separar as amostras em conjunto de treino e teste, pois essa é a melhor maneira de tentar estimar o erro em dados ausentes no conjunto de treinamento. Caso tivéssemos acesso à população da qual as amostras foram retiradas, seria fácil estimar o erro do algoritmo de aprendizado de máquina. Porém, na realidade, dispomos apenas de um espaço amostral e a partir dele devemos estimar quão bem o algoritmo desempenharia na população.

Nesse ponto, é importante destacar uma diferença sutil entre o *viés* e a *variância*, que foram definidos na seção anterior. Anteriormente, a estatística que se pretendia determinar era y_i , e a análise era feita a partir de $\hat{y}_i$. Já para os métodos de avaliação, o *viés* e a *variância* se referem à estatística MAPE [30,31], ou seja, ao erro que estamos estimando para o algoritmo sob determinadas condições.

Novamente, não conseguimos estimar o MAPE real dos algoritmos para a população, então exploraremos alguns métodos mais comuns na separação dos dados disponíveis em conjuntos de treino e teste.

3.1.3.1 Método Holdout

O método Holdout é uma técnica básica para separar dados em conjuntos de treinamento e teste aleatórios, com proporções definidas previamente. Normalmente, a divisão é de 80% para treinamento e 20% para teste. A Figura 9 ilustra esse processo, mostrando um conjunto de dados dividido em duas partes. É relevante destacar que, à medida que o tamanho do conjunto de dados aumenta, a proporção destinada ao treinamento diminui.

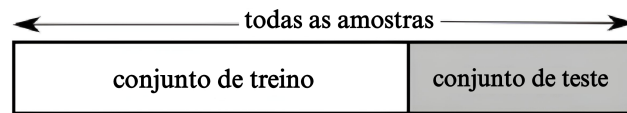


Figura 9 – Representação da divisão Holdout, onde o conjunto inteiro de dados é dividido em duas partes aleatoriamente separadas, a parte de treino e de teste. Fonte: <https://vitalflux.com/hold-out-method-for-training-machine-learning-model/>

Essa técnica tem algumas desvantagens a serem consideradas. A primeira é que, em conjuntos de pouca ou média quantidade de dados, a determinação do erro do algoritmo pode apresentar um viés considerável. Isso ocorre porque uma grande parte das amostras é removidas do treino para ser usada no teste, o que pode limitar o potencial do modelo.

Essa limitação pode levar a um viés negativo, fazendo com que o erro reportado seja pior do que o real. Além disso, há também um viés positivo potencial, já que as amostras selecionadas para teste podem estar sendo escolhidas de forma tendenciosa, favorecendo amostras mais fáceis de prever. Esse problema é mitigado com o aumento do conjunto de dados total, já que então haverá mais amostras para se testar o modelo, tornando mais difícil que tenha ocorrido alguma escolha aleatória de um conjunto de teste favorável.

3.1.3.2 Método de Validação Cruzada

O método de validação cruzada é uma estratégia mais robusta do que o método Holdout. Ele visa superar as limitações do Holdout e funciona da seguinte maneira: Divide-se o conjunto de dados em k partes (também conhecidas como pastas, envelopes ou qualquer termo que se refira a repartições). Em seguida, reserva-se $k - 1$ dessas partes para treinar o modelo e 1 parte para testá-lo. Armazena-se então a métrica de avaliação obtida para esse conjunto de teste. Esse processo é repetido até que todas as partes tenham sido usadas como teste e, finalmente, calcula-se a média das métricas de avaliação para relatar a performance final do modelo.

Na Figura 10 há um exemplo para $k = 5$, tendo então que realizar 5 treinamentos e 5 avaliações de performance e a performance final é calculada como a média das performances.

O método de validação cruzada possui algumas vantagens importantes. A primeira delas é que ele permite a utilização de todos os dados disponíveis como conjunto de teste, o que reduz a possibilidade de variância causada pela seleção inadequada do conjunto de teste. É importante lembrar que essa variância ocorre porque, ao selecionar um conjunto de teste que é fácil para o modelo prever, pode haver outros conjuntos de teste em que isso não é verdade, gerando uma diferença de performance significativa.

No entanto, surge a dúvida sobre qual valor para k deve ser escolhida. Para

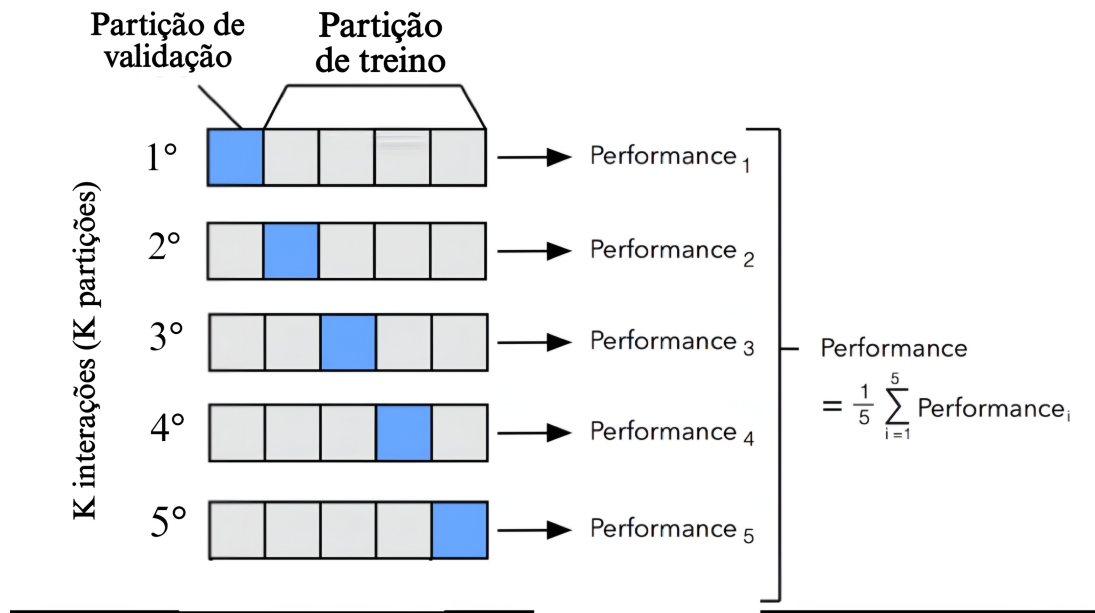


Figura 10 – Método de validação cruzada pra $k = 5$, onde as partes em cinza são as partes usadas para treino e a parte azul para teste. Fonte: https://zitaoshen.rbind.io/project/machine_learning/machine-learning-101-cross-validation/

responder a essa pergunta, é necessário recorrer ao dilema viés-variância, que se refere à estimativa do erro do modelo e não de suas predições. Ou seja, há a tentativa de estimar qual o erro esperado do modelo para dados não fora do conjunto de treinamento, e o dilema viés-variância reitera a imprecisão e dificuldade que temos em estimar esse erro.

Para ilustrar a diferença entre as repartições de treino-teste, pode-se comparar como ficaria a divisão em $k = 5$ e $k = 10$ para um conjunto de dados pequeno, com 50 amostras. Para $k = 5$, seriam feitas 5 rodadas de treinamento, onde cada rodada seria treinada com 40 amostras e testada nas 10 restantes. Já para $k = 10$, seriam feitos 10 treinamentos com 45 amostras e teste nas 5 restantes. É importante destacar que, em ambos os casos, o modelo que será levado à produção será o modelo treinado com todas as 50 amostras.

Nesse contexto, a repartição que utiliza 45 dados para treinamento terá uma estimativa de performance que possuirá um menor viés e se aproximará mais do erro médio do algoritmo treinado com todas as 50 amostras. No entanto, ao diminuir o conjunto de teste, haverá uma maior variabilidade nas performances reportadas, especialmente quando se testa em apenas uma amostra do conjunto de dados. Isso ocorre porque, se a amostra de teste estiver perto da curva preditiva final do modelo, ela apresentará um erro baixo, enquanto que se estiver mais distante dessa curva, apresentará um erro alto, gerando uma variabilidade maior nas performances reportadas para diferentes conjuntos de treino.

Dessa forma, é importante encontrar um equilíbrio para essa questão. Na literatura, normalmente, é utilizado $k = 10$ ou $k = 5$, para um conjunto de dados menor, ou maior respectivamente. Outra questão a se levar em consideração é o custo computacional; quanto maior o k escolhido, maior será a quantidade de treinamentos que será feita. Para conjuntos de dados maiores, não é usual a escolha de um k grande, tanto pelo custo computacional quanto pela possibilidade de se utilizar o método Holdout.

3.1.3.3 Método de Validação Cruzada Aninhada

O método de validação cruzada aninhada tem o objetivo de testar a performance de algoritmos em todo o conjunto de dados disponível, assim como a validação cruzada comum. No entanto, ele também inclui o processo de otimização de hiperparâmetros, que geralmente exige a criação de um conjunto de validação separado do conjunto de treino e teste.

Os hiperparâmetros são parâmetros que precisam ser definidos manualmente pelo usuário, uma vez que o algoritmo não se preocupa em modificá-los. Alguns exemplos incluem a quantidade de árvores em algoritmos baseados em árvores, a quantidade de camadas ou neurônios por camada em uma rede neural e o parâmetro C para máquina de vetor de suporte. Os hiperparâmetros têm um impacto direto na performance de um algoritmo de aprendizado de máquina e podem afetar questões como *overfitting* ou *underfitting*. O espaço de hiperparâmetros é o conjunto de todas as possíveis combinações de hiperparâmetros que deseja-se avaliar.

O processo de validação funciona da seguinte forma: primeiro, treina-se o algoritmo no conjunto de treino e sua performance é avaliada no conjunto de validação. Em seguida, o conjunto de hiperparâmetros do algoritmo é trocado por outro conjunto e é avaliado novamente o desempenho no conjunto de validação. Esse processo é repetido exaustivamente - ou através de outra estratégia - até que seja escolhido o conjunto de hiperparâmetros com melhor desempenho no conjunto de validação.

A partir daí, treina-se o algoritmo com os hiperparâmetros selecionados no conjunto de treinamento somado ao conjunto de validação e seu desempenho será avaliado no conjunto de teste. Essa será a performance final reportada. O modelo final enviado para produção será treinado no conjunto completo de dados (treino + validação + teste) com os hiperparâmetros selecionados durante o processo de validação; o processo está apresentado em diagrama esquemático na Figura 10.

O objetivo da introdução do conjunto de validação é otimizar o desempenho de um algoritmo de aprendizado de máquina. Quando se treina o algoritmo apenas no conjunto de treinamento e valida-se no conjunto de teste, o desempenho resultante pode ser excessivamente otimista. Isso ocorre porque o processo de otimização pode inadvertidamente selecionar hiperparâmetros que se ajustam bem ao conjunto de teste, mas não generalizam

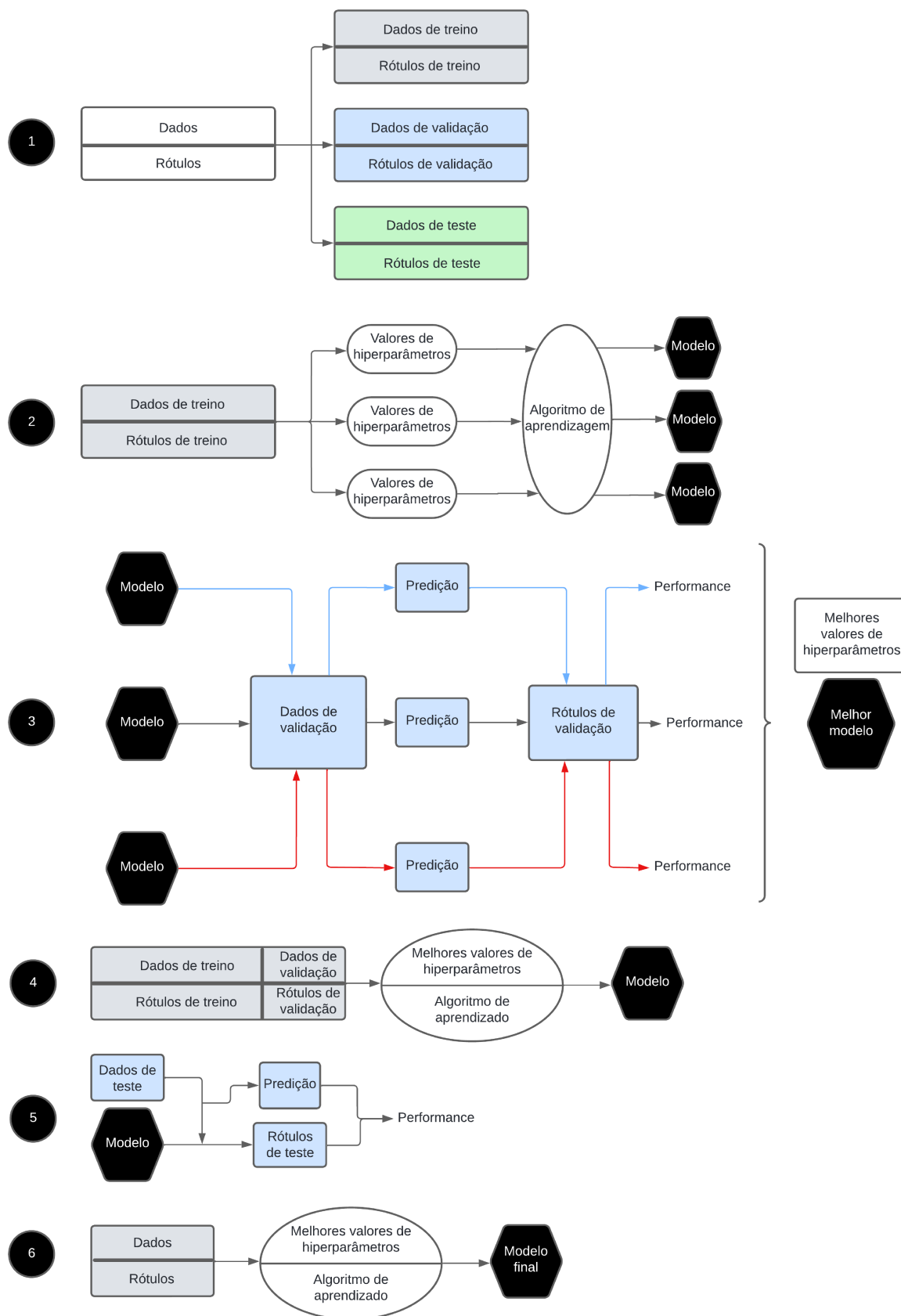


Figura 10 – Seleção de hiperparâmetros: (1) Separação dos dados em conjuntos de treino, validação e teste; (2) Criação de um modelo para cada conjunto de hiperparâmetros (Treinados no mesmo conjunto de treino); (3) Avaliação de desempenho de cada modelo em cima do conjunto de validação e então escolha do conjunto de hiperparâmetros com melhor desempenho; (4) Treinamento do algoritmo com os hiperparâmetros selecionados no conjunto de treino + validação; (5) O desempenho final será obtido pelo modelo que foi treinado na fase anterior, em cima do conjunto de teste; (6) O modelo utilizado em produção seria obtido a partir do treinamento do algoritmo no conjunto de dados completo com o conjunto de hiperparâmetros encontrado. Alterado de: <https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html>

bem para dados externos ao conjunto de treinamento. Portanto, é essencial criar um conjunto de validação como parte do processo de otimização, considerando-o como parte do treinamento do algoritmo.

Essa divisão em 3 partes é o equivalente ao método Holdout para otimização de hiperparâmetros, mas já foi discutido sobre as limitações do método Holdout, principalmente sobre a estaticidade do conjunto de teste. Aqui as mesmas limitações se aplicam e portanto se faz necessário a introdução da validação cruzada. Observa-se que dessa vez há 3 conjuntos, então deveria ser realizado um processo de validação cruzada referente à validação e ao teste para garantir um revezamento entre os dados incluídos no teste e validação de forma a minimizar um possível viés otimista da performance dos algoritmos.

Para realizar a validação cruzada aninhada, primeiro determina-se valores para k_{in} (partições internas) e k_{out} (partições externas), sendo comum utilizar 5 e 10, respectivamente. Normalmente, k_{out} é maior para evitar que o processo de validação ocorra em um conjunto muito pequeno, o que pode causar alta variância e *overfitting* na escolha de modelos [32]. Em seguida, divide-se o conjunto de dados em k_{out} partes, sendo uma separada para testes e as outras $k_{out} - 1$ usadas como treinamento. Os dados de treinamento são divididos novamente em k_{in} partes, sendo uma usada como validação e as outras como treinamento.

Dessa forma, um algoritmo é treinado com um conjunto específico de hiperparâmetros no conjunto de treinamento e realiza-se a validação cruzada interna nas k_{in} partes para determinar a performance média. Em seguida, os hiperparâmetros são modificados e o processo é repetido exaustivamente até a escolha do melhor conjunto de hiperparâmetros. Após essa escolha, essa configuração é utilizada para treinar no conjunto de treinamento e validação juntos, onde avalia-se o desempenho no conjunto de testes. Esse processo é repetido até que todas as k_{out} partes sejam avaliadas e obtenha-se o desempenho esperado do algoritmo escolhido sob a condição de passar por processo de seleção de hiperparâmetros.

O processo é exemplificado pela Figura 11, onde o desempenho estimado do algoritmo *floresta aleatória* em dados não presente no conjunto de treinamento é de

Floresta aleatória

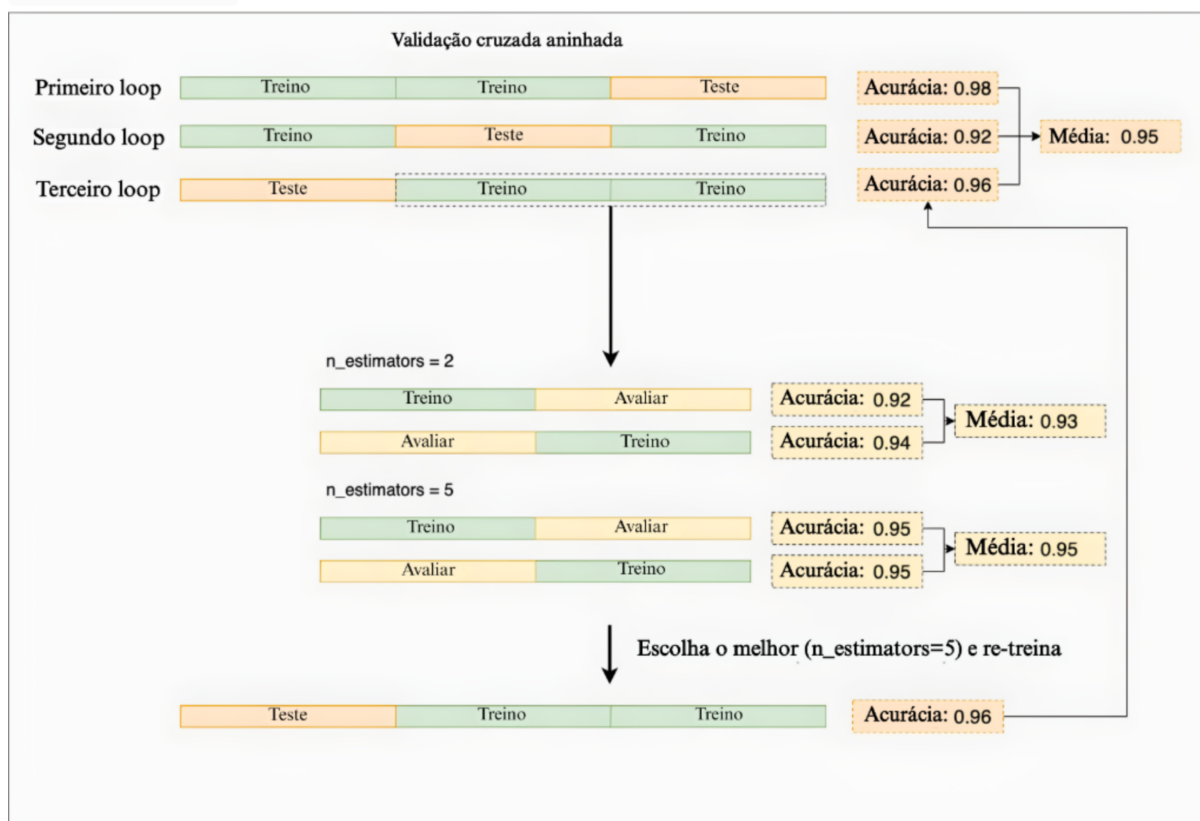


Figura 11 – Representação de um processo de validação cruzada aninhada onde $k_{out} = 3$ e $k_{in} = 2$. O algoritmo cujo o desempenho está sendo avaliado é a *floresta aleatória* com quantidade de árvores ($n_estimators$) sendo os hiperparâmetros a serem otimizados. Fonte: <https://ploomber.io/blog/nested-cv>

0.95 de acurácia. O processo traz a tentativa de diminuir qualquer viés que possa ser gerado pela otimização de hiperparâmetros e costuma apresentar uma estimativa mais robusta da performance do algoritmo. Em suma, são 2 *loops*, interno e externo, sendo o interno o responsável pela decisão do melhor hiperparâmetro e o externo pela estimativa da performance de determinado algoritmo submetido a um processo de otimização de hiperparâmetros.

A validação cruzada aninhada é uma técnica crucial para lidar com conjuntos de dados pequenos ou para garantir maior robustez na avaliação de algoritmos sob a condição de serem otimizados. Em outras palavras, a técnica permite comparar a performance de diferentes algoritmos em um conjunto de dados específico e, em seguida, escolher o que apresentou a melhor média no conjunto de teste externo.

Uma vez selecionado o algoritmo, o próximo passo é determinar o conjunto de hiperparâmetros ideais para treiná-lo. Esse processo pode ser realizado por meio de uma validação cruzada comum na base de dados inteira, utilizando o mesmo espaço

de busca realizado na validação cruzada aninhada. Com o algoritmo e o conjunto de hiperparâmetros finais escolhidos, podemos treiná-lo em todo o conjunto de dados e utilizar o modelo resultante para prever novas instâncias.

3.2 Algoritmos de Aprendizado de Máquina

As próximas subseções apresentarão uma descrição concisa e fundamentada dos algoritmos utilizados neste trabalho. É importante ressaltar que um algoritmo consiste em uma sequência de passos e lógicas que devem ser seguidos para construir um modelo. Por sua vez, o modelo é capaz de, dado um conjunto de variáveis explicativas, retornar um valor para a variável alvo.

3.2.1 Regressão Linear e Polinomial

A regressão linear é um método amplamente utilizado para resolver problemas de regressão. Consiste em modelar uma reta da forma:

$$y_i = a + bx_i + \epsilon_i, \quad (3.5)$$

onde i representa cada uma das N observações na base de dados, e ϵ_i é o erro associado a cada predição em relação ao valor real. O método dos mínimos quadrados é uma técnica famosa para encontrar os coeficientes a e b . Dado um conjunto de N amostras com uma variável explicativa x_i e sua respectiva variável alvo y_i , definimos a função de erro total, $S(a, b)$, como

$$S(a, b) = \sum_{i=1}^N (y_i - a - bx_i)^2. \quad (3.6)$$

O objetivo é minimizar a função $S(a, b)$ para obter os melhores parâmetros a e b . Quando se trata de um problema com mais de uma variável explicativa, trata-se de uma regressão linear múltipla e a forma de se calcular os coeficientes b que minimizam $S(a, b)$ é

$$b = (X^T X)^{-1} X^T y, \quad (3.7)$$

onde y representa as variáveis resposta para cada observação, $b = (b_1, b_2, b_3, \dots, b_n)$ são os coeficientes para as n variáveis, e X é a matriz de i linhas e j colunas. Aqui cada linha representa uma amostra do conjunto de dados e cada coluna representa o valor das variáveis explicativas.

A regressão polinomial é um caso onde ao invés de se trabalhar com as variáveis explicativas $X = (x_1, x_2)$ por exemplo, trabalha-se com as variáveis $X_{poli} = (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$ no caso de um polinômio de segundo grau, a fim de encontrar os 6 coeficientes que minimizariam a função $S(a, b)$ para duas variáveis explicativas.

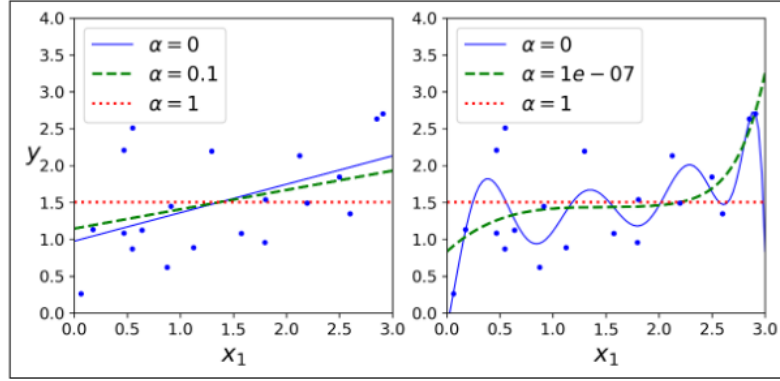


Figura 12 – Comportamento de curvas preditivas conforme se aumenta o coeficiente de regularização α para L1. À esquerda temos uma regressão linear simples e à direita uma regressão polinomial. [33]

3.2.2 Regularizações L1 e L2

Como visto anteriormente, no processo de treinamento de um algoritmo para a produção de um modelo, normalmente é minimizada alguma função custo. Sabendo que o *overfitting* é um problema para modelos mais complexos, os métodos que podem vir a ajudar a evitar o *overfitting* são chamados de métodos de regularização. Dois métodos específicos são as regularizações $L1$ e $L2$, onde ambas se baseiam no acréscimo de um termo na função custo a ser minimizada pelos algoritmos durante o treinamento, sendo definidas como

$$L1 = \alpha \sum_i^N ||b_i||, \quad (3.8)$$

$$L2 = \frac{1}{2} \alpha \sum_i^N b_i^2. \quad (3.9)$$

Nas equações 3.8 e 3.9 α é um termo escolhido a priori e b_i é o i -ésimo peso, ou coeficiente, para uma regressão linear, por exemplo. Aplicável para uma ampla variedade de algoritmos de aprendizado de máquina, o objetivo é transformar o treinamento pelo método dos mínimos quadrados - para o caso de uma regressão linear ou polinomial - em uma minimização da função custo $J(b)$, definida por:

$$J(b) = S(a, b) + W(b), \quad (3.10)$$

onde $S(a, b)$ é a equação 3.6 e W pode ser $L1$ ou $L2$. Modificar a função custo dessa forma visa reduzir pesos elevados em ambos os casos, evitando grandes variações na variável alvo $\hat{y}$ em função de uma ou mais características do conjunto de dados. Isso torna a curva resultante mais suave e menos sensível.

Nas Figuras 12 e 13 pode-se observar como as curvas se tornam mais suaves e menos propensas a “ligar os pontos” conforme α aumenta. Além disso, no caso da regularização $L1$, é possível transformar a regressão linear em uma reta horizontal, pois o peso b associado

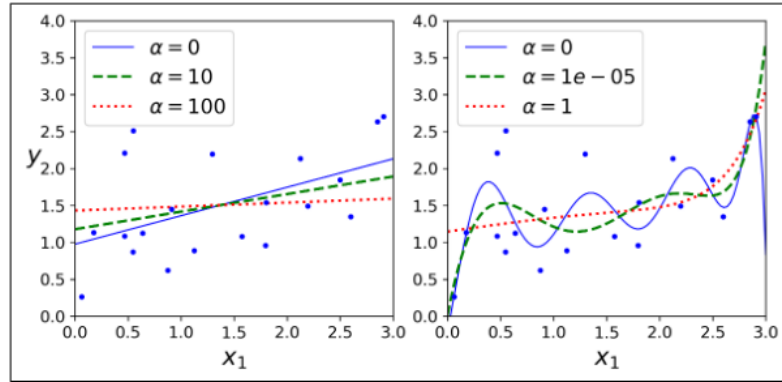


Figura 13 – Comportamento de curvas preditivas conforme se aumenta o coeficiente de regularização α para $L2$. À esquerda temos uma regressão linear simples e à direita uma regressão polinomial. [33]

à variável x_i foi ajustado para 0. A regularização $L1$ pode ser interessante como pré-seleção de atributos, pois há a possibilidade de tornar zero algum coeficiente.

Em termos gerais, isso ocorre devido às características das derivadas de cada termo. A regularização $L1$ sempre sofrerá uma atualização de mesmo valor, o que pode resultar na anulação de algum coeficiente. Por outro lado, o termo $L2$ tem atualizações dependentes do peso atual: quanto menor o peso a ser atualizado, menor será a mudança. Para esclarecer ainda mais, uma das formas comuns de atualizar os pesos é através do cálculo de gradiente:

$$b' = b - \eta \frac{dL}{db}. \quad (3.11)$$

A equação 3.11 significa que o novo peso b' será calculado subtraindo o termo $\eta \frac{dL}{db}$ (onde L pode ser $L1$ ou $L2$), e η normalmente é referido como a *taxa de aprendizagem*, em outras palavras, o quão grande os incrementos serão. Esse algoritmo que busca encontrar um melhor parâmetro b' é chamado *descida do gradiente*.

Esses termos de regularização frequentemente são utilizados em diversos algoritmos como forma de penalizar - tornando os coeficientes dos algoritmos menores - a aprendizagem em situações que hajam possibilidades de *overfitting*; esses parâmetros são utilizados em algoritmos como XGBoost e redes neurais, por exemplo.

Uma regressão linear realizada com a regularização $L1$ é chamada de regressão *LASSO* (*Least Absolute Shrinkage and Selection Operator*), enquanto a regularização $L2$ é chamada regressão *RIDGE* (rígida).

3.2.3 Árvores de Decisão

Árvore de decisão é um algoritmo de aprendizagem de máquina não paramétrico, isto é, não assume uma quantidade fixa de parâmetros a priori, como por exemplo, a regressão linear, que assume uma reta de dimensão especificada pela quantidade de características

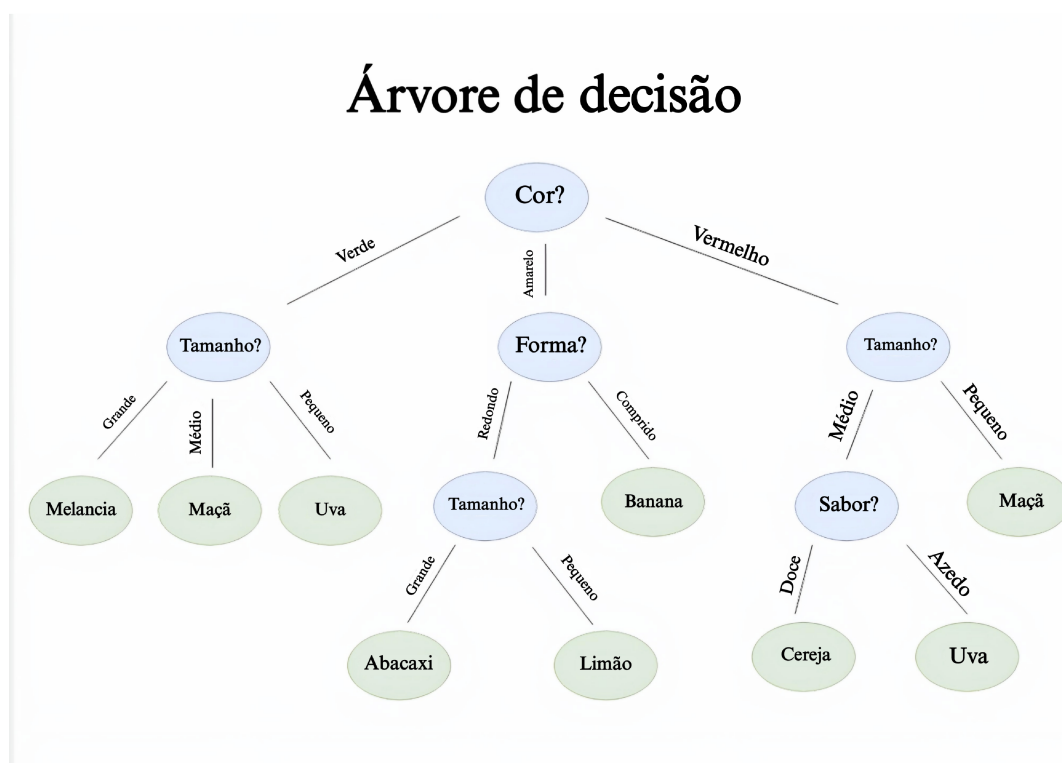


Figura 14 – Ilustração de árvore de decisão cujo objetivo é determinar que fruta corresponde a um conjunto de características. Fonte: <https://www.linkedin.com/pulse/intuition-decision-tree-ensemble-regression-ananlysis-gunji/>

de cada amostra. Um algoritmo não paramétrico comumente é mais complexo e portanto mais suscetível a *overfitting* e menos a *underfitting*.

A Figura 14 é um exemplo de uma árvore de decisão. Começando pelo topo (raiz), deve-se responder a perguntas condicionais - se determinada característica é X ou Y ou se é maior ou menor ao valor Z , por exemplo -. A cada resposta, evolui-se para um nó diferente, onde outra pergunta deverá ser respondida. Repete-se o processo até que se chegue a uma “folha”, que é um nó final contendo a resposta sobre qual classe ou valor a amostra corresponde. Uma estrutura mais hierárquica, com as nomenclaturas apropriadas para cada elemento de uma árvore, é apresentada na Figura 15.

A árvore de decisão possui uma vantagem imediata: é um modelo fácil de interpretar. No entanto, em comparação com algoritmos mais simples, como a regressão linear, ela pode sofrer com *overfitting* com uma maior facilidade.

O processo de treinamento da árvore de decisão consiste em determinar como serão arranjados os nós e quais os valores que serão determinantes para que o dado seja levado a um próximo nó à esquerda ou à direita. Uma forma de se realizar o treinamento é pelo algoritmo *CART* (*Classification and Regression Trees* - Árvores de Classificação e Regressão) pela otimização exaustiva de uma função custo. Podemos definir da seguinte

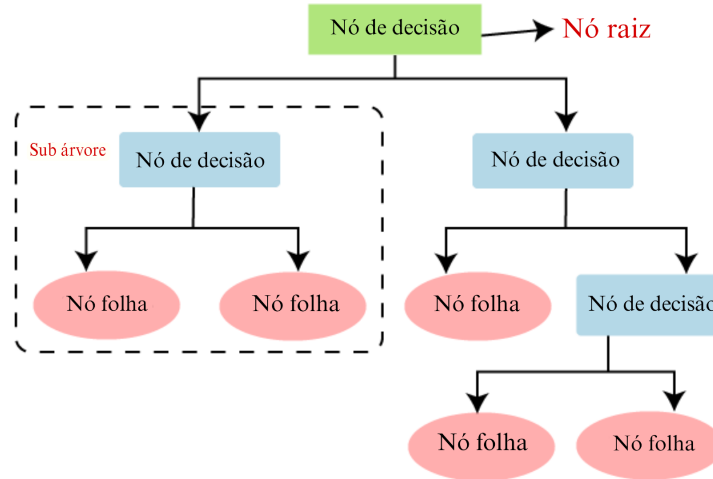


Figura 15 – Ilustração dos elementos de uma árvore de decisão. Fonte: <https://www.devops.ae/decision-tree-classification-algorithm/>

forma uma função custo a ser minimizada para um caso de regressão:

$$J(y; j, S) = \sum_{i \in x_{ij} > S} (y - y_i)^2 + \sum_{i \in x_{ij} \leq S} (y - y_i)^2, \quad (3.12)$$

onde i representa qual amostra individual está sendo analisada, y_i o valor da variável alvo real para esta amostra e N a quantidade total de amostras.

Na equação 3.12, J é função de y e é parametrizada por j e S , ou seja, escolhe-se a priori ambos valores e calcula-se o resto. O valor j significará qual característica da amostra será utilizada como divisor. Na Figura 14 o primeiro nó utiliza, por exemplo, como j o valor correspondente à característica cor. O valor S significa qual o valor de limiar de divisão, isto é, se decidíssemos utilizar uma característica numérica, qual seria o limiar de divisão. Poderia ser um nó que perguntasse se a idade de determinado indivíduo é menor ou igual a 20 anos, por exemplo. Nesse exemplo, a idade é a característica j e 20 é o valor para S . Ou seja, x_{ij} será interpretado como a j -ésima variável previsora referente à amostra i . Então, para um determinado valor de limiar e característica de análise, tem-se um único valor y escolhido. Isso é exemplificado na Figura 16 onde existem os limiares $S = [10, 15, 25]$ aproximadamente. A ideia é achar um valor para y que minimize os somatórios à esquerda e à direita (somatórios esses que se referem à parte superior e inferior de uma divisória criada).

Os nós são construídos de cima para baixo da seguinte forma: é escolhida uma característica do conjunto de treino e um limiar para aquela característica (ambas escolhas arbitrárias inicialmente) e a partir daí é selecionado um y que minimize a função custo. Então é realizado o processo novamente até que seja encontrado um valor de j e S ótimos, que represente o menor $J(y; j, S)$ para o primeiro nó, por isso é chamado de um processo exaustivo, pois os parâmetros serão atualizado exaustivamente até que se encontre o que retorne a menor função custo.

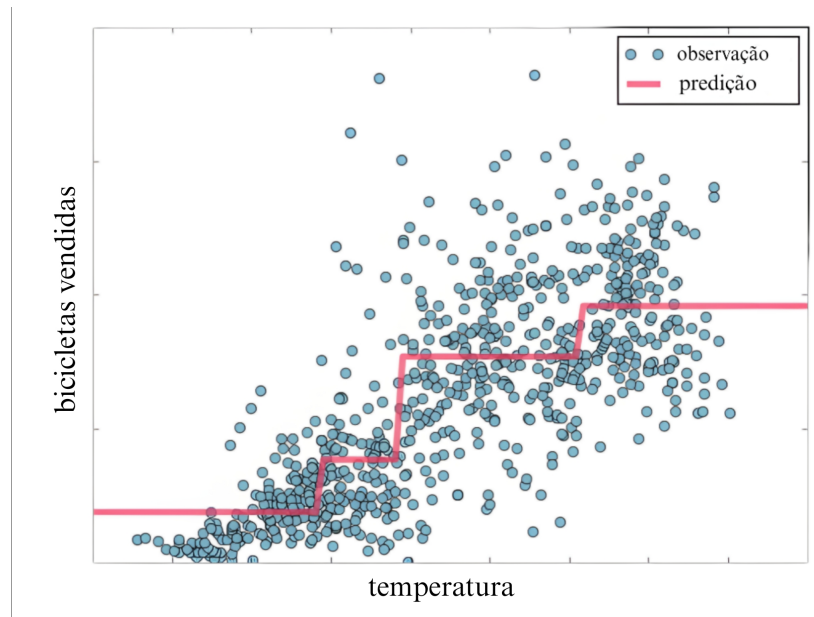


Figura 16 – Exemplo de uma árvore de decisão para um caso de regressão. O exemplo acima tenta prever a quantidade de bicicletas vendidas por dia em função da temperatura do dia. Fonte: <https://www.cambridgespark.com/info/getting-started-with-regression-and-decision-trees>

Para os nós subsequentes será realizado o mesmo processo até que se alcance o critério de parada. Normalmente esse critério é determinado pelo hiperparâmetro *max depth* que diz a profundidade máxima da árvore de decisão. Quanto maior a profundidade máxima, maior a tendência da árvore de decisão a sofrer um *overfitting*, dado que poderá se tornar extremamente específica.

3.2.3.1 XGBoost

Cada árvore de decisão é apenas uma das muitas possibilidades existentes. Juntar várias árvores pode gerar um modelo mais robusto, como um algoritmo que utilize a média do resultado das árvores para um problema de regressão. Esse tipo de algoritmo é chamado de *floresta aleatória*, quando n árvores são combinadas aleatoriamente. Embora ainda possa estar sujeito ao *overfitting*, ele é geralmente mais robusto e preciso do que uma única árvore de decisão.

Outra forma de usar várias árvores de decisão para gerar um algoritmo mais potente é o algoritmo XGBoost, que busca melhorar a performance a cada rodada. O XGBoost é baseado no processo de *boosting*, que é uma técnica de *ensemble* que utiliza vários modelos fracos para gerar um modelo mais robusto a *overfitting* ou com maior capacidade de generalização.

O método em questão aprimora modelos preditivos - especificamente, árvores de decisão - ao corrigir iterativamente os erros cometidos por árvores antecessoras. Ele emprega

um procedimento baseado em gradiente para reduzir esses erros de forma eficiente. O termo *extreme* se refere à capacidade do algoritmo de otimizar o desempenho computacional, valendo-se do paralelismo para processar dados simultaneamente em múltiplos núcleos de CPU. Adicionalmente, o XGBoost incorpora algumas técnicas de regularização, incluindo os termos de penalidade $L1$ e $L2$.

Seja a função custo a ser minimizada por cada árvore $L = \frac{1}{2}(y - \hat{y})^2$. É fácil verificar que seu gradiente é

$$-\frac{\partial L}{\partial \hat{y}} = y - \hat{y}. \quad (3.13)$$

O lado direito da equação 3.13 também é conhecido como *resíduo* (muitas vezes representado com a letra e). O sinal negativo no lado esquerdo tem apenas o propósito de tornar o lado direito em uma expressão que iremos utilizar adiante. Então teremos para a primeira rodada uma árvore gerada tal qual no caso das árvores de decisão comuns, ou em outras palavras, uma função $F_0(x)$. O *boosting* de gradiente gerará uma nova função $h_0(e)$ ou ainda $h_0(-\frac{\partial L}{\partial \hat{y}})$, levando a um próximo estimador na forma

$$F_1(x) = F_0(x) + h_0(e). \quad (3.14)$$

Serão construídos os estimadores seguintes de forma análoga à equação 3.14 até que o critério de parada seja atingido. A ideia de realizar esse processo é que, já que temos $e = y - \hat{y}$ ou, $e = y - F_0(x)$, isso significando que haverá a tentativa de construir uma função $h_n(e)$ que acabará compensando os erros gerados pela $F_n(x)$. De forma mais completa, haverá um termo de *taxa de aprendizagem* também, resultando numa expressão para o estimador final como

$$F_N(x) = F_0(x) + \eta \sum_{i=0}^{N-1} h_i(y - F_{i-1}(x)), \quad (3.15)$$

onde N é a quantidade total de rodadas que serão realizadas e η é a taxa de aprendizagem, ou em outras palavras, a contribuição de cada árvore (h_i). Aqui, por exemplo, são explícitos dois hiperparâmetros que devem ser ajustados manualmente, a taxa de aprendizagem η e a quantidade de árvores total N . Há uma tendência de quanto menor a taxa de aprendizagem, maior deverá ser a quantidade de árvores e vice-versa.

Na Figura 17 exemplifica-se como acontece a iteração de um algoritmo de *boost* de gradiente, onde de cima para baixo, a cada iteração da contabilização dos resíduos e da criação de uma função $h(x_1)$ que se adeque à esses novos resíduos, o modelo preditivo resultante à direita vai se tornando cada vez melhor.

3.2.4 Máquina de Vetores de Suporte

As máquinas de vetores de suporte foram criadas com o objetivo principal de realizar tarefas de classificação. O algoritmo SVM, ou *Support Vector Machine* (máquina

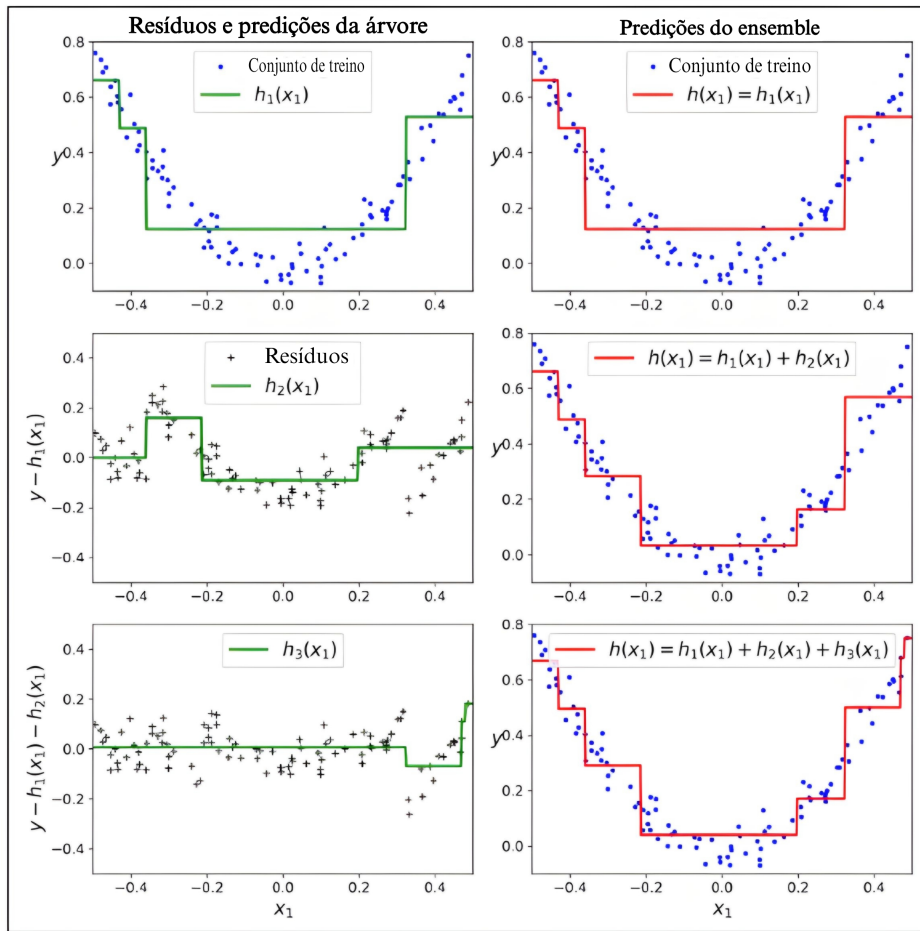


Figura 17 – Exemplo de como fica a regressão via *Boost de Gradiente*. À direita como o modelo preditivo se modifica a cada iteração que modifica o $h(x_1)$ total e à esquerda como os resíduos vão se modificando e como a função h dos resíduos se adequa a eles. [33]

de vetores de suporte) foi inicialmente desenvolvido para resolver problemas binários, em que a variável alvo é categórica e apresenta apenas dois rótulos, como distinguir se uma amostra é um cão ou um gato.

O processo de classificação é o primeiro aspecto a ser explicado antes de discutir a adaptação do algoritmo para realizar tarefas de regressão. Para obter o algoritmo de regressão, modificações são feitas a partir do processo de classificação.

A Figura 18 ilustra um problema linearmente separável, ou seja, quando é possível distinguir as classes por uma reta (ou hiperplano) no espaço gerado pelas variáveis explicativas. No entanto, simplesmente traçar uma reta que separa as classes não é a forma mais inteligente de abordar o problema.

Para encontrar uma forma melhor de separar essas categorias, foram introduzidos os vetores de suporte, representados pelas linhas tracejadas à esquerda e à direita da reta contínua que separa as classes na Figura 18. É importante destacar que, na imagem à

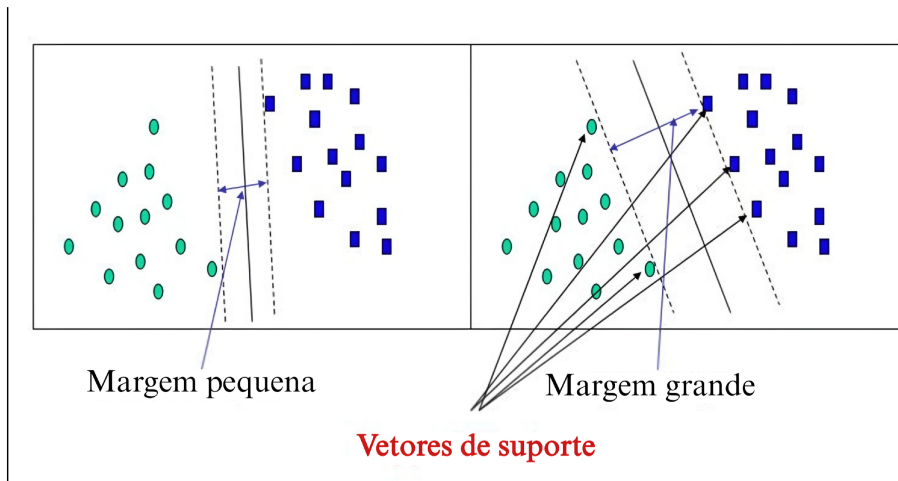


Figura 18 – Ilustração de como funciona o algoritmo SVM para um problema de classificação binário. Fonte: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a4>

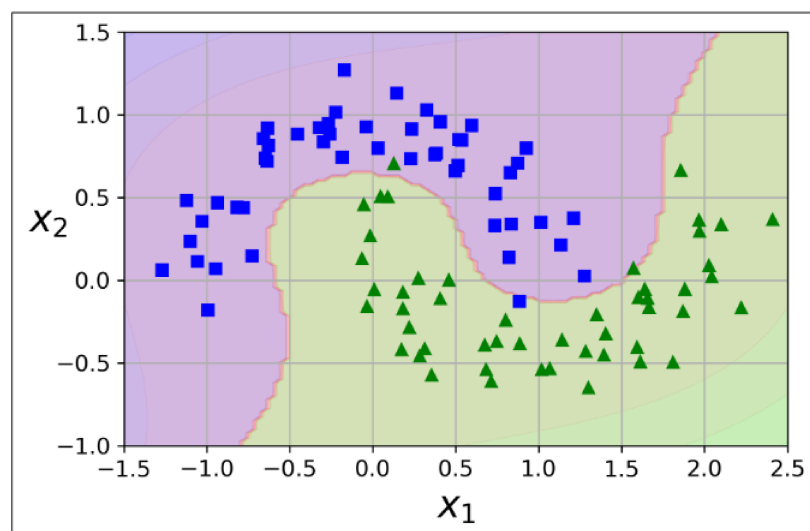


Figura 19 – Exemplo de um modelo criado através de uma transformação polinomial para o algoritmo SVM. [33]

esquerda desta figura, os vetores de suporte estão mais próximos da reta central do que na imagem à direita. O objetivo do algoritmo SVM é maximizar a distância entre os vetores de suporte e a reta central, ou seja, maximizar o tamanho das margens.

Assume-se que, quanto maior a margem encontrada, menor será o erro do modelo preditivo final gerado pelo algoritmo SVM.

Em alguns casos, não é possível criar uma separação linear satisfatória utilizando as variáveis explicativas disponíveis. Nesses casos, uma alternativa é transformar essas variáveis, como no processo explicado anteriormente para a criação de variáveis polinomiais na subseção 3.2.1. Dessa forma, seria possível criar uma reta divisória com margens não-lineares, como na Figura 19.

Se para o problema de classificação a tentativa é de se criar uma separação que contenha a maior margem possível, para a regressão a tentativa é de se criar um modelo que contenha a maior quantidade de pontos possíveis dentro da margem.

Uma primeira diferença é que o tamanho da margem se torna um hiperparâmetro, e daí tem que ser selecionado a priori pelo usuário. Esse hiperparâmetro é chamado ε . Na verdade, o algoritmo de regressão é quase que completamente baseado nesse parâmetro ε . O algoritmo SVR (*Support Vector Regressor* ou Regressor de Vetor de Suporte) normalmente utiliza a seguinte função de custo a ser minimizada:

$$\frac{1}{2} \|b\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*), \quad (3.16)$$

sob as condições:

$$\begin{cases} y_i - (\langle b, x_i \rangle + a) \leq \varepsilon + \xi_i, \\ (\langle b, x_i \rangle + a) - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases} \quad (3.17)$$

O primeiro termo da expressão 3.16 é a norma $L2$ em função dos pesos b para cada variável explicativa (b aqui representa um vetor p -dimensional onde p é a quantidade de variáveis). O segundo termo, por outro lado, representa a função ε -insensível como função perda para cada ponto:

$$|\xi|_\varepsilon = \begin{cases} 0, & \text{se } |\xi| \leq \varepsilon, \\ |\xi| - \varepsilon & \text{se } |\xi| > \varepsilon. \end{cases} \quad (3.18)$$

ξ e ξ^* são chamadas variáveis de folga. A ideia é que para qualquer ponto que se tenha fora da margem delimitada por ε , será contabilizado um erro, como representado na Figura 20,

Existem duas variáveis de folga que facilitam o processo de derivação de otimização. A primeira variável, ξ , é maior do que zero quando a amostra está acima da margem e zero para qualquer região abaixo da margem superior. A segunda variável, ξ^* , é diferente de zero para amostras abaixo da margem inferior e zero para qualquer região acima da margem inferior. Essas variáveis são ilustradas na Figura 20, na qual as amostras fora da margem possuem um valor de ξ ou ξ^* .

Essas condições de existência são traduzidas matematicamente na expressão de restrição 3.17, onde $\langle b, x_i \rangle$, é o produto escalar entre o vetor b de componentes correspondentes aos pesos e x_i o vetor correspondente à i -ésima amostra, ou seja, a variável alvo encontrado pela reta resultante do algoritmo SVR pode ser escrita como $\hat{y}_i = \langle b, x_i \rangle + a$.

Na equação 3.16, o hiperparâmetro C representa o *trade-off* entre a tolerância do distanciamento das amostras fora da margem e o quanto a curva é achatada (neste

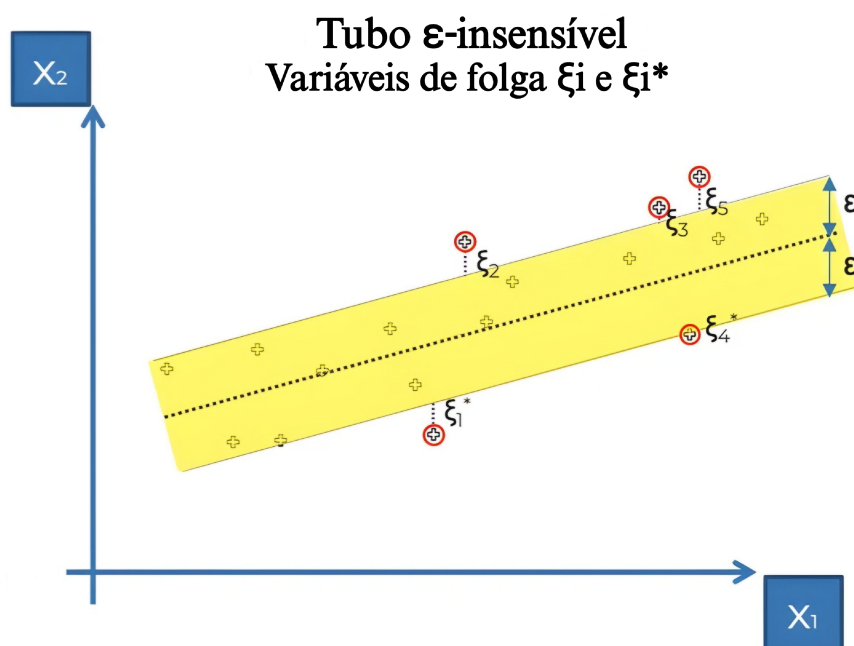


Figura 20 – Exemplo de como funciona a contabilização das variáveis de folga ξ . Fonte: <https://medium.com/analytics-vidhya/support-vector-regression-for-machine-learning-843978ba6279>

contexto uma reta mais achatada seria aquela que possui seus coeficientes mais próximos de 0). Especificamente, quanto maior for o valor de C , maior será o impacto de uma amostra com valor de ξ ou ξ^* , e maior será a tentativa do algoritmo de minimizar esse distanciamento. Por outro lado, se C for pequeno, a norma $L2$ terá maior impacto na função custo, o que levará o algoritmo a encontrar uma curva mais achatada.

A norma $L2$ é incluída na função custo por dois motivos principais. Em primeiro lugar, atua como um termo de regularização, reduzindo a tendência do algoritmo SVR de ajustar-se excessivamente às variáveis explicativas e diminuindo o peso que cada variável tem. Em segundo lugar, garante que, em situações em que seja possível escolher qualquer tubo ε -insensível (por exemplo, quando todos os pontos estão dentro do tubo com uma certa “folga”), seja escolhida a reta com os menores pesos.

Um último detalhe é que as amostras fora das margens de distanciamento ε são os vetores de suporte no caso de regressão, isto é, a distância da margem a esses vetores definirá o formato do modelo final encontrado.

3.2.5 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são algoritmos de aprendizado de máquina bi-informados (algoritmos orientados por sistemas e funcionamentos biológicos). Elas tentam imitar o funcionamento das redes neurais biológicas através de processos matemáticos.

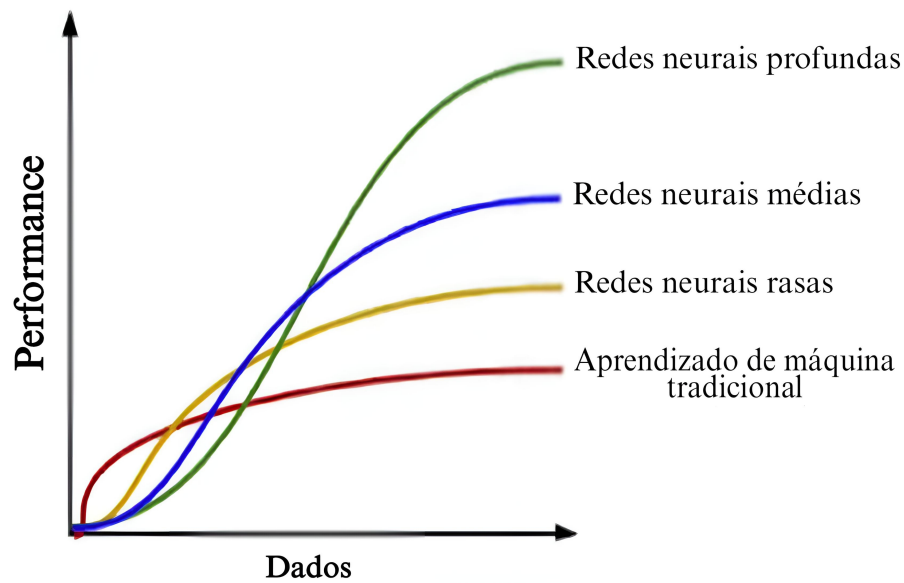


Figura 21 – Quantidade disponível de dados por performance. Em *verde* temos redes neurais profundas, em *azul* redes neurais médias, em *amarelo* redes neurais rasas e em *vermelho* algoritmos de aprendizado de máquina tradicionais. [35]

A história das RNAs começou na década de 1940, quando McCulloch e Pitts [34] propuseram o primeiro modelo de neurônio artificial. Desde então, muitas pesquisas e desenvolvimentos teóricos foram realizados. No entanto, a aplicação prática das RNAs era limitada até o início do século XXI, devido à necessidade de grandes quantidades de dados e poder de processamento computacional.

A Figura 21 ilustra como a profundidade de uma RNA (um conceito que será explicado posteriormente) influencia seu desempenho. Em comparação com algoritmos de aprendizado de máquina tradicionais, as RNAs rasas já apresentam uma vantagem significativa.

Todo algoritmo de aprendizado de máquina atinge um platô de desempenho em algum momento. No entanto, as RNAs têm a vantagem de atingir esse platô em um nível de desempenho mais alto e demoram mais para chegar lá. Mais uma vez, o termo “momento” ao qual nos referimos aqui é sobre a quantidade de dados para treinamento.

Este trabalho concentra-se na utilização do *Perceptron de Multicamadas* (PMC) devido à sua simplicidade de implementação e potencial para alto desempenho.

3.2.5.1 Estrutura de uma RNA

Uma RNA é composta por cinco elementos principais: camada de entrada, camadas ocultas, camadas de saída, pesos e vieses. A Figura 22 ilustra a interação entre esses elementos.

A camada de entrada de uma rede neural é singular e cada nó nela representa um atributo distinto do conjunto de dados. Por exemplo, se possuímos um conjunto de cem amostras com os atributos idade, peso e altura, teríamos três neurônios na camada de entrada, cada um correspondendo a um desses atributos.

Cada neurônio na camada de entrada está associado a uma quantidade de pesos igual ao número de neurônios na camada subsequente, que é uma camada oculta. O valor de saída de cada neurônio é multiplicado por um peso correspondente, somado a um viés e, em seguida, passado por uma função de ativação.

Este processo é conhecido como *feedforward*. O termo *feedforward* é usado porque as informações fluem através da rede na direção “para frente”, ou seja, da camada de entrada para a camada de saída, sem formar ciclos ou *loops*. Isso garante que as informações usadas para alimentar a rede neural são transmitidas de maneira eficiente e direta.

Matematicamente, o processo de *feedforward* pode ser descrito da seguinte maneira. Seja x a entrada para um nó, b os pesos associados a essas entradas e a o viés. A saída y desse nó é calculada pela expressão

$$y = f\left(\sum_i b_i x_i + a\right), \quad (3.19)$$

onde f é a função de ativação. As funções de ativação mais comuns incluem a função sigmóide, ReLU (*Rectified Linear Unit*) e a tangente hiperbólica.

Esse processo é repetido para cada nó em cada camada, com a saída de uma camada servindo como entrada para a próxima. A saída final da rede é então calculada da mesma maneira.

Quando se tem uma matriz de entrada $N \times M$, onde N é o número de registros em um conjunto de dados e M é o número de atributos, a expressão matemática para o *feedforward* é alterada para acomodar a natureza matricial dos dados.

Seja X a matriz de entrada $N \times M$, W a matriz de pesos, e b o viés. A saída Y de uma rede é calculada como:

$$Y = f(WX + b), \quad (3.20)$$

onde f é a função de ativação aplicada elemento a elemento. A multiplicação de matrizes XB realiza a soma ponderada das entradas para cada registro e cada nó. O vetor de viés a é então adicionado a cada soma ponderada (isso é chamado de *broadcasting* em termos de operações de matriz). Finalmente, a função de ativação é aplicada a cada valor resultante para produzir a saída da rede.

O treinamento de redes neurais envolve encontrar os pesos e vieses ótimos, minimizando a função custo. Normalmente, utiliza-se o erro quadrático médio entre o valor real

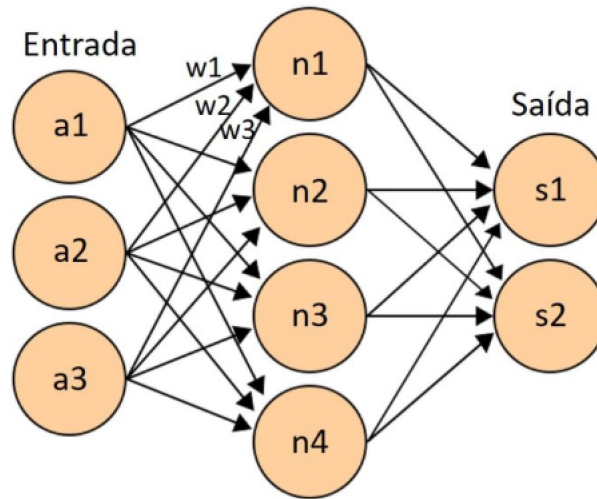


Figura 22 – Diagrama esquemático da estrutura de um *Perceptron de Multicamadas* (PMC). Cada nó representa um neurônio e cada segmento de reta um peso. Os nós com inicial “a” (a1, a2, a3) forma a camada de entrada, os nós com inicial “n” as camadas ocultas e de inicial “s” a camada de saída. Fonte: <https://didatica.tech/introducao-a-redes-neurais-e-deep-learning/>

e o valor previsto pela rede neural. Esse processo inclui a atualização dos pesos após o *feedforward* e a aplicação do *backpropagation* (retropropagação).

O *backpropagation* calcula o gradiente do erro obtido ao final do processo *feedforward* em relação a todos os pesos da rede, propagando esse gradiente de volta através da rede, da saída para a entrada. Isso permite que a rede atualize seus pesos iterativamente a cada rodada.

Matematicamente, o *backpropagation* usa o cálculo para determinar a mudança necessária em cada peso para reduzir o erro. Seja S o erro total da rede, b um peso específico na rede e η a taxa de aprendizado. A atualização do peso é dada por

$$\Delta b = -\eta \frac{\partial S}{\partial w}, \quad (3.21)$$

a derivada $\frac{\partial S}{\partial w}$ é calculada usando a regra da cadeia - uma descrição mais minuciosa de como é feito o cálculo pode ser encontrada em [36] -, propagando o erro de volta através da rede. Isso é feito para cada peso na rede, atualizando todos os pesos para minimizar o erro. O sinal negativo em 3.21 é para que a atualização dos pesos vá em direção ao 0 e o η é para informar o quão grande deve ser cada passo de atualização para o peso em específico.

3.3 Importância de Atributos via *SHAP*

Ao lidar com modelos de aprendizado de máquina, muitas vezes pode ser de interesse de quem os constroi, a interpretação deste, sendo em geral quais os atributos (no caso dessa pesquisa, propriedades moleculares) mais relevantes para cada modelo. Essa análise pode fornecer pistas valiosas para os fenômenos que estão sendo modelados

Em geral, a melhor interpretação é o próprio modelo, por exemplo, para uma regressão linear, a melhor forma de interpretar a relação entre as variáveis explicativas e a variável alvo são os próprios coeficientes da regressão.

Muitas vezes, os modelos de aprendizado de máquina não se tratam de algo que possa ser “aberto”, isto é, obter-se uma relação analítica e seus devidos coeficientes. Então, ao invés de se utilizar o próprio modelo para sua explicabilidade e interpretação, utiliza-se outros modelos e técnicas para os explicar.

Nesta seção se explicará como pode-se calcular as importâncias das variáveis explicativas utilizadas nesse trabalho, através de valores *SHAPLEY*, e qual a diferença para valores *SHAP*.

3.3.1 Valores SHAPLEY e SHAP

Os valores Shapley, desenvolvidos por Lloyd Shapley [37] na teoria de jogos cooperativa, têm como objetivo quantificar a contribuição individual de cada jogador para o resultado esperado de um jogo.

Esses valores são particularmente úteis em situações onde os atores formam coalizões, que podem ser entendidas como equipes, visando a obtenção de um prêmio em um jogo.

Por exemplo, imagine uma situação em que três empresas (A, B e C) estão considerando formar uma aliança para licitar um projeto. Se ganharem, o lucro será distribuído entre elas. No entanto, cada empresa tem diferentes recursos e habilidades, contribuindo de maneira diferente para a vitória. A empresa A tem uma forte rede de contatos, a empresa B tem experiência em projetos semelhantes e a empresa C tem acesso a tecnologia avançada. Usando os valores Shapley, é possível quantificar a contribuição de cada empresa para a vitória, permitindo uma divisão justa do lucro.

Matematicamente, pode-se representar os valores Shapley com a expressão

$$\varphi_i(v) = \frac{1}{n} \sum_{C \subseteq N \setminus \{i\}} \binom{n-1}{|C|}^{-1} (v(C \cup \{i\}) - v(C)). \quad (3.22)$$

Trazendo a interpretação da equação 3.22 para o universo deste trabalho, onde os jogadores aqui se tratam das propriedades moleculares e as coalizões combinações dessas propriedades, $\varphi_i(v)$ representa a contribuição média esperada para uma propriedade i . O termo $v(C \cup$

$\{i\}) - v(C)$ representa a diferença entre o valor da “competição” considerando o atributo i na coalizão e o desconsiderando (no caso de modelos de regressão se trataria do desempenho referente à alguma métrica).

$N \setminus \{i\}$ representa o conjunto de todas as propriedades moleculares excetuando a propriedade i , n a quantidade total de propriedades e C algum subconjunto selecionado de propriedades. A iteração do somatório é realizada nas diversas coalizões (combinações de propriedades moleculares) possíveis no conjunto de todas as propriedades moleculares que não possuam a propriedade molecular i .

O inverso da combinação de $n - 1$ tomados C a C serve para apresentar um peso que representa a probabilidade de incluirmos a propriedade i nas coalizões. (isso pois uma determinada propriedade pode ser incluída numa coalizão de diversas maneiras, como o primeiro ou segundo elemento etc).

A ideia dessa equação é comparar o desempenho dos modelos com e sem a presença de uma determinada propriedade molecular com um peso baseado nas possibilidades de se montar uma coalizão sem sua presença. Dessa forma, consegue-se obter o valor de participação, ou ainda neste caso, o quanto da redução do erro dos modelos se deve a uma determinada propriedade molecular.

Os valores SHAP (*Shapley additive explanations*) surgem com o intuito de sanar algumas deficiências dos valores Shapley. O método SHAP é quem de fato substitui os jogadores das coalizões pelos atributos inseridos nos modelos de aprendizado de máquina. Além disso, em vez de usar os valores estimados como recompensa de uma competição, os valores SHAP consideram os valores preditos ou o desempenho do modelo.

Outra questão importante sobre os valores SHAP é que o cálculo dos valores Shapley é computacionalmente custoso. Portanto, o algoritmo criado por Scott *et al.* [38], que é utilizado neste trabalho, visa ser uma aproximação menos dispendiosa e adaptada para ser agnóstica em relação aos modelos de aprendizado de máquina, ou seja, independente da natureza e dos pressupostos de cada modelo.

3.4 Escalonamento de Atributos

No preparo de dados para algoritmos de aprendizado de máquina, o escalonamento das variáveis explicativas é crucial para garantir um treinamento eficiente e eficaz. Escalas muito distintas podem comprometer a performance e a convergência do modelo, especialmente em algoritmos sensíveis à magnitude dos dados.

Algoritmos como regressões lineares e redes neurais usam gradiente descendente para encontrar o mínimo da função custo. Com variáveis em escalas diferentes, a atualização dos pesos se torna desigual, dificultando a convergência. Isso impacta a velocidade de

treinamento e pode levar a resultados sub-otimizados.

Para algoritmos baseados em distância, como máquinas de vetor de suporte, variáveis com uma maior escala podem acabar sendo favorecidas durante o processo de treinamento, podendo tornar o algoritmo enviesado e podendo comprometer sua performance.

Por via de regra, torna-se uma boa prática durante os processos de treinamento de algoritmos de aprendizado de máquina escalonar os atributos para escalas semelhantes. Veremos a seguir os três tipos de escalonamento que foram utilizados durante o processo de treinamento dos algoritmos nesse trabalho.

Embora cada técnica de escalonamento seja mais indicada para situações específicas, é recomendável testar diversas técnicas em cada conjunto de dados ou problema de modelagem. Isso ocorre porque o comportamento das técnicas pode variar, por exemplo, a que geralmente se supõe ter melhor desempenho em dados normais pode não ser a melhor em todos os casos com distribuições normais.

A interpretação dos valores escalonados individualmente não é usual. O foco está na melhora da performance dos modelos com os dados escalonados. Questões como os valores específicos que cada variável assume após o escalonamento não são tão relevantes para a modelagem via aprendizado de máquina.

3.4.1 Padronização

A padronização, também conhecida como *z-scores*, é uma técnica de escalonamento que transforma os valores de uma distribuição em distâncias múltiplas do desvio padrão em relação à média. Isso significa que cada valor é convertido em uma medida de quantas unidades de desvio padrão ele está distante da média. A expressão para calcular a padronização é

$$z = \frac{X - \theta}{\sigma}, \quad (3.23)$$

onde z é o valor escalonado, X é o valor original, θ e σ são a média e o desvio padrão da distribuição da variável, respectivamente. Uma das particularidades dessa técnica é que em modelos de regressão linear, os coeficientes padronizados representam o efeito de cada variável sobre a variável dependente em unidades de desvio padrão. A padronização também garante que os resíduos em um modelo de regressão linear sigam uma distribuição normal, o que é uma importante suposição para a validação do modelo.

No entanto, a padronização também apresenta algumas limitações. Ela pode levar à perda de informação sobre a distribuição original dos dados, dificultar a interpretação dos coeficientes em distribuições não normais e ser sensível a *outliers*. A técnica também assume que os dados seguem uma distribuição normal, o que pode invalidar os resultados do modelo se essa suposição não for válida.

A padronização é uma técnica útil para preparar dados com distribuições normais para algoritmos de aprendizado de máquina. No entanto, se a distribuição original dos dados é importante e deve ser preservada, se os dados não seguem uma distribuição normal ou se a interpretação dos coeficientes em unidades de desvio padrão não é importante, outras técnicas de escalonamento podem ser mais adequadas.

3.4.2 Normalização

A normalização é uma técnica de escalonamento que transforma os valores de uma variável para um intervalo específico, geralmente entre 0 e 1 ou -1 e 1. Essa técnica visa tornar as variáveis comparáveis entre si e facilitar a interpretação dos resultados. A forma mais comum de se calcular os valores normalizados é:

$$z = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (3.24)$$

onde z é o valor normalizado, X o valor original, X_{min} e X_{max} os valores mínimos e máximos da variável respectivamente. A normalização na equação 3.24 transforma os valores para um intervalo entre 0 e 1.

Assim como a padronização, a normalização permite a comparação direta entre variáveis que originalmente estavam em diferentes escalas, facilitando a interpretação dos resultados, e tornando os dados mais adequados para alguns algoritmos de aprendizado de máquina.

Em geral, tende-se a preferir a normalização ao invés da padronização em situações em que as variáveis não seguem uma distribuição normal ou no caso em que assumam valores negativos. Porém um ponto negativo da normalização, é que pode-se perder relações entre variáveis, enquanto que na padronização as relações entre variáveis se mantêm.

3.4.3 Escalonamento Robusto

As técnicas de escalonamento, como padronização e normalização, visam tornar as variáveis comparáveis e facilitar a interpretação dos resultados. No entanto, a padronização e a normalização podem ser sensíveis a *outliers* e apresentar resultados distorcidos quando os dados não seguem uma distribuição normal (para o caso da padronização).

O escalonamento robusto surge como uma alternativa eficaz para lidar com esses desafios. Em vez de depender da média e do desvio padrão, o escalonamento robusto utiliza medidas de centralidade e dispersão como a mediana e o desvio interquartil (IQR - distância entre o primeiro e terceiro quartil). A expressão para esse escalonamento é

$$z = \frac{X - M}{\text{IQR}}, \quad (3.25)$$

onde z é o valor escalonado, X o valor original, M e IQR se referem à mediana e ao desvio interquartil, respectivamente.

As vantagens desse escalonamento são: o IQR é menos influenciado por valores extremos do que o desvio padrão; não há a suposição de que os dados sigam uma distribuição normal e preserva a relação entre variáveis, o que pode ser relevante para alguns algoritmos de aprendizado de máquina.

Duas desvantagens desse escalonamento é que pode ser menos eficiente para distribuições de dados normais e possuir uma interpretação mais complexa.

3.5 Otimização de hiperparâmetros

O desenvolvimento de modelos de aprendizado de máquina envolve a escolha e ajuste de hiperparâmetros, que são parâmetros que controlam o processo de aprendizagem. Estes parâmetros, como a quantidade de camadas em uma rede neural ou o valor da regularização (ver subseção 3.2.2) em uma regressão linear, podem ter um impacto significativo no desempenho do modelo.

A escolha dos hiperparâmetros é crucial para otimizar o desempenho do modelo em uma tarefa específica. Diversas técnicas de otimização de hiperparâmetros são utilizadas para encontrar o conjunto ideal de parâmetros, como a busca exaustiva e a busca aleatória.

A busca exaustiva testa todas as combinações possíveis de valores dentro do espaço de busca (conjunto de valores possíveis para cada hiperparâmetro). Essa técnica garante encontrar o conjunto de hiperparâmetros que maximiza o desempenho do modelo, mas pode ser computacionalmente inviável para grandes espaços de busca.

A busca aleatória testa combinações aleatórias de valores dentro do espaço de busca. Essa técnica é mais rápida que a busca exaustiva, mas não garante encontrar o conjunto de hiperparâmetros ideal.

Diante das limitações das técnicas de busca exaustiva e aleatória, a otimização bayesiana com processos gaussianos surge como uma alternativa promissora. Essa técnica combina um modelo probabilístico a priori (processo gaussiano) com uma estratégia de busca inteligente para encontrar o mínimo de uma função objetivo (geralmente a métrica de desempenho do modelo).

3.5.1 Otimização Bayesiana

A otimização bayesiana é uma abordagem poderosa para otimização de hiperparâmetros que incorpora incerteza no processo. Essa abordagem utiliza o teorema de Bayes para atualizar a distribuição de probabilidade dos hiperparâmetros à medida que novas informações são coletadas.

A principal vantagem da otimização bayesiana reside em sua capacidade de realizar uma busca mais eficiente e direcionada. Ao contrário da busca exaustiva, que testa todas as combinações de hiperparâmetros, a otimização bayesiana utiliza o conhecimento adquirido em avaliações anteriores para direcionar a busca para regiões com maior probabilidade de conter o conjunto ideal de parâmetros.

Ao mesmo tempo, a otimização bayesiana evita as armadilhas da busca aleatória. Em vez de depender de combinações aleatórias, a otimização bayesiana utiliza um modelo probabilístico para estimar a performance do modelo em diferentes pontos do espaço de busca. Essa estimativa é então utilizada para direcionar a busca para as regiões mais promissoras.

Os processos gaussianos são uma classe de modelos probabilísticos que podem ser utilizados para representar funções desconhecidas. Eles são caracterizados por sua média e função de covariância, que definem a distribuição conjunta de qualquer conjunto de pontos da função.

Na otimização bayesiana, os processos gaussianos são frequentemente utilizados como modelos a priori para a função objetiva. À medida que as avaliações da função objetiva são coletadas, o modelo de processo gaussiano é atualizado utilizando o teorema de Bayes, resultando em uma distribuição a posteriori dos hiperparâmetros.

Não é intuito deste trabalho fornecer uma descrição detalhada sobre a matemática por trás de tal processo, porém Frazier [39] demonstra todo o arcabouço e condições necessárias para quem queira reproduzir passo a passo o processo de otimização.

Em geral algumas das vantagens apresentadas pela otimização bayesiana via processos gaussianos são: os processos gaussianos podem modelar uma ampla variedade de funções, tornando-os adequados à diversos tipos de problemas; a otimização bayesiana pode fornecer uma estimativa da incerteza associada aos hiperparâmetros otimizados, o que pode ser útil em tomadas de decisões mais informadas; a busca por hiperparâmetros pode ser muito mais assertiva do que algoritmos de busca aleatória e como dito anteriormente, costuma apresentar um custo computacional muito menor do que buscas exaustivas.

Algumas de suas possíveis limitações, são que as inferências das funções objetivas podem se tornar complexas em modelos que possuam muitos hiperparâmetros ou ainda, ter aumento em seu custo computacional em casos de conjuntos de dados muito grandes ou com alta dimensionalidade.

Como não há nenhum desses problemas nos conjuntos de dados e modelos utilizados nessa pesquisa, a otimização bayesiana via processos gaussianos se torna uma ferramenta poderosa.

Para exemplificar o processo, considere um modelo de aprendizado de máquina no qual busca-se determinar o valor ideal de um hiperparâmetro para otimizar a performance

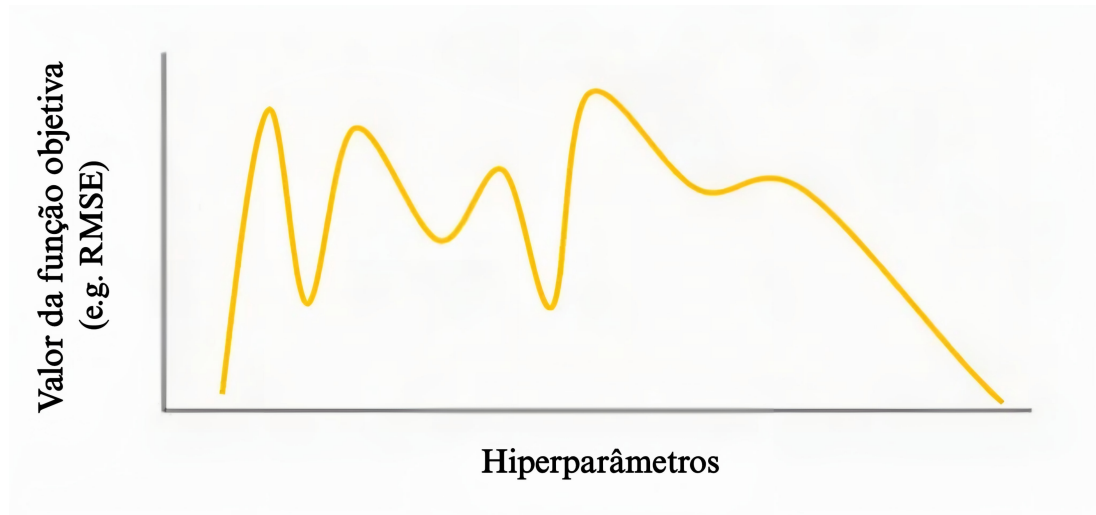


Figura 23 – Função objetiva verdadeira. Fonte: <https://towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f>

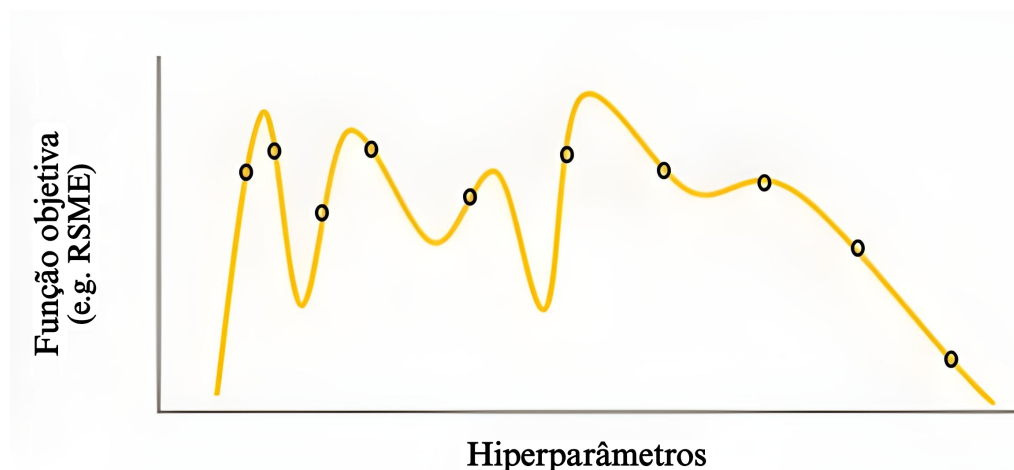


Figura 24 – Amostragem de pontos na função objetiva verdadeira. Fonte: <https://towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f>

do modelo. Em um cenário ideal com poder computacional ilimitado, poderia-se testar todos os valores possíveis desse hiperparâmetro até encontrar o mínimo global (ou máximo, dependendo da métrica de avaliação). Contudo, dada a limitação computacional, é necessário selecionar quais pontos serão avaliados na função objetiva.

Na Figura 23 há o esboço do que poderia ser um gráfico da métrica de avaliação de um modelo em função do valor de um hiperparâmetro. Essa, no caso, se refere à função real.

O próximo passo é amostrar pontos aleatórios dessa função (dado que não se pode realizar o cálculo em todos os pontos, principalmente em funções de domínio em $\mathbb{R}$) e calcular a performance nesses pontos como mostrado em 24. Em seguida, utilizando

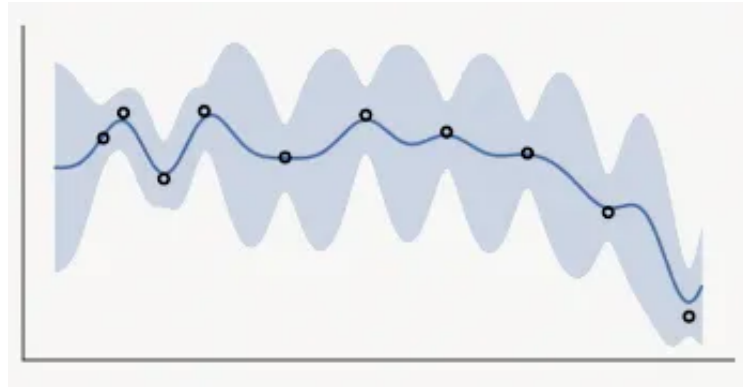


Figura 25 – Inferência da função objetiva verdadeira via processos gaussianos. Fonte: <https://towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f>

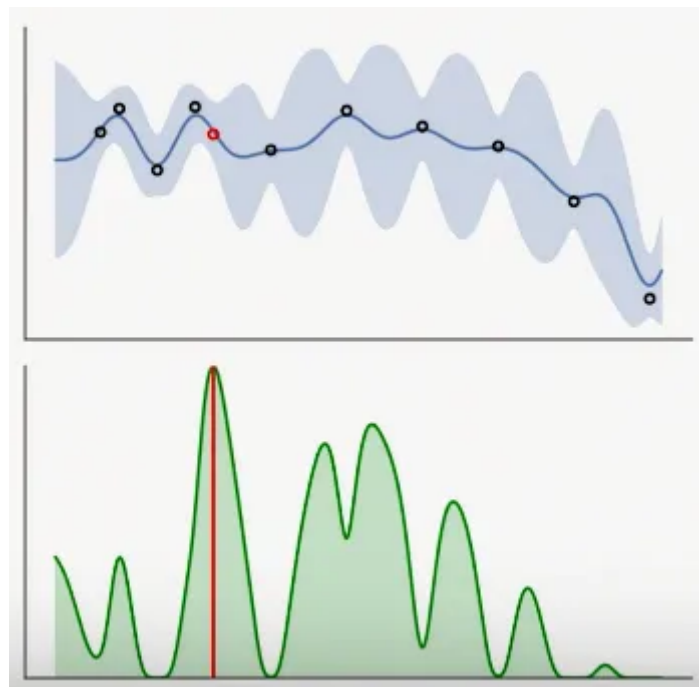


Figura 26 – Ilustração de uma possível função de aquisição, com o ponto em vermelho sendo o eleito para próximo teste de amostragem para repetição do processo via otimização bayesiana. Fonte: <https://towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f>

processos gaussianos, modelamos esses dados para inferir, com um intervalo de confiança, a função objetiva real, conforme ilustrado na Figura 25.

Com base nessa inferência a partir das amostras, calcula-se uma nova função, denominada função de aquisição, que orienta a busca por candidatos com maior probabilidade de representarem pontos de melhor desempenho. Esse processo de amostragem, inferência e seleção do próximo ponto é reiniciado.

A Figura 26 exemplifica a escolha da próxima amostra com base em uma função

de distribuição de probabilidade, onde o próximo valor do hiperparâmetro é aquele que apresenta o máximo valor na função de aquisição.

Esse processo é repetido até que um critério de convergência seja atingido, podendo ser uma quantidade pré-definida de iterações pelo usuário ou a ausência de melhoria na função objetiva após um número determinado de iterações, resultando na identificação do mínimo.

É importante destacar que não há garantia de encontrar o mínimo ou máximo global da função objetiva, mas o objetivo é empregar uma técnica menos aleatória e computacionalmente mais eficiente.

4 Metodologia

O principal objetivo desta pesquisa é explorar como algoritmos de aprendizagem de máquina podem contribuir para o entendimento de fenômenos físicos, mais especificamente investigando quais propriedades ou conjunto de propriedades moleculares exercem maior influência no fenômeno de ligação do pósitron com moléculas. Descreveremos a seguir as metodologias empregadas para o desenvolvimento dos resultados aqui obtidos.

O trabalho foi conduzido em várias etapas. Inicialmente, realizou-se uma revisão bibliográfica abrangente sobre os principais métodos utilizados para descrever as energias de ligação do pósitron com moléculas, detalhado nos capítulos de introdução de propriedades moleculares. Além disso, foram estudados alguns métodos atuais de aprendizado de máquina e suas formas mais adequadas para lidar com conjuntos de dados pequenos, conforme discutido no Capítulo 3.

A etapa seguinte envolveu a coleta de dados para alimentar os modelos de aprendizado de máquina. A qualidade e quantidade de dados são fatores determinantes para o sucesso dos modelos, como mencionado no capítulo 3. Os dados coletados estão apresentados nas Tabelas 3 e 4, com a legenda especificando as fontes das informações.

Tabela 3 – Moléculas e suas propriedades utilizadas na pesquisa. As propriedades moleculares, I_p , α e μ , representando os potenciais de ionização em meV, polarizabilidades em $\text{\AA}^3$ e momentos de dipolo em Debye respectivamente, foram extraídas de [40]. Os valores α_{xx} , α_{yy} e α_{zz} (em $\text{\AA}^3$) foram retirados de cálculos de [41], tendo escolhido a base *aug-cc-pVDZ* e a função *B3LYP*. A polarizabilidade média de α_{xx} , α_{yy} e α_{zz} ($\bar{\alpha}$), é dado também em ($\text{\AA}^3$). Os valores experimentais das energias de ligação do pósitron (ϵ_b , medidos em meV) foram obtidos de [5, 9, 10, 23, 26, 27, 42].

Molécula	Fórmula	I_p	α	α_{xx}	α_{yy}	α_{zz}	$\bar{\alpha}$	N_π	μ	ϵ_b
Butano	C ₄ H ₁₀	10.6	8.1	9.3	7.7	7.2	8.1	0	0	35
Pentano	C ₅ H ₁₂	10.4	10	11.7	9.4	8.7	10	0	0	60
Hexano	C ₆ H ₁₄	10.2	11.8	14.3	11	10.2	11.9	0	0	80
Heptano	C ₇ H ₁₆	9.9	13.7	16.9	12.7	11.7	13.8	0	0	105
Octano	C ₈ H ₁₈	10	15.5	19.6	14.3	13.2	15.7	0	0	115
Nonano	C ₉ H ₂₀	10	17.4	21.3	14.9	14	16.8	0	0	145
Dodecano	C ₁₂ H ₂₆	9.9	22.9					0	0	220
Hexadecano	C ₁₆ H ₃₄	9.9	30.3					0	0	310
Etileno	C ₂ H ₄	10.5	4.2	5.4	3.8	3.3	4.2	1	0	20
Ciclopropano	C ₃ H ₆	9.9	5.7	5.8	5.8	5	5.5	0	0	10
Ciclohexano	C ₆ H ₁₂	9.9	11.8	11.3	11.3	9.8	10.8	0	0	80
Benzeno	C ₆ H ₆	9.3	10.4	12.3	12.3	6.7	10.4	3	0	150
Naftaleno	C ₁₀ H ₈	8.2	16.6	25.7	18.7	9.8	18.1	5	0	300
2-3-butanediona	C ₄ H ₆ O ₂	9.3	8.2	9.8	9.1	6.1	8.3	2	0	77
IsoPentano	C ₅ H ₁₂	10.3	10	10.9	9.9	8.9	9.9	0	0	60
TetraDecano	C ₁₄ H ₃₀	9.9	26.6					0	0	260
Propano	C ₃ H ₈	11.1	6.3	6.8	6.1	5.7	6.2	0	0	10
Tetracloroetileno	C ₂ Cl ₄	9.3	12.3	15.1	13.9	7.8	12.3	1	0	57
1-2-trans-Dicloroetileno	C ₂ H ₂ Cl ₂	9.6	8.2	12.5	6.7	5.6	8.3	1	0	15
Dissulfeto de Carbono	CS ₂	10.1	8.7	13.9	5.2	5.2	8.1	2	0	75
Bromofórmio	CHBr ₃	10.5	11.3	13.2	13.2	8.6	11.7	0	0.9	130
Ciclopentanona	C ₅ H ₈ O	9.3	9	10.3	9.6	7.6	9.2	1	3.3	230
Acetato de Etila(Confórmero 1)	C ₄ H ₈ O ₂	10	8.7	10.8	8.9	7.1	8.9	1	2.1	160
1-Butanol	C ₄ H ₁₀ O	10	8.5	10.4	8.3	7.7	8.8	0	1.6	102
Clorofórmio	CHCl ₃	11.4	8.2	9.4	9.3	6.5	8.4	0	1.1	50
Butanal	C ₄ H ₈ O	9.8	8	9.6	8.4	6.7	8.2	1	2.8	142
2-metilpropionitrila	C ₄ H ₇ N	11.3	8	9.7	7.8	6.8	8.1	2	4.2	274
Dimetilcarbonato	C ₃ H ₆ O ₃	10.3	7.4	7.7	7.7	6.7	7.4	1	0.5	102
Formato de Etila	C ₃ H ₆ O ₂	10.6	6.9	8.7	6.9	5.5	7	1	2.2	103
Acetato de Metila	C ₃ H ₆ O ₂	10.2	6.8	8.4	7.1	5.5	7	1	2	122
2-Propanol	C ₃ H ₈ O	10.2	6.8	7.4	7	6.4	6.9	0	1.6	87
1-Propanol	C ₃ H ₈ O	10.2	6.7	7.9	6.6	6.2	6.9	0	1.7	70
Acetona	C ₃ H ₆ O	9.7	6.2	7.1	7	5.1	6.4	1	3.2	173
Propanal	C ₃ H ₆ O	10	6.2	7	6.9	5.1	6.4	1	2.8	118
Propionitrilo	C ₃ H ₅ N	11.8	6.2	8.1	5.6	5.2	6.3	2	4.2	245
1-Fluoropropano	C ₃ H ₇ F	11.3	6.1	6.9	6	5.6	6.2	0	2.1	35
2-2-difluoropropano	C ₃ H ₆ F ₂	11.4	5.9					0	2.4	25
1-2-Etanodiol	C ₂ H ₆ O ₂	10.2	5.6	6.6	5.6	5	5.7	0	2.5	70
Bromometano	CH ₃ Br	10.5	5.3	6.8	4.6	4.6	5.4	0	2	40
Formato de Metila	C ₂ H ₄ O ₂	10.8	5	7.2	4.8	3.9	5.3	1	2	65
Etanol	C ₂ H ₆ O	10.4	5	5.6	5	4.6	5.1	0	1.7	43

Tabela 4 – Continuação da tabela 3

Molécula	Fórmula	I_p	α	α_{xx}	α_{yy}	α_{zz}	$\bar{\alpha}$	N_π	μ	ϵ_b
Acetaldeído	C ₂ H ₄ O	10.2	4.5	5.6	4.7	3.6	4.6	1	3	88
Acetonitrilo	C ₂ H ₃ N	12.2	4.4	6.1	3.6	3.6	4.4	2	4.1	180
Clorometano	CH ₃ Cl	11.2	4.3	5.4	3.8	3.8	4.3	0	2.1	25
Metanol	CH ₄ O	10.8	3.1	3.5	3.1	3	3.2	0	1.7	20
Dimetoximetano	C ₃ H ₈ O ₂	9.7	7.4					0	0.2	78
2-Butanona	C ₄ H ₈ O	9.5	8					1	3	194
Formato de Propila	C ₄ H ₈ O ₂	10.5	8.7					1	2.4	126
1-Fluorobutano	C ₄ H ₉ F	10.2	7.9					0	2.2	70
1-Fluorohexano	C ₆ H ₁₃ F	9.8	11.6					0	2.2	80
1-Fluorononano	C ₉ H ₁₉ F	9.6	17.2					0	2.2	145
1-Clorohexano	C ₆ H ₁₃ Cl	10.3	13.7					0	2.5	175
Diclorometano	CH ₂ Cl ₂	11.3	6.4	8.3	5.7	5.1	6.4	0	2	32
cis-1-2-Dicloroetileno	C ₂ H ₂ Cl ₂	9.7	8	10.6	7.9	5.6	8	1	2.3	66
1-1-Dicloroetileno	C ₂ H ₂ Cl ₂	9.8	8.1	9.5	9.2	5.6	8.1	1	1.3	30
Tricloroetileno	C ₂ HCl ₃	9.5	10.2	13.6	10.3	6.7	10.2	1	1.1	50
1-Cloropropano	C ₃ H ₇ Cl	10.8	8.2	10.3	7.4	6.9	8.2	0	2.5	98
2-Cloropropano	C ₃ H ₇ Cl	10.8	8.2	9.5	8.1	7.1	8.2	0	2.6	113

Os dados coletados não necessitaram de pré-processamento, pois são valores experimentais recentes ou propriedades moleculares bem estabelecidas. No entanto, foi realizado o escalonamento das propriedades moleculares para uma mesma escala, conforme descrito no capítulo de aprendizagem de máquina. As técnicas de escalonamento foram tratadas como hiperparâmetros. Durante a otimização, foram testadas padronização, normalização e escalonamento robusto, escolhendo-se a técnica que apresentou melhor desempenho para cada algoritmo utilizado.

Embora existam cenários mais recomendados para a utilização de uma técnica de escalonamento em detrimento de outra, mesmo nesses casos, onde usualmente se recomenda uma técnica específica, pode ocorrer que a técnica “não recomendada” se sobressaia. Portanto, optou-se pelo critério de melhor desempenho para selecionar a técnica de escalonamento.

Decidiu-se incluir quase todas as moléculas com valores experimentais de PEBs disponíveis, excluindo apenas duas moléculas polares que apresentaram erro de previsão significativo nas primeiras etapas de modelagem. Essas etapas consistiram na utilização dos algoritmos com seus hiperparâmetros padrões e um processo de validação cruzada LOOCV (*leave one out cross validation*). Nesse processo, treina-se o modelo com todas as moléculas, exceto uma, e compara-se a previsão com o valor real da variável alvo para a molécula excluída. Esse procedimento é repetido até que todas as moléculas sejam testadas. Observou-se que as moléculas excluídas (1,1-dicloroetileno e 2,2-difluoropropano) apresentaram um erro de teste superior a 150%, possivelmente devido ao baixo valor de energia de ligação, 30 e 25 meV, respectivamente.

Foram definidos quatro conjuntos de propriedades moleculares para os modelos de aprendizado de máquina, cada um criado para um tipo de grupo de moléculas: polares, apolares e ambos. Por exemplo, as moléculas apolares, foram divididas em dois grupos: um contendo todas as moléculas apolares e outro somente com as que possuíam polarizabilidade anisotrópica. Para este último grupo, foram testados dois valores de polarizabilidade, isotrópica e anisotrópica. Dessa forma, obtivemos três grupos de dados para serem testados com quatro conjuntos de propriedades moleculares diferentes, como exemplificado na Figura 27.

A mesma lógica foi aplicada para as moléculas polares e para ambos os tipos simultaneamente. O conjunto de propriedades moleculares selecionado para esses dois últimos é exatamente o mesmo.

A propriedade $\bar{\alpha}$ é utilizada na situação em que lidamos com o conjunto de dados que contém as componentes α_{xx} , α_{yy} e α_{zz} (isto é, excluindo aquelas moléculas para as quais não foram encontradas polarizabilidade anisotrópica). Isso foi decidido porque nos permitiu comparar se as componentes individuais ou a média delas foram mais relevantes para melhorar a performance dos algoritmos. Se optássemos por utilizar α , estaríamos utilizando um valor que é próximo, mas não exatamente igual à média das três componentes.

Entendemos que, para as moléculas apolares, não fazia sentido testar como os modelos performariam utilizando apenas uma propriedade como variável independente, dado que o processo de coleta de importância das propriedades já nos forneceria essa informação em cada uma das possibilidades. Alternativamente, para as moléculas polares e polares + apolares simultaneamente, compreendemos que a polarizabilidade elétrica e o momento de dipolo deveriam sempre estar presentes em qualquer conjunto de propriedades moleculares, dada a sua importância bem estabelecida pela literatura.

A fase computacional focou no ajuste de hiperparâmetros por meio da otimização bayesiana e na validação cruzada aninhada com 10 divisões externas (k_{out}) e 5 divisões internas (k_{in}), visando estimar de forma robusta o desempenho dos modelos em dados não vistos pela base de treino. Os hiperparâmetros finais, utilizados para treinar os modelos na base completa, foram determinados a partir de uma validação cruzada simples com 5 divisões, reproduzindo a metodologia do *loop* interno da validação cruzada aninhada. Foram utilizados cinco modelos de aprendizado de máquina: máquinas de vetor de suporte para regressão (SVR), regressão polinomial com regularização *ridge*, XGBoost, perceptron de multicamadas e um modelo *ensemble*, que é a média dos resultados dos modelos anteriores. A métrica de otimização escolhida foi a MAPE. Para cada conjunto de propriedades moleculares, foram comparadas as performances médias dos cinco modelos.

Os hiperparâmetros e os espaços de busca utilizados para cada algoritmo estão detalhados na Tabela 5. A nomenclatura dos hiperparâmetros segue a especificidade de cada algoritmo conforme implementado computacionalmente pelas bibliotecas *Python*.

Hiperparâmetro Otimizado	Espaço de Busca	Tipo do Espaço
Máquina de Vetor de Suporte		
kernel	[rbf, sigmoid, poly]	Categórico
gamma	[scale, auto]	Categórico
degree	(1, 10)	Inteiro
coef0	(0, 10)	Real
C	(0.1, 5)	Real
epsilon	(0.01, 5)	Real
Regressão Polinomial		
alpha	(10^{-5} , 10^2)	Real - log uniforme
fit_intercept	[True, False]	Categórico
copy_X	[True, False]	Categórico
degree	(2, 10)	Inteiro
interaction_only	[True, False]	Categórico
include_bias	[True, False]	Categórico
Perceptron de Multicamadas		
activation	[tanh, relu, logistic]	Categórico
learning_rate_init	(10^{-6} , 10^{-3})	Real - log uniforme
n_hidden_layer	(1, 10)	Inteiro
n_neurons_per_layer	(1, 10)	Inteiro
beta_1	(0.5, 0.9)	Real
beta_2	(0.5, 0.9)	Real
epsilon	(10^{-7} , 10^{-5})	Real
XGBoost		
learning_rate	(10^{-5} , 0.2)	Real - log uniforme
n_estimators	(10^2 , 2×10^3)	Inteiro
max_depth	(1, 15)	Inteiro
min_child_weight	(1, 15)	Inteiro
gamma	(0, 6)	Real
subsample	(0.1, 1)	Real
colsample_bytree	(0.1, 1)	Real
reg_alpha	(0, 1)	Real
reg_lambda	(0, 1)	Real

Tabela 5 – Esta tabela apresenta a relação dos hiperparâmetros otimizados para cada algoritmo empregado na pesquisa. O “Espaço de Busca” refere-se aos valores que os hiperparâmetros podem assumir aleatoriamente na tentativa de minimizar o erro. Os intervalos de busca são representados por parênteses, indicando o limite inferior e superior da busca. Os hiperparâmetros que possuem valores categóricos são representados por colchetes. O “Tipo de Espaço” descreve como cada espaço de busca deve ser explorado. A descrição “Espaços Reais com Log Uniforme” indica que os valores, de 0 a 1, têm a mesma probabilidade de serem selecionados quanto os valores de 1 a 10, e assim sucessivamente para cada potência positiva ou negativa de 10.

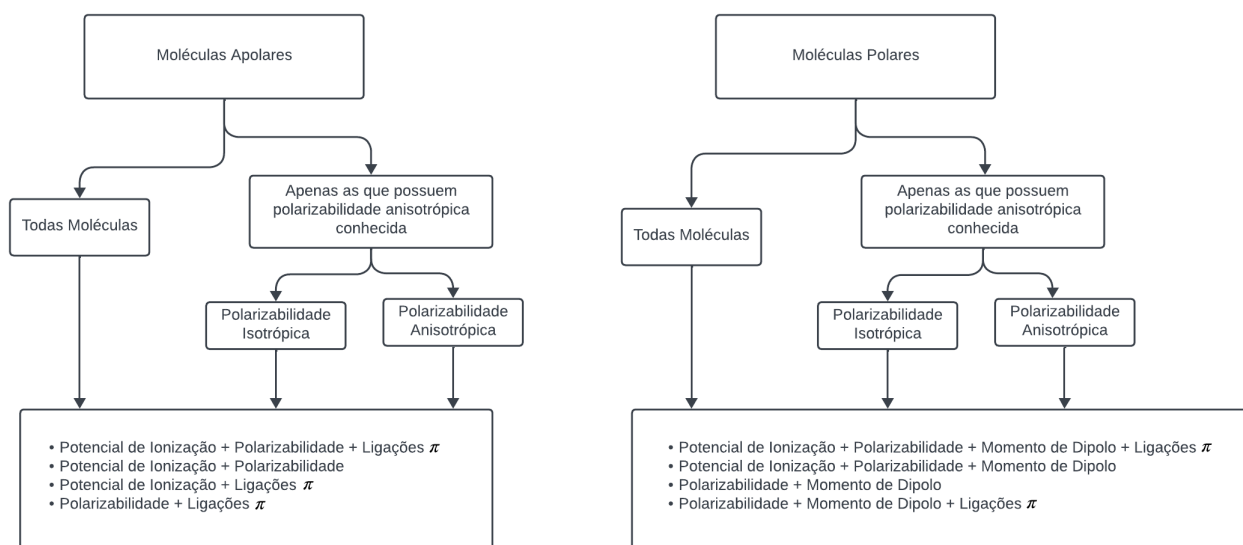


Figura 27 – Fluxograma esquemático representando os conjuntos de dados e conjuntos de propriedades moleculares utilizados.

Em geral, a escolha dos espaços de hiperparâmetros foi realizada da seguinte forma: para os hiperparâmetros categóricos, como o *gamma* para o SVR, foram incluídas todas as categorias possíveis no espaço de busca, pois geralmente não são mais do que três ou quatro opções, e cada uma pode impactar significativamente o desempenho. Os graus para regressão polinomial e máquina de vetor suporte tiveram seus limites superiores escolhidos de forma arbitrária, considerando que valores mais altos aumentam a probabilidade de *overfitting*. A quantidade de camadas ou neurônios por camada das redes neurais, profundidade máxima das árvores do XGBoost e outros hiperparâmetros pertencentes aos $\mathbb{I}$ seguem a mesma lógica. Alguns hiperparâmetros pertencentes a $\mathbb{R}$, como as taxas de aprendizagem do perceptron de multicamadas e do XGBoost, tiveram seus limites superior e inferior definidos com intervalos amplos para garantir que a busca contemplasse diferentes escalas de grandeza. Outros hiperparâmetros com limites superiores e inferiores pré-estabelecidos pela natureza do algoritmo tiveram seus espaços de busca determinados por esses limites (como *subsample* e *colsample_bytree* do XGBoost), visando garantir a identificação do valor ótimo ou, pelo menos, de um valor próximo.

É importante ressaltar que este foi um processo iterativo, onde a cada encontro do melhor conjunto de hiperparâmetros, caso um dos valores encontrados para algum hiperparâmetro específico fosse equivalente ao limite superior ou inferior do espaço de busca, este limite seria aumentado ou diminuído, respectivamente. O intuito é garantir que o encontro dos melhores valores para os hiperparâmetros não sejam limitados pelos valores de espaço de busca. Desta forma, foi garantido que todos os hiperparâmetros selecionados estão contidos no intervalo determinado pelos valores superiores e inferiores e não iguais a

estes.

O modelo *ensemble* foi construído a partir da média aritmética dos resultados dos modelos previamente otimizados. Por exemplo, suponha que os modelos A e B tenham feito previsões de 40 e 50 meV para a energia de ligação do pósitron com a molécula hexano; o modelo *ensemble* faria uma previsão de 45 meV.

Optou-se por não realizar a otimização do modelo *ensemble* após a combinação dos modelos individuais, considerando que tal procedimento acarretaria em um aumento substancial nos graus de liberdade e um elevado custo computacional. A fim de mitigar o aumento dos graus de liberdade, seria necessário restringir os hiperparâmetros otimizados, o que, por sua vez, poderia afetar adversamente a capacidade preditiva dos modelos.

Após a análise do desempenho dos grupos de propriedades moleculares em cada conjunto de moléculas, foi calculado a importância percentual de cada propriedade para os modelos de aprendizado de máquina, utilizando os valores SHAP (Seção 3.3).

A análise de importância SHAP foi aplicada aos algoritmos treinados com a base de dados completa, utilizando o conjunto de todas as propriedades moleculares. O *kernel* para o cálculo das aproximações dos valores *Shapley* foi selecionado automaticamente pelo algoritmo implementado por Lundberg e Lee [38], mantendo-se os demais parâmetros em seus valores padrão. O foco da pesquisa foi compreender a influência global e absoluta das propriedades moleculares para cada algoritmo, sem abordar a direção (positiva ou negativa) da correlação entre as propriedades e a energia de ligação do pósitron com as moléculas.

O objetivo desse cálculo foi avaliar a influência de cada propriedade na capacidade preditiva dos modelos. Em um algoritmo simples, como a regressão linear, uma propriedade de maior importância teria um coeficiente angular mais elevado. Contudo, como os modelos abordados neste estudo não são paramétricos e carecem de uma forma analítica explícita, adotou-se essa abordagem para quantificar a magnitude da influência de cada propriedade no fenômeno de ligação do pósitron com as moléculas.

Com base nos valores calculados para as importâncias das propriedades moleculares e nos conjuntos de propriedades que contribuíram para o melhor desempenho dos algoritmos, foram conduzidas discussões sobre as implicações físicas nos capítulos 5 e 6. Nesses capítulos, além de analisar os resultados obtidos, foram exploradas as relações entre as propriedades moleculares e os fenômenos observados, fornecendo pistas sobre os mecanismos subjacentes e as possíveis aplicações práticas dos modelos desenvolvidos. Ademais, foram delineadas as limitações do estudo e sugeridos possíveis caminhos para pesquisas futuras, visando aprimorar ainda mais a compreensão e aplicabilidade dos resultados apresentados.

Os programas foram todos desenvolvidos em Python 3.8 [43], utilizando algumas de suas principais bibliotecas para tratamento de dados, como Numpy e Scikit-learn [44–47].

Todo o código associado a este trabalho está hospedado no GitHub¹.

Por fim, os experimentos computacionais foram realizados em uma máquina com as seguintes especificações técnicas: Processador Intel® Xeon® CPU E5-2680 v4, operando a uma frequência de 2.40GHz e com um total de 28 núcleos; Placa de vídeo AMD® Radeon RX 570 Series; Memória RAM de 16Gb e Armazenamento SSD de 256Gb.

¹ Repositório no GitHub do Projeto: <https://github.com/Miguilo/Master-Degree-Research>

5 Resultados

Este capítulo apresenta os resultados obtidos a partir dos treinamentos dos algoritmos selecionados para cada conjunto de propriedades moleculares descritos no capítulo de metodologia, lembrando mais uma vez que a métrica que mediu o desempenho em conjuntos de teste para os algoritmos foi o erro percentual absoluto médio. Os resultados são divididos em três seções: Moléculas Apolares, Moléculas Polares e Moléculas Mistas (polares e apolares). Para cada grupo, discutiremos o desempenho desses algoritmos em relação a cada conjunto de propriedades moleculares, bem como a relevância de cada propriedade para os modelos em diferentes contextos. Além disso, conforme julgado necessário, poderemos mostrar outras métricas, cálculos ou outras informações sobre os conjuntos de dados e algoritmos utilizados.

Conforme detalhado no capítulo de metodologia, será apresentado os resultados para cada conjunto de moléculas em três cenários distintos. No primeiro, utiliza-se a polarizabilidade isotrópica para todas as moléculas. No segundo e terceiro cenários, para as moléculas que possuem polarizabilidades anisotrópicas, empregam-se respectivamente estas ou a polarizabilidade isotrópica média $\bar{\alpha}$.

Antes de prosseguir, é importante esclarecer que o modelo RNA representa o perceptron de multicamadas. Além disso, as barras de média ao extremo direito dos gráficos correspondem à média dos cinco modelos, seja para o erro médio de desempenho ou para a importância relativa de propriedades moleculares.

Na Tabela 6 são apresentados os símbolos empregados nos gráficos e seus respectivos significados, para todos os grupos de moléculas, apolares e polares.

5.1 Moléculas Apolares

Um dos primeiros resultados interessantes encontrados para moléculas apolares é a disparidade no desempenho dos algoritmos quando o conjunto de propriedades utilizadas

Tabela 6 – Símbolos utilizados nesse capítulo e seus respectivos significados.

Símbolo	Significado
I_p	Potencial ou Energia de Ionização
α	Polarizabilidade Isotrópica
$\alpha_{(xx,yy,zz)}$	Componentes da Polarizabilidade Anisotrópica
$\bar{\alpha}$	Polarizabilidade Isotrópica Média
π	Ligações π
μ	Momento de Dipolo Elétrico Permanente

são apenas as ligações π e polarizabilidade isotrópica, como mostrado na Figura 28. Essa disparidade pode ocorrer matematicamente pelo fato da propriedade energia de ionização apresentar baixíssima variância (cerca de 0.6 meV de desvio padrão), ou seja, o conjunto de moléculas apolares apresenta baixas mudanças no potencial de ionização de molécula para molécula e portanto, os algoritmos não aprendem muito com essa propriedade em específico.

Fisicamente isso é esperado, dado que o potencial de ionização e a polarizabilidade apresentam uma alta correlação, com um podendo ser aproximado como uma constante multiplicativa pelo inverso do outro. Portanto os algoritmos poderiam vir a enxergar essa propriedade como redundante.

Outra possibilidade para o desempenho superior dos algoritmos que levam em consideração as ligações π é que, apesar das moléculas apolares apresentarem uma nuvem eletrônica uniformemente distribuída, elétrons em orbitais π podem gerar uma maior tendência de ligação do pósitron com a molécula [24].

Através da Figura 29, é possível observar que, para o conjunto de moléculas apolares com polarizabilidades anisotrópicas encontradas (exclusão de Dodecano, Hexadecano e Tetradecano), os desempenhos médios diminuem, mas a hierarquia das propriedades moleculares ainda se mantém. Essa redução pode ser diretamente atribuída à exclusão de um alto percentual de moléculas (vale lembrar que temos um total de 20 moléculas apolares, e a remoção de 3 delas representa mais de 10% do total). Como resultado, os algoritmos não conseguem treinar com sua capacidade máxima. No entanto, ainda é possível observar as outras consequências mencionadas anteriormente, tanto por meio das correlações quanto pela possível influência na localização dos elétrons nos orbitais π .

Observa-se que, para a rede neural artificial, o desempenho das moléculas apolares melhora ao analisar o subconjunto das moléculas com polarizabilidade anisotrópica conhecida em comparação com o conjunto completo de moléculas apolares, considerando o conjunto de propriedades $I_p + \bar{\alpha} + \pi$ (compare-se a Figura 28 com a Figura 29). Este fenômeno pode ser atribuído à possibilidade de a média das polarizabilidades anisotrópicas ($\bar{\alpha}$) constituir um descritor mais eficaz para essas moléculas do que α , ao menos no contexto da rede neural artificial.

As RNAs são algoritmos reconhecidamente versáteis e poderosos, porém de difícil interpretabilidade. Considerando que este resultado diverge significativamente dos demais algoritmos, é plausível interpretá-lo como um possível artefato computacional. No entanto, esta hipótese requer investigação adicional para confirmação.

Ao comparar o grupo das 17 moléculas apolares, observa-se pouca diferença nos desempenhos quando se utiliza as componentes da polarizabilidade anisotrópica (como observado na Figura 30) e a polarizabilidade isotrópica. Na verdade, há até uma ligeira

piora, porém esta não é estatisticamente significativa. Isso se deve, novamente, à baixa quantidade de dados. Ao treinar com uma maior dimensionalidade, espera-se que os modelos tenham piora em seu desempenho, já que as componentes não trouxeram muita informação para os modelos. Dessa forma, a diferença não é expressiva quando os modelos treinam com as componentes da polarizabilidade anisotrópica ao invés da média das 3 componentes ($\bar{\alpha}$).

A análise da Figura 31, que representa os resultados da importância das propriedades através do SHAP, revela uma hierarquia de importância percebida por todos os modelos. Nessa ordem, as propriedades moleculares são classificadas como mais relevantes, com α ocupando a posição mais alta, seguido por ligações π e, por fim, I_p . Isto é totalmente coerente quando se pensa na combinação que produz os melhores resultados, $\alpha +$ ligações π , isto é, os modelos pouco aprendem com o potencial de ionização, que aparenta ser uma propriedade redundante, como já bem intuído na equação 1.5 por *Danielson et al.* em [17].

Ao analisar a Figura 32, porém para o conjunto parcial de moléculas, que incorpora as componentes do tensor de polarizabilidade elétrica, observamos que, individualmente, essas componentes apresentam uma relevância menor em comparação às ligações π . No entanto, é evidente que a soma da relevância das componentes ainda supera a relevância das ligações π . Vale destacar que a componente α_{zz} se destaca com maior relevância em relação às outras componentes.

É interessante notar que as componentes são definidas como os autovalores do tensor de polarizabilidade diagonalizado, de modo que $\alpha_{xx} > \alpha_{yy} > \alpha_{zz}$, indicando que a direção de menor polarizabilidade aparenta ser mais crucial para o cálculo da energia de ligação do que a direção de maior polarizabilidade.

5.2 Moléculas Polares

Ao contrário das moléculas apolares, há pouca diferença hierárquica observada ao analisar o conjunto completo de todas as moléculas polares em diferentes conjuntos de propriedades, como demonstrado na Figura 33.

Da mesma forma, ao compararmos com o conjunto parcial de moléculas polares, que inclui apenas aquelas com polarizabilidades anisotrópicas identificadas, observa-se uma piora singela nos desempenhos. Isso sugere que, neste caso, os algoritmos conseguiram lidar de maneira mais eficaz com a redução do conjunto de dados em comparação com o conjunto de moléculas apolares, que já possuía um conjunto bastante limitado de dados.

Ao comparar a Figura 33 com a Figura 34, nota-se uma piora considerável no desempenho do algoritmo SVR com a redução do conjunto de moléculas. Este comportamento pode ser análogo à diferença de desempenho observada na RNA para moléculas apolares

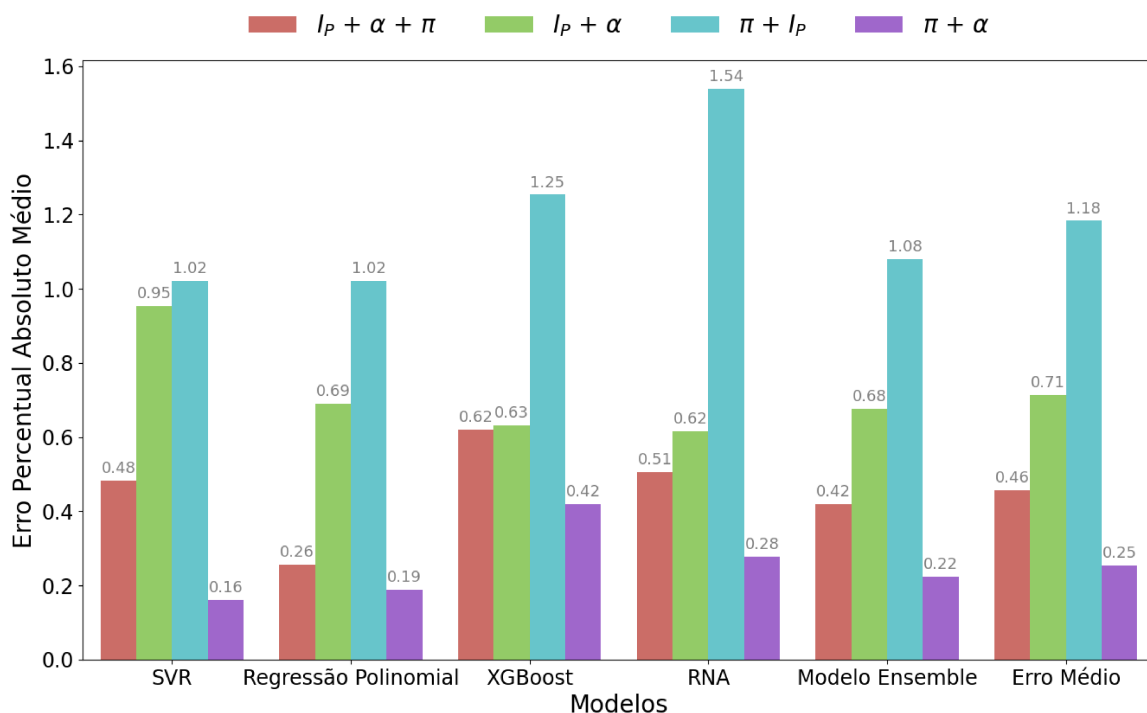


Figura 28 – **Moléculas Apolares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas apolares encontradas.

sob a mesma mudança de escopo de análise. Dado que este resultado é significativamente discrepante e divergente em relação aos outros algoritmos, é possível que seja decorrente de um artefato computacional específico para este algoritmo nestas condições particulares. Uma hipótese plausível é que, para o conjunto de propriedades $l_p + \bar{\alpha} + \mu$, a variação na quantidade de moléculas tenha exercido uma influência mais determinante do que nos outros conjuntos de propriedades.

Essas discrepâncias de desempenho observadas para a RNA e SVR foram um dos motivos que levaram à decisão de restringir as análises à média geral dos desempenhos observados. Entende-se que diferentes modelos que se aproximassem adequadamente da natureza intrínseca do problema e do sistema deveriam apresentar uma convergência na importância de atributos e nos conjuntos de propriedades moleculares com melhores desempenhos, independentemente das diferenças consideráveis na estrutura subjacente dos algoritmos.

Em um cenário com uma quantidade significativa de moléculas disponíveis para treinamento e onde α fosse equivalente a $\bar{\alpha}$, poderia se esperar que as diferenças no desempenho observado ao retirar algumas moléculas fossem insignificantes. As divergências observadas podem, portanto, ser um indicativo da limitação na quantidade de dados

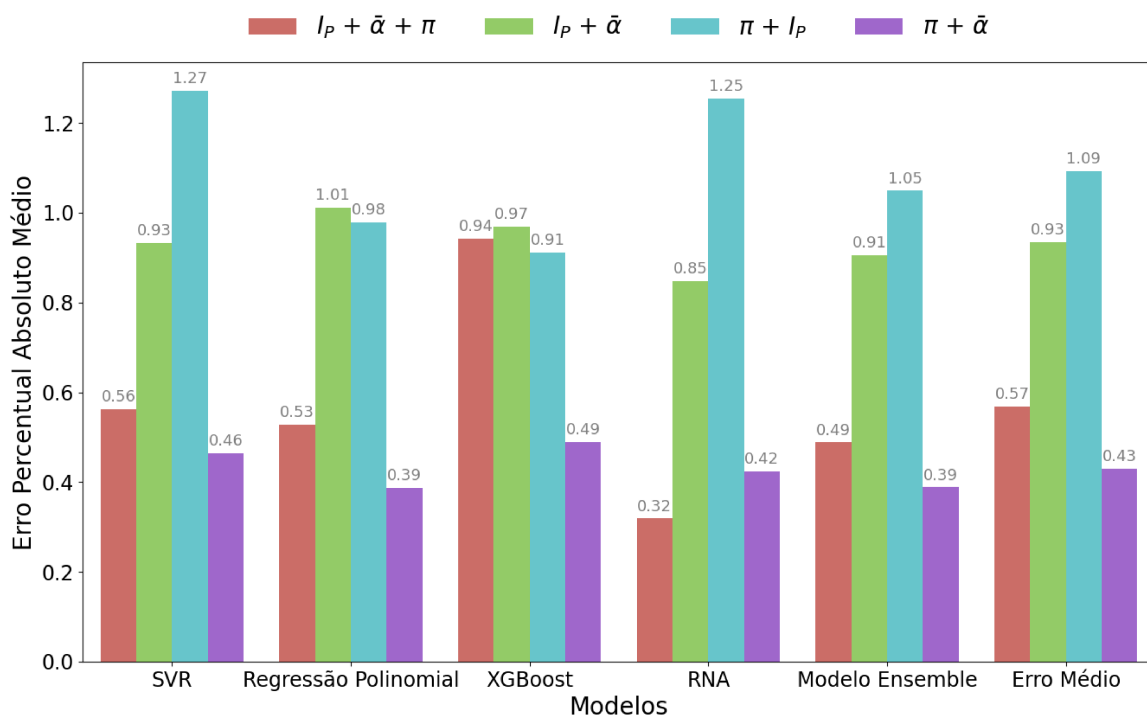


Figura 29 – **Moléculas Apolares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas apolares encontradas que possuem polarizabilidade anisotrópica.

disponíveis e na não correspondência da polarizabilidade isotrópica e média aritmética das polarizabilidades anisotrópicas.

Já para o conjunto completo, não há uma hierarquia clara ou significativa entre os conjuntos de propriedades moleculares no conjunto parcial de moléculas polares. Porém ao compararmos a utilização da média das polarizabilidades anisotrópicas ($\bar{\alpha}$), observa-se uma melhoria bem expressiva ao utilizarmos as componentes $\alpha_{(xx,yy,zz)}$ do tensor de polarizabilidade, evidenciado pela Figura 35.

Essa melhoria pode indicar que as componentes da polarizabilidade anisotrópica são relevantes ou fornecem informações geométricas importantes para o cálculo da energia de ligação dos pósitrons. Além disso, nota-se que o conjunto de propriedades que inclui todas as características apresenta um desempenho superior ao conjunto sem o potencial de ionização (I_P), o que sugere, agora, que esse potencial pode sim ter relevância nos cálculos das energias de ligação.

Nos gráficos de importância de propriedades via *SHAP* na Figura 36, observa-se a seguinte hierarquia: $\mu > \alpha > \text{ligações } \pi > I_P$. Repara-se que é a mesma ordem para as moléculas apolares, com o diferencial apenas de exclusão do momento de dipolo elétrico no segundo caso. Isso é de se esperar, tanto matematicamente quanto fisicamente, onde o

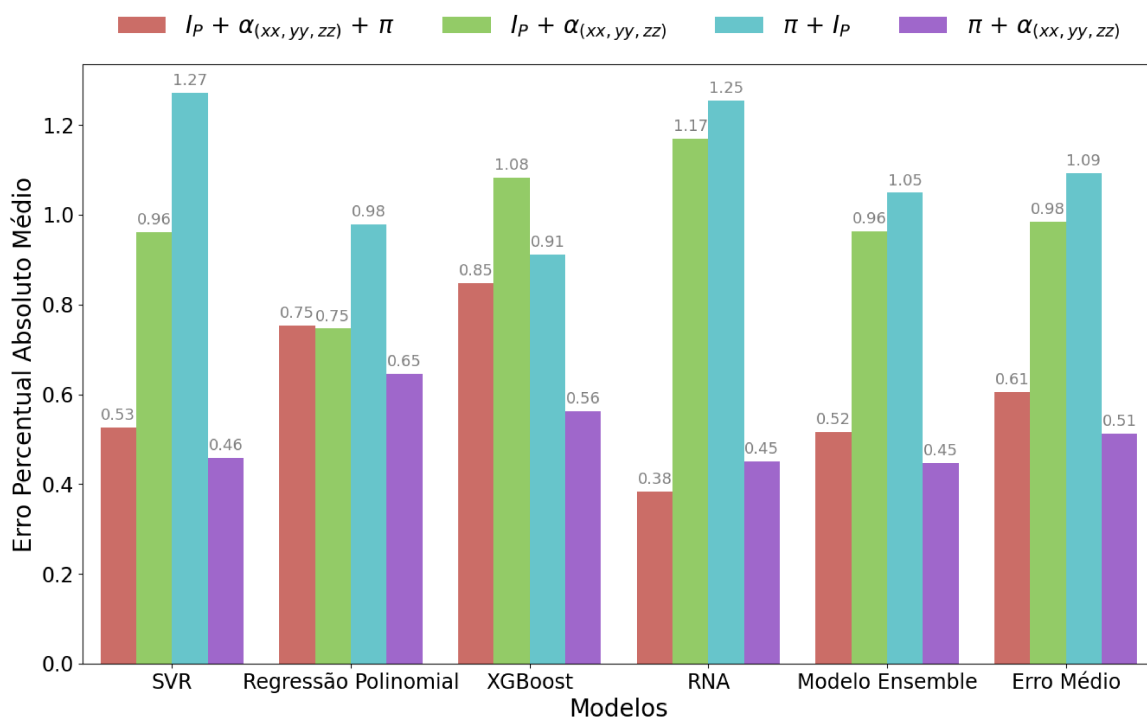


Figura 30 – **Moléculas Apolares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas apolares encontradas que possuem polarizabilidade anisotrópica, substituindo $\bar{\alpha}$ por $\alpha_{(xx,yy,zz)}$.

dipolo já é utilizado como determinante se haverá ligação e para ajudar no cálculo das energias por diversos autores. Matematicamente espera-se que isso ocorra pois o dipolo é a variável com maior correlação (0.68) com a variável alvo e uma alta variância também (0.9).

Além disso, observa-se uma contribuição em percentual ligeiramente maior do potencial de ionização para as moléculas polares do que para as moléculas apolares, apontando mais uma vez para a relevância do potencial de ionização para as moléculas polares. No caso da utilização das componentes do tensor de polarização temos ainda a mesma hierarquia de relevância, dado que o momento de dipolo rouba o protagonismo mais uma vez e a ordem das componentes de polarizabilidade é $\alpha_{zz} > \alpha_{yy} > \alpha_{xx}$, como mostrado na Figura 37, onde novamente a ordem se dá de forma decrescente dos valores de cada componente individualmente.

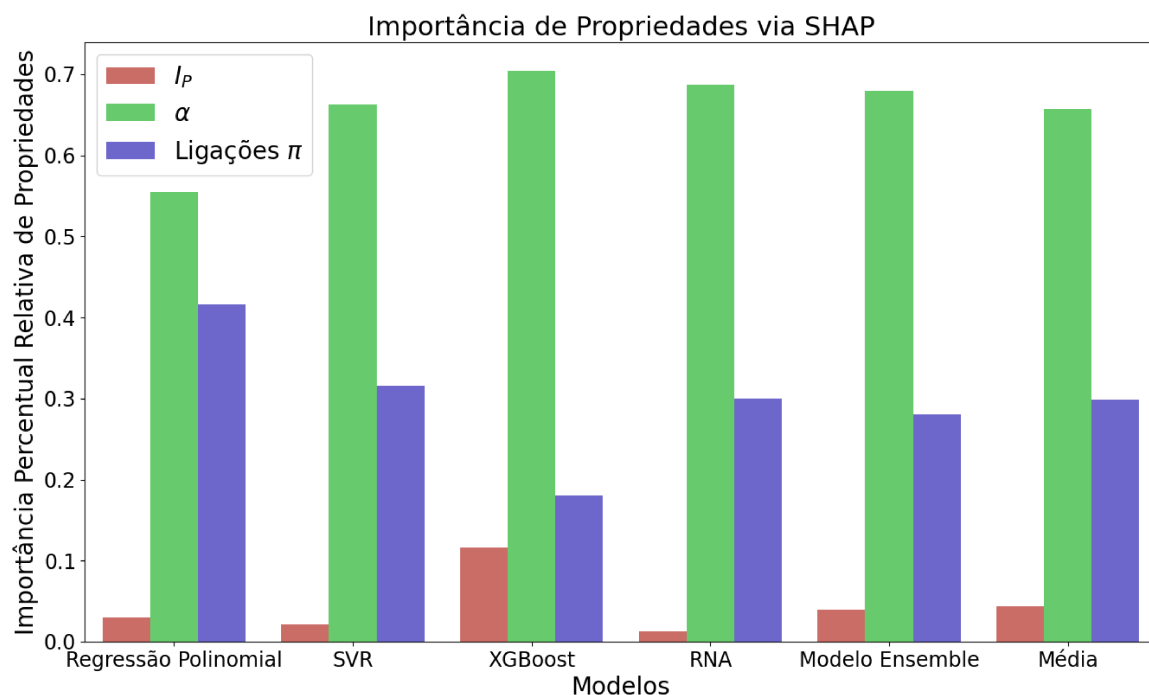


Figura 31 – **Moléculas Apolares**: Relação da importância relativa média das propriedades moleculares individualmente para cada modelo.

5.3 Moléculas Mistas

A tentativa de juntar os grupos de moléculas polares e apolares surge com a ideia de tentar criar modelos que não dependam da segregação entre um ou outro grupo, gerando dessa forma modelos mais generalistas.

Para o grupo de moléculas polares e apolares simultaneamente, observa-se um comportamento misto proveniente dos dois grupos individualmente. Por exemplo, nas Figuras 38 e 39, nota-se uma baixa clareza de hierarquia entre os grupos de propriedades que melhor performam, tal qual no grupo de moléculas polares.

Ao considerar as componentes da polarizabilidade anisotrópica, percebe-se uma acentuação nas hierarquias de desempenho devido às combinações de propriedades. Isso é evidente no caso das moléculas polares, conforme a melhoria aparente ilustrada na Figura 35 para moléculas polares e 40 para os dois grupos juntos.

Além disso, a utilização de todas as propriedades moleculares resulta no melhor desempenho, indicando que a inclusão de ligações π e do potencial de ionização, I_P , aprimora os modelos de aprendizado de máquina. Isso pode acabar significando mais uma vez, que o potencial de ionização pode ser relevante na determinação das energias de ligação com o pósitron, assim como ocorreu com o grupo de moléculas exclusivamente polares.

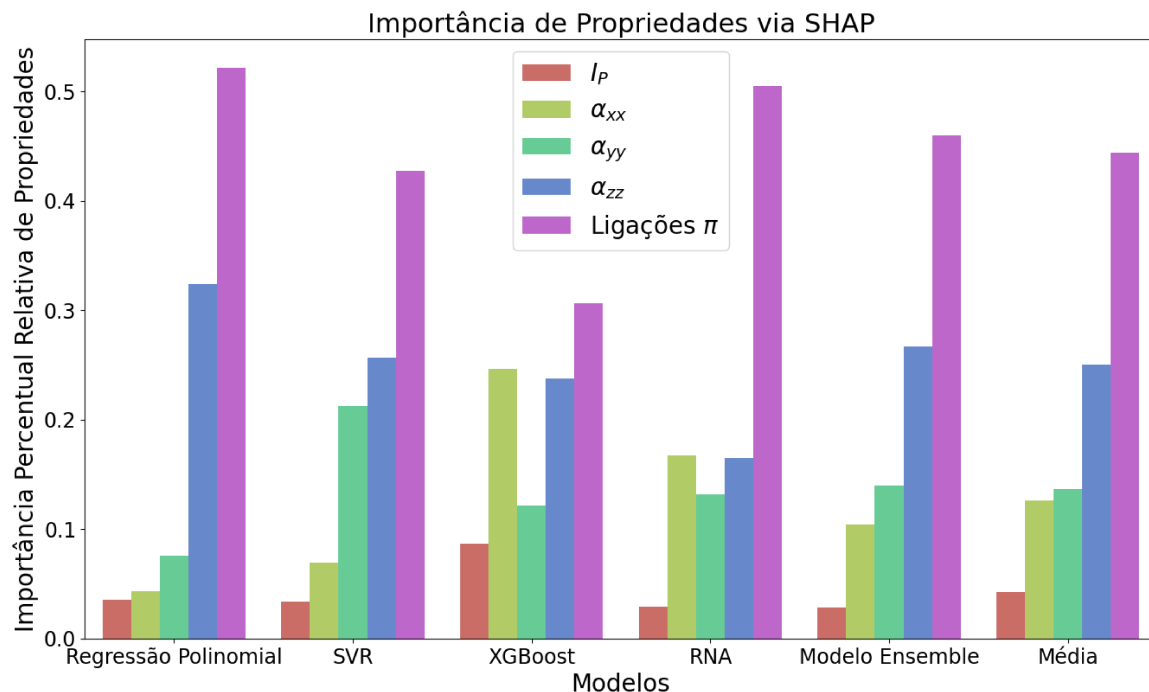


Figura 32 – **Moléculas Apolares**: Relação da importância relativa média das propriedades moleculares individualmente para cada modelo com o conjunto de moléculas parciais, incluindo dessa vez as polarizabilidades anisotrópicas.

Claro, esses efeitos de melhora de desempenho por ambas propriedades moleculares podem acabar sendo mitigados com a inclusão de uma maior volumetria de dados, convergindo para uma importância menor, ou até mesmo nula destas propriedades.

Inicialmente, esperava-se que as hierarquias de importância das propriedades se assemelhassem às dos conjuntos de moléculas polares, dado que estas predominam na base de treinamento. Da mesma forma, esperava-se que os desempenhos por grupos de propriedades moleculares também se aproximassem dos desempenhos das moléculas polares. Quando observa-se a importância de propriedades individuais via SHAP, têm-se que o a polarizabilidade elétrica (α) assume a liderança, tendo portanto uma hierarquia com a seguinte estrutura: $\alpha > \mu > \text{ligações } \pi > I_p$. Isso ocorre por que existem cerca de um terço dos dados com momento de dipolo nulo, ou seja, há a mitigação da relevância desse parâmetro. Porém a hierarquia segue muito semelhante ao caso das moléculas polares.

Porém, ao observar as relevâncias das propriedades moleculares no escopo contendo as polarizabilidades anisotrópicas, o momento de dipolo mais uma vez assume a liderança, possivelmente pela distribuição que ocorre ao dividir a polarizabilidade isotrópica (valor médio) em três outras variáveis informacionais.

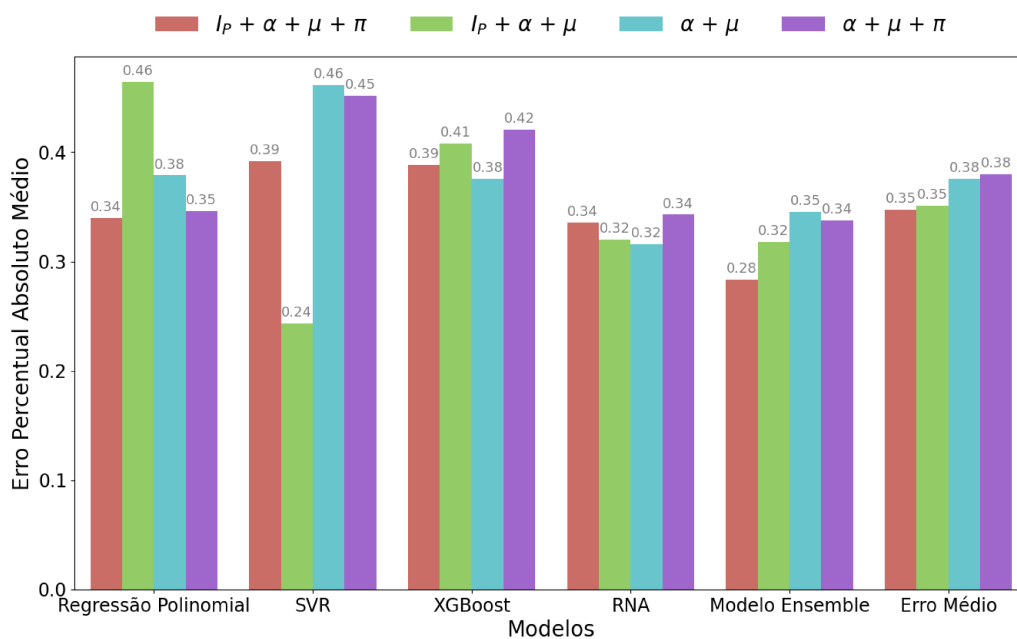


Figura 33 – **Moléculas Polares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares encontradas.

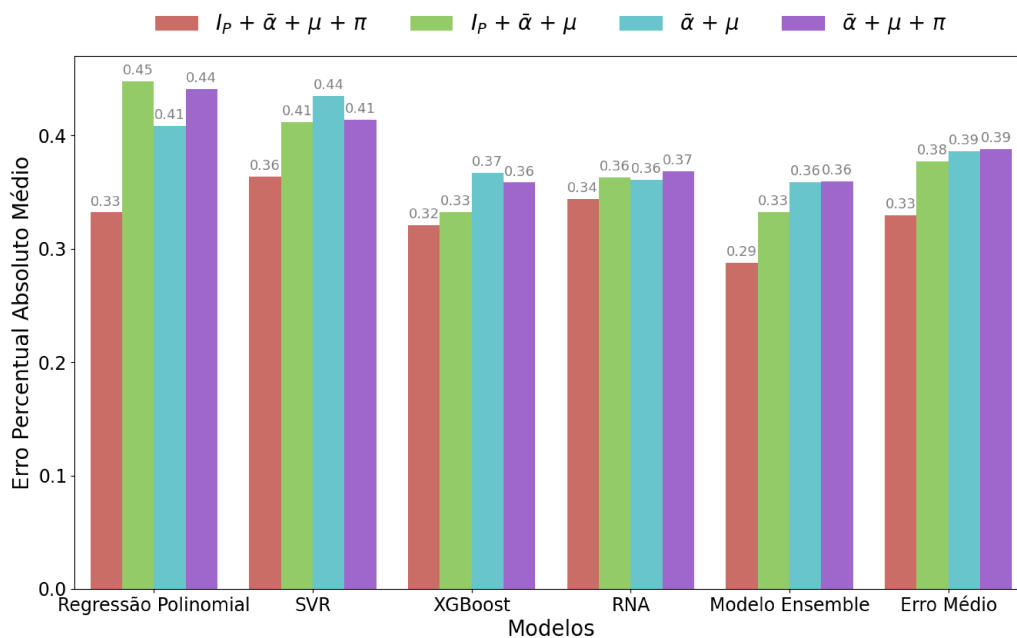


Figura 34 – **Moléculas Polares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares encontradas que possuem polarizabilidade anisotrópica.

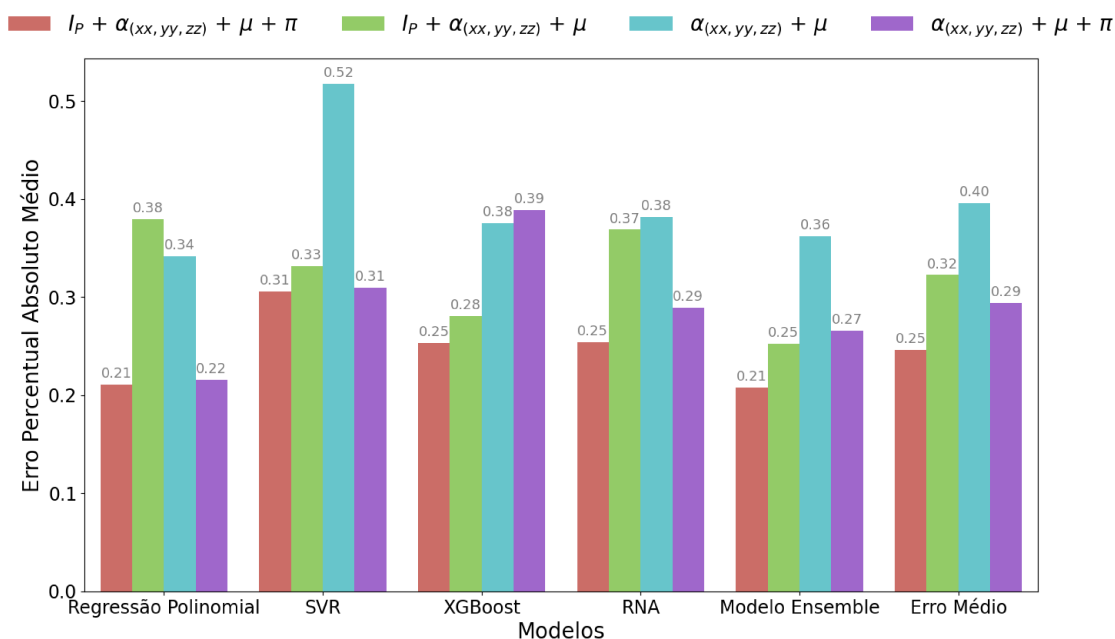


Figura 35 – **Moléculas Polares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares encontradas que possuem polarizabilidade anisotrópica, substituindo $\bar{\alpha}$ por $\alpha_{(xx,yy,zz)}$.

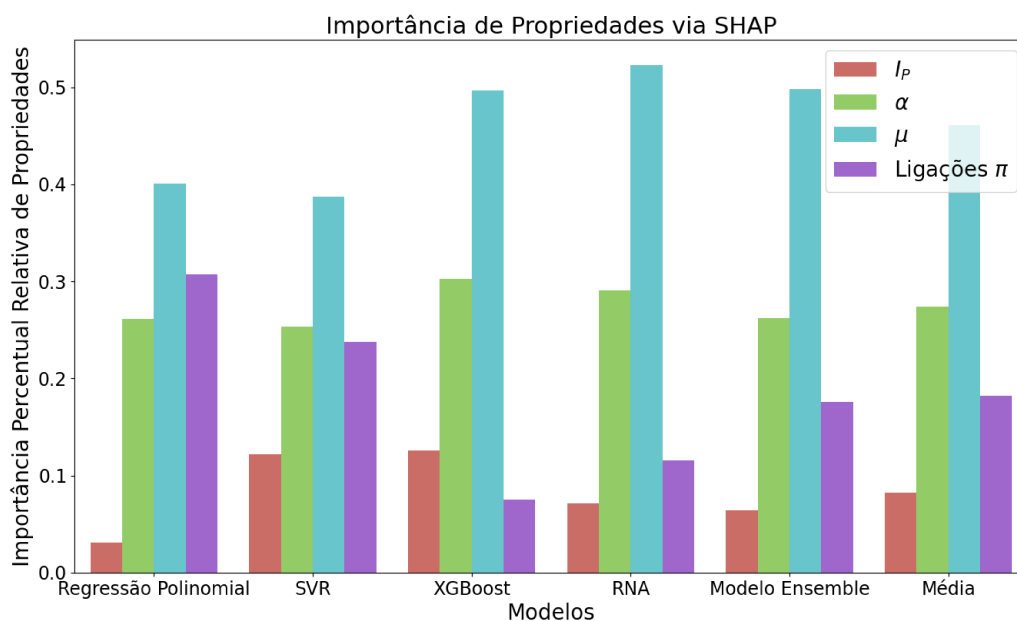


Figura 36 – **Moléculas Polares**: Relação da importância relativa média das propriedades moleculares individualmente para cada modelo.

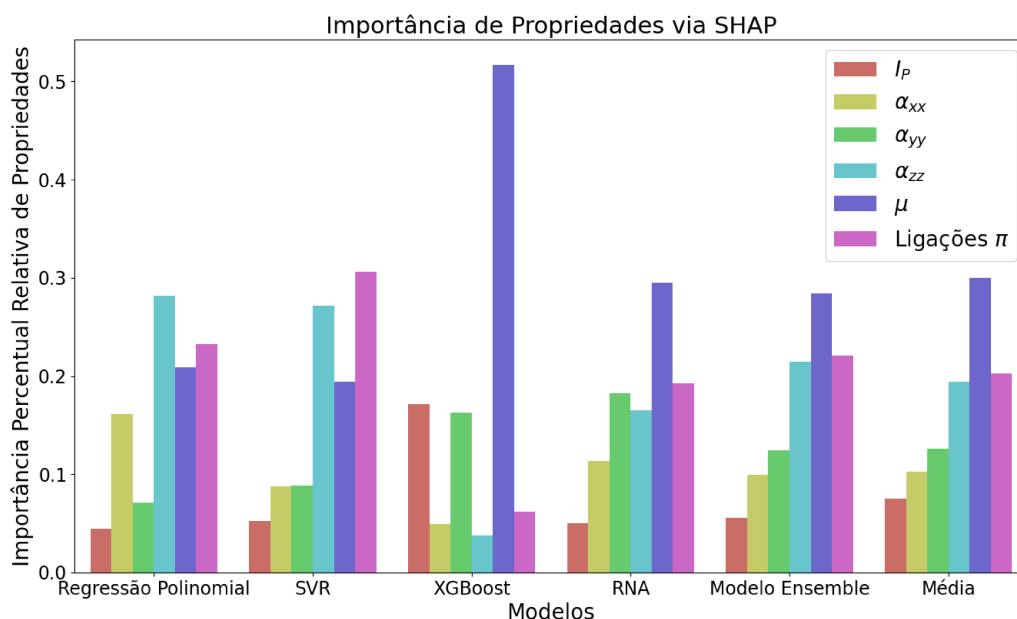


Figura 37 – **Moléculas Polares**: Relação da importância relativa média das propriedades moleculares individualmente para cada modelo com o conjunto de moléculas parciais, incluindo dessa vez as polarizabilidades anisotrópicas.

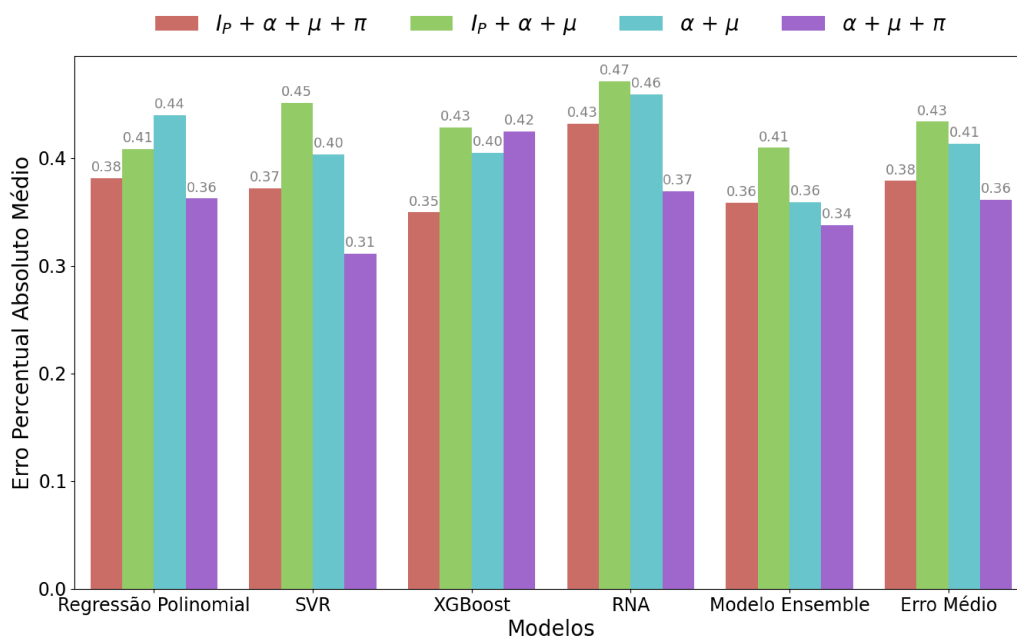


Figura 38 – **Moléculas Polares e Apolares**: Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares e apolares encontradas.

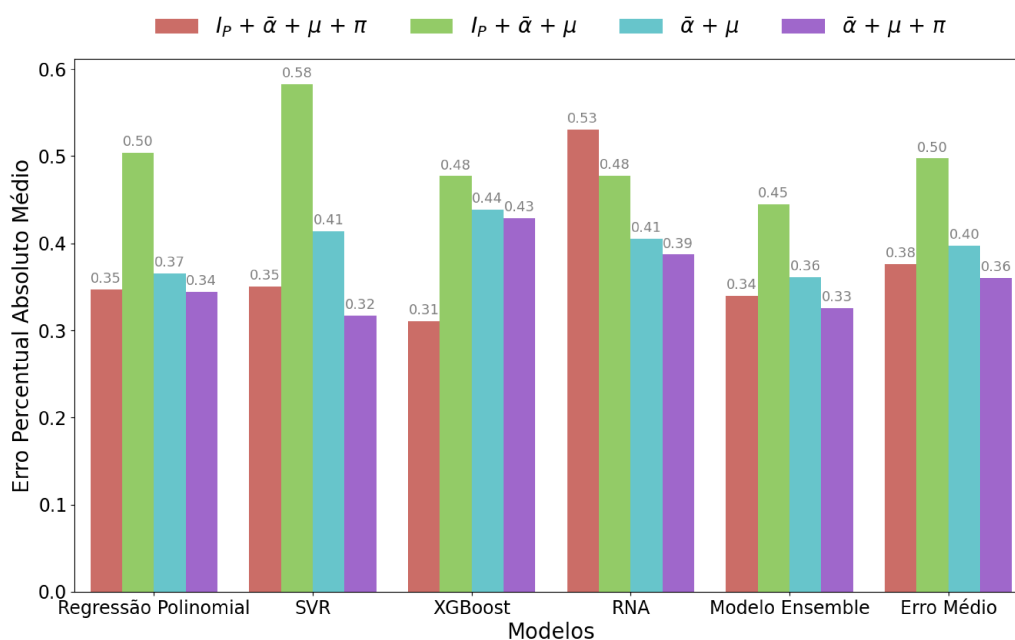


Figura 39 – **Moléculas Polares e Apolares:** Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares e apolares encontradas que possuem polarizabilidade anisotrópica.

5.4 Comparação dos resultados com literatura

Esta seção apresenta comparações entre os desempenhos observados neste trabalho e o modelo empírico de Danielson *et al* [17].

Modelo	MAE	RMSE	MAPE
Danielson ¹	11.55	17.68	0.3
Danielson ²	10.9	18.28	0.33
SVR	8.5	14.87	0.17

Tabela 7 – Comparação de desempenho para moléculas apolares

Modelo	MAE	RMSE	MAPE
Danielson ¹	48.79	65.17	0.42
Danielson ²	34.41	43.85	0.34
Ensemble	16.62	22.69	0.2

Tabela 8 – Comparação de desempenho para moléculas polares

As Tabelas 7, 8 e 9 comparam a equação 1.5 com o modelo de melhor desempenho, identificado pelo conjunto de propriedades moleculares e algoritmo que apresentaram os melhores resultados. O critério de avaliação adotado foi o MAPE, conforme apresentado

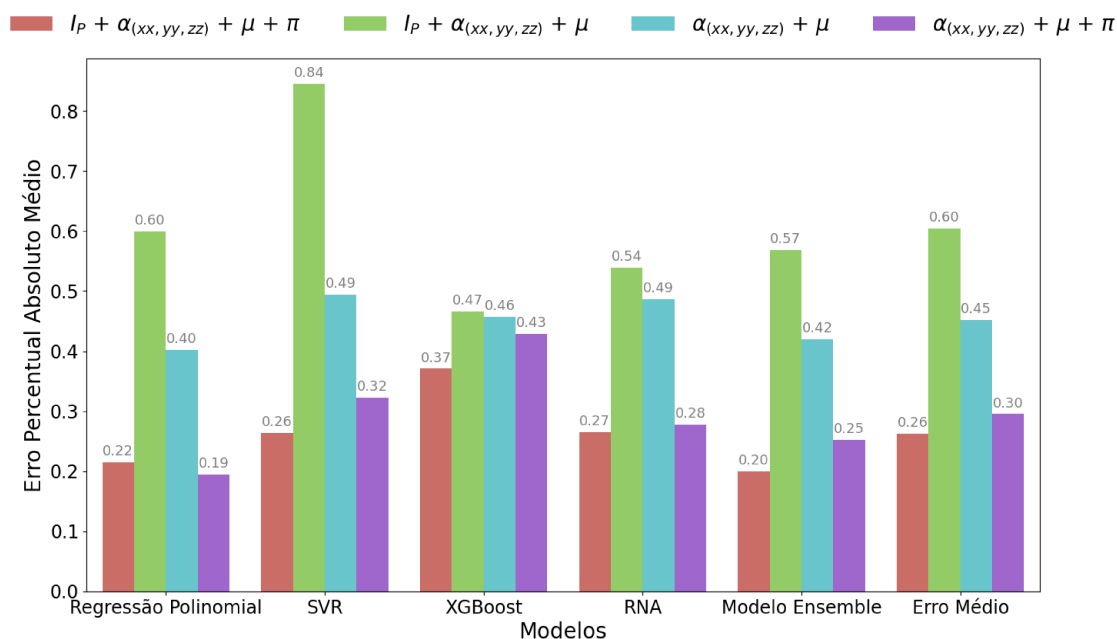


Figura 40 – **Moléculas Polares e Apolares:** Relação do Erro Percentual Absoluto Médio por algoritmo e combinação de propriedades moleculares para todas as moléculas polares e apolares encontradas que possuem polarizabilidade anisotrópica, substituindo $\bar{\alpha}$ por $\alpha_{(xx,yy,zz)}$.

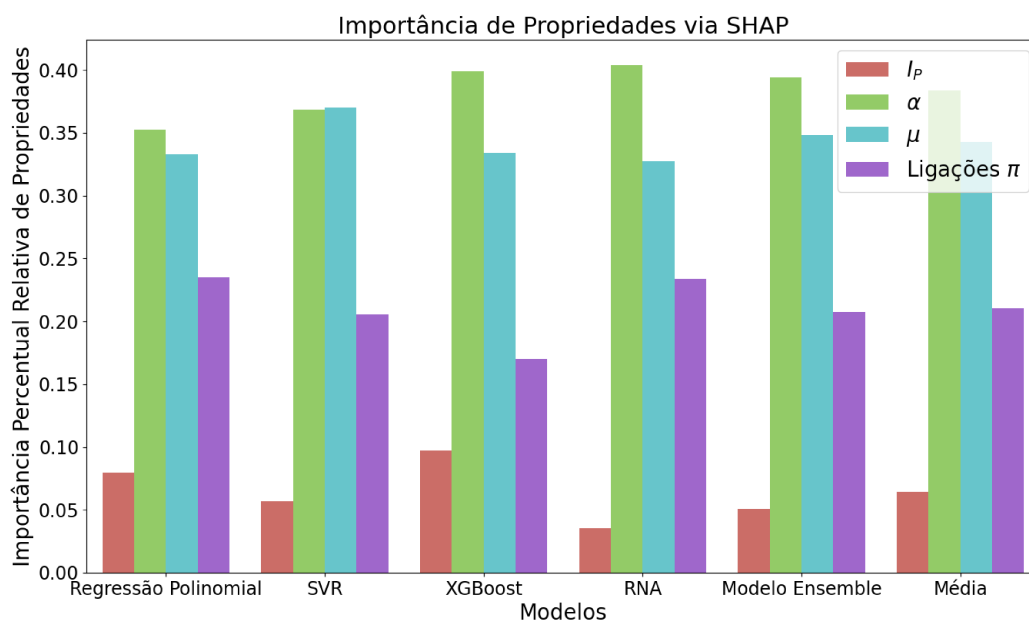


Figura 41 – **Moléculas Polares e Apolares:** Relação da importância relativa média das propriedades moleculares individualmente para cada modelo.

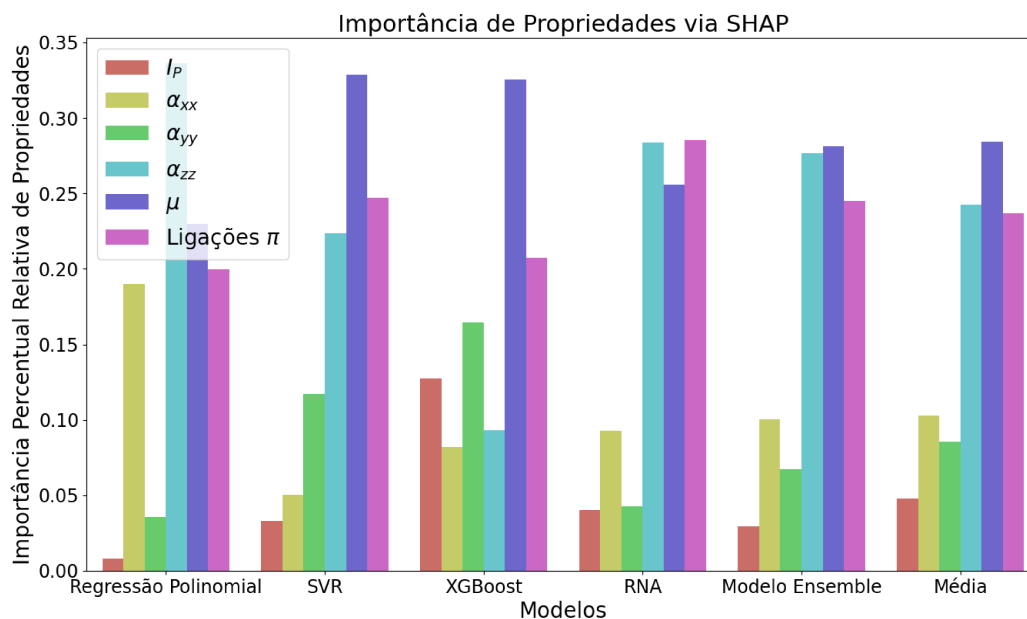


Figura 42 – **Moléculas Polares e Apolares:** Relação da importância relativa média das propriedades moleculares individualmente para cada modelo com o conjunto de moléculas parciais, incluindo dessa vez as polarizabilidades anisotrópicas.

Modelo	MAE	RMSE	MAPE
Danielson ¹	35.59	53.04	0.4
Danielson ²	26.24	36.83	0.36
Ensemble	18.41	27.83	0.2

Tabela 9 – Comparação de desempenho para moléculas mistas

na seção anterior. Esta abordagem comparativa foi escolhida considerando que, para novas previsões, seria utilizado o modelo mais eficaz com o conjunto de propriedades de melhor desempenho. Além disso, as tabelas apresentam outras duas métricas de avaliação comumente utilizadas para problemas de regressão: o erro médio absoluto (MAE) e a raiz do erro quadrático médio (RMSE). Para todas as métricas, valores menores indicam melhor desempenho.

Em casos de empate, priorizou-se o algoritmo *ensemble* (quando este estava entre os empatados), por ser considerado mais robusto e menos sensível a variações nos conjuntos de dados de treinamento. Na ausência do *ensemble* entre os empatados, optou-se pelo algoritmo mais simples em termos de arquitetura.

As tabelas apresentam erros para os modelos denominados Danielson¹ e Danielson². No modelo Danielson¹, a equação 1.5 utiliza as ligações π apenas para moléculas aromáticas, enquanto no modelo Danielson², as ligações π são aplicadas a todas as moléculas.

Esta comparação foi realizada porque Danielson *et al.* na formulação original da equação 1.5, utilizaram as ligações π apenas para moléculas aromáticas. Este trabalho propõe que, para o conjunto de moléculas polares e mistas, a utilização das ligações π resulta em melhores métricas de regressão, conforme observado nas Tabelas 8 e 9. É importante notar que as moléculas estudadas no presente trabalho não possuíam observações experimentais sobre suas energias de ligação com o pósitron no período de proposição da equação 1.5 [17].

Observa-se também um ganho de desempenho considerável em todas as métricas de avaliação nas Tabelas 7, 8 e 9. Apesar do único modelo empírico disponível ser uma regressão linear simples, isso demonstra como a utilização de algoritmos de aprendizado de máquina pode auxiliar a alcançar novos níveis de desempenho.

6 Conclusões

O presente trabalho, teve principalmente como intenção, intuir possibilidades ou novos direcionamentos para a obtenção de melhores modelos físicos que busquem explicar as relações entre o fenômeno de ligação dos pósitrons e propriedades moleculares para moléculas polares e apolares.

A utilização de algoritmos de aprendizado de máquina para explorar e compreender fenômenos físicos oferece benefícios significativos em termos de custo computacional e temporal, especialmente quando comparado a simulações mais tradicionais.

Neste estudo, empregamos algoritmos amplamente reconhecidos na literatura, como redes neurais, árvores de decisão e máquinas de vetores de suporte, para investigar a importância de propriedades moleculares específicas na ligação de pósitrons a moléculas polares e apolares. Essas abordagens iniciais possibilitam a rápida e econômica identificação de pistas relevantes para fenômenos e problemas físicos.

É crucial destacar que a implementação dessas técnicas apresenta desafios e oportunidades de aprimoramento e adaptação que variam conforme o contexto. Podem-se desenvolver modelos e algoritmos de complexidade variável, visando obter resultados mais precisos ou significativos, de acordo com a natureza do problema em questão.

Os resultados apresentados (capítulo 5) reforçam a importância das ligações π em moléculas apolares e ressaltam o papel significativo da polarizabilidade anisotrópica em moléculas polares. A análise revela que as componentes do tensor de polarizabilidade elétrica podem servir como indicadores relevantes de características físicas, como por exemplo a distribuição espacial das nuvens eletrônicas, quando sujeito a campos elétricos, afetando diretamente o cálculo das energias de ligação dos pósitrons com moléculas polares.

Esses resultados indicam que cálculos semi-empíricos ou empíricos para moléculas devem considerar as polarizabilidades anisotrópicas como descritores essenciais em etapas subsequentes. Isso pode ser observado pela melhora considerável na performance dos algoritmos ao incluir a anisotropia como variável explicativa.

É válido enfatizar como o momento de dipolo também foi indispensável para os algoritmos de aprendizado máquina, corroborando com a intuição e teoria desenvolvida na área até então. Embora as ligações π não tenham demonstrado um impacto significativo em moléculas polares neste estudo, investigações futuras não devem ignorar a possibilidade de sua relevância.

Há possibilidade de que, ao aumentar a quantidade de moléculas apolares no estudo, a importância das ligações π para essas moléculas pode se equiparar à relevância

encontrada nas moléculas polares. Isso implica que a importância atribuída às ligações π pode ser um artefato causado pela escassez de dados disponíveis para moléculas apolares.

Os modelos aqui utilizados obtiveram uma boa performance (aproximadamente de 20% de erro para os melhores modelos nas melhores condições), indicando que mesmo com uma baixa disponibilidade de dados, técnicas de aprendizado de máquina com o intuito de reduzir fenômenos como *overfitting* podem ser valiosas.

Os próximos passos que se enxergam com certeza devem envolver a aquisição de uma maior base de dados, entendendo que frequentemente o sucesso de modelos de aprendizado de máquina estão associados a dados de alta qualidade. Além disso pode-se sugerir a utilização de modelos de regressão que são especializados em baixa quantidades de dados, ou até mesmo pode-se pensar na exploração de técnicas e espaços de busca de otimização distintos para os modelos utilizados.

Como a interpretabilidade dos modelos para a física é de extrema importância, outras técnicas como regressões simbólicas [48–50] podem e devem também ser exploradas, com o intuito de buscar relações analíticas que possam ter um encaixe apropriado e de fácil entendimento nos conjuntos de dados de análise. Ou ainda, outras técnicas para além do *SHAP* para a interpretabilidade de modelos caixa preta, como os utilizados nesse trabalho, também podem ser exploradas.

Referências

- [1] P. A. M. Dirac, “On the Annihilation of Electrons and Protons,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 26, pp. 361–375, July 1930. Citado 2 vezes nas páginas 15 e 18.
- [2] C. D. Anderson, “The Positive Electron,” *Physical Review*, vol. 43, pp. 491–494, Mar. 1933. Citado na página 15.
- [3] Y. C. Jean, P. E. Mallon, and D. M. Schrader, eds., *Principles and applications of positron & positronium chemistry*. Singapore ; Hong Kong: World Scientific, 2003. Citado na página 16.
- [4] V. A. Dzuba, V. V. Flambaum, G. F. Gribakin, and W. A. King, “Bound states of positrons and neutral atoms,” *Physical Review A*, vol. 52, pp. 4541–4546, Dec. 1995. Publisher: American Physical Society. Citado 2 vezes nas páginas 17 e 26.
- [5] J. R. Danielson, J. J. Gosselin, and C. M. Surko, “Dipole Enhancement of Positron Binding to Molecules,” *Physical Review Letters*, vol. 104, p. 233201, June 2010. Citado 3 vezes nas páginas 17, 21 e 68.
- [6] J. P. Cassidy, J. Hofierka, B. Cunningham, C. M. Rawlins, C. H. Patterson, and D. G. Green, “Many-body Theory Calculations of Positron Binding to Halogenated Hydrocarbons,” *Physical Review A*, vol. 109, p. L040801, Apr. 2024. Citado na página 17.
- [7] J. A. Young, G. F. Gribakin, C. M. R. Lee, and C. M. Surko, “Role of combination vibrations in resonant positron annihilation,” *Physical Review A*, vol. 77, p. 060702, June 2008. Citado na página 17.
- [8] J. A. Young and C. M. Surko, “Positron attachment to molecules,” *physica status solidi c*, vol. 6, pp. 2265–2271, Nov. 2009. Citado na página 17.
- [9] J. A. Young and C. M. Surko, “Feshbach-resonance-mediated annihilation in positron interactions with large molecules,” *Physical Review A*, vol. 77, p. 052704, May 2008. Citado 2 vezes nas páginas 17 e 68.
- [10] J. A. Young and C. M. Surko, “Feshbach-resonance-mediated positron annihilation in small molecules,” *Physical Review A*, vol. 78, p. 032702, Sept. 2008. Citado 2 vezes nas páginas 17 e 68.
- [11] G. F. Gribakin, “Mechanisms of positron annihilation on molecules,” *Physical Review A*, vol. 61, p. 022720, Jan. 2000. Citado na página 17.

- [12] P. Fraser, “Positrons and Positronium in Gases,” in *Advances in Atomic and Molecular Physics*, vol. 4, pp. 63–107, Elsevier, 1968. Citado na página 18.
- [13] G. F. Gribakin and C. M. Lee, “Positron annihilation in large polyatomic molecules. The role of vibrational Feshbach resonances and binding,” *The European Physical Journal D*, vol. 51, pp. 51–61, Jan. 2009. Citado na página 19.
- [14] S. J. Gilbert, L. D. Barnes, J. P. Sullivan, and C. M. Surko, “Vibrational-Resonance Enhancement of Positron Annihilation in Molecules,” *Physical Review Letters*, vol. 88, p. 043201, Jan. 2002. Citado na página 19.
- [15] L. D. Barnes, S. J. Gilbert, and C. M. Surko, “Energy-resolved positron annihilation for molecules,” *Physical Review A*, vol. 67, p. 032706, Mar. 2003. Citado na página 19.
- [16] G. F. Gribakin, J. A. Young, and C. M. Surko, “Positron-molecule interactions: Resonant attachment, annihilation, and bound states,” *Reviews of Modern Physics*, vol. 82, pp. 2557–2607, Sept. 2010. Citado na página 20.
- [17] J. R. Danielson, J. A. Young, and C. M. Surko, “Dependence of positron–molecule binding energies on molecular properties,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 42, p. 235203, Dec. 2009. Citado 8 vezes nas páginas 18, 19, 23, 24, 29, 77, 86 e 89.
- [18] P. H. R. Amaral and J. R. Mohallem, “Positron binding to atoms and apolar molecules: A convergence of theory and experiment,” *Physical Review A*, vol. 86, p. 042708, Oct. 2012. Citado 3 vezes nas páginas 21, 26 e 29.
- [19] J. Mitroy, M. W. J. Bromley, and G. G. Ryzhikh, “Positron and positronium binding to atoms,” *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 35, pp. R81–R116, July 2002. Citado 2 vezes nas páginas 23 e 29.
- [20] A. R. Swann and G. F. Gribakin, “Calculations of positron binding and annihilation in polyatomic molecules,” *The Journal of Chemical Physics*, vol. 149, p. 244305, Dec. 2018. Citado na página 25.
- [21] A. Swann and G. Gribakin, “Positron Binding and Annihilation in Alkane Molecules,” *Physical Review Letters*, vol. 123, p. 113402, Sept. 2019. Citado na página 25.
- [22] A. R. Swann and G. F. Gribakin, “Model-potential calculations of positron binding, scattering, and annihilation for atoms and small molecules using a Gaussian basis,” *Physical Review A*, vol. 101, p. 022702, Feb. 2020. Citado na página 25.
- [23] A. R. Swann, G. F. Gribakin, J. R. Danielson, S. Ghosh, M. R. Natisin, and C. M. Surko, “Effect of chlorination on positron binding to hydrocarbons: Experiment and

- theory,” *Physical Review A*, vol. 104, p. 012813, July 2021. Citado 3 vezes nas páginas 25, 29 e 68.
- [24] J. R. Danielson, S. Ghosh, and C. M. Surko, “Enhancement of positron binding energy in molecules containing bonds,” *Physical Review A*, vol. 106, p. 032811, Sept. 2022. Citado 2 vezes nas páginas 29 e 76.
- [25] M. Tachikawa, Y. Kita, and R. J. Buenker, “Bound states of the positron with nitrile species with a configuration interaction multi-component molecular orbital approach,” *Phys. Chem. Chem. Phys.*, vol. 13, no. 7, pp. 2701–2705, 2011. Citado na página 29.
- [26] J. R. Danielson, A. C. L. Jones, J. J. Gosselin, M. R. Natisin, and C. M. Surko, “Interplay between permanent dipole moments and polarizability in positron-molecule binding,” *Physical Review A*, vol. 85, p. 022709, Feb. 2012. Citado 2 vezes nas páginas 29 e 68.
- [27] A. C. L. Jones, J. R. Danielson, J. J. Gosselin, M. R. Natisin, and C. M. Surko, “Positron binding to alcohol molecules,” *New Journal of Physics*, vol. 14, p. 015006, Jan. 2012. Citado 2 vezes nas páginas 29 e 68.
- [28] T. G. Dietterich and E. B. Kong, “Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms,” *Technical report, Oreg State University*, 1995. Citado na página 33.
- [29] G. James and T. Hastie, “Generalizations of the Bias/Variance Decomposition for Prediction Error,” *Stanford University*, 1997. Citado na página 34.
- [30] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103 of *Springer Texts in Statistics*. New York, NY: Springer New York, 2013. Citado 3 vezes nas páginas 35, 36 e 37.
- [31] S. Raschka, “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning,” *digital preprint, arXiv 1811.12808*, Nov. 2020. Citado na página 37.
- [32] G. C. Cawley and N. L. C. Talbot, “On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation,” *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010. Citado na página 42.
- [33] A. Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems,” Citado 4 vezes nas páginas 45, 46, 51 e 52.
- [34] W. S. Mcculloch, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, 1943. Citado na página 55.

- [35] S. Laqtib, K. E. Yassini, and M. L. Hasnaoui, “A deep learning methods for intrusion detection systems based machine learning in MANET,” in *Proceedings of the 4th International Conference on Smart City Applications*, (Casablanca Morocco), pp. 1–8, ACM, Oct. 2019. Citado na página 55.
- [36] H. Leung and S. Haykin, “The complex backpropagation algorithm,” *IEEE Transactions on Signal Processing*, vol. 39, pp. 2101–2104, Sept. 1991. Citado na página 57.
- [37] L. S. Shapley, “A value for n-person games,” in *Contributions to the Theory of Games II*, pp. 307–317, Princeton: Princeton University Press, 1953. Citado na página 58.
- [38] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in neural information processing systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017. Citado 2 vezes nas páginas 59 e 73.
- [39] P. I. Frazier, “A Tutorial on Bayesian Optimization,” *digital preprint, arXiv 1807.02811*, Aug. 2018. Citado na página 63.
- [40] W. M. Haynes, ed., *CRC handbook of chemistry and physics*. London, England: CRC Press, 97 ed., July 2016. Citado na página 68.
- [41] N. I. of Standards and Technology, “Computational chemistry comparison and benchmark database.” <https://cccbdb.nist.gov/>, 2023. Citado na página 68.
- [42] A. C. L. Jones, J. R. Danielson, M. R. Natisin, C. M. Surko, and G. F. Gribakin, “Ubiquitous Nature of Multimode Vibrational Resonances in Positron-Molecule Anihilation,” *Physical Review Letters*, vol. 108, p. 093201, Feb. 2012. Citado na página 68.
- [43] Python Software Foundation, *Python Language Reference, version 3.8*, 2023. Citado na página 73.
- [44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, “Scikit-learn: Machine Learning in Python,” *MACHINE LEARNING IN PYTHON*. Citado na página 73.
- [45] W. McKinney, “Data Structures for Statistical Computing in Python,” (Austin, Texas), pp. 56–61, 2010. Citado na página 73.
- [46] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. Van Kerkwijk, M. Brett, A. Haldane, J. F. Del Río, M. Wiebe, P. Peterson,

- P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sept. 2020. Citado na página 73.
- [47] T. scikit-optimize contributors, “Scikit-optimize.” <https://github.com/scikit-optimize/scikit-optimize>, 2023. Citado na página 73.
- [48] M. Lakshminarayanan, R. Dutta, D. V. M. Repaka, S. Jayavelu, W. L. Leong, and K. Hippalgaonkar, “Comparing data driven and physics inspired models for hopping transport in organic field effect transistors,” *Scientific Reports*, vol. 11, p. 23621, Dec. 2021. Citado na página 91.
- [49] M. Lakshminarayanan, R. Dutta, D. V. M. Repaka, S. Jayavelu, W. L. Leong, and K. Hippalgaonkar, “Interpretable physics-based models compared with symbolic regression for hopping transport in organic field-effect transistors,” preprint, In Review, Aug. 2021. Citado na página 91.
- [50] B. Weng, Z. Song, R. Zhu, Q. Yan, Q. Sun, C. G. Grice, Y. Yan, and W.-J. Yin, “Simple descriptor derived from symbolic regression accelerating the discovery of new perovskite catalysts,” *Nature Communications*, vol. 11, p. 3513, July 2020. Citado na página 91.