UNIFACCAMP
Centro Universitário Campo Limpo Paulista

Uma Abordagem para o Uso de Aprendizado de Máquina em Bases Abertas para Apoiar a Predição de Distribuição de Fármacos no Organismo

Helder Pestana

Julho / 2025

Dissertação de Mestrado em Ciência da Computação

Uma Abordagem para o Uso de Aprendizado de Máquina em Bases Abertas para Apoiar a Predição de Distribuição de Fármacos no Organismo

Esse documento corresponde à Dissertação apresentada à Banca Examinadora no curso de Mestrado em Ciência da Computação da UNIFACCAMP- Centro Universitário Campo Limpo Paulista.

Campo Limpo Paulista, 28 de julho de 2025.

Helder Pestana

Orientador: Prof. Dr. Rodrigo Bonacin

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Dedico este trabalho à memória da minha querida avó Hermínia, que sempre acreditou em mim e me apoiou nos momentos mais difíceis; à minha mãe Luci e à minha irmã Cibelle, pelo apoio constante, pelo incentivo, pela paciência e por estarem ao meu lado nos desafios do dia a dia; à minha namorada Silvana, pela compreensão, pelo carinho e pelo amor ao longo desta jornada; e, principalmente, à Nossa Senhora Aparecida, que sempre esteve ao meu lado, me guiando com fé e proteção.

Ficha catalográfica elaborada pela

Biblioteca Central da Unifaccamp

P571a

Pestana, Helder

Uma abordagem para o uso de aprendizado de máquina em bases abertas para apoiar a predição de distribuição de fármacos no organismo / Helder Pestana. Campo Limpo Paulista, SP: Unifaccamp, 2025.
81 f.: il.

Orientador: Prof. Dr. Rodrigo Bonacin

Dissertação (Programa de Mestrado Acadêmico em Ciência da Computação) – Centro Universitário Campo Limpo Paulista – Unifaccamp.

1. Aprendizado de máquina. 2. Aprendizado profundo. 3. Distribuição de fármacos. 4. Bases de dados abertas. 5. Modelagem preditiva. I. Bonacin, Rodrigo. II. Centro Universitário Campo Limpo Paulista. III. Título.

CDD – 005.75

AGRADECIMENTOS

Primeiramente quero agradecer ao exímio e dedicado professor Dr. Ferrucio de Franco Rosa que percebendo o meu interesse no campo da Inteligência Artificial me encaminhou a orientação do professor Dr. Rodrigo Bonacin que também pela excelência como professor e orientador viabilizou a realização deste trabalho. O professor Rodrigo sempre muito solicito em qualquer dia da semana, feriados ou finais de semana e praticamente qualquer hora do dia ou da noite, ajudando nas duvidas, na execução dos modelos nos servidores do CTI Renato Archer, na resolução de problemas nos scripts e principalmente no redirecionamento da pesquisa quando a mesma chegou num ponto sem respostas assertivas.

Quero também agradecer imensamente a Dra. Mariângela Dametto pelas inúmeras horas de dedicação em reuniões no intuito de me ajudar a compreender inicialmente os conceitos bioquímicos e farmacológicos que serviram como ponto inicial no desenvolvimento desta pesquisa. Agradeço também ao Professor Dr. Sérgio Modesto Vechi, da Universidade Federal do Alagoas, pelo apoio fundamental na abordagem dos desafios relacionados aos descritores bioquímicos e modelos preditivos. A imprescindível colaboração de ambos fez toda a diferença nesta pesquisa.

Também quero agradecer a Biomédica Dra. Ana Carolina que forneceu as ideias primordiais para o desenvolvimento desta pesquisa e ao especialista com doutorado em IA para a saúde Dr. Reinaldo Padilha França pelo valioso auxílio na configuração do ambiente de desenvolvimento dos modelos de IA, sempre se mostrando muito prestativo e atencioso.

Para concluir quero também deixar o meu agradecimento a todos os dedicados professores deste programa de mestrado que ampliaram minha visão crítica, refinaram meu senso acadêmico e acima de tudo me ajudaram a crescer como docente.

“Nossa maior fraqueza está em desistir. O caminho mais certo de vencer é tentar mais uma vez.”

Thomas Edison

*“*Tudo vale a pena quando a alma não é pequena.*”*

Fernando Pessoa

Resumo. *O desenvolvimento de medicamentos eficazes e seguros depende de processos complexos que envolvem muitas vezes alto grau de incerteza, exigindo assim muitos experimentos in vitro e in vivo que consomem muitos recursos e tempo. Um dos principais desafios desse processo é garantir uma adequada distribuição dos fármacos no organismo. Uma estratégia eficaz para acelerar essa etapa e otimizar o desenvolvimento de novos medicamentos consiste em compreender como diferentes compostos interagem com o organismo humano e de que maneira suas propriedades físico-químicas influenciam tanto a distribuição quanto a eficácia terapêutica. Essa abordagem permite identificar candidatos promissores de maneira mais rápida e eficiente, além de otimizar suas formulações para maximizar o potencial clínico. Tendo em vista este contexto e o avanço dos algoritmos de aprendizado de máquina e a crescente disponibilidade de dados em bases abertas, este trabalho visa pesquisar e desenvolver modelos preditivos que possam otimizar o processo de seleção e formulação de novos medicamentos. O projeto inclui etapas de revisão da literatura, seleção de bases de dados, pré-processamento de dados, desenvolvimento de modelos preditivos e análise dos resultados, visando contribuir para a melhoria da eficiência na indústria farmacêutica por meio da predição de propriedades ligadas à distribuição de fármacos. Os protótipos construídos são destinados a pesquisadores e profissionais que queiram otimizar e aprimorar o desenvolvimento de medicamentos. Os resultados com o uso de técnicas de aprendizado de máquina e aprendizado profundo são promissores e justificam a continuidade desta pesquisa.*

Palavras-chave: *Aprendizado de Máquina, Aprendizado Profundo, Distribuição de Fármacos, Bases de Dados Abertas, Modelagem Preditiva.*

Abstract: *Developing effective and safe drugs depends on complex processes that often involve a high degree of uncertainty, thus requiring many in vitro and in vivo experiments that consume a lot of resources and time. One of the main challenges of this process is to ensure an adequate distribution of drugs in the body. An effective strategy to accelerate this step and optimize the development of new drugs is to understand how different compounds interact with the human body and how their physicochemical properties influence both distribution and therapeutic efficacy. This approach allows us to identify promising candidates more quickly and efficiently, in addition to optimizing their formulations to maximize clinical potential. Given this context and the advancement of machine learning algorithms and the increasing availability of data in open databases, this work aims to research and develop predictive models that can optimize the process of selecting and formulating new drugs. The project includes stages of literature review, database selection, data preprocessing, development of predictive models, and analysis of results, aiming to contribute to improving efficiency in the pharmaceutical industry by predicting properties linked to drug distribution. The prototypes built are intended for researchers and professionals who want to optimize and improve the drug development process. The results using machine learning and deep learning techniques are promising and justify the continuation of this research.*

Keywords: *Machine Learning, Deep Learning, Drug Distribution, Open Databases, Predictive Modeling.*

Sumário

Glossário

AB	*AdaBoost*
ADMET	Absorção, Distribuição, Metabolismo, Excreção e Toxicidade
ANVISA	Agência Nacional de Vigilância Sanitária
CNN	*Convolutional Neural Network*
CSV	*Comma-Separated Values*
DL	*Deep Learning*
DNN	*Deep Neural Network*
Drug Delivery	Distribuição de Fármacos
DT	*Decision Tree*
ETL	Extração, Transformação e Carga
FDA	*Food and Drug Administration*
GB	*Gradient Boosting*
GNB	*Gaussian Naive Bayes*
GP	*Gaussian Process*
IA	Inteligência Artificial
JSON	*JavaScript Object Notation*
KNN	*K-Nearest Neighbors*
KR	*Kernel Ridge Regression*
LGBM	*Light Gradient Boosting Machine*
LR	*Logistic Regression*
LRG	*Linear Ridge Regression*
LRM	*Multiple Linear Regression*
ML	*Machine Learning*
MLP	*Multilayer Perceptron*
MLR	*Multiple Linear Regression*
MSE	*Mean Squared Error*
NB	*Naive Bayes*
NCBI	*National Center for Biotechnology Information*
Overfiting	Modelo aprende demais os dados de treino e erra nos novos.
P&D	Pesquisa e Desenvolvimento
PLS	*Partial Least Squares*
PLSR	*Partial Least Squares Regression*

ReLU	*Rectified Linear Unit*
R^2 Score	*R-squared*
RF	*Random Forest*
RN	*ResNet DNN*
RNA	Rede Neural Artificial
RR	*Ridge Regression*
SEDDS	*Self-Emulsifying Drug Delivery Systems*
SMILES	*Simplified Molecular Input Line Entry System*
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*
SVN	*Support Vector Network*
XGB	*eXtreme Gradient Boosting*
XML	*eXtensible Markup Language*

Lista de Tabelas

Lista de Figuras

1 Introdução

A eficácia terapêutica de um medicamento envolve fatores como a biodisponibilidade, ou seja, a quantidade de um medicamento que realmente é absorvida pelo corpo humano e se torna disponível para uso, e a absorção que é processo pelo qual um medicamento é transferido do local de administração para a corrente sanguínea, tornando-se disponível para distribuição para os tecidos e órgãos do corpo. Estes fatores, ligados à distribuição de fármacos, dependem da estrutura química do fármaco, suas propriedades físico-químicas, interações com proteínas e tecidos biológicos, entre outros (Shargel, 1999).

Pesquisadores do campo da farmacologia desenvolveram bases de dados abertas contendo conjuntos de dados acessíveis publicamente que fornecem informações sobre uma ampla gama de drogas, incluindo aquelas desenvolvidas pela indústria farmacêutica, bem como substâncias experimentais e drogas ilícitas (Wishart, 2006). Essas bases de dados geralmente incluem uma variedade de informações, como estrutura química dos fármacos, propriedades físico-químicas, dados de ensaios clínicos e pré-clínicos, interações medicamentosas, etc. A análise dessas bases de dados tem o potencial de identificar padrões e correlações, assim os algoritmos de Aprendizado de Máquina (ML – *Machine Learning*) podem ajudar a prever como os medicamentos são distribuídos nos tecidos e órgãos do corpo humano, permitindo avaliar a eficácia e a segurança dos compostos em estudo, contribuindo para a tomada de decisões e para a redução do tempo e dos custos envolvidos na produção farmacêutica.

O propósito desta pesquisa é avaliar a capacidade dos algoritmos de ML, incluindo Aprendizagem Profunda (DL – *Deep Learning*), com as informações contidas em bases abertas no campo da farmacologia para apoiar a predição de propriedades relevantes à distribuição de fármacos. Esta pesquisa é base para o desenvolvimento de sistemas para apoio a pesquisadores, profissionais e a indústria farmacêutica no processo de desenvolvimento de medicamentos.

1.1 Contexto e Motivação

Predizer os efeitos clínicos de um determinado medicamento é essencial para o desenvolvimento de medicamentos eficazes e seguros. A distribuição de fármacos (*Drug*

Delivery, como é conhecida no contexto farmacêutico em inglês) concentra-se na compreensão de como os medicamentos são transportados pelo corpo após sua administração e a concentração do fármaco nos tecidos-alvo (Pattni, 2015). O propósito é evitar que na administração destes medicamentos ocorra subdosagens em locais críticos ou toxicidades elevadas em locais não desejados, o que compromete a ação do fármaco e aumenta o risco de efeitos adversos.

Na abordagem mais comumente utilizada o desenvolvimento de formulações depende de processos incertos de tentativa e erro, exigindo assim um grande número de experimentos *in vitro* e *in vivo* que consomem muitos recursos e tempo (Shargel, 1999). Entre as dificuldades deste processo está o desenvolvimento de medicamentos com boa distribuição de fármacos. A Figura 1 ilustra as etapas desse processo, que atualmente é feito de forma manual, que segundo Moore *et al.* (2018) além de demorado envolve custos elevados.

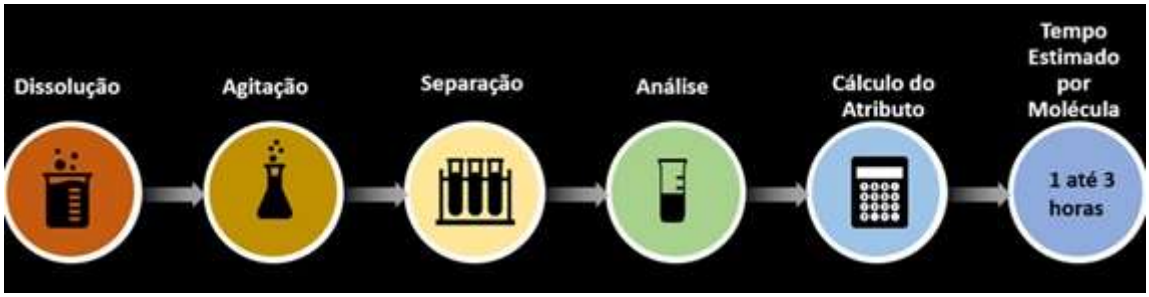

Figura 1. Fases do Processo típico *in vitro* e tempo estimado por molécula. Fonte: autor.

Assim, uma forma de acelerar o processo de desenvolvimento de medicamentos é otimiza-lo, identificando como diferentes compostos interagem com o corpo humano e como suas características físico-químicas influenciam sua distribuição e eficácia terapêutica. Isso pode acelerar o processo ao identificar candidatos promissores de maneira mais rápida e eficiente, e otimizar suas formulações para maximizar sua eficácia clínica.

O aumento da capacidade computacional, a evolução dos algoritmos de ML e o conjunto de dados disponíveis ofertados em bases abertas motivaram o desenvolvimento deste projeto. Espera-se que por meio destes recursos será possível ofertar soluções que otimizem o tempo e reduzam o custo, possibilitando a fabricação de fármacos de forma mais eficiente e mais acessível.

1.2 Problemática e Justificativa

Uma abordagem baseada em ML para o apoio à predição de propriedades ligadas a distribuição de fármacos encontra obstáculos como a falta de qualidade dos dados, qualidade do modelo de aprendizagem e uma compreensão profunda da natureza dos modelos de Inteligência Artificial (IA) (Tran *et al.,* 2023). Embora as fontes de dados públicos para Pesquisa e Desenvolvimento (P&D) de medicamentos tenha crescido nos últimos anos, os algoritmos de IA precisam de volume de dados ,qualidade e integração para gerar modelos precisos.

Para além da qualidade dos dados, é crucial ter um modelo de aprendizagem apropriado para explorar o potencial da IA na previsão de propriedades distributivas. Isso inclui investigar e construir novos modelos baseados em ML e DL.

Outro desafio enfrentado é a dificuldade em compreender a natureza dos modelos de IA. Apesar do desempenho dos modelos de previsão de propriedades distributivas baseados em IA, ainda há uma carência de interpretações mais precisas. Isso torna difícil avaliar e confiar nas hipóteses geradas pela IA devido à sua natureza de "caixa preta", dificultando o aprimoramento do modelo e a otimização de compostos com propriedades de distribuição indesejáveis. Assim como na descoberta de medicamentos, o estudo da distribuição de fármacos é uma disciplina multidisciplinar, sendo preciso ter conhecimentos relevantes em áreas como biologia, bioinformática, farmacologia, química e informática química.

Uma abordagem crucial para enfrentar os desafios na previsão da distribuição de fármacos com base em IA é otimizar a parametrização dos algoritmos utilizados. Isso requer um foco específico na melhoria tanto da qualidade quanto da quantidade dos dados empregados para treinar e validar os modelos. A implementação de técnicas avançadas de limpeza e pré-processamento de dados se mostra essencial para mitigar o ruído e o viés inerentes aos conjuntos de dados, resultando em previsões mais precisas e confiáveis.

A literatura atual sobre o tema tem abordado principalmente aspectos relacionados ao uso de algoritmos de ML e DL e sua aplicabilidade no desenvolvimento de soluções para distribuição de fármacos em contextos específicos, no entanto, uma lacuna

significativa persiste em relação ao entendimento dos algoritmos utilizados, bem como a forma de parametrização e ajuste destes algoritmos na construção de um modelo confiável. Muitos estudos também não evidenciaram os resultados obtidos a partir do uso desses modelos. O presente trabalho visa preencher essas lacunas e investigar detalhadamente a relação entre os algoritmos utilizados e as medidas de desempenho, utilizando métodos e abordagens inovadoras para explorar essas interações. Isso contribuirá para uma compreensão mais completa do uso de ML e DL em bases abertas como ferramenta de apoio a predição na distribuição de fármacos.

1.3 Objetivos, Contribuições e Métodos

O objetivo geral deste trabalho é pesquisar e propor uma solução que possa predizer propriedades ligadas à distribuição de fármacos por meio de modelos de ML e DL treinados utilizando bases internacionais abertas para otimizar a produção de fármacos. O foco inicial, apresentado nesta dissertação, está nas propriedades de biodisponibilidade e no coeficiente de partição octanol-água (LogP) atributos relacionados a capacidade de distribuição dos medicamentos no organismo (Aliagas *et al*., 2022). Podendo este ser expandido para outras propriedades em pesquisas futuras.

Este trabalho se propõe a responder a seguinte pergunta: “Como criar uma abordagem e solução baseada em ML (e DL) que faça uso de bases internacionais abertas para apoiar a predição das propriedades de biodisponibilidade e LogP ?”. A partir da pergunta principal, outras perguntas específicas são abordadas:

- “Quais bases de dados abertas serão úteis ao treinamento de modelos de ML e DL?”
- “Quais técnicas de seleção e pré-processamento deverão ser aplicadas sobre estes dados antes de utiliza-los para treinar os modelos?”
- “Como codificar (*embedding*) as informações (moléculas) das bases abertas de modo a serem utilizadas pelos algoritmos?”
- “Como escolher os algoritmos adequados para o modelo de aprendizagem de máquina segundo o objetivo proposto?”
- “Como ajustar os hiper parâmetros dos algoritmos de ML e DL?”

- “Como mensurar a eficácia do modelo? Quais métricas serão necessárias para este propósito?”
- “De que forma apresentar os resultados e maximizar a interpretação dos resultados?”

A partir deste objetivo principal e das questões de pesquisa, definimos as seguintes metas:

1. Revisar a literatura sobre o uso de algoritmos de ML e DL para a predição de propriedades ligadas a distribuição de fármacos.
2. Avaliar e selecionar bases abertas nacionais e internacionais existentes que podem ter dados de interesse para treinamento, validação e testes dos modelos.
3. Aplicar o processo de ETL (Extração, Transformação e Carga) sobre estes dados no intuito de deixá-los formatados aos algoritmos.
4. Conceber e criar modelos candidatos (de ML e DL) e ajustar adequadamente os hiper parâmetros dos algoritmos.
5. Validar, ajustar e efetuar os testes finais nos algoritmos com o melhor desempenho.
6. Analisar os resultados obtidos e modelos com maior eficácia.

Assim o presente trabalho se propõe a criação e otimização de modelos de ML e DL específicos para analisar dados de fármacos em bases abertas, isso inclui testes e parametrização dos modelos para lidar com os desafios únicos apresentados por esses dados. Para tanto, se faz necessário avaliar de forma abrangente o desempenho de diferentes algoritmos de aprendizado de máquina na tarefa de análise de dados de fármacos em bases abertas, incluindo comparações de precisão e robustez em diferentes cenários de aplicação.

Esta pesquisa irá contribuir para a aplicação prática de algoritmos e desenvolvimento de modelos de ML (e DL) na descoberta de fármacos, envolvendo a identificação de candidatos a medicamentos promissores do ponto de vista de distribuição de fármacos. Com isso, espera-se fornecer a pesquisadores e profissionais da indústria farmacêutica (bioquímicos, biomédicos, farmacêuticos e químicos) um modelo preditivo para algumas das principais características relacionadas ao processo de distribuição de medicamentos no organismo humano.

1.4 Estrutura da Proposta

A estrutura dos capítulos restantes desta dissertação é a seguinte:

- **Capitulo 2 – Referencial Teórico e Metodológico:** Apresenta a fundamentação dos principais conceitos e tecnologias abordados nesta pesquisa, tais como ADMET (Absorção, Distribuição, Metabolismo e Excreção e Toxicidade), ML e DL, bem como Bases de Dados Abertas contendo dados químicos e farmacológicos.
- **Capitulo 3 - Revisão da Literatura e Trabalhos Relacionados:** Apresenta um estudo abrangente sobre a pesquisa em temas relacionados por meio de uma revisão da literatura, enfatizando a metodologia utilizada e os resultados mais significativos alcançados, bem como a discussão dos trabalhos relacionados.
- **Capitulo 4 - O Uso de Aprendizado de Máquina no Apoio à Predição de Distribuição de Fármacos:** Apresenta a metodologia de pesquisa, a base de dados e a seleção de atributos, as fases de pré-processamento e codificação, os modelos, a implementação, a execução e as métricas de avaliação.
- **Capitulo 5 - Análise dos Resultados e Discussão:** Apresenta os resultados experimentais obtidos, bem como uma discussão sobre as implicações dos resultados, a aplicabilidade do modelo na prática, a possíveis vieses e desafios futuros.
- **Capitulo 6 – Conclusão:** Apresenta as contribuições, limitações, trabalhos futuros e considerações finais desta dissertação.

2 Referencial Teórico e Metodológico

Neste capítulo, é apresentada a fundamentação teórica e metodológica, além das tecnologias utilizadas no desenvolvimento desta dissertação. O conteúdo deste capítulo é embasado na revisão exploratória de trabalhos relacionados ao tema central desta pesquisa, bem como na consulta a especialista no domínio.

A seção 2.1 aborda os conceitos básicos sobre o desenvolvimento de fármacos, as propriedades farmacológicas e a distribuição dos fármacos no organismo humano; na seção 2.2 são explicitados conceitos sobre ML e DL; na seção 2.3 são detalhadas as principais bases de dados abertas sobre dados químicos e farmacológicos e o SMILES (*Simplified Molecular-Input Line-Entry System*) , padrão de representação de estruturas químicas de compostos orgânicos e inorgânicos por meio de uma linha de texto simples.

2.1 Desenvolvimento de fármacos, propriedades farmacológicas e distribuição de fármacos

O desenvolvimento de fármacos é um processo complexo e de elevado custo financeiro que envolve diversas etapas, desde a descoberta inicial até a disponibilização do medicamento no mercado. Segundo Franke (2020) para compreendermos o processo inicialmente é importante entender quatro conceitos básicos:

- Droga: Qualquer substancia que interaja com o organismo produzindo algum efeito;
- Fármaco: Uma substancia definida com propriedades ativas produzindo efeito terapêutico;
- Medicamento: É quando no fármaco são adicionados todos os componentes (incipientes) para que este seja administrado terapeuticamente;
- Forma Farmacêutica: É a forma final de como um medicamento se apresenta: comprimido, capsulas, injetáveis, etc.

Quanto às etapas no desenvolvimento de fármacos (Franke, 2020), podem ser subdivididas nos respectivos períodos:

- Período de descoberta da droga:

    - Descoberta de Fármacos: Nesta fase, os pesquisadores identificam e caracterizam moléculas que têm potencial para se tornarem medicamentos. Isso pode envolver triagem de compostos químicos, estudos de biologia molecular e modelagem computacional.
    - Desenvolvimento Pré-clínico: Os compostos promissores identificados na fase de descoberta passam por testes pré-clínicos em laboratório e em animais para determinar sua segurança, eficácia e toxicidade. Nesta fase também incluem a otimização da formulação do medicamento.
- Período de desenvolvimento do fármaco:
    - Ensaios Clínicos: Se um composto passa nos testes pré-clínicos, ele avança para ensaios clínicos em seres humanos. Estes ensaios são divididos em três fases (Fase I, Fase II e Fase III) e visam determinar a segurança, eficácia e dosagem adequada do medicamento.
    - Submissão Regulatória: Após concluir os ensaios clínicos, os dados são submetidos às agências regulatórias, como a *Food and Drug Administration* (FDA) nos Estados Unidos ou a Agência Nacional de Vigilância Sanitária (ANVISA) no Brasil, para revisão e aprovação.
- Período de fabricação, comercialização e farmacovigilância (Fase IV):
    - Fabricação: Uma vez aprovado, o medicamento é fabricado em larga escala de acordo com padrões rigorosos de qualidade e boas práticas de fabricação (GMP, *Good Manufacturing Practices*).
    - Comercialização: Após a aprovação regulatória e a fabricação em escala comercial, o medicamento é lançado no mercado, onde é distribuído para farmácias, hospitais e outros pontos de venda.
    - Farmacovigilância: Após o lançamento no mercado, o medicamento continua sendo monitorado para identificar e avaliar quaisquer efeitos colaterais ou problemas de segurança que possam surgir.

Para compreendermos como ocorre a distribuição de fármacos no organismo, precisamos entender o conceito de farmacocinética (Pereira, 2007) que consiste no estudo do movimento de uma substância química, em particular, um fármaco no interior de um organismo vivo. Na farmacocinética existem propriedades farmacológicas importantes como a ADMET.

A absorção é processo pelo qual o fármaco entra na corrente sanguínea a partir do local de administração. A distribuição é como o fármaco se espalha pelos tecidos do corpo, influenciado por fatores como perfusão sanguínea, ligação a proteínas plasmáticas e características da própria molécula do fármaco. O metabolismo é o processo pelo qual o fármaco é modificado no organismo, geralmente ocorrendo no fígado, para facilitar sua excreção. A excreção é a remoção do fármaco e seus metabólitos do organismo, principalmente pelos rins e a toxicidade refere-se à capacidade de uma substância causar danos a um organismo vivo.

Na distribuição de fármacos, uma série de fatores desempenham um papel crucial, incluindo barreiras biológicas, biodisponibilidade e meia-vida. No organismo, os fármacos enfrentam desafios como a superação da barreira hematoencefálica e da barreira placentária, conforme sua aplicação. A biodisponibilidade refere-se à fração do fármaco administrado que alcança a circulação sistêmica, pronta para desencadear seus efeitos farmacológicos. Por exemplo, quando um medicamento é administrado por via oral, parte dele pode ser degradada no trato gastrointestinal ou metabolizada no fígado antes de chegar à corrente sanguínea, reduzindo sua biodisponibilidade (Hosseini *et al*., 2024). Por sua vez, a meia-vida representa o período necessário para que a concentração do fármaco no sangue seja reduzida pela metade, o que pode impactar diretamente na frequência requerida para sua dosagem.

2.2 Aprendizado de Máquina e Aprendizagem Profunda

O ML é um campo que se destaca por sua abordagem de aprendizagem da IA focando no desenvolvimento de algoritmos capazes de representar eficientemente os conjuntos de dados (Choi *et al.*, 2020). Ao contrário da programação clássica, onde algoritmos são codificados de forma explícita utilizando características conhecidas, o ML emprega conjuntos de dados para criar modelos.

O aprendizado profundo, uma subárea do aprendizado de máquina, utiliza redes neurais com múltiplas camadas para modelar dados complexos. Essas redes são capazes de aprender por meio dos dados construir modelos, o que as torna particularmente eficazes em tarefas como reconhecimento de imagem e processamento de linguagem natural (Goodfellow *et al*., 2016).

Na ML, quatro métodos de aprendizagem são comumente utilizados, cada um adequado para resolver diferentes tipos de tarefas: supervisionada (utiliza dados rotulados), não supervisionada (utiliza dados não rotulados), semi-supervisionada (combinam dados rotulados e não rotulados) e aprendizado por reforço (utiliza mecanismos de punição e recompensa).

2.2.1 Conceitos de Aprendizagem Supervisionada

Rimal (2024) ilustra a aprendizagem supervisionada em um cenário de uma empresa imobiliária que deseja prever o preço de uma casa com base em características específicas. Para iniciar, a empresa coletaria um conjunto de dados contendo diversas instâncias, cada uma representando uma observação singular de uma casa e suas características associadas. Essas características incluem propriedades registradas da casa que podem ser relevantes para a predição de preços, como área total, número de andares e presença de quintal. O alvo a ser previsto é o preço do imóvel.

Geralmente, os conjuntos de dados são divididos em conjuntos de treinamento, validação e teste, sendo que os modelos tendem a ter um desempenho melhor nos dados nos quais foram treinados. A aprendizagem supervisionada utiliza padrões presentes no conjunto de treinamento previamente rotulados para mapear características para o alvo, permitindo que um algoritmo faça previsões, tais como preços de imóveis, em conjuntos de dados futuros. Ou seja, nesse exemplo é necessário construir um conjunto de dados com preços reais dos imóveis, para então treinar o algoritmo que constrói um modelo a ser utilizado em predições futuras.

O resultado da execução do algoritmo de aprendizado de máquina pode ser expresso como uma função $f(x)$ que recebe uma nova entrada x e gera um vetor de saída y, codificado da mesma forma que os vetores alvo. A forma precisa da função $f(x)$ é determinada durante a fase de treinamento, também conhecida como fase de aprendizado,

com base nos dados de treinamento previamente rotulada. Uma vez que o modelo é treinado, ele pode ser usado para prever a categoria de novas entradas, que compõem um conjunto de teste rotulado que é utilizado para prever medidas de performance, tais como a acurácia. A capacidade de categorizar corretamente novos exemplos que diferem daqueles usados no treinamento é conhecida como generalização (Bishop ,2006).

Essa abordagem é denominada supervisionada pois o algoritmo infere um modelo a partir de pares de características e alvo, sendo informado pelo alvo se suas previsões foram corretas. Em outras palavras, as características x são mapeadas para o alvo y, aprendendo a função de mapeamento $f(x)$, de modo que os preços futuros dos imóveis possam ser aproximados utilizando o algoritmo desenvolvido.

Um conjunto de validação é utilizado para avaliar e melhorar o modelo gerado pelo algoritmo de ML. A avaliação da performance do modelo ocorre no conjunto de dados de teste, que consiste em dados não vistos pelo algoritmo durante o treinamento e validação do modelo.

Os passos fundamentais do ML supervisionado são os seguintes:

- Aquisição e Divisão dos Dados: O primeiro passo é adquirir um conjunto de dados e dividi-lo em conjuntos separados de treinamento, validação e teste.
- Modelagem e Ajuste: Em seguida, utiliza-se os conjuntos de treinamento e validação para informar o modelo sobre a relação entre as características e o alvo.
- Avaliação da Performance: O modelo é avaliado utilizando o conjunto de dados de teste para determinar sua capacidade de prever com precisão os preços dos imóveis para instâncias não vistas.

Durante cada iteração, o desempenho do algoritmo nos dados de treinamento é comparado com seu desempenho no conjunto de validação. O algoritmo é então ajustado com base na avaliação feita no conjunto de validação. No entanto, é importante ressaltar que a generalização do desempenho do algoritmo pode variar, uma vez que o conjunto de validação pode diferir do conjunto de teste.

A avaliação da performance do desempenho de modelos preditivos é um componente essencial no desenvolvimento de sistemas de aprendizado de máquina. Um dos métodos mais amplamente utilizados para essa finalidade é a validação cruzada, que visa estimar a capacidade de generalização do modelo a partir de partições do conjunto de dados disponíveis. Nessa abordagem, os dados são divididos aleatoriamente em k subconjuntos (ou folds) de tamanhos aproximadamente iguais. Em seguida, o modelo é treinado k vezes, sendo que, a cada iteração, k−1 folds são utilizados para treinamento e o fold remanescente é reservado para validação. Ao final das k iterações, calcula-se a média das métricas de desempenho obtidas (tais como *MSE* (*Mean Squared Error*), R^2 *Score*, etc.), fornecendo uma estimativa mais confiável da performance do modelo em dados não vistos.

A utilização da validação cruzada, segundo Bishop (2006), é particularmente importante para reduzir o risco de sobreajuste (*overfitting*), um fenômeno comum em modelos de alta complexidade que apresentam desempenho elevado no conjunto de treinamento, mas falham ao generalizar para novos dados. Esse fenômeno ocorre quando o modelo passa a representar não apenas os padrões relevantes dos dados, mas também características específicas e não generalizáveis do conjunto de treinamento, prejudicando sua capacidade de generalização.

Além disso, é fundamental considerar o papel dos hiperparâmetros no processo de ajuste do modelo. Diferentemente dos parâmetros internos, que são aprendidos diretamente a partir dos dados durante o treinamento, os hiperparâmetros são parâmetros externos definidos previamente e controlam aspectos fundamentais da estrutura ou do comportamento do modelo. Sua escolha influencia diretamente a capacidade preditiva e a generalização do modelo. Em muitos casos, a combinação entre validação cruzada e técnicas de busca por hiperparâmetros como grid search, constitui uma estratégia eficaz para alcançar modelos mais robustos e generalizáveis (Bishop ,2006).

2.2.2 Modelos de Classificação e Regressão

As tarefas mais comuns no aprendizado supervisionado são a regressão e a classificação. Na regressão, o objetivo é prever dados numéricos (quantitativos), que podem estar num intervalo continuo como por exemplo altura, salário, inflação,

semelhante ao exemplo de preços de imóveis ou valores inteiros como idade, número de empregados ou produção de veículos.

Por outro lado, na classificação, o objetivo é prever a categoria à qual uma instância pertence e este tipo de dado é qualitativo e pode ser do tipo nominal (sem uma ordem definida) como raça, cor, sexo ou profissão ou do tipo ordinal (com ordem definida) como escolaridade, faixa etária ou ranking de reclamações.

Os dados também podem ser classificados como do tipo binário que podem assumir apenas um de dois estados e pode ser binário simétrico como por exemplo o gênero de uma pessoa ou binário assimétrico como por exemplo a biodisponibilidade de uma determinada molécula (valor 1 para com biodisponibilidade e valor 0 para sem biodisponibilidade) ou a presença ou não de uma determinada propriedade bioquímica.

Os cientistas de dados também podem transformar a variável alvo numérica em uma variável categórica. Usando como exemplo a previsão de faixas de preços para venda de imóveis em um mercado volátil, agrupando os preços dos imóveis em classes separadas, como (0, 125 mil), (125 mil, 250 mil), (250 mil, 375 mil) e (375 mil, ∞). Essas classes seriam ordinais, ou seja, há uma ordem natural associada às categorias. No entanto, se a tarefa fosse determinar se as casas tinham revestimento de madeira, plástico ou metal, estes dados seriam classificados como nominais, independentes uns dos outros e sem uma ordem natural.

2.2.3 Métricas de Classificação e Regressão

2.2.3.1 Medidas de Performance para Classificação

Segundo Provost e Fawcett (2013), a análise do desempenho de um classificador tem por base um conjunto de métricas que comparam as previsões do modelo com os valores reais, e estas métricas derivam da matriz de confusão (Tabela 1).

Tabela 1 – Matriz de Confusão

	Previsto Positivo (P)	**Previsto Negativo (N)**
Real Positivo (P)	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Real Negativo (N)	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Métricas Derivadas da Matriz de Confusão:

- Acurácia: Proporção de previsões corretas sobre o total.
  Acurácia = (VP + VN) / (VP + VN + FP + FN)
- Precisão: Proporção de verdadeiros positivos entre todas as previsões positivas.
  Precisão = VP / (VP + FP)
- Especificidade: Proporção de verdadeiros negativos corretamente identificados.
  Especificidade = VN / (VN + FP)
- Revocação (Recall / Sensibilidade): Mede a capacidade do modelo de detectar todos os casos positivos.
  Revocação = VP / (VP + FN)
- Medida F (F1-Score): Combina Precisão e Revocação em uma única métrica
  F1 = 2 x (Precisão x Revocação) / (Precisão + Revocação)
- AUC: Avalia a capacidade do modelo de distinguir entre classes

2.2.3.2 Medidas de Performance para Regressão

Segundo James *et al.* (2013), modelos de regressão são avaliados com base no erro entre valores previstos e reais, utilizando métricas que penalizam erros de forma diferenciada.

Métricas de Avaliação para Modelos de Regressão:

- Erro Absoluto Médio (MAE - Mean Absolute Error): Média das diferenças absolutas entre os valores previstos ($\widehat{y_i}$) e valores reais (y_i).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \qquad (1)$$

- Erro Quadrático Médio (MSE - Mean Squared Error): Média dos quadrados dos erros, penalizando mais severamente previsões distantes do valor real. Ideal para treinar os modelos de regressão.

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

- Raiz do Erro Quadrático Médio (RMSE - Root Mean Squared Error): Raiz quadrada do MSE.

$$RMSE = \sqrt{\left(1/n \sum (y_i - \hat{y}_i)^2\right)} \qquad (3)$$

- Coeficiente de Determinação (R^2 - R-Squared): Mede o quanto da variação dos dados é explicada pelo modelo (valor: 0, modelo não explica nada a valor:1, modelo perfeito). Mede o quão bem o modelo se ajusta aos dados.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \qquad (4)$$

2.2.4 Algoritmos de Aprendizado de Máquina

Os algoritmos abordados nesta dissertação podem ser classificados como: Modelos de Regressão, Modelos Baseados em Árvore, Modelos Probabilísticos, Modelos Baseados em Distância, Modelo Baseado em Margem Máxima e Modelo Baseado em Redes Neurais.

2.2.4.1 Modelos de Regressão

- **Regressão Linear**
  Modelo básico que assume uma relação linear entre as variáveis (James *et al*., 2013).
- **Regressão Ridge**
  Segue o mesmo princípio da regressão linear, mas com penalização para reduzir o efeito de variáveis correlacionadas (Hoerl & Kennard, 1970).
- **Mínimos Quadrados Parciais (PLS)**

Técnica que projeta *features* em um espaço latente de menor dimensão (reduz dimensionalidade), maximizando a covariância com a variável resposta, adequada para dados com alta dimensionalidade (Wold *et al*., 2001).

2.2.4.2 Modelos Baseados em Árvores

- **Árvore de Decisão**

  Regras hierárquicas simples, com uma estrutura no formato de arvore invertida (Breiman, 1984).
- **Random Forest**

  Combina múltiplas árvores para maior precisão, adequados a dados tabulares para evitar *overfitting* (Breiman, 2001).
- **XGBoost/LightGBM**

  Otimizados para velocidade e performance. Usa regras complexas (árvores profundas) e o modelo se ajusta a partir dos erros (Ke *et al*., 2017).
- **AdaBoost**

  Método de *ensemble* que combina vários modelos fracos (como árvores rasas), ajustando iterativamente o foco nos erros cometidos em rodadas anteriores. Dá mais peso aos exemplos mal classificados, tornando o modelo final mais robusto e preciso (Freund & Schapire, 1997).

2.2.4.3 Modelos Probabilísticos

- **GNB - Gaussian Naive Bayes**

  Classificador probabilístico que supõe que cada variável de entrada segue uma distribuição Gaussiana (normal) e que todas as variáveis são mutuamente independentes (Murphy, 2006).
- **NB - Naive Bayes**

  Modelo mais geral do Naive Bayes, que não exige que as variáveis tenham distribuição normal. Ele apenas assume que as variáveis são condicionalmente independentes dado a classe (McCallum & Nigam, 1998).

- **GP - Gaussian Process**
  Modelo probabilístico não paramétrico que define uma distribuição sobre funções. Permite realizar previsões com estimativas de incerteza associadas a cada ponto previsto. (Rasmussen & Williams, 2006).

2.2.4.4 Modelos Baseados em Distância

- **KNN (K-Nearest Neighbors)**
  Modelo supervisionado que classifica ou estima valores com base nos k exemplos mais próximos. Usa métricas de distância (ex.: Euclidiana, Manhattan) para identificar vizinhos no espaço de atributos (Cover & Hart, 1967).
- **SVM/SVR (Support Vector Machines/Regression)**
  Encontra um hiperplano ótimo que separa classes ou ajusta dados usando *kernels* (ex.: RBF) para mapear características em espaços de alta dimensão (Cortes & Vapnik,1995).
- **Kernel Ridge (KR)**
  Combina Ridge Regression com kernels para modelar relações não lineares entre as variáveis. Mede semelhança entre amostras com base em funções kernel, que envolvem distância implícita (Saunders *et al*.,1998).

2.2.4.5 Modelo Baseado em Margem Máxima

- **Support Vector Regression (SVR)**
  Baseado em Support Vector Machines, mapeia dados para um espaço de alta dimensão, encontrando um hiperplano que minimize erros dentro de uma margem (Smola & Schölkopf, 2004).

2.2.4.6 Modelo de Machine Learning Baseado em Redes Neurais

Multilayer Perceptron (MLP)

- Rede neural *feedforward* com camadas ocultas não-lineares (e.g., ReLU), treinada via *backpropagation*. Capaz de aproximar funções universais, mas exige *tuning* de hiperparâmetros (Goodfellow *et al*., 2016).

2.2.5 Redes Neurais e Aprendizagem Profunda

As Redes Neurais Artificiais (RNA) incluem um conjunto de algoritmos de aprendizado de máquina que se inspira nas redes neurais encontradas no cérebro humano. Em uma RNA, cada nó desempenha um papel semelhante ao de um neurônio biológico, comunicando-se com outros nós por meio de conexões, que podem ser comparadas aos axônios e dendritos encontrados no sistema nervoso (Ferneda, 2006).

Assim como no cérebro humano, onde as sinapses entre neurônios são fortalecidas quando os neurônios têm saídas correlacionadas (de acordo com a teoria Hebbiana (Abreu *et. al*, 2020) que postula "células nervosas que se disparam juntas, se conectam juntas"), as conexões entre nós em uma RNA são ponderadas com base na contribuição que cada uma oferece para alcançar o resultado desejado. Esse processo permite que a rede aprenda e adapte seus pesos ao longo do tempo, otimizando seu desempenho na realização de tarefas específicas. Em um modelo treinado, os pesos são ajustados de acordo com hiperparâmetros e o conjunto de treinamento e validação utilizados.

2.2.6 Redes Neurais *Feedforward*

Um *perceptron* é um algoritmo de aprendizado de máquina que recebe um conjunto de características e seus alvos como entrada, buscando identificar uma linha, plano ou hiperplano que separe as classes em um espaço bidimensional, tridimensional ou hiper dimensional, respectivamente. Essas características são então transformadas usando a função sigmoide.

Quando vários *perceptrons* são conectados (Figura 2), formando um modelo conhecido como *perceptron* multicamadas RNA, é comum incluir uma camada de nós de entrada, uma camada de nós de saída e uma ou mais "camadas ocultas" entre elas. Enquanto em RNA simples pode haver uma camada de entrada, zero a três camadas

ocultas e uma camada de saída, redes neurais profundas podem conter dezenas ou até centenas de camadas ocultas.

Em redes neurais *feedforward*, que são as mais comuns, as informações fluem em uma única direção, da camada de entrada para a camada de saída. Cada nó em uma camada subsequente recebe informações de todos os nós na camada anterior, realiza transformações e repassa os resultados para a próxima camada. Já em redes neurais recorrentes, as informações podem ser passadas entre nós dentro da mesma camada ou retroceder para camadas anteriores, permitindo a retroalimentação de dados.

Cada camada em uma RNA pode conter qualquer número de nós, sendo que o número de nós na camada de saída geralmente corresponde ao número de classes sendo previstas. Por exemplo, em classificação multiclasse, há um nó de saída para cada classe, enquanto em classificação binária há um único nó com uma função de ativação sigmoide. Para tarefas de regressão, a camada de saída pode usar uma função de ativação linear. Essas funções de ativação transformam a entrada de um nó em uma saída desejada, e cada nó em uma RNA possui uma função de ativação associada, não se limitando apenas à camada de saída.

Essas funções de ativação, embora não sejam sempre lineares, não precisam ser intrinsecamente complexas. Por exemplo, a unidade linear retificada (ReLU - *Rectified Linear Unit*) aplica uma transformação linear às entradas ≥ 0 e define as entradas < 0 como 0. Essa simplicidade é sua força: ao longo das camadas de uma RNA, as entradas são progressivamente ajustadas, de modo que, na camada final, elas podem não se parecer mais com o estado original. No entanto, essa representação final da entrada é teoricamente a mais preditiva do resultado desejado.

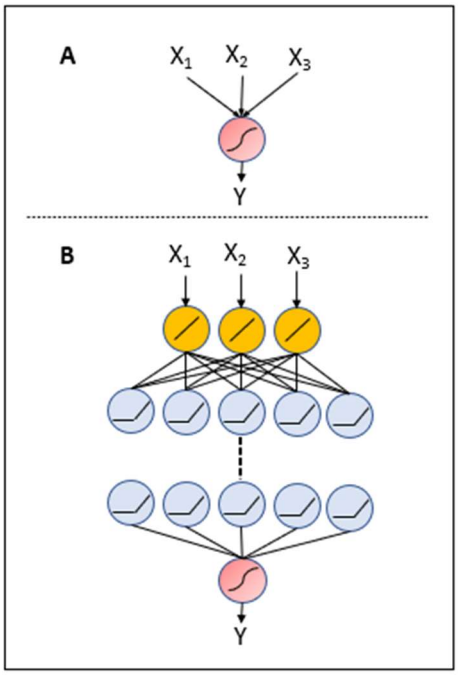

Figura 2. -*Perceptron*(A) e Rede Neural Profunda(B) (Adaptado de Choi *et al*. 2020)

2.2.7 Redes Neurais Convolucionais

Quando utilizamos uma rede neural *feedforward* para reconhecer imagens, cada pixel é tratado como uma entrada separada. No entanto, essa abordagem tem uma limitação: não leva em consideração a relação entre os pixels vizinhos. Por exemplo, em uma imagem, pixels próximos tendem a estar mais correlacionados do que pixels distantes, o que significa que a informação espacial importante pode ser perdida.

É aqui que entra a rede neural convolucional (CNN - *Convolutional Neural Network*). Ao contrário da RNA tradicional, a CNN (Silva, 2018) preserva o contexto espacial das características da imagem. Em vez de alimentar cada pixel separadamente, a CNN utiliza "*patches*" ou pequenas regiões da imagem, permitindo que ela capture relações espaciais entre os *pixels*. Esses *patches* são processados por filtros convolucionais, que aprendem a extrair características específicas da imagem.

As convoluções são amplamente utilizadas no campo do processamento de imagens por sua capacidade de desfocar, realçar ou detectar bordas. Por exemplo, ao aplicar um filtro convolucional, podemos realçar as bordas de um objeto na imagem. Uma imagem digital em escala de cinza é representada por uma única matriz, enquanto uma

imagem colorida é representada por três matrizes empilhadas para os canais de cor vermelho, verde e azul.

Um filtro convolucional é uma matriz menor que é deslizada sobre a imagem original. Em cada posição, ocorre uma multiplicação elemento a elemento entre o filtro e a região correspondente da imagem. A saída dessa operação é um novo conjunto de valores que representam características específicas da imagem. Em resumo, as CNNs são poderosas para o reconhecimento de imagens, pois preservam o contexto espacial e permitem a extração eficiente de características importantes para a tarefa em questão.

Nas CNNs, os filtros são treinados para identificar características específicas em imagens, como linhas verticais ou objetos com formato de U, e registrar sua localização no mapa de características. Uma CNN profunda usa esses mapas de características como entrada para a camada seguinte, que aplica novos filtros para gerar outro mapa de características. Esse processo continua por várias camadas, e à medida que avança, as características extraídas se tornam mais abstratas, mas também mais úteis para a previsão. No final, os mapas de características são comprimidos e passados para uma rede neural *feedforward*, onde ocorre a classificação da imagem com base nas características e texturas extraídas. Esse processo é conhecido como aprendizado profundo (DL - *Deep Learning*).

Além da classificação de imagens, o DL para dados tabulares aplica redes neurais profundas para analisar e identificar padrões em conjuntos de dados estruturados, como planilhas e bancos de dados relacionais. Ao contrário dos dados não estruturados, como imagens ou texto, os dados tabulares possuem colunas com tipos de dados específicos (numéricos, categóricos, etc.). Modelos de DL, como redes neurais totalmente conectadas (*fully connected networks*) e arquiteturas avançadas como TabNet (Arik & Pfister,2021), são treinados para capturar relações complexas e interações entre as colunas, frequentemente superando métodos tradicionais de ML em precisão e capacidade de generalização. Esses modelos são utilizados em tarefas como classificação, regressão e previsão, proporcionando uma abordagem poderosa para a análise de grandes volumes de dados tabulares.

2.3 Bases de Dados Abertas de dados químicos e farmacológicos

As bases de dados abertas de química e farmacologia servem como ferramentas essenciais para a pesquisa, descoberta e desenvolvimento de novos medicamentos, bem como para o avanço do conhecimento científico nessas áreas. Elas proporcionam acesso a uma vasta quantidade de informações que são fundamentais para avançar no entendimento e tratamento de doenças.

Existem diversas bases de dados importantes, abaixo algumas das principais:

- PubChem: É uma base de dados mantida pelo National Center for Biotechnology Information (NCBI) que armazena informações sobre compostos químicos, incluindo estruturas, propriedades químicas, atividades biológicas e referências bibliográficas (Sunghwan *et al*., 2016).
- ChEMBL: É uma base de dados que se concentra em dados de atividade biológica para compostos químicos, oferecendo informações sobre alvos moleculares, bioensaios, atividades biológicas e referências (Anna *et al*., 2017)
- DrugBank: É uma base de dados que contém informações detalhadas sobre medicamentos, incluindo suas estruturas químicas, mecanismos de ação, interações, propriedades farmacocinéticas, indicações clínicas e referências bibliográficas (David *et al*., 2018).
- ChemSpider: É uma base de dados de propriedades químicas mantida pela Royal Society of Chemistry, que contém informações sobre milhões de compostos químicos, incluindo estruturas, propriedades físicas e químicas, identificadores e referências cruzadas para literatura científica (Williams *et al*., 2010).
- ZINC: É uma base de dados fundamental para a triagem virtual e descoberta de novos medicamentos (Irwin e Shoichet, 2005), oferecendo uma vasta quantidade de informações sobre compostos comercialmente disponíveis. O banco de dados é especialmente útil para biólogos estruturais e químicos medicinais, permitindo triagem rápida e eficiente de hipóteses. As informações dos compostos estão disponíveis em vários formatos, como SMILES, mol2, SDF 3D, entre outros, facilitando a

integração com diferentes ferramentas de triagem virtual e *docking* molecular (técnica computacional utilizada para prever a orientação preferencial de uma molécula, geralmente uma pequena molécula chamada ligante, quando ligada a uma segunda molécula geralmente uma proteína, chamada receptor).

Portanto, é possível identificar características especificas que diferenciam cada uma das bases de dados abertas. Para o propósito deste projeto foi utilizada a base de dados DrugBank como fonte principal, e as outras como apoio.

2.3.1 DrugBank

O DrugBank surgiu no ano de 2006 como uma plataforma digital com um banco de dados habilitado para a web que contém informações moleculares abrangentes sobre drogas, os mecanismos, as interações e respectivos alvos. O DrugBank contém informações detalhadas sobre 1.480 medicamentos aprovados pela FDA dos EUA, abrangendo 28.447 nomes de marcas e sinônimos (Wishart, 2008). Esta coleção inclui 1.281 medicamentos sintéticos de pequenas moléculas, 128 medicamentos biotecnológicos e 71 suplementos ou medicamentos nutracêuticos (substâncias derivadas de alimentos que têm efeitos positivos na saúde além do seu valor nutricional básico). Além disso, dispõe de dados sobre 1.669 alvos diferentes (moléculas de proteínas, lipídios ou DNA) e enzimas metabolizadoras com as quais esses medicamentos interagem. A base de dados também cobre 187 drogas ilícitas (aquelas legalmente proibidas ou seletivamente proibidas na maioria dos países desenvolvidos) e 64 drogas retiradas do mercado devido a questões de segurança.

As informações químicas, farmacêuticas e biológicas contidas no DrugBank são essenciais não só para a compreensão das reações adversas dos medicamentos, mas também para prever se uma nova entidade medicamentosa pode apresentar semelhanças químicas ou funcionais inesperadas com uma droga perigosa ou altamente viciante. A base de dados é manualmente curada por uma equipe de bioinformáticos, farmacêuticos, bioquímicos e médicos. Atualizações do DrugBank são normalmente lançadas a cada seis meses, incluindo informações sobre medicamentos recentemente aprovados, correções ou

atualizações de medicamentos antigos, a adição de novos campos de dados e melhorias na interface ou nos utilitários de pesquisa.

Atualmente, estão em andamento esforços para expandir as capacidades de consulta da base de dados (com pesquisas de estrutura aprimorada), aumentar a cobertura de nutracêuticos ou medicamentos fitoterápicos, incluir plug-ins para facilitar a triagem virtual de drogas e a modelagem farmacológica (absorção, distribuição, excreção, metabolismo e toxicidade).

2.3.2 SMILES (*Simplified Molecular-Input Line-Entry System*)

Método de notação que permite representar a estrutura química de moléculas utilizando uma linha de texto ASCII (Weininger, 1988). Esse sistema é amplamente utilizado em química computacional e bioinformática para a entrada e manipulação de dados químicos.

Weininger (1989), Heller *et al*. (2015) e Rogers *et al*.(2010) descrevem respectivamente o algoritmo para gerar notações SMILES únicas, garantindo que cada estrutura molecular tenha uma representação única, discute o InChI, um identificador químico internacional desenvolvido pela IUPAC, e sua relação com o SMILES e explora o uso de *fingerprints* de conectividade estendida (ECFP), que podem ser derivados de SMILES, para aplicações em quimioinformática.

O uso de SMILES facilita a troca de informações químicas e o processamento computacional de dados químicos, sendo uma ferramenta essencial na bioinformática, descoberta de medicamentos e química computacional. A Tabela 2 apresenta exemplos de SMILES, como o metano ("C"), etanol ("CCO") e benzeno ("C1=CC=CC=C1"), além de destacar suas vantagens, como ser uma notação compacta, fácil de interpretar por sistemas computacionais, e amplamente aceita em bancos de dados e softwares de química computacional. Ferramentas como RDKit[1], Open Babel, ChemDraw e ChemSketch suportam SMILES, permitindo a conversão, visualização e análise de moléculas de forma eficiente.

[1] RDKit: https://www.rdkit.org, Open Babel: https://openbabel.org , ChemDraw: https://revvitysignals.com/products/research/chemdraw
ChemSketch: https://www.acdlabs.com/products/chemsketch

Tabela 2. - Exemplos, vantagens e ferramentas relacionadas ao SMILES

Categoria	Descrição
Exemplos de SMILES	
Metano	"C"
Etanol	"CCO"
Benzeno	"C1=CC=CC=C1"
Ácido acético	"CC(=O)O"
Glucose	"C(C1C(C(C(O1)O)O)O)O"
Vantagens do SMILES	
Compacto e Eficiente	A notação SMILES é compacta, o que facilita o armazenamento e a manipulação de grandes conjuntos de dados químicos.
Facilidade de Interpretação	Os sistemas computacionais podem facilmente converter SMILES em representações gráficas ou outras formas de dados químicos.
Amplamente Aceito	SMILES é suportado por muitos softwares e bancos de dados químicos, tornando-o um padrão de fato na química computacional.
Ferramentas Relacionadas	
RDKit	Biblioteca de código aberto para química computacional que suporta SMILES e outras representações moleculares.
Open Babel	Ferramenta de química computacional que permite a conversão entre diferentes formatos de representação química, incluindo SMILES.
ChemDraw	Software de desenho químico que pode interpretar e gerar SMILES para representar estruturas moleculares.
PubChem	Banco de dados de compostos químicos que usa SMILES para representar as estruturas dos compostos disponíveis em sua base de dados.
ChemSketch	Software para desenhar estruturas químicas e gerar SMILES a partir dessas estruturas.

3 Revisão da Literatura e Trabalhos Relacionados

O desenvolvimento de produtos farmacêuticos envolve tempo e um alto custo financeiro, e uma fase crítica nesse processo é a formulação (Yang *et al*., 2019). As formulações farmacêuticas envolvem um grande conjunto de fatores, e a abordagem tradicional para desenvolver essas formulações depende de processos incertos baseados em tentativa e erro. Esse processo requer um grande número de experimentos *in vitro* e *in vivo*, que consomem muitos recursos e tempo (Bannigan *et al*., 2021). Outro fator chave na indústria farmacêutica é o rápido crescimento dos avanços tecnológicos, incluindo o aumento da quantidade de dados científicos disponíveis, o que resulta em vários desafios computacionais, como aqueles ligados ao armazenamento e análise de dados (Kamerzell e Middaugh, 2021)

Nesse cenário, o ML tem se destacado como uma ferramenta para otimizar e acelerar o desenvolvimento de formulações farmacêuticas. Por exemplo, técnicas de ML podem ser usadas para analisar informações complexas sobre as propriedades físico-químicas de ingredientes ativos e excipientes, bem como dados de estabilidade e eficácia de medicamentos existentes. O ML permitiu várias melhorias no domínio da saúde, por exemplo, no diagnóstico de doenças (Dametto *et al*., 2022), na previsão de estruturas de proteínas e na identificação de novos medicamentos (Bannigan *et al*., 2021).

A falta de revisões da literatura focadas na relação do uso de ML e suas aplicações, bem como o potencial e rápido desenvolvimento do uso de ML para apoiar a formulação de medicamentos, motivou a conduzir esta revisão da literatura. Os principais objetivos e contribuições esperadas da revisão realizada são: (1) identificar as técnicas de ML e sua relação com grupos de aplicações no contexto de formulações farmacêuticas; e (2) destacar os aspectos positivos e limitações dos estudos analisados, apontando lacunas na literatura e desafios de pesquisa atuais e futuros.

3.1 Trabalhos relacionados a revisão da literatura e diferencial desta revisão

Existem diversas revisões da literatura sobre a aplicação de ML e DL na saúde, e muitas revisões sobre tecnologias e métodos para formulação farmacêutica e descoberta de medicamentos (por exemplo, Shetti *et al*. (2022)). Esta seção é focada em apresentar e

analisar revisões sobre o uso de ML ou DL na formulação de medicamentos e fases relacionadas.

Bannigan *et al*. (2021) apresentaram uma revisão da literatura sobre os métodos e ferramentas aplicados ao desenvolvimento de formulações químicas usando ML. Segundo os autores, a formulação de medicamentos orientada por ML oferece oportunidades para acelerar os esforços de desenvolvimento, descobrir novos materiais e formulações inovadoras, e gerar novos conhecimentos na ciência da formulação de medicamentos. A revisão enfatiza as mais recentes tecnologias de *IA*, incluindo modelos generativos, DL, modelos *bayesianos*, aprendizado por reforço e laboratórios autônomos.

Puranik *et al*. (2022) apresentaram uma revisão da literatura com o objetivo de analisar o potencial do ML em várias áreas do desenvolvimento biofarmacêutico. O foco deles foi nos desenvolvimentos e aplicações de ML nos desafios relacionados à adoção da Indústria 4.0 na indústria biofarmacêutica.

Wang *et al*. (2021) propõe o uso de algoritmos de IA e ML em conjunto com modelagem molecular, modelagem matemática, simulação de processos e modelagem farmacocinética baseada em fisiologia para aprimorar o desenvolvimento da técnica de distribuição de fármacos.

Outras revisões focam em diferentes fases e abordagens específicas na formulação de medicamentos. Martinelli (2022), por exemplo, focou no uso de ML generativo para descoberta de novos medicamentos, ou seja, no design de novas entidades químicas que se encaixam em um conjunto de restrições usando algoritmos computacionais.

Gholipour e Bastas (2023) revisaram o estado da arte sobre o uso de redes neurais na fabricação farmacêutica. Os estudos selecionados foram classificados em análise e melhoria de processos, controle de qualidade e fabricação aditiva.

Esta revisão se difere das outras ao fornecer uma nova visão focada no uso de técnicas de ML e sua aplicação em três grupos: distribuição de fármacos, desenvolvimento de proteínas e formulações.

3.2 Metodologia e Execução da Revisão

Esta revisão é metodologicamente baseada nas diretrizes apresentadas por Kitchenham (2004) e no protocolo PRISMA (Page *et al*., 2021).

Uma pesquisa exploratória preliminar foi realizada para obter os insumos necessários para esta revisão. Isso resultou na definição dos parâmetros de pesquisa, período de cobertura da pesquisa, bases de dados científicas, palavras-chave a serem usadas e área de busca nos artigos. Foram utilizadas as seguintes bases de dados na pesquisa exploratória: Google Scholar, IEEE Xplore, Springer Link, ACM DL e PubMed[2].

A partir dessa pesquisa exploratória, uma Questão Principal da revisão (MQ) foi definida: "Quais são as abordagens, aplicações, técnicas, limitações e desafios ao usar ML para apoiar a seleção e desenvolvimento de formulações farmacêuticas?"

Como focamos em um tópico de pesquisa recente, o período de busca foi definido de 2019 a 2023. As seguintes bases de dados foram escolhidas porque fornecem resultados relevantes e baixa sobreposição na revisão exploratória preliminar: PubMed, Springer Link e IEEE Xplore. A seguinte *string* de busca foi usada (adaptada à sintaxe de cada base de dados): *("formulation") AND ("machine learning") AND ("pharmaceutical") AND ("predicting" OR "prediction")*.

Os critérios de inclusão e exclusão detalhados na Tabela 3, foram definidos por três pesquisadores da área de pesquisa em computação, em um processo iterativo de leitura de artigos e proposição de critérios até alcançar um consenso.

A pesquisa inicial resultou em um total de 71 artigos, distribuídos da seguinte forma: 2 artigos encontrados no IEEE Xplore, 22 no Springer Link e 47 no PubMed.

Tabela 3 – Critérios de inclusão e exclusão de artigos

Tipo	Sigla	Critério
Inclusão	I1	Pesquisas sobre Machine Learning , Farmacêutica e Predição.
	I2	Estudos que utilizam aprendizado de maquina em formulações farmacêuticas.
Exclusão	E1	Artigos escritos em idiomas diferentes do Inglês e do Português.
	E2	Artigos que não estejam relacionados com Machine Learning e formulações farmacêuticas.
	E3	Artigos que não sejam da área de computação ou multidisciplinar com computação.
	E4	Textos que não sejam publicações científicas.
	E5	Resumos com menos de 4 páginas e que não tenham profundidade ou resultados relevantes.
	E6	Revisões sistemáticas e Livros.

[2] SpringerLink**:** https://link.springer.com/, ACM Digital Library**:** https://dl.acm.org/
PubMed**:** https://pubmed.ncbi.nlm.nih.gov/ , Google Scholar: https://scholar.google.com/
IEEE Xplore: https://ieeexplore.ieee.org/

No diagrama da Figura 3 são apresentados o quantitativo dos registros (que podem incluir metadados de livros, relatórios técnicos, artigos e outros, de acordo com o PRISMA) retornados das buscas nas bases científicas supracitadas durante a fase de identificação. Os registros duplicados foram excluídos durante a identificação resultando em 70 registros. Os registros foram filtrados na fase de triagem, eliminando os de natureza diversa e que não continham versão completa disponível. Com isso foram recuperados 65 documentos completos, identificados como relatórios (documentos científicos sob triagem).

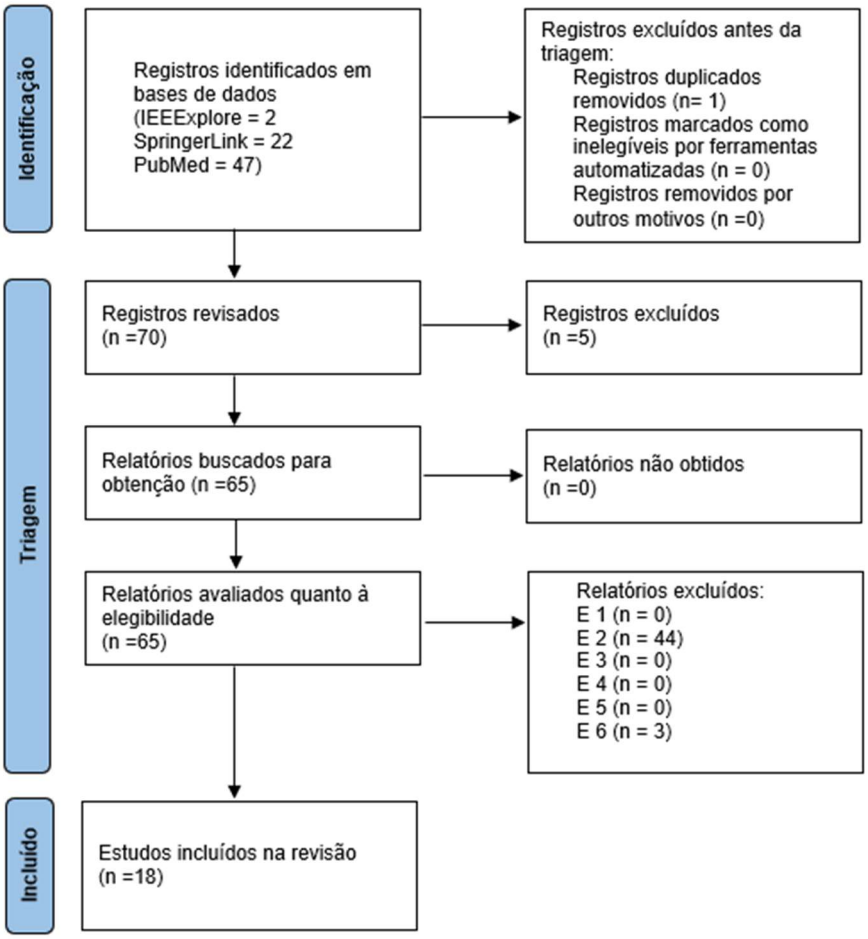

Figura 3-Diagrama Prisma com resultados quantitativos

Durante a fase de triagem (Figura 3), foram aplicados os critérios de inclusão e exclusão (Tabela 3) aos relatórios, sendo 44 relatórios excluídos pelo critério E2 e 3 pelo critério E6. Dois pesquisadores de computação e um pesquisador de biomedicina analisaram os artigos selecionados e criaram a lista final por consenso após discussões, resultando assim em 18 estudos incluídos e analisados.

3.3 Análise e Discussão dos Resultados

A Tabela 4 apresenta uma visão geral da classificação dos 18 artigos de acordo com suas técnicas e aplicações. As siglas referentes às técnicas de ML, usadas no restante deste artigo, também são definidas na Tabela 4. Esta seção está estruturada da seguinte forma: A Seção 3.3.1 apresenta soluções baseadas em ML para otimizar a distribuição de fármacos; A Seção 3.3.2 apresenta soluções baseadas em ML para o desenvolvimento de proteínas; e a Seção 3.3.3 apresenta o restante dos artigos, que usam o ML como ferramenta para otimizar a formulação farmacêutica.

Tabela 4 - Resumo das aplicações de ML e técnicas utilizadas nos trabalhos selecionados

Referencia	Aplicação				Principais técnicas de ML
	M	D	P	F	
Deng *et al.*,2023	X	X			XGB,RNA,DT,PLS DNN,KNN,SVM,RF LGBM,MLR,RN,RR
Lou & Hageman,2021	X	X			AB,RF,DT,RNA GNB,KNN
He *et al.*,2020	X	X			LGBM,RF,KNN,PLS DNN,SVM,DT,MLR
Gao *et al.*,2021	X	X			RF, KNN, AD, NB, SVM, LGBM, XGB
Noorain *et al.*,2023	X	X			GP
Damiati & Damiati,2021	X	X			RNA
Gentiluomo *et al.*,2019	X		X		RNA
Gentiluomo *et al.*,2020	X		X		RNA
Lai *et al.*,2022	X		X		LR,SVM,KNN,DT
Kamerzell & Middaugh,2021	X		X		KNN,DT,LR,NB,RNA,RF,SVN...
Zhao *et al.*,2019	X			X	DNN, LGBM, RF
Schmitt *et al.*,2022	X			X	PLS,R,LRG KR,XGB,RNA
Patel *et al.*,2023	X			X	RNA
Gao *et al.*,2021	X			X	LGBM
Dong *et al.*,2021	X			X	RF, SVM, LGBM, XGB, SVM
Glišić *et al.*,2023	X			X	AB, NB, RL
Yang *et al.*,2019	X			X	MLR,PLSR,SVM RNA,KNN,DNN
Yoo *et al.*,2023	X			X	CNN

Aplicações: M: Aprendizado de Máquina; D: Distribuição de fármacos; P: Desenvolvimento de Proteínas; F:Formulações.

Técnicas de ML: AB: AdaBoost; RNA: Rede Neural Artificial; CNN: Rede Neural Convolucional; DNN: Rede Neural Profunda; DT: Árvore de Decisão; GNB: Gaussian Naive Bayes; GP: Processo Gaussiano; KNN: K-Nearest Neighbors; KR: Regressão Kernel Ridge; LGBM: Máquina de Gradiente LightBoost; LRG: Regressão Linear Ridge; LR: Regressão Logística; LRM: Regressão Linear Múltipla; NB: Naive

Bayes; PLS: Mínimos Quadrados Parciais; RF: Random Forest; RN: ResNet DNN; RR: Regressão Ridge; SVM: Máquina de Vetores de Suporte; SVR: Regressão de Vetores de Suporte; XGB: eXtreme Gradient Boosting.

3.3.1 Soluções baseadas em aprendizado de máquina para otimizar o desenvolvimento de distribuição de fármacos

Após analisar o texto completo dos 18 artigos incluídos nesta revisão, 6 foram selecionados para serem apresentados nesta subseção. Deng *et al.* (2023) apresentaram um modelo de previsão para acelerar o desenvolvimento de produtos à base de microesferas para medicamentos de moléculas pequenas por meio de técnicas de ML. Para tanto, foram utilizadas as seguintes técnicas de ML: XGB, RNA, DT, PLS, DNN, KNN, SVM, RF, LGBM, MLR, RN e RR.

Lou e Hageman (2021) usaram as técnicas AB, RF, DT, RNA, GNB e KNN para prever a biodisponibilidade, após administração subcutânea de anticorpos monoclonais, mesmo sem entender completamente o mecanismo e a causalidade entre entradas e saídas.

He *et al.* (2020) apresentaram o uso de técnicas LGBM, RF, KNN, PLS, DNN, SVM, DT e MLR para o desenvolvimento de formulações de nanocristais substituindo processos de tentativa e erro que consomem tempo e recursos.

Gao *et al.* (2021) propôs o uso de uma metodologia computacional integrada com o objetivo de reduzir o trabalho tradicional de design de formulações de medicamentos e trazer novas ideias para futuros projetos de formulação, modelagem molecular e abordagens experimentais para o design racional de formulações SEDDS (*Self-Emulsifying Drug Delivery Systems*). Para tanto, foi utilizada a técnica de LGBM.

Noorain *et al.* (2023) abordaram a combinação nano terapêutica e técnica de GP para simplificar os sistemas de desenvolvimento de medicamentos antivirais automatizando a análise.

Por fim, Damiati e Damiati (2021) utilizaram a combinação das técnicas microfluídicas e LR, SVM, KNN e DT, juntamente com o uso de biomateriais, a fim de gerar micropartículas poliméricas carregadas com medicamentos.

3.3.2 Soluções baseadas em aprendizado de máquina para desenvolvimento de proteínas

Após a leitura de todos os artigos na íntegra, 4 trabalhos foram selecionados para serem apresentados nesta subseção. O rápido crescimento dos avanços tecnológicos e a quantidade de dados científicos na última década levaram a vários desafios, incluindo armazenamento e análise de dados. Isso se reflete no estudo e desenvolvimento de modelos precisos de conjuntos de dados complexos de proteínas.

Kamerzell e Middaugh (2021) utilizou vários algoritmos ML populares (por exemplo, KNN, DT, LR, NB, RNA, RF e SVN) focados no desenvolvimento de proteínas farmacêuticas. Os autores afirmam que os modelos ML aplicados podem ser usados para entender a viscosidade não linear dependente da concentração de soluções proteicas, prever taxas de oxidação e desamidação de proteínas, classificar partículas subvisíveis e comparar a estabilidade física.

Gentiluomo *et al.* (2019) abordaram o desenvolvimento de plataformas para a formulação de proteínas usando uma abordagem baseada na técnica RNA com o objetivo de diminuir os custos financeiros e também facilitar o desenvolvimento de medicamentos à base de proteínas.

Gentiluomo *et al.* (2020) aplicou RNAs para prever a estabilidade a longo prazo em condições reais de armazenamento a partir de estudos de estabilidade acelerada e outras propriedades biofísicas de alto rendimento, por exemplo, a primeira temperatura aparente de desdobramento.

Finalmente, Lai *et al.* (2022) usou modelos preditivos baseados em LR, SVM, KNN e DT para prever taxas de agregação de anticorpos terapêuticos e viscosidade em altas concentrações no desenvolvimento de proteínas terapêuticas.

3.3.3 Soluções baseadas em aprendizado de máquina com uma variedade de aplicações

Entre os 18 estudos avaliados, 8 trabalhos foram selecionados para serem apresentados nesta subseção.

Zhao *et al.* (2019) usaram técnicas de DNN, LGBM e RF para construir modelos preditivos para simplificar e reduzir o número de experimentos no desenvolvimento de formulações farmacêuticas para sistemas de medicamentos com ciclodextrina.

Schmitt *et al.* (2022) usaram modelos preditivos baseados em PLS, RR, LRG, KR, XGB e RNA para obter estimativas iniciais para valores de parâmetros de processo que melhor atingem um tamanho de partícula alvo para uma determinada formulação.

Patel *et al.* (2023) desenvolveram uma ferramenta preditiva para identificar incompatibilidades de excipientes de medicamentos usando modelos de RNAs. Gao *et al*. (2021) integraram ferramentas, incluindo modelagem molecular e métodos de ML, com base em um modelo LGBM, para desenvolver uma nova formulação ternária de Andrographolide-Ciclodextrinas-Tocoferol Polietileno Glicol Succinato (AG-CD-TPGS), obtendo melhorias na solubilidade aquosa, taxa de dissolução e biodisponibilidade.

Dong *et al.* (2021) apresentaram uma plataforma de previsão de formulação de dispersão sólida usando técnicas de RF, SVM, LGBM, XGB e SVM. O processamento de sistemas líquido-sólido, uma abordagem para melhorar a biodisponibilidade oral de medicamentos pouco solúveis, tem se mostrado desafiador devido à quantidade relativamente alta de fase líquida incorporada a eles. Glišić *et al*. (2023) utilizaram ferramentas baseadas nas técnicas AB, NB e LR, para melhor compreender os efeitos dos fatores de formulação e dos parâmetros do processo de compressão nas propriedades de fluidez e compressão do sistema líquido-sólido. Yang *et al*. (2019) usaram as técnicas de MLR, PLSR, SVM, RNA, KNN e DNN para prever formulações farmacêuticas. Os resultados mostram que as precisões das redes de DL foram maiores do que outras.

O complexo fármaco-fosfolipídio é uma tecnologia de fórmula promissora para melhorar a baixa biodisponibilidade de ingredientes farmacêuticos ativos. No entanto, identificar o fosfolipídio fosfórico e o candidato a medicamento por meio de testes *in vitro* pode ser caro e demorado devido às propriedades físico-químicas e ao ambiente experimental. Nesse contexto, Yoo *et al.* (2023) usaram modelos de CNN para prever a formação do complexo fármaco-fosfolipídio.

3.3.4 Discussão sobre a Revisão da Literatura

De acordo com He *et al.* (2020), o uso de ML na criação e seleção de formulações farmacêuticas tem o potencial de impulsionar a indústria, pois tem a capacidade de acelerar um extenso processo manual de tentativa e erro, reduzindo custos e tempo. Neste capítulo, apresentamos uma revisão da literatura sobre o uso de ML na criação e seleção de formulações farmacêuticas. Este estudo identifica e analisa tendências, por exemplo, o uso de redes neurais treinadas com uma grande quantidade de dados. De um total de 71 artigos, 18 trabalhos foram selecionados, classificados e sintetizados a fim de analisar as abordagens do estado da arte, com o objetivo de avaliar o uso de ferramentas computacionais de ML e sua aplicabilidade no campo do desenvolvimento de medicamentos.

Os trabalhos analisados mostram um cenário muito promissor para a aplicação de algoritmos de ML, pois o volume de dados científicos (Kamerzell e Middaugh, 2021) está crescendo e tem possibilitado a criação de modelos preditivos mais confiáveis nos últimos anos. Em consonância com Bender *et al.* (2021), destaca-se que os estudos devem se concentrar mais na qualidade das decisões sobre qual composto levar adiante, incluindo a eficácia e segurança dos medicamentos. No que diz respeito às técnicas de ML adotadas, destacamos que a maioria dos trabalhos utilizou técnicas baseadas em redes neurais (profundas ou não), estando presente em 12 dos 18 trabalhos analisados. Esta revisão também destaca que vários trabalhos experimentaram muitas técnicas, arquiteturas e parâmetros diferentes, revelando que não há uma solução dominante ou previamente estabelecida para todos os casos. As pesquisas analisadas também apontam que os modelos preditivos devem ser utilizados de forma criteriosa e complementar à experimentação oratória laboratorial, pois as validações experimentais são essenciais para confirmar as previsões do modelo, bem como para garantir a qualidade das formulações. A interação de ML e cientistas farmacêuticos é essencial para explorar plenamente o potencial dessa abordagem e garantir o desenvolvimento responsável de medicamentos, considerando questões éticas e de segurança relacionadas à privacidade de dados e ao uso responsável da tecnologia. Ao mesmo tempo em que os artigos analisados apontam para o uso de diversas técnicas para diferentes aplicações com bons resultados, destacamos

também desafios relacionados à necessidade de criação de bancos de dados integrados. O uso de novas técnicas generativas e de DL também está na agenda de pesquisas futuras.

3.4 Trabalhos Relacionados e Diferencial da Pesquisa

Após a análise crítica dos estudos selecionados, identificou-se que cinco deles possuem propósitos semelhantes aos deste projeto. Sendo esses classificados como trabalhos relacionados, esta seção os apresenta com mais detalhes e compara com esta dissertação.

Deng *et al.* (2023) utilizaram um conjunto de técnicas de ML para prever e acelerar o desenvolvimento de microesferas para medicamentos de moléculas pequenas e obteve um resultado satisfatório em suas métricas de avaliação com coeficientes de determinação (R^2) de *0,880* e *0,958*, indicando um bom ajuste do modelo preditivo aos dados. No entanto, a pesquisa de Deng *et al.* (2023) focava somente neste tipo de aplicação.

Lou e Hageman (2021) utilizaram métodos baseados em árvore, incluindo RF, AB e DT que apresentaram melhor previsibilidade e poder de generalização na classificação de biodisponibilidade com acurácia de *0,780*. Além disso, foram investigados modelos baseados em algoritmos MLP, GNB e KNN que também forneceram precisão de predição aceitável com acurácia de *0,670*. O propósito do estudo foi prever a biodisponibilidade subcutânea humana de anticorpos monoclonais utilizando modelos de ML. É importante ressaltar que o estudo aborda um único parâmetro como referência na análise da distribuição de fármacos, com um número limitado de modelos de ML e também é especifico para este tipo de problema.

He *et al.* (2020) utilizaram nesta pesquisa, técnicas de ML para prever o tamanho da partícula e o índice de polidispersão de nanocristais. Os nanocristais possuem uma grande vantagem, pois aumentam a taxa de dissolução de medicamentos insolúveis em água devido ao tamanho reduzido para nanoescala e estes atributos influenciam na capacidade de absorção dos medicamentos. O estudo apresentou resultados satisfatórios com valores de MAE (Erro Absoluto Médio) de *0,298* para LightGBM e *0,299* para Random Forest, para o conjunto de modelos de ML :LGBM, RF, KNN, PLS, DNN, SVM, DT e MLR com aplicabilidade restrita a produção de nanocristais.

Em Gao *et al.* (2021) foram utilizadas as técnicas de ML: RF, KNN, DT, NB, SVM, LGBM e XGB com o propósito de facilitar o desenvolvimento de Sistemas de Administração de Fármacos Autoemulsionantes (SEDDS). Entre os modelos utilizados o modelo Random Forest (RF) apresentou desempenho superior, com *0,913* de acurácia. Na pesquisa foram utilizados 10 descritores moleculares (variáveis independentes) no intuito de prever a dosagem correta dos excipientes: óleos, surfactantes e cosurfactantes (variáveis dependentes). O propósito do estudo é especifico e trabalha com um conjunto limitado de descritores moleculares para identificar características especificas para o problema.

Damiati e Damiati (2021) utilizou a combinação de tecnologias de microfluídica e o modelo MLP, juntamente com o uso de biomateriais, para gerar micropartículas poliméricas (MPs) carregadas com o fármaco indometacina (IND), para controlar sua liberação sustentada e reduzir seus efeitos colaterais. O propósito de utilizar o modelo MLP consistiu em prever o tamanho das MPs, que é um fator-chave para manter a estabilidade das partículas e, portanto, ficou limitado a apenas um único modelo de ML para identificar a melhor predição. A capacidade preditiva deste modelo, avaliada pelo coeficiente de determinação (R^2), alcançou o valor de *0,997*, indicando um excelente ajuste entre as previsões do modelo e os dados observados.

Os estudos analisados destacam a aplicação de técnicas de ML em contextos específicos do desenvolvimento farmacêutico, tais como a previsão de tamanhos de partículas e biodisponibilidade de fármacos. Embora esses trabalhos tenham obtido resultados satisfatórios, eles se restringem a conjuntos limitados de descritores moleculares e a um número reduzido de modelos de ML, focando em parâmetros isolados ou tipos específicos de formulações.

Em contraste, o presente projeto adota uma abordagem mais abrangente no que diz respeito à distribuição de fármacos. Além disso, ela difere quanto à estratégia de codificação dos descritores moleculares e ao abordar duas variáveis dependentes (biodisponibilidade e LogP). Espera-se que com a abordagem proposta seja possível superar as principais limitações observadas nos estudos anteriores, oferecendo um modelo preditivo mais robusto e generalizável, e oferecer base para análise futura de outras variáveis relacionadas à distribuição de fármacos.

4 O Uso de Aprendizado de Máquina no Apoio à Predição de Distribuição de Fármacos

Conforme apresentado no capítulo anterior, o uso de aprendizado de máquina tem se mostrado uma ferramenta valiosa na predição de características farmacológicas, incluindo a distribuição de fármacos. Esta seção está estruturada da seguinte forma: seção 4.1apresenta uma análise exploratória sobre os dados; seção 4.2 apresenta a etapa de extração dos atributos LogP, Biodisponibilidade e SMILES; seção 4.3 apresenta uma forma de padronização dos dados; seção 4.4 apresenta a técnica de codificação do SMILES; seção 4.5 apresenta a implementação e validação dos modelos de ML e DL e na seção 4.6 são apresentadas as técnicas para geração das visualizações e a análise dos resultados.

Para este estudo foi selecionado o DrugBank por sua abrangência, curadoria e por incluir informações farmacológicas, alvos biológicos e dados clínicos, comumente preferidos em tarefas de ML.

As etapas do processo, ilustradas na Figura 4, são descritas a seguir:

1. *Análise exploratória.* Esta etapa inclui a familiarização com os dados, aplicação de estatística descritiva e identificação de características relevantes, conforme detalha a seção 4.1.
2. *Extração dos atributos LogP, Biodisponibilidade e SMILES:* A DrugBank é disponibiliza por meio de um arquivo XML com toda a base em formatos diversos. Para tanto foi necessário o desenvolvimento de ferramenta de extração, conforme apresentado na seção 4.2.
3. *Padronização dos dados.* Esta etapa foca na padronização dos dados, incluindo a identificação e remoção de *outliers* (seção 4.3)
4. *Codificação do SMILES.* Esta etapa inclui a definição de uma estratégia para o a codificação do SMILES, de modo a possibilitar a construção de modelos eficientes. A seção 4.4 apresenta as alternativas estudadas, bem como a escolhida para ser aplicada nesta pesquisa.
5. *Implementação e validação dos modelos.* Esta etapa descreve como os modelos foram elaborados a partir do uso de diversas técnicas de ML e DL. A seção 4.5 detalha esta etapa.

6. *Geração das visualizações e análise dos resultados:* Nesta etapa foram avaliadas diferentes visualizações, bem como as métricas utilizadas (apresentadas na seção 4.6).

A separação do projeto em etapas tem como propósito simplificar a pesquisa e tornar possível seu desenvolvimento, cuja natureza envolve a busca pelo melhor resultado de predição com modelos de aprendizado de máquina. Inicialmente a analise exploratória permite compreender a natureza dos dados e definir os atributos chave do projeto. A etapa de extração que por sua complexidade foi destacada de forma independente, assim como a etapa de codificação, responsável por converter as representações moleculares em vetores binários, foram tratadas separadamente para evidenciar seu papel específico. As etapas subsequentes também foram apresentadas de modo individual, pois envolvem conjuntos distintos de tarefas essenciais para a solução do problema, como a padronização dos dados, a implementação e validação dos modelos, e, por fim, a apresentação dos resultados e a seleção dos melhores modelos com base em um conjunto de métricas predefinidas.

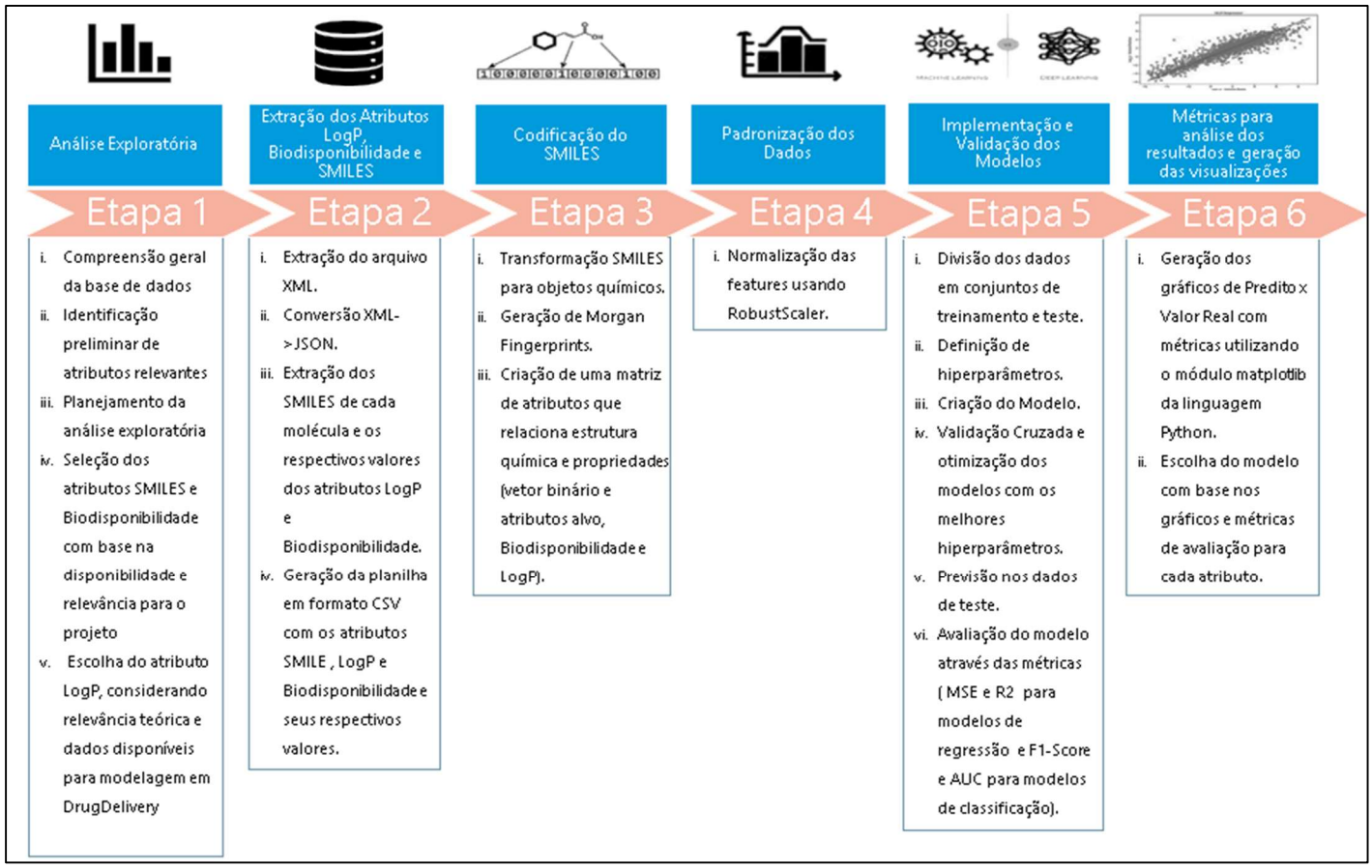

Figura 4- Etapas do processo de desenvolvimento do projeto. Fonte: autor

4.1 Análise exploratória

A análise exploratória iniciou com estudo da base DrugBank para compreensão geral dos dados disponibilizados e identificação preliminar dos atributos relevantes. Na sequência, foi identificada a necessidade de reuniões com especialistas da área. Este estudo baseia-se nas contribuições de uma especialista em biomedicina com doutorado em química computacional, e um especialista com doutorado em química e experiência profissional em desenvolvimento de medicamentos e inteligência artificial.

A Tabela 5 apresenta o registro das reuniões realizadas durante a etapa de análise exploratória, destacando os respectivos momentos, objetivos, participantes e os principais resultados obtidos que contribuíram para a definição do projeto de pesquisa. Inicialmente, em 05/02/2024, foi realizada reunião no CTI Renato Archer em Campinas, com foco na apresentação da equipe de trabalho, definição dos objetivos do projeto, definição sobre

uso do DrugBank e análise preliminar dos atributos disponíveis. Com isso foi possível obter uma compreensão geral da base de dados, identificar os atributos preliminarmente relevantes e planejar a análise exploratória.

Durante o período de 20/02/2024 a 13/06/2024 foram realizadas duas reuniões online, além de uma troca constante de comunicação via WhatsApp com o foco de discutir os atributos da base de dados, verificação da qualidade dos dados e avaliação da relevância para o contexto de DrugDelivery. Com isso, foi possível selecionar os atributos (a descrição em SMILES) e a Biodisponibilidade como atributo alvo (a ser predito).

Durante o período de 06/09/2024 a 12/09/2024 foram realizadas diversas reuniões visando o refinamento dos atributos selecionados e a verificação de variáveis alternativas com maior impacto no modelo preditivo. Como resultado temos a escolha do LogP como relevância a DrugDelivery e disponibilidade no DrugBank, sendo assim também selecionado como atributo a ser predito. Outras variáveis (ex: CYP450 2C9 e CYP450 2D6) foram consideradas relevantes, no entanto não se encontram padronizadas ou em alguns casos estão em campos textuais no DrugBank.

Tabela 5 - Registro das reuniões realizadas durante a etapa de análise exploratória

Data	Local	Atividades Executadas	Participantes	Resultados Principais
05/02/2024	Reunião presencial no CTI Renato Archer	- Apresentação da equipe de trabalho - Definição inicial dos objetivos do projeto - Definição da base de dados (DrugBank) - Análise preliminar dos atributos disponíveis	Mestrando, Orientador, Especialistas em Bioquímica e IA aplicada à Saúde	- Compreensão geral da base de dados - Identificação preliminar de atributos relevantes - Planejamento da Análise exploratória
20/02/2024 a 13/06/2024	Reuniões on-line via Google Meet e WhatsApp	- Discussão dos atributos da base de dados - Verificação da qualidade e disponibilidade dos dados - Avaliação da relevância dos atributos para o contexto de DrugDelivery	Mestrando e Especialista em Bioquímica	- Seleção dos atributos SMILES e Biodisponibilidade com base na disponibilidade e relevância para o projeto
06/09/2024 a 12/09/2024	Troca de mensagens via WhatsApp	- Refinamento dos atributos selecionados - Verificação de variáveis com maior impacto no modelo preditivo	Mestrando, Orientador, Especialista em Bioquímica	- Escolha do atributo LogP, considerando relevância teórica e dados disponíveis para modelagem em DrugDelivery

4.2 Extração dos atributos LogP, Biodisponibilidade e SMILES

O processo de extração dos atributos a serem preditos, ou seja, LogP e Biodisponibilidade e da descrição *SMILES* da molécula ocorreu por meio de algoritmos específicos implementados na linguagem Python e foi dividido em 3 etapas (Figura 5):

1. Gerar um arquivo em formato *JSON* a partir do arquivo em formato *XML* extraído da base de dados do DrugBank no intuito de simplificar e viabilizar a extração dos valores dos atributos escolhidos (Figura 6).
2. Ler o arquivo *JSON* que contém as informações sobre as moléculas, percorrer os dados em busca de dois elementos principais: a estrutura química representada por SMILES e um atributo específico (como "Bioavailability" ou “LogP”), salvar esses pares em uma lista e depois converter em tabela usando a biblioteca pandas, permitindo visualizar e, se desejado, exportar os dados para um arquivo CSV.
3. Ler o arquivo *JSON* com dados químicos, percorrer suas estruturas procurando pares contendo a representação molecular (*SMILES*) e o valor de um atributo específico (como "Bioavailability" ou “LogP”), salvar esses pares em uma lista e depois converter em tabela usando a biblioteca pandas, permitindo visualizar e, se desejado, exportar os dados para um arquivo CSV.

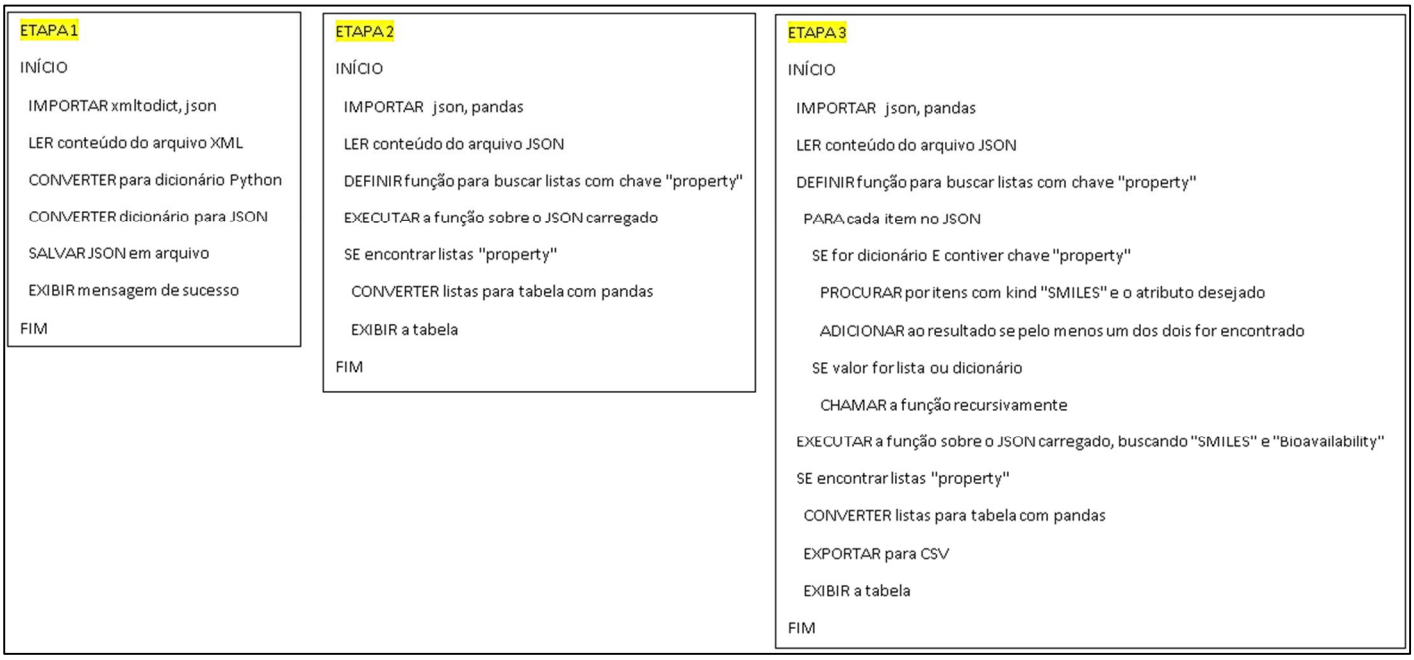

Figura 5- Etapas do processo extração dos atributos LogP, Biodisponibilidade e SMILES. Fonte: autor

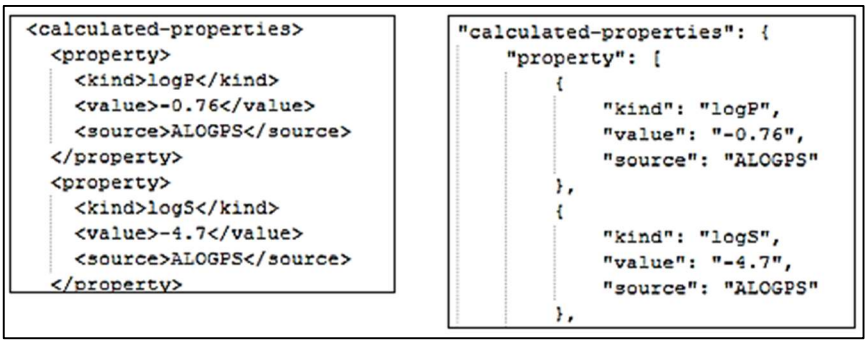

Figura 6- Arquivo *XML* a esquerda da figura(origem) e arquivo *JSON* (gerado após conversão). Fonte: autor

Observa-se nos algoritmos descritos nas etapas 1,2 e 3 que existe uma similaridade na etapa 2 e 3 que pode ser compreendida como a sequência para a extração do atributo ("Bioavailability" ou “LogP”) e seu respectivo valor. Na prática, as etapas 2 e 3 ocorreram de forma complementar durante o processo de extração desses atributos, mas foram apresentadas separadamente com propósito didático e para facilitar futuras implementações com outros tipos de atributos.

Na implementação dos algoritmos na linguagem Python, fez-se uso de bibliotecas especificas para:

- **Conversão XML para JSON :**
  - Xmltodict[3]: Biblioteca para conversão dos dados em XML para o formato de uma estrutura de dicionário (dict)**.**
  - Json[4]: Biblioteca para trabalhar com dados no formato JSON, que permite converter objetos Python para strings JSON e vice-versa.
- **Conversão de Listas para Tabelas:**
  - Pandas[5]: módulo para manipulação dos dados

4.3 Codificação do SMILES

Foram investigadas duas alternativas para a codificação do SMILES: (1) extração de descritores moleculares e (2) geração de *Morgan Fingerprints*. Nesta seção, são

[3] https://pypi.org/project/xmltodict/ ;
[4] https://docs.python.org/pt-br/3/library/json.html;
[5] https://pypi.org/project/pandas/;

apresentadas as duas alternativas de codificação, bem como a escolha da alternativa mais promissora.

Como supracitado, a abordagem inicial utilizou descritores moleculares, que são valores numéricos que representam características físico-químicas, estruturais ou topológicas de uma molécula. Eles são usados para transformar a estrutura química (como uma cadeia SMILES) em números que possam ser utilizados por algoritmos de aprendizado de máquina (Figura 7).

A Figura 8 apresenta os descritores moleculares gerados e utilizado como variáveis independentes para a predição de valores. Esta abordagem apresentou diversos problemas com relação ao valor continuo calculado para cada descritor. Por exemplo: determinados descritores não se aplicavam a certas moléculas e retornavam uma grande quantidade de valores *nan* (*not a number*) (conforme Figura 9), dificultando a execução do modelo preditivo.

Foram realizados diversos ajustes e experimentações visando explorar esta abordagem. Isso envolveu múltiplas iterações e reuniões técnicas com especialistas e resultou em sucessivas revisões do código. A Tabela 6 detalha a interação com especialistas no domínio e resultados das reuniões que exploraram essa alternativa.

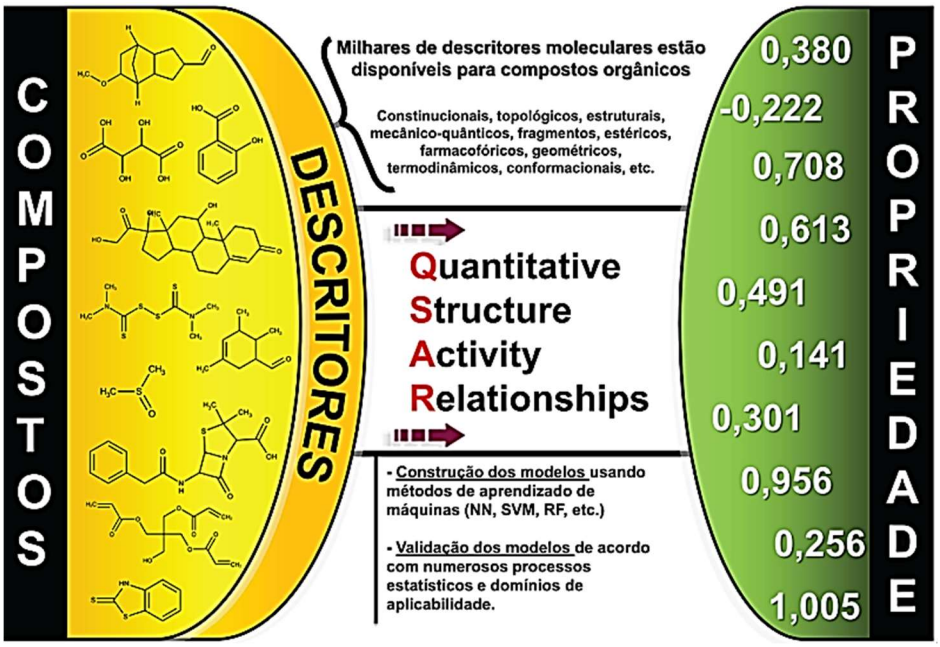

Figura 7 - Exemplo de representação de descritores moleculares utilizados em QSAR. Fonte: Adaptado de SILVA, J. R. *et al*. (2022).

```
descriptor_names = ['MolWt', 'MolLogP', 'NumHDonors', 'NumHAcceptors','BalabanJ', 'BertzCT',
                    'Chi0', 'Chi0n', 'Chi0v','EState_VSA1''Chi1', 'Chi1n', 'Chi1v', 'Chi2n', 'Chi2v',
                    'Chi3n', 'Chi3v', 'Chi4n' , 'Chi4v', 'EState_VSA1', 'EState_VSA10', 'EState_VSA11',
                    'EState_VSA2', 'EState_VSA3', 'EState_VSA4', 'EState_VSA5', 'EState_VSA6',
                    'EState_VSA7', 'EState_VSA8', 'EState_VSA9', 'ExactMolWt', 'FpDensityMorgan1',
                    'FpDensityMorgan2', 'FpDensityMorgan3', 'FractionCSP3']

calc = MoleculeDescriptors.MolecularDescriptorCalculator(descriptor_names)
```

Figura 8-Conversão dos descritores moleculares como variáveis independentes para os modelos preditivos de cada molécula representada no SMILE. Fonte: autor

```
[Sr++].[Sr++].[O-]C(=O)CN(CC([O-])=O)C1=C(C#N)C(CC([O-])=O)=C(S1)C([O-])=O
(11.05244425547997, -1.7955092592592587, 11.05244425547997, 0.0,
0.3026063522460547, 513.4930000000004, 507.4450000000004,
513.7957103519999, 122, 0, 2.0, -0.5498044794385336, 2.0,
0.5498044794385336, 0.96, 1.44, 1.76, nan, nan, nan, nan, nan, nan, nan,
nan, 0, 711.432751335566, 17.731686328639373, 12.604195711107423,
```

Figura 9-SMILES e os valores dos descritores químicos com o erro de *Not a Number* (**nan**). Fonte: autor

Data	Local	Objetivos Principais	Integrantes	Principais Resultados
09/07/2024	on-line atraves de mensagens no whatsApp	Discutir dificuldades encontradas com descritores quimicos	Mestrando, Orientador	Exclusão temporaria dos descritores que apresentavam erro de not a number'nan' para testar o modelo de regressão
12/07/2024	reunião on-line utilizando Google Meeting	Identificar os descritores que apresentavam erros e a necessidade de utiliza-los para o modelo	Mestrando, Orientador,Especialista em Bioquimica.	Identificação de descritores topologicos que possivelmente apresentavam erro pois não identificavam a caracteristica na molécula.
19/07/2024	reunião on-line utilizando Google Meeting	Identificar os descritores que apresentavam erros e a necessidade de utiliza-los para o modelo com base na planilha com o conjunto de erros identificados	Mestrando, Orientador,Especialista em Bioquimica.	Sem resultados aparentes , alguns erros não tem relação com a falta ou ausencia da caracteristica quimica.
04/09/2024	on-line atraves de mensagens no whatsApp	identificar uma solução definitiva para o erro de not a number (nan)	Mestrando, Orientador	Orientador conseguiu converter os dados com 'nan' para '0' sem prejudicar a execução do modelo utilizando um metodo especifico resolvendo o problema.
06/09/2024	reunião on-line utilizando Google Meeting	Resolver o problema das metricas com valores ruins e R^2 negativo.	Mestrando, Orientador	Mudança de abordagem ,substituindo descritores quimicos por fingerprints.

Tabela 6 - Registro das reuniões realizadas durante a etapa de Codificação do SMILES

Na sequência, foram realizadas implementações iniciais exploratórias para determinar a melhor alternativa de codificação. Os resultados preliminares utilizando técnicas como SVM, NB e MLP resultaram em valores inadequados. Por exemplo, na

Figura 10, pode-se notar o resultado para LogP, onde o R^2 Score retorna valores negativos em todos os casos.

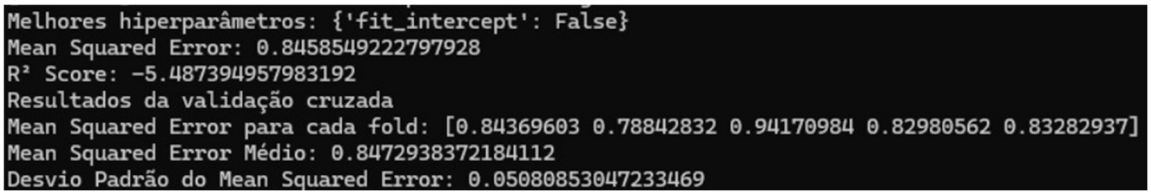

Figura 10-Métrica do modelo utilizando descritores químicos (R^2 negativo). Fonte: autor

Como apresentado anteriormente, a segunda técnica utilizada de codificação de moléculas para análise quantitativa transforma sequências SMILES em vetores numéricos, permitindo a representação das propriedades químicas para modelos de aprendizado de máquina (Alves *et al*., 2018). Por meio deste procedimento, *fingerprints* moleculares capturam padrões estruturais através de representações vetoriais binárias. Os *fingerprints* moleculares que codificam as propriedades estruturais e físico-químicas de uma molécula (e com isso permitem a comparação entre moléculas viabilizando previsões sobre seu comportamento) são ferramentas valiosas para a representação de moléculas em modelos de inteligência artificial, pois transforma estruturas químicas em vetores numéricos (He, 2022).

Os passos para codificação de moléculas aplicados foram os seguintes:

- Transformação SMILES para objetos químicos: SMILES são convertidos em objetos Mol, um formato que facilita a manipulação de informações químicas computacionais.
- Geração de *Morgan Fingerprints*: Utilizando uma função especifica do modulo RDKit[6] da linguagem Python (AllChem.GetMorganGenerator) que gera vetores que representam subestruturas ao redor de cada átomo (Laudrum, 2013). Cada bit do vetor binário representa a presença ou ausência de subestruturas específicas (utilizadas para modelagem química).
- Formato de Saída: Esses vetores binários são unidos aos atributos alvos (Biodisponibilidade e LogP), criando uma matriz de atributos que relaciona

[6] https://pypi.org/project/rdkit/

estrutura química e propriedades alvos. Isso permite análises exploratórias e preditivas com métodos de ML e DL. Inicialmente, foi analisado o atributo Biodisponibilidade, que é um atributo binário e na sequencia o LogP que é um atributo numérico contínuo (como apresentado nas próximas seções).

Essa técnica gera um vetor binário de 2048 bits identificando a presença (bit 1) ou a ausência (bit 0) de cada característica em cada molécula, que por sua vez representam as *features* de entrada utilizadas no treinamento para a predição da Biodisponibilidade e do LogP. Para isso foram utilizados dois módulos do RDKit:

- Chem**:** que lida com estruturas moleculares e operações químicas.
- ML.Descriptors: que fornece funções para calcular descritores moleculares, que são características numéricas usadas em modelos de aprendizado de máquina para prever propriedades químicas e biológicas.

Com resultados iniciais melhores, optou-se por se aprofundar nos estudos com o uso da codificação utilizando *Morgan Fingerprints*.

4.4 Padronização dos dados

A padronização de dados foi apenas necessária durante a investigação da primeira opção de codificação (descritores). Para tanto. a técnica intitulada *RobustScaler,* foi utilizada. Essa técnica ajuda o modelo a convergir mais rapidamente e evita que variáveis com escalas maiores dominem a análise (Pedregosa *et al.,* 2011). A implementação da ferramenta *RobustScaler* normalizava as *features* que são extraídas do dataset .

No caso da codificação por *Morgan Fingerprints* ela se torna desnecessária, pois as características são representadas por vetores binários. As próximas seções apresentam a implementação e análise dos resultados utilizando esta codificação.

4.5 Implementação e validação dos modelos

O desenvolvimento e a avaliação dos modelos preditivos propostos nesta pesquisa foram conduzidos de forma sistemática, seguindo um fluxo metodológico estruturado.

Visando seguir este fluxo, o desenvolvimento e a validação dos modelos preditivos foram divididos nas etapas descritas nas subseções a seguir.

4.5.1 Divisão dos dados

Inicialmente, os dados foram divididos em dois subconjuntos: treinamento/validação e teste. Essa separação é uma etapa fundamental no desenvolvimento de modelos preditivos, pois permite avaliar o desempenho do modelo em dados não vistos durante o treinamento e validação, garantindo uma estimativa mais realista da sua capacidade de generalização. A divisão foi realizada com a função `train_test_split`, disponibilizada pela biblioteca *scikit-learn*. Nesta etapa, foi especificado que 80% dos dados deveriam ser utilizados para o treinamento (`X_train`, `y_train`) e os 20% restantes para o teste (`X_test`, `y_test`) conforme Figura 11 para os atributos *LogP* e *Biodisponibilidade*. Esse particionamento foi controlado por um valor fixo de semente aleatória (`random_state=42`), garantindo a reprodutibilidade dos resultados. O conjunto de treinamento/validação foi então submetido à validação cruzada para evitar *overfitting*.

```
# Divisão dos dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Figura 11- Divisão dos dados para os Atributos logP e Bioavailability Fonte: autor

4.5.2 Definição das Técnicas Utilizadas e Definição de Valores de Hiperparâmetros

Com base nos resultados mais promissores nos artigos selecionados na revisão bibliográfica (Capítulo 3), as seguintes técnicas de ML foram selecionadas para a predição do atributo LogP: AB, DT, SVR, GP, LightGBM, RNA-MLP, RF, RL, PLS, RR e XGBoost. Para Biodisponibilidade foram avaliadas as técnicas de MLP, RF e SVM, pois essas apresentaram bons resultados em estudos preliminares com o uso de um conjunto pequeno com 150 moléculas. Além disso, foi investigado o uso de técnicas de DL para predição do LogP, sendo inicialmente avaliada redes CNN e RNN com camadas densas. Sendo que esta última obteve resultados mais promissores e foi escolhida para um estudo mais aprofundado. Para predição com o atributo-alvo Biodisponibilidade utilizando técnicas de DL, utilizou-se o modelo DNN (*Deep Neural Network* com camadas densas).

O método *GridSearchCV*[7] foi aplicado na otimização de hiperparâmetros para as seguintes técnicas de ML: RF, RNA-MLP, SVR e GPR para regressão (LogP) e RNA-MLP e SVM-SVC para classificação binária (Biodisponibilidade). A Tabela 7 apresenta uma síntese dos ajustes para as técnicas de ML, incluindo os hiperparâmetros ajustados, detalhes e implementação em Python. A seguir, são apresentados os detalhes.

Tabela 7 - Síntese de ajustes de hiperparâmetros de técnicas de ML

Modelo	Hiperparâmetros Ajustados	Detalhes/Testes
Random Forest (RF)	n_estimators: [100, 200, 300] max_depth: [10, 20, None] min_samples_split: [2, 5] min_samples_leaf: [1, 2] bootstrap: [True, False]	Define um dicionário param_grid com hiperparâmetros do RandomForestClassifier. Inclui número de árvores (n_estimators), profundidade máxima e critérios de divisão. Permite buscar combinações com ou sem bootstrap.
Rede Neural Artificial (RNA)	hidden_layer_sizes: [(80, 60), (120, 80)] activation: ['relu', 'logistic'] alpha: [0.0001, 0.001] solver: ['adam'] max_iter: [400, 600}	Define parâmetros de uma rede neural com hidden_layer_sizes pré-definidos. Configura função de ativação, regularização (alpha), solver e número máximo de iterações.
Support Vector Regressor (SVR)	C: [1, 10, 100, 500, 1000] epsilon: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] kernel: ['linear', 'rbf']	Define param_grid para Support Vector Regressor. Inclui valores de penalidade C, epsilon e tipos de kernel ('linear', 'rbf'). Serve para ajuste fino de regressão por vetores de suporte.
Gaussian Process Regressor (GPR)	kernel: - C(1.0, (1e-4, 1e1)) * RBF(1.0, (1e-4, 1e1)) - C(1.0, (1e-4, 1e1)) * Matern(nu=2.5) - Matern(nu=2.5) n_restarts_optimizer: [10, 20, 30]	Define diferentes kernels combinados (RBF e Matern) Cria uma lista kernels com funções para modelagem da covariância. Configura número de reinicializações do otimizador.
MLP - Classificação Binária	hidden_layer_sizes: - Todas combinações entre: first_layer_neurons: [80, 100, 120, 140] second_layer_neurons: [80, 100, 120, 140] solver: ['adam', 'lbfgs', 'sgd'] max_iter: [400, 600, 1000] alpha: np.arange(1e-5, 1e-3, 1e-5) random_state: [3, 5] activation: ['logistic', 'relu']	Gera combinações de tamanhos de camada escondida usando product(). Define lista hidden_layer_sizes com todas as combinações possíveis. Configura solver, iterações, alpha, random_state e ativação.
Support Vector Classifier (SVC)	C: [1, 10, 100, 500, 1000] kernel: - 'linear' - 'rbf' gamma: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] - 'poly' gamma: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] degree: [2, 3, 4]	Define listas de valores para C, gamma e degree. Configura dicionários com diferentes kernels: linear, rbf e polinomial. Permite busca combinando kernel, regularização e parâmetros específicos.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

No caso do RF, os principais hiperparâmetros ajustados (Figura 12) foram:

- *n_estimators*: número de árvores no modelo.
- *max_depth:* profundidade máxima das árvores.
- *min_samples_split:* número mínimo de amostras necessárias para dividir um nó.
- *min_samples_leaf:* número mínimo de amostras necessárias para estar em um nó folha.
- *bootstrap:* define se as amostras de treinamento são retiradas com reposição.

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
    'bootstrap': [True, False]
}

grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, scoring='accuracy', n_jobs=-1)
```

Figura 12- Parâmetros ajustados para RF no Grid Search para regressão Fonte: autor

Para as RNA-MPL, para regressão, a otimização envolveu a variação de parâmetros da arquitetura, como o número de camadas ocultas e as funções de ativação (Figura 13). Os principais hiperparâmetros ajustados foram:

- *hidden_layer_sizes***:** número e tamanho das camadas ocultas.
- *activation:* função de ativação a ser utilizada.
- *alpha:* parâmetro de regularização.
- *solver:* algoritmo de otimização a ser utilizado.
- *max_iter:* número máximo de iterações para o treinamento.

```
parameters = {
    'hidden_layer_sizes': [(80, 60), (120, 80)],
    'activation': ['relu', 'logistic'],
    'alpha': [0.0001, 0.001],
    'solver': ['adam'],
    'max_iter': [400, 600]
}

grid_search = GridSearchCV(MLPClassifier(random_state=42), parameters, cv=5, scoring='accuracy', n_jobs=-1)
```

Figura 13- Parâmetros ajustados para RNA no Grid Search para regressão Fonte: autor

No caso do SVR (Figura 14) os principais hiperparâmetros ajustados foram:

- *C*: parâmetro de regularização que controla o equilíbrio entre *bias* e *variance* (valores menores permitem maior regularização, enquanto valores maiores resultam em modelos mais ajustados aos dados de treinamento).
- *epsilon*: margem de tolerância ao erro, determinando a faixa em torno das predições onde os erros não são penalizados. Valores menores tornam o modelo mais rigoroso, enquanto valores maiores o tornam mais tolerante a pequenas imprecisões.
- *kernel*: especifica a função de *kernel* utilizada para transformar os dados em um espaço de maior dimensão. Foram testados os *kernels* 'linear', adequado para relações lineares entre os descritores, e 'rbf' (*Radial Basis Function*), apropriado para capturar relações não lineares.

```
param_grid = {
    'C': [0.1, 1, 10, 100],
    'epsilon': [0.01, 0.1, 0.5, 1],
    'kernel': ['linear', 'rbf']
}
```

Figura 14- Parâmetros ajustados para Support Vector Regressor (SVR) Fonte: autor

Os principais hiperparâmetros ajustados para o GP Regressor (Figura 15) foram:

- *kernel*: funções de *kernel* que determinam a similaridade entre os pontos no espaço dos descritores. Foram testados três tipos de *kernels*:
    - *C * RBF*: combina o *kernel* RBF (Radial Basis Function), que captura relações não lineares suaves, com um coeficiente de escala *C*, permitindo ao modelo ajustar a amplitude das variações na predição.
    - *C * Matern*: combina o *kernel* Matern, que é uma generalização do RBF, com um coeficiente de escala C. O *kernel* Matern, com *nu=2.5*, oferece maior flexibilidade ao capturar relações menos suaves e mais complexas nos dados.
    - *Matern*: *kernel* Matern isolado, sem coeficiente de escala, adequado para modelar padrões complexos com maior eficiência computacional.

- *n_restarts_optimizer*: número de reinicializações do otimizador durante o ajuste do modelo. Foram testados os valores *10, 20 e 30*, com o objetivo de encontrar uma solução globalmente melhor, evitando mínimos locais durante a otimização dos hiperparâmetros.

```
# Definir kernels para busca
kernels = [
    C(1.0, (1e-4, 1e1)) * RBF(1.0, (1e-4, 1e1)),
    C(1.0, (1e-4, 1e1)) * Matern(nu=2.5),
    Matern(nu=2.5)
]

# Definir parâmetros para busca
param_grid = {
    'kernel': kernels,
    'n_restarts_optimizer': [10, 20, 30]
}
```

Figura 15- Parâmetros ajustados para Gaussian Process Regressor Fonte: autor

No caso do RNA-MLP aplicada a classificação binária, os principais hiperparâmetros ajustados (Figura 16) foram:

- *hidden_layer_sizes*: define a quantidade de neurônios em cada camada oculta. Neste projeto, foram testadas combinações de neurônios nas duas primeiras camadas ocultas, variando entre *80* e *160* para a primeira camada, e entre *60* e *120* para a segunda.
- *solver*: método de otimização usado para o ajuste dos pesos. Os algoritmos testados foram: 'adam', 'lbfgs' e 'sgd'.
- *max_iter*: número máximo de iterações realizadas durante o treinamento, com os valores *400, 600 e 1000*.
- *alpha*: parâmetro de regularização L2, que penaliza grandes valores de peso para evitar *overfitting*. Os valores testados foram *[0.1, 0.001, 0.00001, 0.0000001, 0.0000000001]*.
- *random_state*: define a semente do gerador aleatório, garantindo a reprodutibilidade dos resultados. Foram utilizados os valores *3 e 6*.

- *activation*: função de ativação utilizada pelos neurônios. Foram avaliadas as funções 'logistic' (sigmoide) e 'relu' (reta linear retificada).

```
# Definir a grade de hiperparâmetros para MLP
first_layer_neurons = np.arange(80, 160, 20)
second_layer_neurons = np.arange(60, 120, 20)
hidden_layer_sizes = list(product(first_layer_neurons, second_layer_neurons))

parameters = {'solver': ['adam','lbfgs','sgd'], 'max_iter': [400,600,1000],
              'alpha': 10.0 ** -np.arange(1,10,2), 'hidden_layer_sizes':hidden_layer_sizes,
              'random_state':[3,6], "activation": ["logistic","relu"]}
```

Figura 16- Parâmetros ajustados para MLP com 3 camadas para classificação binaria Fonte: autor

No caso do SVM-SVC para classificação binária, os principais hiperparâmetros ajustados (Figura 17) foram:

- *C*: parâmetro de regularização que controla o equilíbrio entre maximizar a margem do hiperplano e minimizar o erro de classificação. Valores mais altos de C priorizam a minimização do erro de classificação, enquanto valores mais baixos favorecem a margem maior. Os valores testados foram [1.0, 10.0, 100.0, 500.0, 1000.0].
- *kernel*: especifica o tipo de função de *kernel* utilizada para transformar os dados em um espaço de maior dimensionalidade, onde se espera que eles se tornem mais separáveis. Foram avaliados três tipos de kernel:
- *'linear':* utiliza um hiperplano linear para separar as classes.
- *'rbf': kernel* radial baseado em funções gaussianas, adequado para dados que não são linearmente separáveis.
- *'poly': kernel* polinomial, que cria um hiperplano polinomial para a separação.
- *gamma*: parâmetro específico dos *kernels* 'rbf' e 'poly', que define a influência de um único exemplo de treinamento. Valores menores de gamma resultam em modelos mais simples, enquanto valores maiores

geram modelos mais complexos. Os valores testados foram *[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]*.

- *degree*: parâmetro específico do *kernel* polinomial, que define o grau do polinômio utilizado. Os valores testados foram *[2, 3, 4]*.

```
# Definir a grade de hiperparâmetros para SVC
xparm = [1.0,10.0,100.0,500.0,1000.0]
yparm = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
zparm = [2,3,4]
parameters=[{'C': xparm,'kernel': ['linear']},
            {'C': xparm,'kernel': ['rbf'],'gamma': yparm} ,
            {'C': xparm,'kernel': ['poly'],'gamma': yparm,'degree': zparm}
           ]
```

Figura 17- Parâmetros ajustados para SVC para classificação binaria Fonte: autor

No caso da DNN, para o atributo LogP (regressão) foi utilizado ajuste automatizado por busca bayesiana (Figura 18), os principais hiperparâmetros definidos no otimizador *BayesianOptimization* foram:

- *build_model:* função responsável pela construção do modelo de rede neural a ser ajustado. Nela são definidos os hiperparâmetros a serem explorados, como número de camadas ocultas, quantidade de neurônios por camada, taxa de aprendizado, função de ativação, entre outros. A função deve retornar um modelo *Keras* compilado, adequado ao problema proposto tanto para regressão (Figura 19), quanto para classificação (Figura 20). Essa abordagem permite testar diferentes combinações de parâmetros com o objetivo de identificar a configuração que oferece o melhor desempenho nos dados.
- *objective = 'val_loss'*: métrica utilizada como critério de desempenho para guiar a otimização. Neste caso, o objetivo foi minimizar a perda *(loss)* no conjunto de validação, buscando

modelos que tenham melhor capacidade de generalização. Essa métrica é especialmente adequada para tarefas de regressão ou classificação com desequilíbrio.

- *max_trials = 10:* define o número máximo de combinações de hiperparâmetros que serão testadas. Cada tentativa *(trial)* representa um conjunto único de valores escolhidos pelo otimizador para os hiperparâmetros definidos na função *build_model.* Um valor de 10 indica que até 10 modelos diferentes serão treinados e avaliados.
- *directory = '~/helderDL/DNN/Few':* caminho do diretório onde serão armazenados os resultados do processo de *tuning,* incluindo logs, pesos dos modelos treinados, histórico de métricas e o melhor modelo encontrado.
- *project_name = 'ANN_hyperparam_tuning':* nome do projeto associado à execução do *tuning.* Serve para organizar os experimentos dentro do diretório especificado, permitindo que diferentes execuções sejam mantidas separadas**.**

```
#  🔹 Criar o otimizador de hiperparâmetros
tuner = kt.BayesianOptimization(
    build_model,
    objective='val_loss',
    max_trials=10,  # Número de modelos a testar
    directory='~/helderDL/DNN/Few',
    project_name='ANN_hyperparam_tuning'
)
```

Figura 18- Hiperparâmetros utilizados para a técnica de DL, com camadas densas. Fonte:autor

```python
# Função para construção do modelo com hiperparâmetros variáveis
def build_model(hp):
    model = keras.Sequential()
    model.add(keras.layers.Dense(hp.Int('units_1', min_value=64, max_value=256, step=32),
    activation='relu',input_shape=(X_train.shape[1],)))

    # Adicionar camadas ocultas variáveis (1 a 3 camadas)
    for i in range(hp.Int('num_layers', 1, 3)):
        model.add(keras.layers.Dense(hp.Int(f'units_{i+2}', min_value=32, max_value=128, step=32),
        activation='relu'))

    model.add(keras.layers.Dense(1))  # Saída única para regressão

    # Escolher taxa de aprendizado ideal
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=hp.Choice('learning_rate', [0.01, 0.001, 0.0001])),
    loss='mse',metrics=['mae'])
    return model
```

Figura 19- Função build_model com código de ajuste de camadas e taxa de aprendizado para regressão.

Fonte:autor

```python
def build_model(hp):
    model = keras.Sequential()
    model.add(keras.layers.Dense(hp.Int('units_1', 64, 256, step=32),
        activation='relu',
        input_shape=(X_train.shape[1],)))

    for i in range(hp.Int('num_layers', 1, 3)):
        model.add(keras.layers.Dense(hp.Int(f'units_{i+2}', 32, 128, step=32),
        activation='relu'))

    model.add(keras.layers.Dense(1, activation='sigmoid'))  # Saída binária

    model.compile(optimizer=keras.optimizers.Adam(
            learning_rate=hp.Choice('learning_rate', [0.01, 0.001, 0.0001])),
            loss='binary_crossentropy',metrics=['accuracy', keras.metrics.AUC(name='auc')])
    return model

tuner = kt.BayesianOptimization(
    build_model,
    objective='val_accuracy',
    max_trials=10,
    directory='~/helderDL/DNN/Few',
    project_name='ANN_biodisp_classification'
)
```

Figura 20- Função *build_model* com código de ajuste de camadas e taxa de aprendizado para classificação.

Fonte:autor

4.5.3 Validação Cruzada

Foram utilizadas abordagens distintas de validação cruzada para tarefas de regressão e classificação, considerando a natureza específica de cada tipo de problema. Para a regressão a técnica de validação cruzada com cinco partições (*5-fold cross-validation*) foi empregada, utilizando como métrica o erro quadrático médio negativo (*neg_ mean_squared_error*), conforme ilustra a Figura 21. A métrica, oferecida pela função *cross_val_score* da biblioteca *scikit-learn* retorna os valores negativos do erro quadrático médio como convenção, sendo posteriormente convertidos em valores absolutos para interpretação. O resultado final foi calculado como a média dos erros obtidos em cada uma das cinco partições, refletindo a capacidade do modelo de estimar valores contínuos com mínima variação entre os *folds*. Essa abordagem permitiu avaliar a consistência do desempenho do modelo ao longo de diferentes subconjuntos de dados.

Para o problema de classificação foi utilizada a técnica de validação cruzada com busca em grade de hiperparâmetros (*Grid Search with Cross-Validation*), também com cinco partições (Figura 22). A estratégia consistiu na aplicação do *GridSearchCV*, combinando diferentes valores de parâmetros do modelo, avaliados com base na métrica de acurácia. O uso de *cv=5* garante que cada conjunto de treinamento seja avaliado cinco vezes, reduzindo a possibilidade de *overfitting* durante o processo de ajuste fino. O melhor conjunto de parâmetros identificado pela maior média de acurácia nos *folds* foi então utilizado para reconfigurar o modelo final a ser treinado.

```
# Validação cruzada com 5 folds
cv_scores = cross_val_score(mlp_model, X_train,
y_train, cv=5, scoring='neg_mean_squared_error')
mean_cv_score = np.mean(np.abs(cv_scores))
```

Figura 21- Exemplo de validação cruzada com 5 folds para modelo de regressão. Fonte: autor

```
# Realizar a busca com validação cruzada
grid_search = GridSearchCV(mlp_model, param_grid=parameters,
cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
mlp_model = grid_search
```

Figura 22- Exemplo de validação cruzada integrada a busca em grade de hiperparâmetros com 5 folds para modelo de classificação. Fonte: autor

4.5.4 Predição

A fase de predição visou verificar a capacidade do modelo treinado em generalizar padrões a partir de novos dados. Para ambos os casos, classificação e regressão, a função *predict()* da biblioteca *scikit-learn* (Figura 23) foi empregada para gerar as predições do modelo sobre o conjunto de teste (*X_test*).

```
# Previsão nos dados de teste
y_pred = mlp_model.predict(X_test)
```

Figura 23- Criação do modelo para o atributo de regressão LogP. Fonte: autor

4.6 Métricas para análise dos resultados e geração das visualizações

A avaliação dos modelos desenvolvidos nesta pesquisa foi conduzida com base em métricas específicas para as tarefas de regressão e classificação, permitindo uma análise quantitativa do desempenho obtido em cada abordagem. A escolha das métricas considerou tanto a natureza do problema, quanto a necessidade de interpretação objetiva dos resultados.

No caso do modelo de regressão, foram utilizados quatro métricas: o erro quadrático médio (*Mean Squared Error - MSE*), o erro médio absoluto (*Mean Absolute Error - MAE*), a raiz do erro quadrático médio (*Root Mean Squared Error - RMSE*) e o coeficiente de determinação (R^2).

As métricas foram calculadas utilizando os valores previstos (*y_pred*) e os valores reais do conjunto de teste (*y_test*). O uso combinado dessas métricas permite uma

avaliação mais abrangente da qualidade das predições, fornecendo subsídios para a identificação de possíveis limitações do modelo, como tendência a subestimar ou superestimar determinados valores.

Para o modelo de classificação, após o processo de treinamento, o desempenho preditivo foi avaliado por meio das métricas Acurácia (*Accuracy*), F1-Score e Área sob a Curva ROC *(AUC)*, amplamente reconhecidas na literatura de aprendizado de máquina para problemas de classificação. Essas métricas fornecem uma visão abrangente da qualidade do modelo, considerando tanto a taxa global de acerto quanto o equilíbrio entre precisão e revocação. Adicionalmente, foi gerada a matriz de confusão, que fornece uma visão detalhada das previsões corretas e incorretas, permitindo identificar padrões de erro e possíveis vieses do modelo.

Na sequência, foram geradas visualizações e análises descritivas a fim de interpretar o desempenho dos modelos treinados, tanto para a tarefa de classificação (*bioviabilidade*) quanto para a de regressão (*LogP*). No caso da tarefa de regressão para predição do valor de *LogP*, foi empregada uma visualização por meio de um gráfico de dispersão, no qual os valores reais foram plotados no *eixo x* e os valores previstos no *eixo y* (Figura 24). Esse tipo de gráfico é particularmente útil para observar o alinhamento entre predições e realidade: quanto mais próximos os pontos estiverem da linha de identidade (traçada em vermelho), melhor é o desempenho do modelo.

A visualização evidencia a capacidade do modelo em capturar a tendência geral dos dados e identificar possíveis desvios sistemáticos. Como forma de análise adicional, também foram analisadas as dez primeiras predições e os valores reais correspondentes, permitindo a verificação da precisão ponto a ponto.

Foi utilizada a biblioteca Pandas[8] para a manipulação e análise dos dados e a biblioteca matplotlib[9] para visualização dos resultados. A Figura 25 apresenta exemplo de código para visualização da predição do LogP.

[8] https://pypi.org/project/pandas/;
[9] https://pypi.org/project/matplotlib/;

```
# Plotar resultados
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred)
plt.xlabel("Log_p - Valores Reais")
plt.ylabel("Log_p - Valores Preditos")
plt.title("MLP Regressor")
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red')  # Linha de identidade
plt.savefig("MLP_logP_Bin.png", dpi=300, bbox_inches='tight')
```

Figura 24- Exemplo de código para visualização da predição do valor de LogP por meio de um gráfico de dispersão Fonte: autor

5 Análise dos Resultados

Neste capítulo serão apresentados os resultados obtidos a partir da aplicação de modelos de regressão de ML e DL na predição do LogP e modelos de classificação para a predição da biodisponibilidade.

Na Seção 5.1 serão apresentados, a partir do uso de métricas especificas para análise de modelos de regressão, os principais resultados da aplicação da ML para predição do atributo LogP. Enquanto a Seção 5.2 foca resultados obtidos com DL para o LogP. Na Seção 5.3 são apresentados os resultados obtidos na aplicação de modelos de ML para a predição da biodisponibilidade utilizando as métricas Acurácia, F1-Score e AUC comumente usadas para a avaliação de modelos classificatórios. A Seção 5.4 descreve os resultados objetivos com DL para Biodisponibilidade. Por fim, a seção 5.5 discute os resultados obtidos a partir de uma minuciosa análise.

5.1 Resultados da aplicação da ML para predição do LogP

Como pode-se observar na Figura 26, as principais métricas para regressão utilizadas foram: *Mean Squared Error* (MSE) e R^2 *Score*, os quais indicam, respectivamente, o erro quadrático médio e a proporção da variância explicada pelo modelo.

Considerando os resultados para o atributo LogP, destaca-se que os modelos de regressão linear obtiveram métricas competitivas (Regressão SVR com MSE de *1,6082* e R^2 de *0,7824*), embora com desempenho inferior aos modelos de *boosting*, cujo melhor resultado foi o modelo LightGBM que alcançou um MSE de *1,4913* e R^2 de *0,7982*, posicionando-se como uma alternativa mais interessante que o modelo linear. Quanto aos modelos baseados em arvores de decisão para o atributo LogP, apesar de serem robustos, apresentaram um desempenho pior, alcançando MSE de *1,9944* e R^2 de *0,7301*.

Já o modelo MLP com 120 neurônios em duas camadas internas (após o ajuste de hiperparâmetros) apresentou o melhor desempenho geral, sugerindo que a capacidade de modelagem não linear foi determinante para capturar padrões nos dados analisados e o mais indicado, até então, para aplicações neste contexto quando o atributo alvo é o LogP.

A Figura 25 também apresenta gráficos individuais para cada modelo. Nesses gráficos são plotados os valores reais no eixo x e os valores preditos no eixo y para cada molécula analisada no conjunto de testes. Assim, é possível interpretar o comportamento dos modelos para cada molécula do conjunto. Uma quantidade maior de pontos próximos à linha vermelha, bem como uma menor dispersão dos pontos em relação à essa linha representa um melhor desempenho do modelo. Assim, é possível observar, por exemplo, que o modelo AdaBoost obteve um desempenho ruim com uma repetição grande de valores preditos, onde valores maiores não foram corretamente preditos. Já os modelos SVR e MLP, por exemplo, apresentaram uma distribuição mais próxima da linha de referência (vermelha), com vantagem no MLP para valores mais altos e mais baixos.

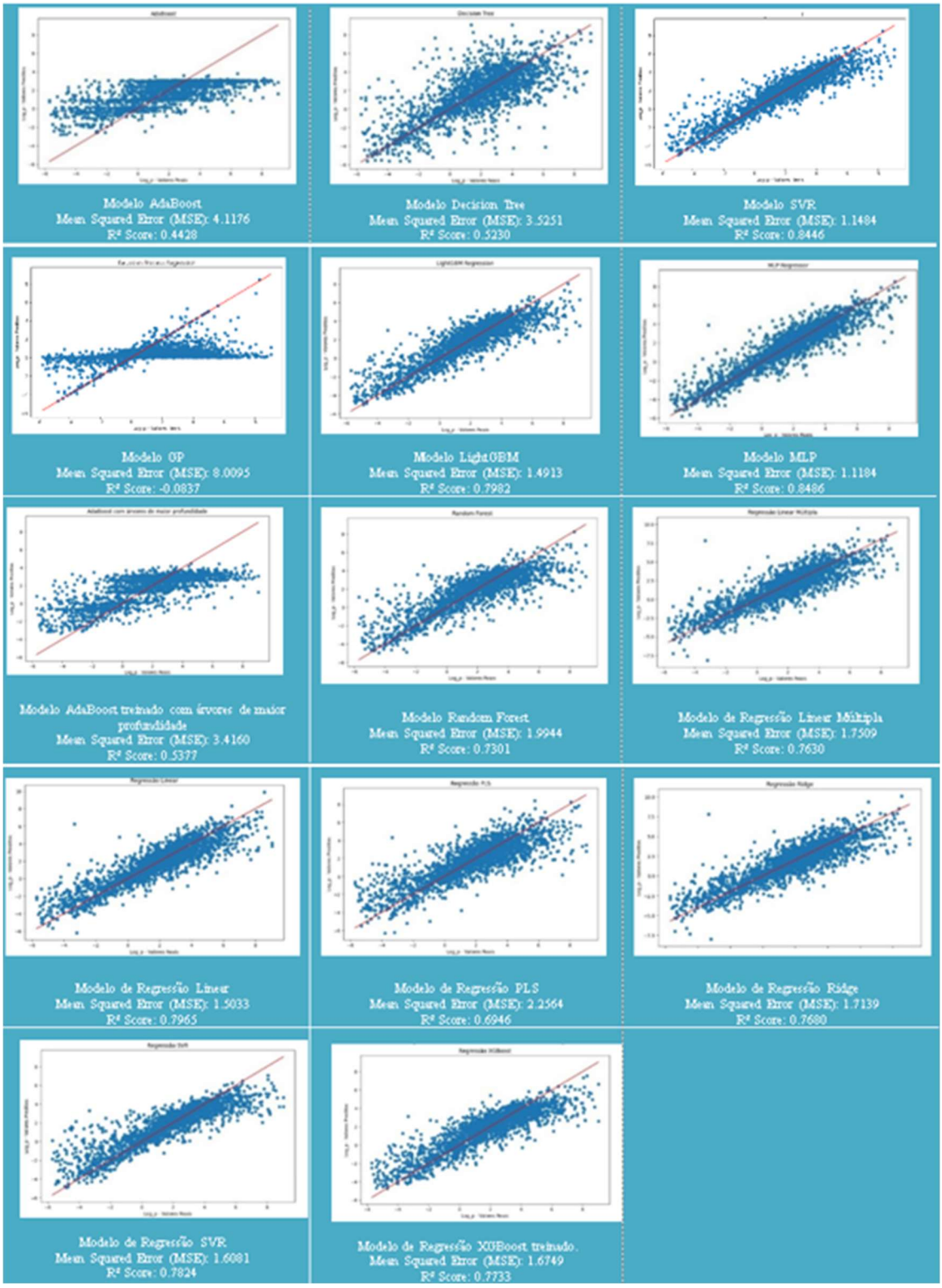

Figura 25- Comparação entre Modelos de ML com métricas para LogP

5.2 Resultados da aplicação da DL para predição do LogP

Como pode ser observado na Figura 26, as principais métricas utilizadas para avaliar o desempenho do modelo DNN na predição do atributo LogP foram: *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE) e R^2 Score, que indicam, respectivamente, o erro quadrático médio, o erro absoluto médio e a proporção da variância explicada pelo modelo.

O modelo DNN, baseado em camadas densas, foi ajustado por meio de otimização de hiperparâmetros e apresentou os melhores resultados entre todos os modelos testados (incluindo os de ML), alcançando MSE de *0,9978*, MAE de *0,7039* e R^2 de *0,8649*.

Em comparação com modelos lineares e de árvores de decisão previamente analisados, a DNN demonstrou desempenho superior. Enquanto modelos como a Regressão Linear ou SVR apresentaram métricas competitivas, mas limitadas pela linearidade ou sensibilidade, a arquitetura com várias camadas densas foi capaz de capturar padrões mais complexos superando também as métricas do modelo MLP (com poucas camadas internas), resultando em menor erro e maior explicabilidade da variância dos dados (R^2).

Na Figura 26 a concentração dos pontos próximos à linha de referência (vermelha) reflete o bom desempenho do modelo. Dessa maneira, os resultados sugerem que a abordagem baseada em DL, especialmente por meio de redes densas profundas, representa uma estratégia robusta e altamente eficaz para predição do LogP, superando a capacidade preditiva dos modelos de ML previamente testados. Vale lembrar que foram realizados testes preliminares com redes CNN e RNN, mas as redes exclusivamente com camadas densas foram mais promissoras.

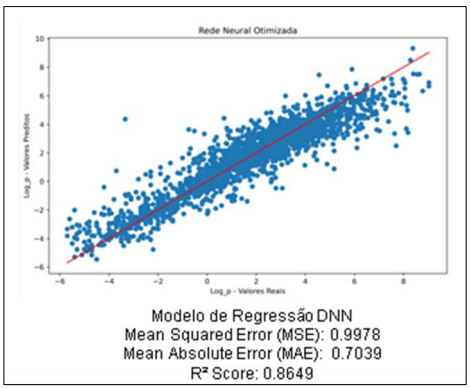

Figura 26- Deep Neural Network com camadas densas. Fonte: autor

5.3 Resultados da aplicação da ML para predição da Biodisponibilidade

Conforme apresenta a Tabela 8, quando o atributo alvo é a Biodisponibilidade o modelo MLP é muito eficaz mostrando maior capacidade de acerto global (Acurácia) e equilíbrio entre precisão e revocação (F1-Score). Além disso, o modelo teve o maior AUC, reforçando sua superioridade na separação das classes.

Os modelos RF e SVM obtiveram um desempenho muitos próximos quando as medidas de acurácia (*0,9274* e *0,9287*, respectivamente) e F1-Score (*0,9587* e *0,9595*, respectivamente), mas os valores mais baixos de AUC (cerca de 5% inferior a MLP) evidenciaram limitações em termos de separação de classes quando comparados a MLP (Tabela 8).

Tabela8– Comparação entre modelos de ML com métricas para Biodisponibilidade

Modelo	Accuracy	F1-Score	AUC
MLP	0.9326	0.9611	0.8304
Random Forest	0.9274	0.9587	0.7815
SVM	0.9287	0.9595	0.7810

5.4 Resultados da aplicação da DL para predição da Biodisponibilidade

Conforme apresentado na Figura 27, ao ser utilizado para predição com o atributo-alvo Biodisponibilidade, o modelo DNN mostrou desempenho ligeiramente inferior ao modelo MLP. A acurácia obtida foi de *0,9277*, indicando uma capacidade global de acerto um pouco menor. O F1-score de *0,9586* reflete um bom equilíbrio entre precisão e revocação, mas ainda ligeiramente inferior ao observado no modelo MLP. Além disso, o valor de AUC (*0,8040*) também foi inferior, sugerindo uma menor capacidade do modelo DNN em distinguir corretamente as classes, quando comparado ao modelo MLP.

Dado o desempenho próximo ao modelo MLP comum, foi investigado outras faixas de justes de hiperparâmetros, incluindo modelos de até *5* a *20* camadas densas em 30 tentativas no otimizador BayesianOptimization. No entanto, não foram obtidas melhorias.

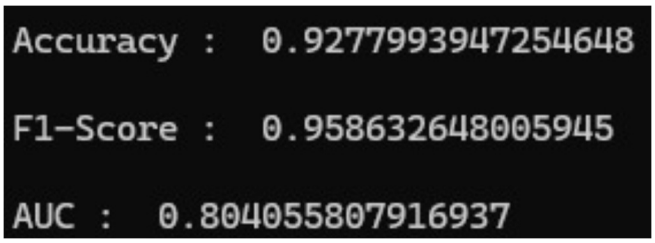

Figura 27- Acurácia, F1-score e AUC obtidos após a execução do modelo de DL para Biodisponibilidade. Fonte: autor

5.5 Análise e Discussão sobre os resultados

Nesta pesquisa foram selecionados modelos de classificação para predizer Biodisponibilidade e modelos de regressão para estimar o LogP. Foram testadas métricas, hiperparâmetros e técnicas com base em uma revisão sistemática da literatura, alinhados ao tema de estudo e às características dos dados.

Para a análise dos resultados, foi elaborada uma tabela de apoio à interpretação, uma vez que, conforme descrito por Géron (2019), modelos de regressão e classificação utilizam métricas específicas de avaliação. Esses indicadores descritos na tabela 9 têm

como objetivo fornecer uma base comparativa que permita interpretar os resultados obtidos e realizar uma análise crítica do desempenho preditivo de cada modelo avaliado.

Tabela 9 – Valor ideal das métricas para modelos de classificação e regressão

Tipo de Modelo	Métrica	Descrição	Valor Ideal / Interpretação
Regressão	MSE (Erro Quadrático Médio)	Média dos quadrados dos erros entre valores reais e preditos.	Quanto mais próximo de 0, melhor. Comparar com o desvio padrão da variável alvo.
	RMSE (Raiz do Erro Quadrático Médio)	Raiz quadrada do MSE.	Ideal: < 10% do intervalo da variável alvo ou próximo ao desvio padrão.
	MAE (Erro Absoluto Médio)	Média dos erros absolutos.	Quanto mais próximo de 0, melhor.
	R^2 (Coeficiente de Determinação)	Proporção da variância explicada pelo modelo.	≈ 1.0: Perfeito
			> 0.9: Excelente
			0.7–0.9: Bom
			0.5–0.7: Aceitável
			< 0.5: Fraco
Classificação	Accuracy (Acurácia)	Percentual de acertos do modelo.	> 90%: Ótimo
			80–90%: Bom
			70–80%: Razoável
			< 70%: Ruim ou modelo fraco
	Precision / Recall / F1-score	Avaliam qualidade das classes positivas preditas, recuperadas e harmonia entre elas.	> 0.9: Excelente
			0.8–0.9: Bom
			0.7–0.8: Aceitável
			< 0.7: Fraco
	AUC (Área sob a Curva ROC)	Mede a capacidade do modelo em distinguir entre classes.	= 1.0: Perfeito
			≥ 0.9: Excelente
			0.8–0.9: Muito bom
			0.7–0.8: Aceitável
			< 0.7: Fraco

A partir da análise das métricas apresentadas na Tabela 9, observamos que o modelo DNN obteve o melhor desempenho entre os modelos preditivos de regressão, alcançando um MSE de *0,9978*, MAE de *0,7039* e R^2 de *0,8649*. Esses resultados superam todos os modelos baseados em aprendizado de máquina "comum" (ML). Notadamente, o DNN também superou o MLP, cuja arquitetura é mais rasa, indicando que a maior profundidade da rede contribuiu para uma melhor captura de padrões complexos nos dados.

Analisando os outros modelos de ML, o SVR e o LightGBM apresentaram bons resultados (MSE entre *1,14* e *1,49* e R^2 entre *0,79* e *0,84*), reforçando a adequação desses métodos ao problema. Contudo, modelos como AdaBoost e árvores de decisão

apresentaram desempenho inferior (MSE entre *3,52* e *4,11* e R^2 entre *0,44* e *0,52*), possivelmente devido à menor capacidade de generalização e maior sensibilidade ao *overfitting*, especialmente diante de variáveis contínuas como o LogP.

Tabela 10 – Resultados dos Modelos de Regressão Avaliados

Modelo	MSE	MAE	R^2	Interpretação Geral
DNN (Rede Neural Profunda)	0,9978	0,7039	0,8649	Melhor desempenho geral. Baixos erros (MSE e MAE) e R^2 excelente.
MLP	1,1184		0,8486	Erros baixos e R^2 alto. Desempenho próximo ao DNN.
SVR	1,1484		0,8446	MSE baixo e ótimo R^2. Modelo consistente.
LightGBM	1,4913		0,7982	Bom desempenho. MSE razoavelmente baixo e R^2 elevado.
Regressão Linear	1,5033		0,7965	Modelo simples, mas eficaz. MSE competitivo.
Regressão SVR	1,6081		0,7824	Desempenho sólido, com erro ainda controlado.
XGBoost	1,6749		0,7733	Modelo robusto, mas com MSE um pouco mais alto.
Regressão Ridge	1,7139		0,768	Bom R^2 e erro moderado. Benefício da regularização.
Regressão Linear Múltipla	1,7509		0,763	Simples e eficaz. Desempenho comparável à regressão Ridge.
Random Forest	1,9944		0,7301	R^2 bom, mas com MSE um pouco elevado.
Regressão PLS	2,2564		0,6946	MSE moderado e R^2 aceitável.
AdaBoost (profundidade maior)	3,4160		0,5377	R^2 fraco. Erro considerável.
Decision Tree	3,5251		0,523	Poder preditivo limitado. MSE alto.
AdaBoost	4,1176		0,4428	Desempenho fraco. MSE elevado e baixo R^2.
GP (Gaussian Process)	8,0095		-0,0837	Pior desempenho. MSE muito alto e R^2 negativo.

No caso da Biodisponibilidade (Tabela 11), tratada como um problema de classificação binária, o modelo MLP se destacou com uma boa capacidade discriminativa entre as classes (aplicando as faixas de referência da Tabela 9 nos resultados da Tabela 11). Essa performance indica que a arquitetura neural foi mais eficaz na identificação dos padrões que separam as classes, superando modelos como SVM e Random Forest, que apresentaram desempenho apenas intermediário. Apesar disso, o valor relativamente baixo do AUC para alguns modelos reforça a dificuldade em separar corretamente as classes, o que pode ser atribuído à um eventual desbalanceamento do conjunto de dados ou à presença de ruído nas variáveis explicativas.

Tabela 11 – Resultados dos Modelos de Classificação Avaliados

Modelo	Accuracy	F1-Score	AUC	Interpretação
MLP	0.9326	0.9611	0.8304	O MLP apresentou o melhor desempenho geral. Alta acurácia e F1 indicam boa classificação, e o AUC elevado mostra boa capacidade discriminativa entre classes.
Random Forest	0.9274	0.9587	0.7815	Bom desempenho, mas inferior ao MLP. Leve queda no AUC sugere menor separação entre classes.
SVM	0.9287	0.9595	0.7810	Desempenho semelhante ao Random Forest. Acurácia levemente superior, mas com o menor AUC dos três, o que pode indicar menor poder de discriminação.

Considerando o desempenho observado em alguns modelos de regressão, é provável que intervenções pontuais, como um pré-processamento mais refinado dos dados, o ajuste criterioso de hiperparâmetros e a aplicação de técnicas de seleção automática de atributos, como a eliminação recursiva de variáveis (*RFE*), poderiam contribuir para melhorias na capacidade preditiva desses algoritmos. No entanto, dadas as características do conjunto de dados (muitos atributos binários esparsos) e a complexidade dos padrões subjacentes, é pouco provável que tais ajustes sejam suficientes para superar significativamente o desempenho já alcançados pela DNN, os quais mostraram maior eficácia na extração de relações não lineares e estruturais entre as variáveis.

Ao analisar o desempenho dos modelos de classificação frente ao atributo biodisponibilidade em um contexto farmacocinético, observa-se que a métrica AUC apresenta valores relativamente baixos (entre 78,10% e 83,04%), mesmo quando a acurácia e o F1-Score estão acima de 92%. Isso ocorre porque *datasets* com esse atributo costumam apresentar um número significativamente maior de amostras pertencentes a uma classe, aproximadamente 70% do total de amostras em detrimento de outra, caracterizando um desbalanceamento de classes. Tal desproporção compromete a capacidade do modelo em aprender adequadamente os padrões da classe minoritária, dificultando sua correta distinção.

Técnicas de balanceamento de classes, como o *undersampling* (que reduz a quantidade de amostras da classe majoritária mantendo todas as da classe minoritária) e o *oversampling* (que aumenta artificialmente o número de amostras da classe minoritária até equilibrar a distribuição), podem contribuir para a melhoria da métrica *AUC*. No

entanto, essas abordagens podem impactar negativamente o desempenho geral do modelo, seja pela perda de informação relevante (no caso do *undersampling*) ou pela introdução de redundância e risco de *overfitting* (no caso do *oversampling*). Diante disso, pesquisas adicionais devem ser realizadas para verificar se o esforço necessário para ajustar o *dataset* se justifica (ou não), especialmente quando modelos baseados em redes neurais profundas apresentam desempenho superior mesmo em cenários de desbalanceamento.

Por fim, também é necessário o estudo com novas arquiteturas de redes neurais profundas. Isto inclui redes baseadas em *transformers* (Vaswani *et al*., 2017) incluindo arquiteturas de dados tabulares (TabNet) e imagens. Também deve ser explorado o uso de redes pré-treinadas em conjunto com técnicas de aprendizado de transferência.

6 Conclusão e Trabalhos Futuros

O processo tradicional de produção de novos medicamentos pela indústria farmacêutica necessita de ferramentas que possam minimizar o tempo e o custo no desenvolvimento de novos produtos. Este trabalho explorou o uso de algoritmos de ML aplicados à predição de propriedades relacionadas à distribuição de fármacos, com ênfase no coeficiente de partição octanol-água (LogP) e na biodisponibilidade. Desta forma modelos preditivos serviram como suporte à seleção de candidatos a novos fármacos.

Os resultados obtidos mostraram a eficácia ao utilizar técnicas de aprendizado supervisionado e redes neurais profundas para prever as propriedades de distribuição dos compostos com alta precisão. Além disso, a metodologia empregada demonstrou a viabilidade de utilizar bases de dados abertas, como o DrugBank, com algoritmos de aprendizado de máquina para otimizar a seleção de compostos promissores.

Este capítulo de conclusão está organizado em seções da seguinte maneira: a seção 6.1 apresenta as contribuições da pesquisa; a seção 6.2 expõe os trabalhos futuros; e a seção 6.3 faz as considerações finais.

6.1 Contribuições

Este trabalho viabilizou uma forma de otimização dos processos de desenvolvimento de medicamentos utilizando a capacidade dos modelos de ML (e DL) na predição de propriedades importantes no processo de distribuição de fármacos no organismo. Dessa forma, o projeto envolveu a execução de um fluxo que abrangeu a extração, transformação e carga dos dados necessários ao treinamento de modelos preditivos, além da aplicação de técnicas especificas de avaliação de resultados para cada tipo especifico que culminou na identificação do modelo com os melhores resultados.

Com o objetivo de responder à questão de pesquisa apresentada no início desta dissertação - “Como criar uma abordagem e solução baseada em ML (e DL) que faça uso de bases internacionais abertas para apoiar a predição de propriedades ligadas à distribuição de fármacos?” -, são destacadas a seguir as principais contribuições obtidas:

- **Desenvolvimento de Modelos Preditivos**: A necessidade de prever com precisão satisfatória o coeficiente de partição octanol-água (LogP) e a biodisponibilidade de fármacos levou ao desenvolvimento de modelos de aprendizado de máquina e aprendizado profundo.
- **Uso de Bases de Dados Abertas**: Através deste trabalho foi possível mostrar como bases de dados abertas, como o DrugBank, podem ser utilizadas de maneira eficaz no treinamento de modelos de aprendizado de máquina para predição de propriedades farmacocinéticas.
- **Algoritmos para Extração e Formatação dos Valores dos Atributos**: Desenvolvimento de algoritmos para extração e formatação dos valores dos atributos relevantes à predição, utilizando a notação SMILES e descritores moleculares.
- **Contribuição para a Redução de Custos e Tempo**: O modelo desenvolvido e selecionado pode ser aplicado para simplificar e otimizar o processo de seleção de candidatos a novos medicamentos, proporcionando uma redução de tempo e de recursos.
- **Base para Pesquisas Futuras**: O trabalho oferece uma base para pesquisas futuras que pretendam explorar a aplicação de técnicas de aprendizado de máquina em outras etapas do desenvolvimento de fármacos ou o uso de outros atributos relacionados ao processo de distribuição de fármacos.

Como resultado deste estudo, foram elaborados 2 artigos, destacados a seguir:

- Workshop de Computação da UNIFACCAMP (WCF) 2023, evento promovido pelo Centro Universitário Campo Limpo Paulista (UNIFACCAMP). Título: O uso de Aprendizado de Máquina para Criação e Seleção de Formulações Farmacêuticas (PESTANA; BONACIN; FERRUCIO, 2023).
- Artigo submetido e aceito para a 21st International Conference on Information Technology-New Generations (ITNG 2024)
  Titulo: *A Review on the Use of Machine Learning for Pharmaceutical Formulations.*

Além disso, está prevista mais uma submissão de artigo à publicação internacional com os resultados obtidos.

6.2 Trabalhos Futuros

Como próximo passo desta pesquisa está previsto o estudo de como algoritmos de IA podem apoiar na análise as interações com as enzimas do citocromo P450, especialmente as isoformas CYP450 2C9 e CYP450 2D6 (Smith *et al*., 1998). Essas enzimas desempenham um papel fundamental no metabolismo de diversas drogas, influenciando a velocidade de biotransformação dos compostos e, consequentemente, sua permanência e distribuição no organismo. Dado o impacto dessas isoformas no metabolismo de fármacos, esses trabalhos poderão explorar com maior profundidade o desenvolvimento de modelos preditivos para apoio à distribuição de fármacos como um todo.

Além disso, dado o bom desempenho das redes neurais em ambos atributos preditos, se justifica trabalhos futuros com foco no estudo de outras arquiteturas de DL, em especial aquelas que fazem uso de múltiplas camadas densas.

Pretende-se também explorar campos textuais e desestruturados em bases abertas. Durante o estudo, foi possível identificar campos de relevância no conjunto de dados do DrugBank, mas eles se encontravam desestruturados. A pesquisa pelo uso de técnicas de processamento de linguagem natural, em particular modelos de linguagem em grande escala, podem ser úteis para extrair conhecimento e aprimorar o apoio ao processo de desenvolvimento de medicamentos.

A predição dos atributos Biodisponibilidade e LogP, bem como de potenciais atributos adicionais em estudos futuros, viabiliza o desenvolvimento de uma ferramenta computacional que facilite o processo de identificação desses valores para uma ou mais moléculas, por meio de uma interface visual acessível via aplicação web.

Outras bases abertas (veja capítulo 2), também devem ser exploradas em trabalhos futuros. Em particular, o uso dessas bases em conjunto para predição de propriedades de

fármacos constitui em um desafio de pesquisa em aberto a ser abordado como continuidade desta pesquisa.

6.3 Considerações Finais

Este projeto de pesquisa apresentou uma abordagem e solução baseada em ML e DL e mostrou por meio de resultados e análises, diferentes modelos para a predição de determinadas propriedades ligadas à distribuição de fármacos, utilizando como referência os dados da base aberta DrugBank. Esses dados foram fundamentais para o treinamento e validação dos modelos desenvolvidos, permitindo contribuir para o conhecimento e para a construção de algoritmos capazes de identificar padrões nas moléculas e realizar predições de valores contínuos e discretos. Espera-se que este trabalho, em longo prazo, possa contribuir no processo de desenvolvimento de medicamentos, proporcionando redução de custos e tempo.

7 Referências

Aliagas, I., Gobbi, A., Lee, M. L., *et al*. (2022). Comparison of logP and logD correction models trained with public and proprietary data sets. *Journal of Computer-Aided Molecular Design, 36*, 253–262. https://doi.org/10.1007/s10822-022-00450-9

Alves, V. M., Braga, R. C., Muratov, E. N., & Andrade, C. H. (2018). **QUIMIOINFORMÁTICA: UMA INTRODUÇÃO.** Química Nova, 41(2), 202–212. https://doi.org/10.21577/0100-4042.20170145

Arik, S. Ö., & Pfister, T. (2021, May**). Tabnet: Attentive interpretable tabular learning. In Proceedings of the AAAI conference on artificial intelligence** (Vol. 35, No. 8, pp. 6679-6687).

Bannigan, P., Aldeghi, M., Bao, Z., Häse, F., Aspuru-Guzik, A., & Allen, C. (2021). **Machine learning directed drug formulation development.** *Advanced Drug Delivery Reviews, 175*, 113806.

Bender, A., & Cortés-Ciriano, I. (2021). **Artificial intelligence in drug discovery: What is realistic, what are illusions? Part 1: Ways to make an impact, and why we are not there yet.** *Drug Discovery Today, 26*(2), 511–524.

Bishop, C. M. (2006). **Pattern Recognition and Machine Learning**. Springer.

Breiman, L. (2001). **Random forests. Machine learning**, 45, 5-32.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). **Classification and regression trees. Routledge.**

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). **Introduction to machine learning, neural networks, and deep learning**. Translational Vision Science & Technology, 9(2), 14. https://doi.org/10.1167/tvst.9.2.14

Cortes, C., & Vapnik, V. (1995). **Support-vector networks. Machine Learning**, 20(3), 273-297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). **Nearest neighbor pattern classification. IEEE Transactions on Information Theory**, 13(1), 21-27.

Dametto, M., Vechi, S. M., & Bonacin, R. (2022**). Predicting cancer relapse with machine learning from an open Brazilian database.** In *2022 E-Health and Bioengineering Conference (EHB)* (pp. 1–4).

Damiati, S. A., & Damiati, S. (2021). **Microfluidic synthesis of indomethacin-loaded PLGA microparticles optimized by machine learning.** *Frontiers in Molecular Biosciences, 8*. https://www.frontiersin.org/articles/10.3389/fmolb.2021.677547

Deng, J., Ye, Z., Zheng, W., Chen, J., Gao, H., Wu, Z., ... Ouyang, D. (2023). **Machine learning in accelerating microsphere formulation development.** *Drug Delivery and Translational Research, 13*(4), 966–982 https://doi.org/10.1007/s13346-022-01253-z

Dong, J., Gao, H., & Ouyang, D. (2021). **Pharmsd: A novel AI-based computational platform for solid dispersion formulation design.** *International Journal of Pharmaceutics, 604*, 120705.

Ferneda, E. (2006). **Redes neurais e sua aplicação em sistemas de recuperação de informação.** Ciência da Informação, 35(1), 25-30. https://doi.org/10.1590/S0100-19652006000100003

Franke, F. S. (2020). **Como são desenvolvidos os medicamentos?** farmacoLÓGICA. Universidade Federal do Rio Grande do Sul. Disponível em: https://www.ufrgs.br/farmacologica/2020/06/30/como-sao-desenvolvidos-os-medicamentos/

Freitas, I. W. S. (2019). **Um estudo comparativo de técnicas de detecção de outliers no contexto de classificação de dados.** Dissertação de Mestrado.

Gao, H., Jia, H., Dong, J., Yang, X., Li, H., & Ouyang, D. (2021). **Integrated in silico formulation design of self-emulsifying drug delivery systems.** *Acta Pharmaceutica Sinica B, 11*(11), 3585–3594.

Gao, H., Su, Y., Wang, W., Xiong, W., Sun, X., Ji, Y., ... Ouyang, D. (2021). **Integrated computer-aided formulation design: A case study of andrographolide/cyclodextrin ternary formulation.** *Asian Journal of Pharmaceutical Sciences, 16*(4), 494–507.

Gaulton, A., *et al*. (2017). **The ChEMBL database in 2017.** Nucleic Acids Research, 45(D1), D945-D954. https://doi.org/10.1093/nar/gkw1074

Gentiluomo, L., & Roessner, D. (2020). **Application of machine learning to predict monomer retention of therapeutic proteins after long term storage.** *International Journal of Pharmaceutics, 577*, 119039

Gentiluomo, L., Roessner, D., Augustijn, D., Svilenov, H., Kulakova, A., Mahapatra, S., Frieß, W. (2019). **Application of interpretable artificial neural networks to early monoclonal antibodies development.** *European Journal of Pharmaceutics and Biopharmaceutics, 141*, 81–89.

Géron, A. (2019). **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow** (2nd ed.). O'Reilly Media.

Gholipour, E., & Bastas, A. (2023). **State-of-the-art review of neural network applications in pharmaceutical manufacturing: Current state and future directions.** *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-023-02206-0

Glišić, T., Djuriš, J., Vasiljević, I., Parojčić, J., & Aleksić, I. (2023). **Application of machine-learning algorithms for better understanding the properties of liquisolid systems prepared with three mesoporous silica-based carriers.** *Pharmaceutics, 15*(3), 741. https://www.mdpi.com/1999-4923/15/3/741

Goodfellow, I., Bengio, Y., & Courville, A. (2016). ***Deep learning***. MIT Press.

Guido, R. V. C., Andricopulo, A. D., & Oliva, G. (2010). **Planejamento de fármacos, biotecnologia e química medicinal: Aplicações em doenças infecciosas.** Revista Estudos Avançados, 24(70), 81-98. https://doi.org/10.xxxx

He K. (2022). **Pharmacological affinity fingerprints derived from bioactivity data for the identification of designer drugs.** Journal of cheminformatics, 14(1), 35. https://doi.org/10.1186/s13321-022-00607-6

He, Y., Ye, Z., Liu, X., Wei, Z., Qiu, F., Li, H.-F., ... Ouyang, D. (2020). **Can machine learning predict drug nanocrystals?** *Journal of Controlled Release, 322*, 274–285 https://doi.org/10.1016/j.jconrel.2020.03.043

Hosseini, M. A. H., Alizadeh, A. A., & Shayanfar, A. (2024). Prediction of the first-pass metabolism of a drug after oral intake based on structural parameters and physicochemical properties. *European Journal of Drug Metabolism and Pharmacokinetics, 49*(4), 449–465. https://doi.org/10.1007/s13318-024-00892-6

Irwin, J. J., & Shoichet, B. K. (2005). **ZINC--A free database of commercially available compounds for virtual screening.** Journal of Chemical Information and Modeling, 45(1), 177-182. https://doi.org/10.1021/ci049714+

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). **An introduction to statistical learning with applications in R**. Springer. https://doi.org/10.1007/978-1-4614-7138-7

Kamerzell, T. J., & Middaugh, C. R. (2021). **Prediction machines: Applied machine learning for therapeutic protein design and development.** *Journal of Pharmaceutical Sciences, 110*(2), 665–681.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). **LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems**, 30, 3146-3154.

Kim, S., Thiessen, P. A., Bolton, E. E., Chen, J., Fu, G., Gindulyte, A., ... & Bryant, S. H. (2016). **PubChem substance and compound databases.** Nucleic Acids Research, 44(D1), D1202-D1213. https://doi.org/10.1093/nar/gkv951

Kitchenham, B. (2004). **Procedures for performing systematic reviews.** Keele, UK: Keele University.

Lai, P. K., Gallegos, A., & Trout, B. L. (2022). **Machine learning prediction of antibody aggregation and viscosity for high concentration formulation development of protein therapeutics.** *mAbs,14*(1),2026208. https://doi.org/10.1080/19420862.2022.2026208

Landrum, G. (2013). **RDKit: Open-source cheminformatics software**. Retrieved from https://www.rdkit.org

Lou, H., & Hageman, M. J. (2021). **Machine learning attempts for predicting human subcutaneous bioavailability of monoclonal antibodies.** *Pharmaceutical Research, 38*(3), 451–460. https://doi.org/10.1007/s11095-021-03022-y

Martinelli, D. D. (2022). **Generative machine learning for de novo drug discovery: A systematic review.** *Computers in Biology and Medicine, 145*, 105403.

McCallum, A., & Nigam, K. (1998). **A comparison of event models for Naive Bayes text classification.** AAAI Workshop on Learning for Text Categorization, 41-48.

Moore, T. J., Zhang, H., Anderson, G., & Alexander, G. C. (2018). **Estimated Costs of Pivotal Trials for Novel Therapeutic Agents Approved by the US Food and Drug Administration**, 2015-2016. JAMA internal medicine, 178(11), 1451–1457. https://doi.org/10.1001/jamainternmed.2018.3931

Moore, T. J., Zhang, H., Anderson, G., & Alexander, G. C. (2018). **Estimated Costs of Pivotal Trials for Novel Therapeutic Agents Approved by the US Food and Drug Administration**, 2015-2016. JAMA internal medicine, 178(11), 1451–1457. https://doi.org/10.1001/jamainternmed.2018.3931

Noorain, L., Nguyen, V., Kim, H.-W., & Nguyen, L. T. B. (2023). **A machine learning approach for PLGA nanoparticles in antiviral drug delivery.** *Pharmaceutics, 15*(2), 495 https://www.mdpi.com/1999-4923/15/2/495

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D.,Moher, D. (2021). **The PRISMA 2020 statement: An updated guideline for reporting systematic reviews.** *BMJ, 372*, n71. https://www.bmj.com/content/372/bmj.n71

Patel, S., Patel, M., Kulkarni, M., & Patel, M. S. (2023). **De-interact: A machine learning-based predictive tool for the drug-excipient interaction study during product development—validation through paracetamol and vanillin as a case study.** *International Journal of Pharmaceutics, 637*, 122839

Pattni, B. S., & Torchilin, V. P. (2015). **Targeted drug delivery systems: Strategies and challenges.** In P. Devarajan & S. Jain (Eds.), **Targeted Drug Delivery: Concepts and Design** (pp. xx-xx). Springer. https://doi.org/10.1007/978-3-319-11355-5_1

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). **Scikit-learn: Machine learning in Python.** Journal of Machine Learning Research, 12, 2825–2830.

Pence, H. E., & Williams, A. (2010). **ChemSpider: An online chemical information resource.** Journal of Chemical Education, 87(11), 1123-1124.

Pereira, D. G. (2007). **Importância do metabolismo no planejamento de fármacos.** Química Nova, 30(1), 171-177.https://doi.org/10.1590/S0100-40422007000100029

Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking (1st ed.). O'Reilly Media.

Puranik, A., Dandekar, P., & Jain, R. (2022). **Exploring the potential of machine learning for more efficient development and production of biopharmaceuticals.** *Biotechnology Progress, 38*(6), e3291. https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/btpr.3291

Rasmussen, C. E., & Williams, C. K. I. (2006). **Gaussian processes for machine learning. MIT Press.** https://doi.org/10.7551/mitpress/3206.001.0001

Rimal, R., Rimal, B., & Bhandari, H. N. (2024). **Real estate market prediction using deep learning models.** Annals of Data Science. https://doi.org/10.1007/s40745-024-00543-2

Saunders, C., Gammerman, A., & Vovk, V. (1998). **Ridge regression learning algorithm in dual variables. Proceedings of the 15th International Conference on Machine Learning** (pp. 515-521). Morgan Kaufmann.

Schmitt, J. M., Baumann, J. M., & Morgen, M. M. (2022). **Predicting spray-dried dispersion particle size via machine learning regression methods.** *Pharmaceutical Research, 39*(12), 3223–3239.https://doi.org/10.1007/s11095-022-03370-3

Shargel, L., Andrew, B. C., & Wu-Pong, S. (1999). **Applied biopharmaceutics & pharmacokinetics.** Stamford: Appleton & Lange.

Shetti, J., Pickl, S., Bein, D., & Nistor, M. S. (2022). **Using software for computational fluid dynamics and molecular dynamics.** In *ITNG 2022 19th International Conference on Information Technology-New Generations* (pp. 35–38). Springer.

Silva, R. E. V. (2018). **Um estudo comparativo entre redes neurais convolucionais para a classificação de imagens.** Dissertação de Mestrado.

Singh, J., & Banerjee, R. (2019). **A study on single and multi-layer perceptron neural network.** Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 35-40. https://doi.org/10.1109/ICCMC.2019.8819775

Smith, G., Stubbins, M. J., Harries, L. W., & Wolf, C. R. (1998). **Molecular genetics of the human cytochrome P450 monooxygenase superfamily**. Xenobiotica, 28(12), 1129-1165. https://doi.org/10.1080/004982598238868

Smola, A. J., & Schölkopf, B. (2004). **A tutorial on support vector regression. Statistics and computing**, 14, 199-222.

Tran, T. T. V., Tayara, H., & Chong, K. T. (2023). **Recent studies of artificial intelligence on in silico drug distribution prediction.** International Journal of Molecular Sciences, 24(3), 1815. https://doi.org/10.3390/ijms24031815

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). **Attention is all you need**. *Advances in Neural Information Processing Systems*, *30*.

Wang, W., Ye, Z., Gao, H., & Ouyang, D. (2021**). Computational pharmaceutics: A new paradigm of drug delivery.** *Journal of Controlled Release, 338*, 119–136.

Weininger, D. (1988). **SMILES, a chemical language and information system**. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences, 28(1), 31-36. https://doi.org/10.1021/ci00057a005

Wishart, D. S. (2008). **DrugBank and its relevance to pharmacogenomics. Pharmacogenomics**, 9(8), 1155-1162. https://doi.org/10.2217/14622416.9.8.1155

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., ... & Wilson, M. (2018). **DrugBank 5.0: A major update to the DrugBank database for 2018.** Nucleic Acids Research, 46(D1), D1074-D1082. https://doi.org/10.1093/nar/gkx1037

Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., & Woolsey, J. (2006). **DrugBank: A comprehensive resource for in silico drug discovery and exploration.** Nucleic Acids Research, 34(Database issue), D668-D672. https://doi.org/10.1093/nar/gkj067

Yang, Y., Ye, Z., Su, Y., Zhao, Q., Li, X., & Ouyang, D. (2019). **Deep learning for in vitro prediction of pharmaceutical formulations.** *Acta Pharmaceutica Sinica B, 9*(1), 177–185.

Yoo, S., Lee, H., & Kim, J. (2023). **Deep learning for identifying promising drug candidates in drug–phospholipid complexes**. *Molecules, 28*(12), 4821. https://www.mdpi.com/14203049/28/12/4821

Zhao, Q., Ye, Z., Su, Y., & Ouyang, D. (2019). **Predicting complexation performance between cyclodextrins and guest molecules by integrated machine learning and molecular modeling techniques.** *Acta Pharmaceutica Sinica B, 9*(6), 1241–1252.