



UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Comparação de *Ensemble* de Funções de Escore em Docagem Molecular Proteína-Ligante

Maurício Dorneles Caldeira Balboni

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Profa. Dra. Karina dos Santos Machado
Co-orientador: Prof. Dr. Adriano Velasque Werhli

Rio Grande, 2024

Ficha Catalográfica

B173c Balboni, Maurício Dorneles Caldeira.
Comparação de *ensemble* de funções de escore em docagem molecular proteína-ligante / Maurício Dorneles Caldeira Balboni. – 2024.
67 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2024.
Orientadora: Dra. Karina dos Santos Machado.
Coorientador: Dr. Adriano Velasque Werhli.

1. Docagem molecular 2. *Ensemble* 3. Bioinformática
4. Aprendizado de máquina 5. Funções de escore I. Machado, Karina dos Santos II. Werhli, Adriano Velasque III. Título.

CDU 004

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344



Universidade Federal do Rio Grande
Centro de Ciências Computacionais
Programa de Pós-Graduação em Computação
Curso de Mestrado em Engenharia de Computação



DISSERTAÇÃO DE MESTRADO

Comparação de Ensemble de Funções de Escore em Docagem Molecular Proteína-Ligante

Maurício Dorneles Caldeira Balboni

Banca examinadora:

 Documento assinado digitalmente
DIANA FRANCISCA ADAMATTI
Data: 01/04/2024 17:57:36-0300
Verifique em <https://validar.jti.gov.br>

Profa. Dra. Diana Francisca Adamatti (FURG)

 Documento assinado digitalmente
FREDERICO SCHMITT KREMER
Data: 24/06/2024 15:24:44-0300
Verifique em <https://validar.jti.gov.br>

Prof. Dr. Frederico Schmitt Kremer (UFPel)

 Documento assinado digitalmente
KARINA DOS SANTOS MACHADO
Data: 01/04/2024 15:43:52-0300
Verifique em <https://validar.jti.gov.br>

Profa. Dra. Karina dos Santos Machado (FURG)
Orientadora

RESUMO

BALBONI, Maurício Dorneles Caldeira . **Comparação de *Ensemble* de Funções de Escore em Docagem Molecular Proteína-Ligante**. 2024. 67 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

A inclusão de uma abordagem *in silico* na descoberta de novos medicamentos visa otimizar e reduzir os custos de um processo que costuma ser demorado e custoso. Estudos computacionais sobre as interações entre pequenas moléculas (ligantes) e proteínas-alvo (receptores) são de grande importância, principalmente nessa busca por novos fármacos. Esses estudos podem ser realizados utilizando algoritmos de docagem molecular que buscam encontrar a melhor conformação e posição que um ligante no sítio de ligação de um receptor. Nesse contexto, as funções de escore são funções matemáticas utilizadas com a finalidade de estimar aproximadamente a afinidade de ligação entre essas duas moléculas. O desenvolvimento de funções de escore cada vez mais precisas vem recebendo crescente destaque na literatura, destacando-se atualmente quatro categorias principais: baseadas em física *Physics-based*, empíricas, baseadas em conhecimento *Knowledge-based* e baseadas em aprendizado de máquina. O uso de *ensemble* é uma técnica que combina as previsões de vários modelos de aprendizado de máquina, e tem se mostrado fundamental para aprimorar o desempenho e a robustez em diversas aplicações. Com isto, o presente trabalho tem o objetivo de comparar a eficiência do uso dessas técnicas de *ensemble learning* em funções de escore para docagem molecular proteína-ligante. Para isso, é utilizado o PDBbind, o qual é um repositório de dados biológicos que armazena coleções de afinidade de ligações medidas de forma experimental para complexos proteína-ligante. São utilizadas as funções de escore Korp-PL, Vina, rb_score_GBT, rb_score_RF, ECIF-GBT, ECIF-RF, lin_F9, Convex-PL e rb_score_ET. Ao incorporar estas à um modelo de *ensemble*, podemos avaliar se a combinação desses métodos pode resultar em melhorias significativas na precisão das estimativas do escore de ligação. Neste trabalho foram utilizados os algoritmos XGboost e *Random Forest* como técnicas de *ensemble* assim como foi proposta uma função baseada em uma média ponderada pela correlação de cada função de escore. As métricas de avaliação utilizadas foram os valores de *Root Mean Squared Error* (RMSE) e Correlação de Pearson comparando o valor predito pelas funções de escore individuais e pelos modelos *ensemble* com o valor experimental disponível no PDBBind. Com base nos resultados gerados, é possível inferir que o *ensemble* apresentou o melhor desempenho nas métricas de avaliação, RMSE e correlação de Pearson, destacando-se especialmente quando utilizado o algoritmo *Random Forest*. O que evidencia a eficácia do *ensemble* como uma estratégia eficiente para aprimorar a precisão das predições quando comparado

às funções de escore isoladas. Esses resultados demonstram a vantagem dos modelos que utilizam *ensemble* a fim de combinar previsões individuais, proporcionando maior estabilidade nos resultados e capacidade de generalização, especialmente em aplicações complexas como a previsão de afinidade molecular, onde a precisão é fundamental para acelerar o desenvolvimento de novos compostos e reduzir o tempo de experimentação.

Palavras-chave: Docagem molecular, ensemble, bioinformática, aprendizado de máquina, funções de escore.

ABSTRACT

BALBONI, Maurício Dorneles Caldeira . **Comparative Analysis of Scoring Functions using Ensemble Learning in Molecular Docking**. 2024. 67 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

The inclusion of an *in silico* approach in drug discovery aims to optimize and reduce the costs of a process that is often time-consuming and expensive. Computational studies on the interactions between small molecules (ligands) and proteins (receptors) play a crucial role in drug discovery experiments. These studies can be conducted using molecular docking algorithms that try to find the best conformation and position for a ligand in the binding site of a receptor. In this context, scoring functions are mathematical functions used to estimate the binding affinity between these two molecules. The development of increasingly accurate scoring functions has been receiving growing attention in the literature. Currently four main categories standing out: physics-based, empirical, knowledge-based, and machine learning-based. The use of ensemble methods, which combine predictions from multiple machine learning models, have proven to be essential for improving performance and robustness in various applications. This study aims to compare the effectiveness of ensemble techniques applied to protein-ligand docking scoring functions. For this purpose, the PDBbind is used, which is a repository of biological data that stores collections of experimentally measured binding affinities for protein-ligand complexes. The scoring functions Korp-PL, Vina, rb_score_GBT, rb_score_RF, ECIF-GBT, ECIF-RF, lin_F9, Convex-PL, and rb_score_ET are used. By incorporating these into an ensemble model, we can evaluate whether the combination of these methods can result in significant improvements in the accuracy of binding score estimates. In this study, the XGBoost and Random Forest algorithms were used as ensemble techniques as well as it was proposed a weighted average based on the accuracy of each scoring function. The evaluation metrics used were the Root Mean Squared Error (RMSE) and Pearson correlation comparing the predicted value by the individual scoring functions and by the ensemble models with the experimental value available in PDBBind. Based on the results generated, it is possible to infer that the ensemble consistently showed the best performance in the evaluation metrics, RMSE and Pearson correlation, especially when using the Random Forest algorithm. This highlights the effectiveness of the ensemble as an efficient strategy to enhance the accuracy of predictions when compared to individual scoring functions. These results demonstrate the advantage of models that use ensemble to combine individual predictions, providing greater stability in results and generalization capacity, especially in complex applications such as molecular affinity prediction,

where precision is essential to accelerate the development of new compounds and reduce experimentation time.

Keywords: docking molecular, Ensemble, Bioinformatics, Machine Learning, Scoring Functions.

LISTA DE FIGURAS

Figura 1	Fluxograma do <i>ensemble bagging</i>	21
Figura 2	<i>Random Forest</i>	22
Figura 3	Fluxograma do <i>ensemble boosting</i>	23
Figura 4	Exemplo de docagem molecular. A proteína tem código PDB 5TMN. A proteína está no formato <i>New Cartoon</i> em ciano e o ligante no formato <i>Licorice</i> colorido por tipo de átomo. A figura foi gerada com o software VMD.	26
Figura 5	Pesos e termos da função Lin.F9	29
Figura 6	Funcionamento de uma função de escore baseada em aprendizado de máquina.	30
Figura 7	Fluxograma para calcular o <i>Scoring Power</i> . Fonte Arrúa (2020) [3]	32
Figura 8	Metodologia da dissertação.	37
Figura 9	Organização e exemplo de dados do PDDBind 2020.	40
Figura 10	<i>Datasets</i> gerados sem a limpeza de dados.	43
Figura 11	<i>Datasets</i> após tratamento dos dados.	43
Figura 12	Distribuição dos dados preditos pela função de escore Convex-PL.	45
Figura 13	Distribuição dos dados preditos pela função de escore Korp-PL.	46
Figura 14	Distribuição dos dados preditos pela função de escore ECIF-GBT.	46
Figura 15	Distribuição dos dados preditos pela função de escore ECIF-RF.	47
Figura 16	Distribuição dos dados preditos pela função de escore Lin.F9.	48
Figura 17	Distribuição dos dados preditos pela função de escore rb_score_ET.	48
Figura 18	Distribuição dos dados preditos pela função de escore rb_score_GBT.	49
Figura 19	Distribuição dos dados preditos pela função de escore rb_score_RF.	49
Figura 20	Distribuição dos dados preditos pela função de escore vina.	50
Figura 21	RMSE das funções de escore.	51
Figura 22	Correlação de Pearson das funções de escore.	52
Figura 23	Top 10 melhores predições das funções de escore.	52
Figura 24	Top 10 piores predições das funções de escore.	53
Figura 25	Parâmetros utilizados no modelo <i>Random Forest</i>	53
Figura 26	Parâmetros utilizados no modelo XGBoost	54
Figura 27	Distribuição dos dados preditos pelo <i>ensemble</i> utilizando <i>Random Forest</i>	55
Figura 28	Distribuição dos dados preditos pelo <i>ensemble</i> utilizando XGBoost	55
Figura 29	Distribuição dos dados preditos pela função de ponderação	56
Figura 30	Comparativo do RMSE dos modelos e das funções de escore.	57

Figura 31	Comparativo da correlação de Pearson dos modelos e das funções de escore	57
-----------	---	----

LISTA DE TABELAS

Tabela 1	Resumo das funções de escore utilizadas.	38
Tabela 2	Resultado da FS RB_Score_RF, uma das colunas do <i>dataset</i>	42
Tabela 3	Pesos utilizados na função de ponderação.	54
Tabela 4	Top 10 melhores predições dos modelos gerados	58
Tabela 5	Top 10 piores predições dos modelos gerados	58

SUMÁRIO

1	Introdução	13
1.1	Objetivo	15
1.1.1	Objetivo geral	15
1.1.2	Objetivos específicos	15
1.2	Organização	16
2	Referencial teórico	17
2.1	Aprendizado de máquina	17
2.1.1	Aprendizado supervisionado	18
2.1.2	Aprendizado não-supervisionado	19
2.2	Ensemble	20
2.2.1	Voting	20
2.2.2	Bagging	20
2.2.3	Boosting	22
2.3	Bioinformática	24
2.4	Banco de dados biológicos	24
2.4.1	BD de Proteínas: Protein Data Bank	24
2.4.2	BD de complexos proteína-ligante: PDBbind	24
2.5	Docagem Molecular	25
2.5.1	Busca	26
2.5.2	Funções de Score	27
2.5.3	Tipos de funções de escore	27
2.5.4	<i>Benchmarks</i> para validação de funções de escore	31
2.5.5	Aplicações da <i>Docagem Molecular</i> e Funções de Score	34
2.6	Linguagem de programação, pacotes e ferramentas	35
2.6.1	Linguagem de programação Python e dependências	35
2.6.2	Open Babel	36
3	Materiais e métodos	37
3.1	Obtenção e instalação das ferramentas	38
3.1.1	Funções de Escore	38
3.1.2	Instalação de dependências	38
3.2	Seleção do conjunto de treinamento	38
3.2.1	PDBbind <i>general set</i> 2020	39
3.2.2	PDBbind <i>refined set</i> 2020	40
3.3	Geração do <i>dataset</i>	40

3.3.1	Preparo da proteína e do ligante	40
3.3.2	Algoritmo para geração dos resultados das FS	41
3.3.3	Pré-processamento dos dados	43
3.4	Geração dos modelos preditivos	44
3.5	Avaliação da função proposta	44
4	Resultados	45
4.1	Análise exploratória de dados	45
4.2	Função de Ensemble	53
5	Conclusão	59
	Referências	61

ANEXOS

1 INTRODUÇÃO

A descoberta e desenvolvimento de novos fármacos é um processo de grande complexidade, visto que um medicamento, para ser eficaz, deve resistir às condições adversas produzidas pelo organismo humano, como exemplo o pH estomacal, a trajetória até a corrente sanguínea e seu percurso até atingir seu alvo principal. Além disto, deve-se ter o cuidado de controlar os efeitos adversos promovidos pelos mesmos, em um processo que envolve um alto custo financeiro e um longo tempo para seu desenvolvimento [45].

A pandemia da COVID-19, declarada em 2020 pela Organização Mundial da Saúde, ressaltou ainda mais a urgência de encontrar soluções rápidas para enfrentar doenças infecciosas emergentes. O surgimento do novo agente do coronavírus (nCoV-2019) no final de 2019 e sua rápida disseminação resultaram na declaração oficial de pandemia pela OMS em março de 2020 [15]. Diante desse cenário de emergência global, os pesquisadores se viram diante da necessidade premente de encontrar soluções e tratamentos eficazes para as vítimas da COVID-19.

Com base nisso, é de suma importância que sejam elaboradas estratégias para aceleração do processo de descoberta de novos fármacos. Neste contexto, a abordagem *in silico* apresenta-se como uma alternativa consistente, visto sua capacidade de gerar previsões confiáveis no que diz respeito às novas ações das drogas e até sobre os mecanismos ligados aos seus efeitos colaterais [57]. Esta etapa de simulação visa otimizar o processo e reduzir custos, além de prever a melhor conformação e estimar a afinidade de um ligante (molécula pequena) em um sítio de ligação de um receptor (proteína) [41].

Por volta de 1960, os programas de descoberta de novos medicamentos eram baseados na triagem de milhares de compostos tanto naturais quanto sintéticos. Uma vez que os compostos eram selecionados, os químicos sintetizavam centenas de compostos relacionados a fim de obter o medicamento mais eficaz. Entretanto, este custo é muito elevado, e estima-se que o custo total deste processo poderia variar entre 200 e 500 milhões de dólares, o que destacou a necessidade urgente de abordagens mais eficientes e econômicas na descoberta de medicamentos. Para auxiliar o Desenho Racional de Fármacos, do inglês *Rational Drug Design* (RDD) [32], avanços substanciais nas áreas de biologia molecular e cristalografia de proteínas tornaram-se imprescindíveis [62]. O entendimento mais

profundo das estruturas moleculares das proteínas-alvo possibilita uma abordagem mais direcionada e eficiente na concepção de novos medicamentos. Essa transição representa não apenas uma economia significativa de recursos financeiros, mas também uma melhora na precisão e eficácia no processo de desenvolvimento de novos fármacos.

Na necessidade de uma abordagem mais robusta e fidedigna, recorre-se aos ensaios *in vitro*. Esta abordagem é realizada em um ambiente controlado e permite uma aproximação mais fiel ao comportamento humano, resultando na identificação de alguns dos melhores candidatos entre as centenas de moléculas utilizadas no estudo [74]. O controle rigoroso do ambiente proporciona condições que simulam aspectos específicos do organismo humano, tornando os ensaios *in vitro* uma ferramenta valiosa na seleção de compostos [74], entretanto demandando uma maior quantidade de recurso e tempo. Com isto, têm-se a necessidade de aproximação cada vez mais entre as abordagens *in silico* e *in vitro* para uma maior capacidade de filtragem de moléculas mais promissoras.

No RDD, a docagem molecular é uma etapa fundamental. Este método realiza a avaliação de diferentes conformações e posições que uma determinada pequena molécula (ligante) pode ter no sítio de ligação de uma proteína-alvo. Os resultados são expressos por um escore que estima a energia de ligação entre o complexo. Desta forma, a molécula que apresentar a menor quantidade de energia necessária para se ligar ao sítio ativo apresentará o melhor resultado de atividade biológica [41].

Existem diversas ferramentas computacionais desenvolvidas para a docagem molecular [54]. Diferentes abordagens podem ser utilizadas em cada uma dessas ferramentas, porém o princípio básico delas pode ser dividido nas etapas a seguir: na etapa conhecida como amostragem, são propostas diferentes posições e conformações do ligante no sítio de ligação; na etapa seguinte, os complexos receptor-ligante são qualificados de acordo com sua afinidade de ligação por uma função de escore [21].

As funções de escore são funções matemáticas utilizadas com a finalidade de estimar aproximadamente a afinidade de ligação entre duas moléculas. Uma gama de variedade de funções de escore tem sido desenvolvidas. Visando obter um resultado preciso em um tempo de execução tolerável, surgem diferentes abordagens que objetivam-se a encontrar um equilíbrio entre velocidade de execução e precisão [29]. Uma maneira de classificar as funções de escore se dá a partir de quatro categorias [37], sendo elas: *Physics-based*: a qual utiliza os valores dos campos de forças disponíveis para calcular as interações diretas; *Empirical*: calcula o valor associado à adaptabilidade do alvo ao ambiente da ligação proteína-ligante, sendo esses cálculos obtidos através dos termos individuais que representam um fator referente à energia da interação; *Knowledge-based*: soma os potenciais estatísticos da interação proteína-ligante, sendo os cálculos realizados a partir de uma função de escore derivada de uma análise com base em estruturas já conhecidas; *Machine Learning*: utiliza-se de descritores as quais podem ser características do ligante, proteína e da interação, para treinar um modelo utilizando métodos de aprendizado de máquina e

com isso calcular as interações proteína-ligante. Esta última abordagem é bastante utilizada na descoberta de fármacos [2].

para garantir que a modelagem molecular simule com precisão sistemas com a maior proximidade da realidade, é crucial definir um modelo que descreva adequadamente o comportamento do sistema, considerando as interações intermoleculares e intramoleculares. A definição do método de modelagem pode ser baseado em três critérios, sendo eles: o número de átomos do sistema, as propriedades a serem estudadas e o custo computacional. Essas etapas se tornam fundamentais, uma vez que a escolha do método mais otimizado para a aplicação permite uma diminuição do custo computacional do projeto e do seu tempo de execução [5].

A utilização do *ensemble* no campo de aprendizado de máquina tem ganhado popularidade devido à sua eficiência na busca por maior precisão nos seus resultados [11]. O *ensemble* realiza a combinação de múltiplos modelos de aprendizado de máquina com o intuito de formar uma previsão conjunta, superando as limitações dos modelos individuais. Além disso, essa abordagem reduz o risco de ocorrência de *overfitting* [16], uma vez que a combinação de diferentes modelos contribui para uma generalização mais eficaz.

Com base nisso, a aplicação de modelos baseados em *ensemble* surge como uma estratégia promissora para aprimorar a capacidade de generalização dos resultados das funções de escore no contexto de docagem molecular. Dada a complexidade inerente a esse campo, onde existem diversas variáveis, cada função de escore pode lidar com uma característica específica do problema. A utilização de um modelo de *ensemble* visa atingir um equilíbrio entre essas funções, fortalecendo as vantagens individuais de cada uma delas. Cada função de escore pode oferecer percepções valiosas em aspectos específicos da docagem molecular como interações moleculares, propriedades químicas e características estruturais.

1.1 Objetivo

1.1.1 Objetivo geral

Desenvolver e avaliar modelos de *ensemble* para a estimar a afinidade de ligação em docagem molecular proteína-ligante.

1.1.2 Objetivos específicos

- Aquisição e Pré-processamento de Dados: obter dados a partir de repositórios de dados, como o PDBbind, e realizar o pré-processamento necessário para garantir a consistência e qualidade dos dados, considerando diferentes tipos de resultado e diferença nos dados utilizados para cada função.
- Obter uma função ponderada das funções de escore, atribuindo pesos específicos a

cada função de escore, considerando sua performance e contribuição na predição da afinidade de ligação.

- Realizar uma análise exploratória detalhada dos dados com o objetivo de identificar padrões consistentes entre os resultados obtidos.
- Comparar as métricas de desempenho (RMSE, correlação de Pearson) do modelo de *ensemble* com os resultados individuais de cada função de escore e destacar diferenças significativas nas performances.

1.2 Organização

O presente trabalho está dividido da seguinte forma: na Seção 2 é possível encontrar a fundamentação teórica necessária para entendimento do trabalho. Seção 3 apresenta a metodologia utilizada para o desenvolvimento do trabalho. Na Seção 4 encontram-se os resultados preliminares. A Seção 5 apresenta a conclusão e trabalhos futuros.

2 REFERENCIAL TEÓRICO

Nesta seção será apresentado o referencial teórico necessário para o entendimento do presente trabalho. Na seção 2.1 é descrito o que concerne o aprendizado de máquina onde são detalhados os conceitos de aprendizado supervisionado (classificação e regressão) e aprendizado não supervisionado. Na seção 2.2, são apresentados os fundamentos e conceitos relacionados à *ensemble* descrevendo as técnicas de Voting, Bagging e Boosting e os respectivos algoritmos considerados neste trabalho. Na seção 2.3 é descrito o que é a bioinformática. Na seção 2.4 são apresentados os banco de dados biológicos com alguma relação com este trabalho: PDB, EBI-AlphaFold e PDBeBind. Na seção 2.5 é explicado o básico sobre docagem molecular, detalhando a função de busca (seção 2.5.1 e as funções de escore 2.5.2. Nesta seção 2.5.2 são detalhados os tipos de funções de escore existentes e, para cada tipo, são descritas as funções que foram utilizadas neste trabalho. Na seção 2.5.4 são descritas as *benchmarks* utilizadas para validação dos dados em funções de escore. Para finalizar o referencial sobre docagem molecular, a seção 2.5.5 descreve algumas aplicações desta importante área de pesquisa em bioinformática. Por fim, na seção 2.6 são apresentados os softwares e pacotes utilizados no presente trabalho.

2.1 Aprendizado de máquina

O aprendizado de máquina (AM) é uma subárea da inteligência artificial a qual tem como objetivo desenvolver técnicas computacionais capazes de adquirir conhecimento de forma automática. Um algoritmo de AM aprende tomando decisões a partir da identificação de padrões em dados e prediz comportamentos baseados em dados de problemas anteriores. AM estuda métodos computacionais capazes de adquirir novos conhecimentos, novas habilidades e meios de tratar e organizar o conhecimento já existente [44].

Existem diversas abordagens de aprendizado que podem ser utilizadas por um sistema computacional que simulam o aprendizado humano como, por exemplo, o aprendizado por hábito, por instrução, por dedução, por analogia e por indução [50].

A indução é uma forma de inferência lógica a qual possibilita tomar conclusões ba-

seadas em um conjunto particular de dados. Na indução, um conceito é aprendido realizando a inferência indutiva sobre os dados apresentados [46]. Porém, as hipóteses geradas através de inferência indutiva nem sempre podem prever o resultado correto. Ainda assim, a inferência indutiva pode ser considerada um dos principais métodos utilizados para prever eventos futuros. Apesar da indução ser o recurso mais utilizado nas redes neurais de um humano, ela é limitada em relação à dados amostrais insuficientes ou dados que não tragam vantagem ao modelo preditivo [33]. Esse tipo de aprendizado indutivo pode ser dividido em duas categorias: supervisionado e não supervisionado.

2.1.1 Aprendizado supervisionado

O aprendizado supervisionado são técnicas cujo aprendizado ocorre a partir de um conjunto de treinamento de entrada do tipo exemplo entrada-saída. Cada exemplo é um par que consiste em um objeto de entrada, normalmente um vetor de características ou atributos, e um valor de saída (variável alvo). O atributo alvo pode ser um valor numérico (regressão linear) ou um rótulo de classe (classificação). Utiliza-se os valores de entrada passados da variável alvo para prever quais devem ser seus resultados de saída. Estes mesmos valores servem como “supervisão” destas previsões, permitindo o ajuste nas previsões com base nos erros, ou seja, o modelo possui uma referência daquilo que está certo ou errado.

2.1.1.1 Classificação

A classificação é uma tarefa fundamental no campo de aprendizado de máquina, na qual visa realizar a atribuição de rótulos ou categorias a diferentes instâncias com base em suas características distintas. Esse processo desempenha um papel importante em diversas áreas, como diagnóstico médico, reconhecimento de padrões e identificação de objetos em imagens.

Nessa abordagem, as classes representam as diferentes categorias que o modelo tenta prever, enquanto as instâncias de dados são descritas por um conjunto específico de características ou atributos. A escolha criteriosa desses atributos é vital para o desempenho eficaz do modelo de classificação.

A classificação pode ser dividida em dois principais tipos: binária, quando envolve a categorização entre duas classes distintas, e multiclasse, onde o modelo precisa classificar instâncias em mais de duas categorias.

Temos como exemplos de algoritmos de classificação: Dois exemplos notáveis são o *Support Vector Machine* (SVM) e Redes Neurais (RN).

O *support vector machine* é um classificador linear binário não probabilístico. Sua abordagem é baseada na criação de hiperplanos em um espaço multidimensional dos dados, buscando encontrar uma linha de separação eficiente. Essa linha é projetada para maximizar a distância entre os pontos mais próximos de cada classe. Essa característica

torna o SVM particularmente eficaz em problemas de classificação linear [13, 66].

Redes Neurais artificiais são classificadores conexionistas inspirados no cérebro humano, e são aplicados em várias tarefas de aprendizado, incluindo problema de classificação. São modelos muito flexíveis, e são compostas de uma topologia e seus parâmetros. A topologia define o número de camadas, o número de nós em cada camada e as camadas de entrada e de saída. Os parâmetros para tal topologia são aprendidos a partir dos dados durante o processo de formação [73].

2.1.1.2 Regressão

A regressão é uma técnica usada particularmente quando o objetivo é prever um valor contínuo com base em dados de entrada. Diferentemente da classificação, onde o objetivo é atribuir rótulos ou categorias a instâncias, a regressão busca encontrar uma relação matemática que represente a dependência entre variáveis.

Em problemas de regressão, o conjunto de dados é composto por pares ordenados, consistindo em entradas e as saídas correspondentes, que são valores contínuos. O objetivo é treinar um modelo capaz de generalizar essa relação para fazer previsões precisas em novos dados não vistos.

2.1.2 Aprendizado não-supervisionado

No aprendizado não-supervisionado, não existem informações da variável alvo desejada, e esses dados são chamados de dados não-rotulados. O algoritmo não recebe durante o treinamento os resultados esperados, ou seja, o algoritmo descobre por si só, por meio de exploração de dados os possíveis relacionamentos entre eles [46]. Neste caso, o processo de aprendizado visa identificar regularidades entre os dados com o intuito de agrupá-los em funções similares entre si. As principais técnicas baseadas em aprendizado não supervisionado são:

2.1.2.1 Agrupamento

Agrupamento é uma abordagem em aprendizado de máquina que visa identificar padrões intrínsecos e estruturas em conjuntos de dados, sem a presença de rótulos ou categorias predefinidas. Em vez de prever saídas específicas, o objetivo é explorar a estrutura interna dos dados e agrupar instâncias semelhantes.

Um exemplo de algoritmo utilizado para realizar o agrupamento é o *k-means* [30], o qual é um algoritmo de clusterização que visa particionar n observações em k clusters, na qual cada uma das observação pertence ao cluster com a média mais próxima do grupo pertencente, resultando em uma divisão de espaço chamado Diagrama de Voronoi [25].

2.1.2.2 Associação

A técnica de mineração de dados definida como regras de associação podem trabalhar com um número grande de atributos e gerando diversas alternativas de combinações entre eles. Essa técnica consiste em explorar relações frequentes entre diferentes atributos em grandes conjuntos de dados. Seu objetivo principal é identificar padrões e associações entre itens, fornecendo *insights* valiosos sobre a ocorrência de características específicas [8].

Um exemplo de algoritmo utilizado para realizar a associação é o *Apriori*. [1] Esse algoritmo realiza buscas sucessivas em toda a base de dados, mantendo um ótimo desempenho em termos de tempo de processamento.

2.2 Ensemble

O *ensemble* no contexto de aprendizado de máquina refere-se à estratégia de combinar as previsões de diversos modelos para aprimorar a precisão. A essência dessa abordagem consiste na diversidade e interdependência entre os modelos individuais, visando reduzir limitações e melhorar a capacidade de generalização. O *ensemble* oferece vantagens significativas, incluindo a redução de *overfitting*, melhoria na precisão das previsões e maior estabilidade diante de dados ruidosos [19].

Existem várias técnicas de *ensemble*, cada uma com suas características distintas. Sendo elas:

2.2.1 Voting

Nesse método de ensemble, cada modelo contribui na votação para a decisão final sobre a classificação de um determinado dado de entrada. Nesse método, a votação em geral, pode ser feita com base na maioria (a classe é determinada pela maioria dos votos dos modelos individuais) ou uma votação ponderada (nesse caso, cada modelo pode ter um peso na decisão final).

2.2.2 Bagging

O *ensemble bagging* [53] é uma técnica de *ensemble* que busca melhorar a estabilidade e a generalização dos modelos de aprendizado de máquina. Sua abordagem envolve a criação de múltiplos conjuntos de dados de treinamento, selecionados aleatoriamente com reposição a partir do conjunto de dados original. Modelos individuais são treinados em cada subconjunto, e suas previsões são combinadas para produzir uma decisão final, como pode ser visualizado na Figura 1.

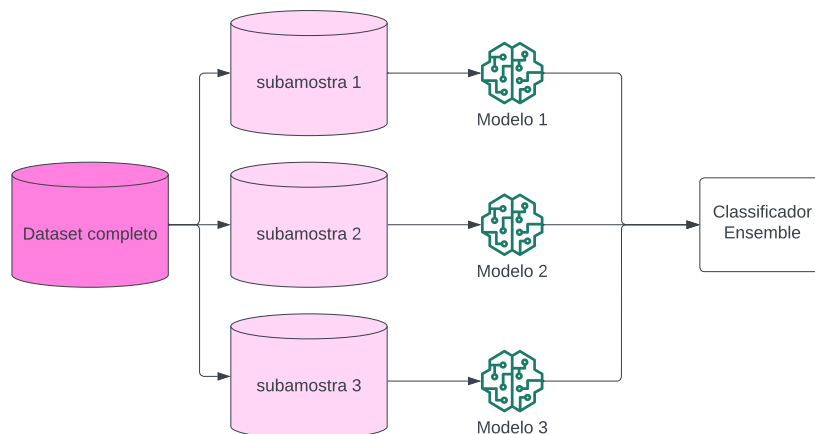


Figura 1: Fluxograma do *ensemble bagging*.

O principal objetivo do *ensemble bagging* é reduzir a variância do modelo, reduzindo a sensibilidade a padrões específicos do conjunto de dados de treinamento. Ao introduzir aleatoriedade na formação de subconjuntos, o *bagging* promove a diversidade entre os modelos, resultando em uma combinação menos propensa a *overfitting*, visto que a aleatoriedade pode ser o principal simulador de dados externos [9].

A sua principal diferença em relação ao método *voting* está na forma como as previsões dos modelos individuais são combinadas: o *bagging* usa uma abordagem de votação majoritária com pesos iguais, enquanto o *voting* pode usar uma abordagem de votação majoritária ou ponderada, onde os pesos são atribuídos com base na confiabilidade dos modelos.

2.2.2.1 *Random Forest*

O *Random Forest*, proposto por Breiman [10], é uma combinação de " n " árvores preditivas, de tal forma que cada uma destas árvores depende dos valores de um vetor aleatório de amostra independente e com a mesma distribuição de todas as árvores na floresta. Com isso, são gerados diversos resultados diferentes, e estes são englobados em uma função que calcula um único resultado final, a qual será nossa predição.

Para problemas de classificação, a saída do *Random Forest* é a classe selecionada pela maioria das árvores. Enquanto nos problemas de regressão, o retorno da saída se dá através da média ou da previsão média das árvores individuais, como pode ser visualizado na Figura 2.

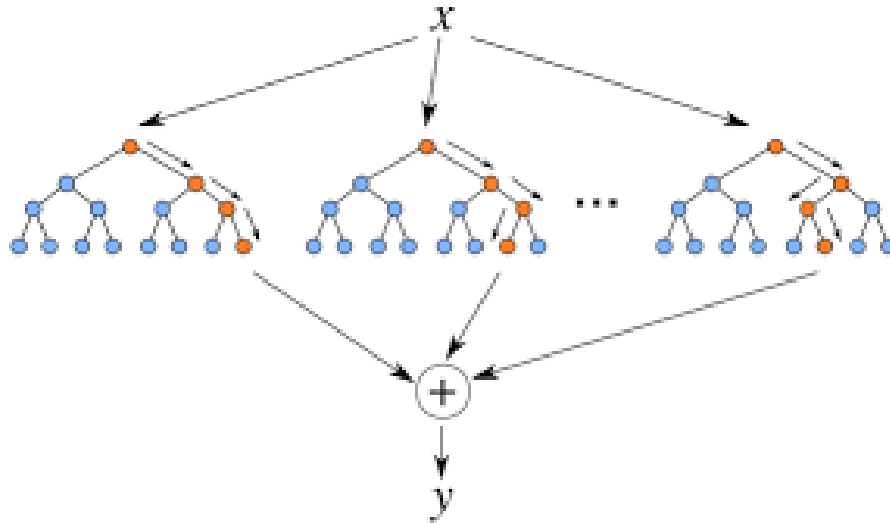


Figura 2: *Random Forest*

Uma vez que o presente trabalho prediz os valores utilizando uma regressão linear, a função de previsão utilizada será a média (*mean prediction*). Essa função, além de melhorar a precisão preditiva final, ajuda no controle do sobreajuste das árvores geradas.

2.2.3 *Boosting*

O *ensemble boosting* [53] é uma técnica de combinação de modelos que visa melhorar a capacidade preditiva de algoritmos de aprendizado de máquina por meio da criação sequencial de modelos, onde cada novo modelo é treinado para corrigir as deficiências dos modelos anteriores, como pode ser visualizado na Figura 3.

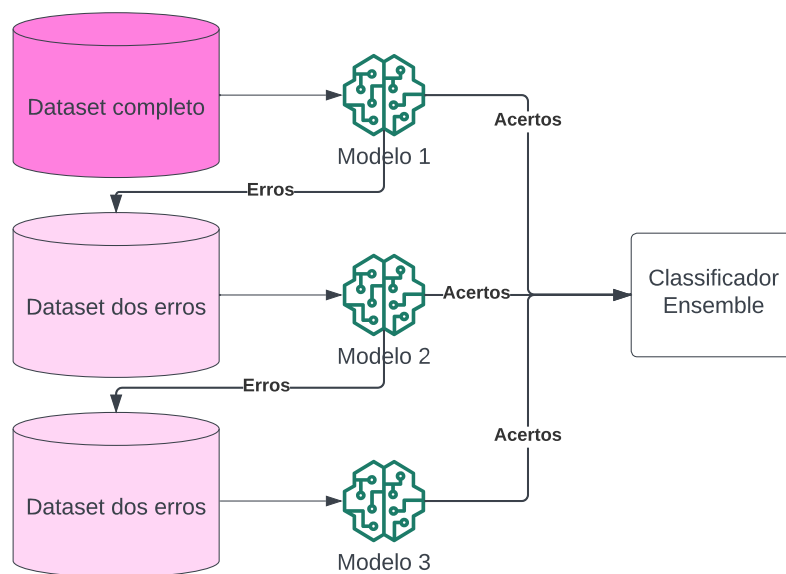


Figura 3: Fluxograma do *ensemble boosting*.

Os modelos são treinados iterativamente, com um foco maior nas instâncias classificadas incorretamente pelos modelos anteriores. Isso direciona o aprendizado para as áreas onde o modelo está tendo dificuldades.

2.2.3.1 XGBoost

O XGBoost (eXtreme Gradient Boosting) [14] é utilizado no presente trabalho, e é um algoritmo que se destaca por sua eficácia em problemas complexos de aprendizado de máquina. Ele representa uma evolução no conceito de *ensemble boosting*, combinando diversas técnicas para otimizar o desempenho.

Uma das principais características que faz o XGBoost uma ótima escolha para o presente trabalho é sua capacidade de lidar com conjuntos de dados de grande escala e alta dimensionalidade. Ele consegue fazer isso através de uma implementação eficiente que aproveita otimizações computacionais e técnicas de paralelismo. Outra vantagem do XGBoost é sua capacidade de lidar com dados desbalanceados e ruídos nos dados, através de técnicas de regularização e ajuste nos seus parâmetros que colaboram no sobreajuste e melhora a generalização do modelo.

Contudo, o XGBoost se torna uma ferramenta essencial devido a sua versatilidade e a sua capacidade de lidar com uma ampla variedade de problemas, combinando com sua alta eficiência computacional.

2.3 Bioinformática

A bioinformática consiste na análise computacional de grande variedade de dados biológicos, como sequências de DNA, RNA e proteínas [22]. Essa linha de pesquisa tem um caráter multidisciplinar, podendo ser definida como a combinação de conhecimentos da biologia, computação, estatística e a aplicação de técnicas de computação para a manipulação, organização, análise e interpretação de dados biológicos como sequências de DNA, proteínas, redes de interação entre genes e proteínas, entre outros [40].

2.4 Banco de dados biológicos

Na bioinformática aplicada, a coleta de dados e de informações biológicas é um dos fundamentos mais importantes. O armazenamento de dados torna-se essencial diante do significativo aumento no volume de dados ao longo dos anos, além da necessidade de tratamento desses dados através de buscas, processamentos e análise de resultados. Esses bancos de dados biológicos utilizam sistemas de gerenciamento de dados para manter acessíveis e funcionais informações de sequências, anotações genômicas, dados de complexos proteicos e suas interações, dentre outros exemplos, tendo uma vasta utilidade para a comunidade científica, principalmente na parte de avaliação do uso de fármacos, interações moleculares e funcionais de proteínas.

Entre os muitos bancos de dados biológicos, os que estão sendo utilizados no contexto deste trabalho são bancos de dados de proteínas, de moléculas e de complexos proteína-ligante, conforme descritos a seguir:

2.4.1 BD de Proteínas: Protein Data Bank

O *Protein Data Bank* (PDB) [4, 7] é um banco de dados de estruturas 3D de macromoléculas biológicas determinadas experimentalmente. Em março de 2024 tem mais de 216 mil estruturas determinadas experimentalmente e mais de 1 milhão de estruturas determinadas computacionalmente.

2.4.2 BD de complexos proteína-ligante: PDBbind

O PDBbind tem como objetivo fornecer uma coleção de dados de afinidade de ligação medidos experimentalmente e coletados da literatura para complexos proteína-ligante depositados no *Protein Data Bank*. O mesmo é um repositório de dados biológicos que armazena de maneira lógica e estruturada as coleções de afinidade de ligações medidas de forma experimental para complexos biomoleculares. Os conjuntos de dados que são mais utilizados são denominados por "PDBbind refined set" e "PDBbind general set", os quais serão utilizados no presente trabalho. Esses dados são imprescindíveis para estudos *in silico* sobre reconhecimento molecular e na descoberta de novos fármacos.

O banco de dados PDBbind foi originalmente desenvolvido por Wang [72], e tem sido

atualizado anualmente para acompanhar o crescimento do *Protein Data Bank* [4].

2.4.2.1 Afinidade de ligação medida experimentalmente pKd

Os dados do PDBBind correspondem a um conjunto de complexos proteína-ligante (estruturas tridimensionais da proteína, do ligante e do sítio de ligação em diferentes formatos) acompanhados da afinidade de ligação do complexo medida experimentalmente.

A afinidade de ligação determina a força com a qual uma pequena molécula é incorporada dentro de um sítio de ligação da proteína-alvo. Ela é medida pela constante de dissociação de equilíbrio Kd , onde valores menores (mais negativos) representam maior estabilidade na interação [70]. Como os valores das afinidades de ligação são representados em potências de 10, para sua melhor interpretação, eles podem ser expressos como o negativo do logaritmo da constante de dissociação de equilíbrio Kd , ou seja, $pKd = -\log_{10}Kd$. Nos dados do PDBBind, independente da versão, os valores de energia de interação experimental pKd dos complexos são fornecidos e neste trabalho corresponde ao atributo-alvo no treinamento das funções de escore

2.5 Docagem Molecular

A docagem molecular é uma ferramenta para prever a melhor conformação e orientação de uma pequena molécula (ligante) no sítio de ligação de um receptor (proteína-alvo), caracterizando assim a interação de um complexo proteína-ligante [42]. Essa prática se torna cada vez mais importante na descoberta de novos fármacos, onde uma das principais etapas é a análise da interação entre um candidato a inibidor (ligante) e a proteína-alvo realizada pela docagem molecular.

A figura 4 representa um exemplo de resultado de docagem molecular mostrando a posição final de uma determinada pequena molécula (na figura desenhada no formato *licorice*, colorido por tipo de átomo) no sítio de ligação do receptor (em ciano no formato *New Cartoon*). Em destaque na Figura, tem-se uma aproximação dessa melhor posição do ligante após a busca e de acordo com o escore de ligação do software de docagem utilizado.

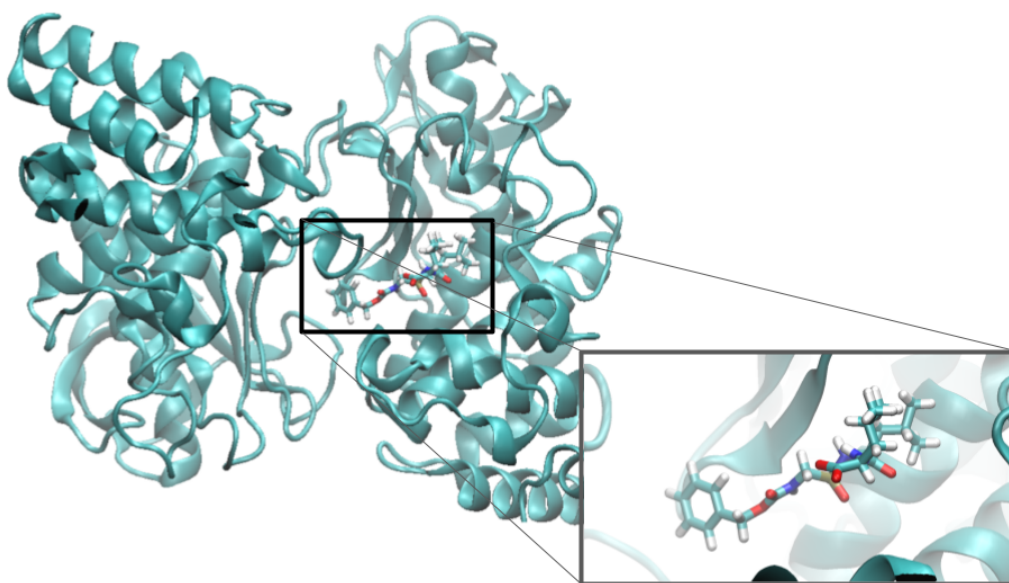


Figura 4: Exemplo de docagem molecular. A proteína tem código PDB 5TMN. A proteína está no formato *New Cartoon* em ciano e o ligante no formato *Licorice* colorido por tipo de átomo. A figura foi gerada com o software VMD.

O processo de desenvolvimento de métodos de docagem molecular está relacionado à realização de suposições e otimizações que estimem modelos preditivos aceitáveis para calcular essa interação entre a proteína-alvo e o ligante [49]. Essas otimizações são necessárias para se propor conformações/posições do ligante e interações com o receptor dentro de um tempo computacional aceitável.

O processo de docagem molecular envolve duas etapas, as quais serão descritas a seguir:

2.5.1 Busca

Na etapa de busca, os ligantes são posicionados em diferentes posições possíveis no sítio de ligação do receptor. A partir disto, o algoritmo de *docagem* avalia essas diferentes posições e conformações que o ligante pode adquirir no sítio de ligação do receptor-alvo. Porém, com a restrição de recursos computacionais, se torna inviável de avaliar todas essas possibilidades. Para contornar isto, podem-se utilizar algoritmos sistemáticos, estocásticos ou determinísticos.

Os algoritmos de busca para docagem molecular sistemáticos consistem em uma grade de valores para cada grau de liberdade e esses valores são explorados de forma combinatória durante a execução do algoritmo. Porém, conforme o número de graus de liberdade aumenta, consequentemente o número de avaliações também aumenta. Com isto, é importante que existam critérios de parada a fim de evitar que o algoritmo apresente soluções errôneas [23].

Os algoritmos estocásticos utilizam-se da aleatoriedade, alternando o grau de liber-

dade do sistema por vez, o que resulta na incerteza da convergência no seu resultado. Com o intuito de melhorar a convergência, o mesmo realiza diversas execuções independentes [47].

2.5.2 Funções de Score

A outra etapa da docagem molecular se refere ao cálculo da afinidade de ligação estimada, avaliado com a utilização de funções de escore. A função de escore é um modelo que calcula a pontuação da energia livre de ligação estimada entre um receptor e um ligante [28]. Sua utilização se dá na área de triagem virtual e otimização de descoberta de fármacos.

2.5.3 Tipos de funções de escore

As funções de escore foram desenvolvidas e otimizadas a partir de diferentes abordagens. Essas abordagens podem ser classificadas nas seguintes categorias:

2.5.3.1 *Physics-based*

As funções de escore *Physics-based* incluem métodos baseados em campos de força, os quais calculam a energia de ligação e interações de Van der Waals e eletrostáticas entre os pares de átomos da proteína-ligante, considerando a entalpia para a energia [43]. Entretanto, essa abordagem não considera a entropia e o efeito do solvente, pois não são precisos o suficiente para um resultado confiável [23]. Com a otimização dessas funções ao longo do tempo, foi introduzida a entropia de torção dos átomos dos ligantes e modelos de solvatação, porém a dificuldade de localizar os parâmetros relacionados à forma funcional da energia potencial impacta na precisão da predição da energia de ligação [35].

Além disto, existem as abordagens baseadas em mecânica quântica (QM), cujo intuito é lidar com a dificuldade de interações covalentes, polarização e transferência de carga no docking. Essas funções de escore apresentam uma maior precisão em relação às citadas anteriormente, porém o seu custo computacional é maior. A partir disto, se usa uma abordagem híbrida utilizando a mecânica molecular para reduzir seu custo computacional, e elas são denominadas como mecânica quântica/mecânica molecular (QM/MM) [58]. Em razão disto, esses tipos de funções de escore baseadas na física podem calcular diretamente as interações entre átomos de proteínas e ligantes com um potencial de predição consideravelmente maior sobre os outros métodos. Isso ocorre devido a consideração de entalpia, solvatação e entropia [35].

2.5.3.2 *Empirical*

As funções de escore empíricas foram desenvolvidas para reproduzir dados experimentais de afinidades [55], onde utiliza-se a ideia de que é possível correlacionar a energia

livre de ligação à um conjunto de variáveis não relacionadas. Os coeficientes associados aos termos funcionais são obtidos por meio de análise de regressão a partir da utilização de dados de afinidade já conhecidos de estruturas experimentais. Usualmente é utilizado um conjunto de treinamento de afinidades de ligação com o intuito de otimizar os pesos de fatores de energia nas suas regressões lineares [26].

Comparada às funções de escore do tipo *physics-based*, são muito mais rápidas em cálculos de afinidade, e isso se dá por conta de seus termos de energia simples e sua forma funcional intuitiva [34]. Um dos seus gargalos é a dependência do conjunto de treinamento, devido ao seu caráter de ajuste das afinidades de enlace de um pequeno conjunto de dados. Entretanto, com o aumento do número de complexos proteína-ligante conhecidos e afinidades de ligação, se torna possível o desenvolvimento de uma função relativamente genérica [35]. Neste trabalho foram utilizadas duas funções do tipo Empírica, a função do programa Autodock Vina e a função Lin_F9 que são escritas a seguir.

Vina

A função de escore da ferramenta AutoDock Vina [68] foi utilizada neste trabalho e é uma função de escore do tipo híbrida (Empírica e Knowledge-based). Ela é composta por um somatório de termos que modelam diferentes tipos de interação entre os átomos do ligante e do receptor como por exemplo termos que contêm funções gaussianas que modelam a interação entre diferentes tipos de átomos, termos para considerar a interação hidrofóbica e interações de ligação de hidrogênio, um termo que modela a repulsão entre diferentes tipos de átomos e um termo que penaliza no caso da rotação excessiva de certas ligações entre átomos [68].

Lin_F9

Lin_F9 [75] é uma função de escore empírica linear. Utiliza-se de uma combinação linear de 9 termos empíricos, e sua pontuação total é calculada somando ponderadamente esses termos. A função disponibiliza os pesos e os termos, como podemos visualizar na figura 9.

```

Weights      Terms
-0.032879    gauss(o=0,_w=0.5,_c=8)
0.244318     repulsion(o=0,_c=8)
-0.007459    hydrophobic(g=0.5,_b=1.5,_c=8)
-0.170027    non_dir_h_bond(g=-0.7,_b=0,_c=8)
-0.000896    gauss(o=2.5,_w=0.5,_c=12)
-0.006915    gauss(o=5,_w=0.5,_c=12)
-3.356       non_dir_m_bond(g=-2.7,_b=-1.3,_c=8)
0.240385     num_tors_add
0.02         num_tors_Nlog

Affinity: -4.57471  -2.56163 (kcal/mol)
RMSD: 1.04438
Refine time 0.01508
Loop time 0.01581

```

Figura 5: Pesos e termos da função Lin_F9

2.5.3.3 Knowledge-based

As funções de escore *knowledge-based*, também conhecidas como função de escore baseadas em estatística, são baseadas na análise estatística de pares de átomos em interações de complexos proteína-ligante com estruturas 3D disponíveis. Os átomos são utilizados par-a-par e são convertidos em um potencial de força, a qual descreve a a melhor geometria dos pares de átomos da proteína-ligante. São obtidos diretamente da frequência de ocorrência dos pares de átomos em um banco de dados utilizando a relação inversa de Boltzmann [67]. Temos como exemplo dessas funções o DrugScore [69] e PMF [48].

Em comparativo às funções baseadas em *physics* e *knowledge-based*, essas funções fornecem um meio termo entre precisão e velocidade [24]. Neste trabalho estão sendo utilizadas duas FS do tipo Knowledge-based KORP-PL e Convex-PL.

KORP-PL

KORP-PL [27] é uma função de escore do tipo *knowledge-based*. Foi treinada utilizando o conjunto de dados do PDDBind 2016. Nesta função de escore foram coletadas estatísticas a respeito das distâncias e ângulos entre um plano orientado construído na cadeia principal de um resíduo e um átomo de ligante para cada resíduo e átomo de ligante dentro de um certo limiar de corte. A partir dessas estatísticas de contato calculadas para aproximadamente 40 diferentes combinações de tipos de átomos na proteína e no ligante são derivados potenciais de interação.

Convex-PL

Convex-PL [26] é uma função de escore *knowledge-based* para interações proteína-ligante. Sua abordagem utiliza informações estruturais para desenvolver uma função de escore que avalia a afinidade entre proteínas e ligantes durante interações moleculares. O procedimento de treinamento da função é realizado através do conjunto de dados extraídos

do banco de dados PDDBindCN 2015, onde são gerados decoys com deformações rígidas do ligante.

2.5.3.4 Machine learning

As funções de escore baseadas em aprendizado de máquina criam modelos preditivos a partir de descritores que podem ser obtidos das propriedades do ligante, da proteína e do complexo [37]. Esses descritores podem ser interações específicas, interações eletrostáticas, ligação de hidrogênio [36], descritores geométricos [28], peso molecular, número de ligações rotacionais, estrutura da proteína [31], entre outras.

Diversos algoritmos de aprendizado de máquina podem ser utilizados com base nesses descritores, como os citados na seção 2.1, entre outros. Com isso, surge a necessidade de um conjunto de treinamento de complexos proteína-ligante com estruturas conhecidas e com o valor experimental da interação proteína-ligante a fim de obter seus modelos [2].

Na figura 6 podemos visualizar um fluxograma de execução de uma função de escore baseada em aprendizado de máquina.

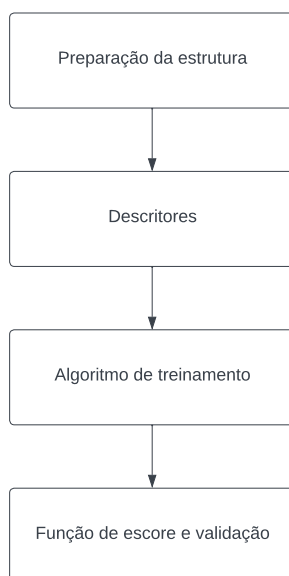


Figura 6: Funcionamento de uma função de escore baseada em aprendizado de máquina.

As funções de escore baseadas em aprendizado de máquina diferem dos tipos *knowledge-based* e *empirical*, que também se utilizam de métodos de aprendizado de máquina. Porém, esses tipos não se utilizam de métodos necessariamente lineares [64]. Também se utilizam de um número de descritores maiores que os termos das funções *empirical*. Além disto, podem utilizar a seleção de atributos a fim de obter seu modelo final [71]. E por fim, ao invés de utilizar termos que tenham um significado físico interpretável

[17], partem da razão de combinar um conjunto de dados de descritores de modo que uma função de escore baseada em descritores em uma busca cega [18].

Neste trabalho foram considerados os escores gerados por duas FS baseadas em AM: GB_score e ECIF-GBT que são descritas a seguir.

GB-score

O GB-Score [61] é uma função de escore baseada em aprendizado de máquina a qual utiliza a distância entre diferentes tipos de átomos de ligante e proteína para a obtenção dos seus descritores. Utiliza-se do conjunto de dados PDBbind-v2019 e o do algoritmo *Gradient Boosting Trees* para seu treinamento. Nesta função, é possível encontrar outros modelos treinados pela função, embora os pesos específicos dos modelos não sejam disponibilizados. No entanto, os atributos utilizadas por esses modelos são conhecidas, permitindo a sua geração de pesos. Um exemplo presente na própria função são os algoritmos *Random Forest* e *Extra-Tree*. Isso significa que, mesmo sem a exposição direta dos pesos, é possível inferir informações sobre o modelo com base nos atributos utilizados. Para o presente trabalho foi utilizado ambos algoritmos gerando as funções RF-score e ET-score, respectivamente.

ECIF

Extended Connectivity Interaction Features (ECIF) [63] é uma função de escore baseada em aprendizado de máquina. A mesma consiste em uma série de descritores que utilizam pares de tipos de átomos proteína-ligante que levam em consideração a conectividade de cada átomo para definir seus valores. A função fornece, junto ao seu modelo, um conjunto de dados que pode ser utilizado para testar outros tipos de treino, permitindo que outros pesquisadores ou desenvolvedores avaliem a performance do modelo em diferentes cenários de treinamento. Para o presente trabalho foi utilizado os algoritmos *Gradient Boosting Trees* e *Random Forest* gerando as funções ECIF-GBT e ECIF-RF, respectivamente.

2.5.4 Benchmarks para validação de funções de escore

É imprescindível que existam *Benchmarks* para validação de desempenho das funções de escore, pois a partir delas podemos avaliar os resultado e realizar comparações de desempenho entre as funções de escore. Um desses *benchmarks* é o *Comparative Assessment of Scoring Functions* (CASF-2016) [65]. O mesmo utiliza-se de três conjuntos de teste para qualificar as funções de escore conforme quatro métricas, sendo elas:

2.5.4.1 Scoring power

Esta métrica utiliza-se de dados das afinidades medidas experimentalmente [65], e verifica a capacidade da função de escore de prever a afinidade de ligação por meio de uma correlação linear entre o valor experimental e o valor predito. Para medir os resultados, é utilizado o conjunto de teste PDBbind core set [38], o qual contém 285 complexos

proteína-ligante com os dados experimentais de suas estruturas cristalográficas originais [39].

O fluxo do processo para o cálculo do *Scoring Power* pode ser visualizado na figura 7

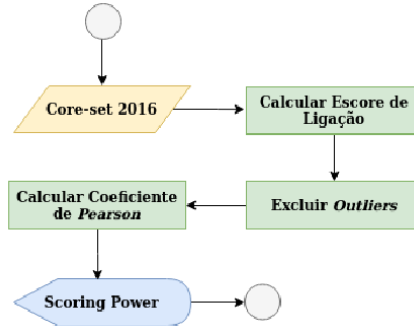


Figura 7: Fluxograma para calcular o *Scoring Power*. Fonte Arrúa (2020) [3]

Para a avaliação dos resultados é necessária a exclusão dos dados discrepantes. O CASF-2016 utiliza-se de um *script* para calcular suas métricas através do Coeficiente de Correlação de Pearson (R) [6] demonstrado na equação 1 e do Desvio Padrão (SD) demonstrado na equação 2.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Onde o x é o valor previsto e y é o valor experimental.

$$SD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}. \quad (2)$$

2.5.4.2 Ranking power

O *ranking power* se refere à capacidade de uma função de escore ranquear corretamente os ligantes conhecidos de uma determinada proteína a partir das poses de ligações (*binding poses*) fornecidas. Diferente do *scoring power*, o *ranking power* não exige que exista uma correlação linear entre as funções de escore e os dados experimentais de ligação [65]. É realizada a divisão do conjunto de treinamento PDBbind *core set* em 58 *cluster* de complexos, onde em cada um deles tem-se 5 complexos formados para uma proteína-alvo. Com isto, o objetivo da função de escore é classificar cada *cluster* de acordo com seu valor de energia estimado.

O *Spearman's rank correlation coefficient* demonstrado na equação 3, *Kendall's rank correlation coefficient* e *Predictive Index*, são utilizados para quantificar o *ranking power*. Essas três métricas são escolhidas pois existem uma boa correlação entre eles.

$$p = \frac{\sum_i (rx_i - r\bar{x})(y_i - r\bar{y})}{\sqrt{(\sum_i (x_i - r\bar{x})^2)} \sqrt{(\sum_i (y_i - r\bar{y})^2)}} \quad (3)$$

Onde rx é o *ranking* de ligação do complexo i e ry_i é a constante do ligante experimental do complexo.

2.5.4.3 Docking power

Refere-se à capacidade da função de escore em identificar a posição do ligante nativo entre os decoys gerados por computador. Para a avaliação do *Docking Power*, é utilizado o conjunto de CASF-2016 *docking set*, o qual é composto por 100 *decoys* com os seus valores de *binding pose* em RMSD para cada um dos complexos proteína-ligante do conjunto PDBbind *core set* [39]. Segundo Su et al. [65], o RMSD é a distância entre o centro de massa do ligante nativo e o ligante calculado, e pode ser visualizado na equação 4.

$$RMSD = \sqrt{\frac{\sum_{i=1}^N [(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2]}{N}} \quad (4)$$

A medida utilizada como métrica é a taxa de sucesso para identificar poses de ligação com RMSD abaixo do limite estabelecido pelo CASF-2016. A partir disto, os resultados de afinidade de cada grupo de complexos proteína-ligante são classificados de acordo com os melhores resultados e cruzados com os valores do RMSD. Se os valores de RMSD dos três primeiros lugares da classificação estiverem dentro do limite definido pelo CASF-2016, então esses resultados podem ser considerados verdadeiros. É então calculada a taxa de sucesso, definida como a soma de todas as vezes que os três primeiros colocados na classificação tem o RMSD menor do que o limite definido pelo CASF-2016. Na equação 5, podemos visualizar a taxa de sucesso.

$$TaxaDeSucesso = \frac{\sum_{i=1}^N x_i}{N} \quad (5)$$

2.5.4.4 Screening power

O *screening power* avalia dois tipos de desempenho, sendo eles: *forward screening power* e *reverse screening power*. O grupo de teste CAFS-2016 *screening set* é usado para calcular as duas funções de *screening power*, que inclui 57 grupos de teste composto por 285 conjuntos de complexos proteína-ligante, com cada conjunto incluindo 100 *decoys* para cada complexo.

O *forward screening power* refere-se à capacidade de uma função de escore proposta em identificar ligantes verdadeiros para uma proteína-alvo entre proteínas aleatórias. O *reverse screening power* é a capacidade do modelo preditivo para identificar proteínas-alvos potenciais para um ligante em específico [65]. O *forward screening power* é avaliado através da taxa de sucesso calculada pela média de ligantes verdadeiros entre moléculas aleatórias caso se enquadrem no topo das classificações, podendo ser 1%, 5% e 10%, e também é medido por *Enhancement Factor* (EF). Na equação 6 é possível visualizar o EF.

$$EF_{\alpha} = \frac{NTB_{\alpha}}{NTB_{total} * \alpha} \quad (6)$$

Onde NTB_{α} é o número de ligantes verdadeiros observados entre os candidatos com melhor classificação com $\alpha = 1\%$, 5% ou 10% , selecionados por uma função de pontuação. NTB_{total} é o número total de verdadeiros-ligantes para a proteína alvo dada, normalmente definido em cinco.

O *reverse screening power* é avaliado através da taxa de sucesso calculada pela média das proteínas-alvos potenciais identificadas que se enquadram nos melhores classificados, devendo estar nos topo 1% , 5% e 10% para serem consideradas verdadeiras.

O CASF-2016 considera o *screening power* como o aspecto mais fraco das funções de escore avaliadas, onde os modelos preditivos com pontuação mais altas atingem taxas de sucesso em torno de 40% [65].

2.5.5 Aplicações da Docagem Molecular e Funções de Score

2.5.5.1 Modo de ligação

Esta aplicação é responsável por determinar o modo de ligação do ligante no sítio de ligação da proteína. São geradas diversas conformações no sítio ativo da proteína. Com isto, as funções de escore agrupam os resultados avaliando a iteração em cada um dos complexos resultantes. Com base nesses resultados, pode-se entender o funcionamento molecular a fim de projetar medicamentos mais eficientes modificando a proteína ou o ligante [59].

Essa aplicação de docagem molecular é avaliada através da sua capacidade de distinguir modos de ligação nativos de estruturas falsas ou *decoys*. Os *decoys* se caracterizam como ligantes gerados computacionalmente, com estruturas geométricas incorretas cujo objetivo de uso é treinar as funções de score para identificar quais dessas moléculas são falsos positivos, utilizando isso nas avaliações do desempenho dos algoritmos de docagem molecular [20].

2.5.5.2 Afinidade de ligação

O conhecimento sobre as afinidades de ligação é de suma importância para se compreender as interações intermoleculares e estimar os processos biológicos [2]. A afinidade de ligação é a força da interação de ligação entre uma única proteína ou DNA e seu inibidor. A afinidade de ligação é usualmente medida pela constante de dissociação de equilíbrio, a qual é utilizada para avaliar e classificar as forças de interação biomoleculares. Quanto maior o seu valor, maior é a afinidade de ligação da complexo. A ligação de hidrogênio, as interações eletrostáticas e as forças de Van der Waals entre as moléculas são fatores que influenciam nessas interações.

2.5.5.3 *Triagem virtual*

Esta aplicação pode ser considerada a mais importante no desenho de medicamentos, pois devido ao alto custo da detecção experimental e a falta de disponibilidade de testes de alto rendimento, esse processo virtual desempenha um papel imprescindível na descoberta de fármacos [23].

Essa técnica tem como objetivo identificar estruturas com maior probabilidade de se ligarem a um determinado receptor ou enzima. Existem duas abordagens utilizadas na triagem virtual [23]. Na primeira, chamada de *Target-Based Virtual Screening*, os algoritmos de triagem de ligantes dentre um conjunto de moléculas dependem de informações sobre os alvos para que se possa estimar as probabilidades das interações. Na segunda abordagem, chamada de *Ligand Based Virtual Screening*, não existe esta dependência de informações sobre os alvos. Nesta abordagem, os descritores envolvem os próprios ligantes como atributos discriminantes nos algoritmos de seleção e classificação [12].

2.6 Linguagem de programação, pacotes e ferramentas

2.6.1 Linguagem de programação Python e dependências

O python é uma linguagem de programação de alto nível, imperativa, interpretada, orientada a objetos, funcional, de tipagem dinâmica e forte. É altamente abstrata e expressiva, uma vez que suas expressões são simples de serem entendidas e a combinação delas definem instruções mais complexas, afirmando assim o conceito de ortogonalidade. Por se tratar de uma linguagem de instruções simples e de fácil uso, a mesma se torna um ambiente agradável para o usuário [60].

O scikit-learn ou sklearn [56] é uma biblioteca da linguagem de programação Python desenvolvida de forma simples e eficiente para análise de dados e utilização de algoritmos de aprendizado de máquina, além de ser reutilizável em diferentes situações, possuindo o seu código aberto e sendo construída sobre os pacotes NumPy, SciPy e matplotlib.

O sklearn está organizado em muitos módulos, cada um desenvolvido para uma finalidade específica, dentre eles podemos citar:

- Pré-processamento – A parte mais trabalhosa e demorada no desenvolvimento de um modelo de aprendizado de máquina. Nessa etapa, normalmente se utilizam os pacotes NumPy e Pandas. Entretanto, o sklearn apresenta funcionalidades que auxiliam nesta etapa.
- Classificação – Apenas funções que desenvolvem modelos capazes de detectar a qual classe pré-determinada uma instância pertence. Podemos identificar se uma pessoa possui ou não determinada doença, ou qual doença uma pessoa pode ter dentre várias possíveis.

- Regressão – Refere-se aos modelos que podem atribuir um valor contínuo a uma instância. Possibilita prever o preço de um medicamento, quantidade de vendas de um produto, energia de interação entre uma proteína e um ligante.
- Clusterização – Desenvolvimento de modelos para detecção automática de grupos com características similares em suas instâncias. Pode-se encontrar grupos de risco de determinada doença, bem como verificar padrões entre sintomas de um paciente.
- Redução de dimensionalidade – Redução do número de variáveis de uma aplicação. Muito utilizado quando o conjunto de dados apresenta grande dimensionalidade. Com esta redução, podemos diminuir consideravelmente a quantidade de cálculos necessários em um modelo, por vezes aumentando a eficiência e reduzindo seu custo computacional.

2.6.2 Open Babel

O Open Babel [52] é uma biblioteca química de código aberto que oferece uma ampla gama de funcionalidades para manipulação de estruturas químicas, incluindo conversão entre formatos, cálculos de descritores moleculares e outras operações relacionadas à química computacional. Uma das principais funcionalidades do Open Babel é a capacidade de converter estruturas químicas entre diferentes formatos de arquivo, tornando-se fundamental no preparo de estruturas.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta as principais ferramentas e dados utilizados no trabalho e a seguir a metodologia proposta. A figura 8 apresenta a metodologia utilizada na elaboração deste trabalho, e cada uma das suas etapas será detalhada a seguir.

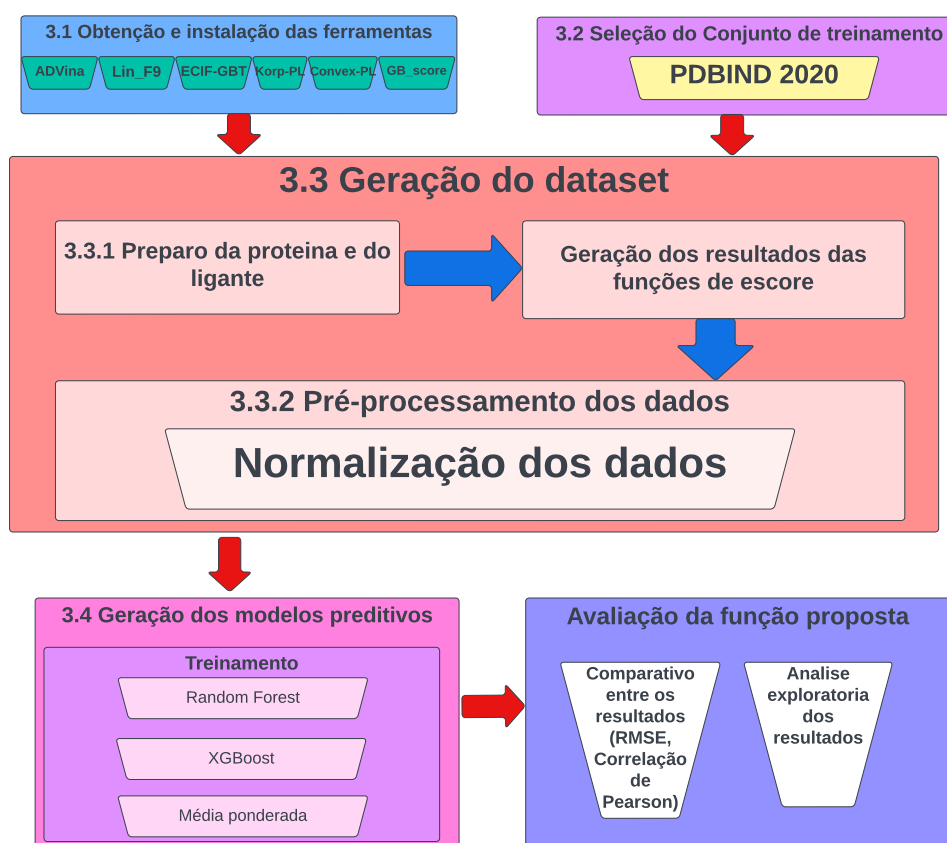


Figura 8: Metodologia da dissertação.

3.1 Obtenção e instalação das ferramentas

3.1.1 Funções de Escore

Na tabela 5 encontra-se um resumo das funções de escore utilizadas no presente trabalho.

Tabela 1: Resumo das funções de escore utilizadas.

Função de escore	Tipo da função	Ano	Autor
Korp-PL	Knowledge-based	2021	Kadukova, Grudin
Vina	Empirical + Knowledge-based	2010	Trott, Olson
GB_score	Machine learning	2023	Rayka, Firouzi
ECIF-GBT	Machine learning	2021	Sanchez et.al
lin_F9	Empirical	2021	Yang, Zhang
Convex-PL	Knowledge-based	2017	Kadukova et.al

Foram consideradas três versões da função GB-Score, uma considerando o algoritmo *Random Forest* (RF), que chamada neste trabalho de RB_Score_RF, uma utilizando o algoritmo *Extra-Tree*, definida neste trabalho como RB_score-ET e uma executada parametrizando como modelo o algoritmo *Gradient Boosting Trees* (GBT), nomeada neste trabalho de RB_Score_GBT.

Para a função de escore ECIF, foram considerados dois diferentes algoritmos de treinamento: ECIF-GBT, corresponde a versão desta função de escore considerando o algoritmo GBT, e ECIF-RF, cuja função de escore foi treinada considerando o algoritmo RF. Assim, um total de 9 funções de escore aplicadas neste trabalho para a geração do *dataset* para o *ensemble*: Convex-PL, Korp-PL, Vina, Lin_F9, RB_Score_RF, RB_score-ET, RB_Score_GBT, ECIF-GBT e ECIF-RF.

3.1.2 Instalação de dependências

Nesta etapa é realizada a instalação dos softwares e bibliotecas necessárias para a execução completa do objetivo proposto. Foi utilizado o python para a maioria das execuções se adequando aos "requirements.txt" de cada função individualmente. Esse arquivo é encontrado na maioria das funções de escore. Ele contém uma lista de pacotes Python necessários para executar os códigos das funções de escore, juntamente com suas versões específicas, se necessário. Isso permite que outros desenvolvedores e sistemas reproduzam facilmente o ambiente de desenvolvimento, garantindo que todas as dependências corretas estejam instaladas.

3.2 Seleção do conjunto de treinamento

As funções de escore baseadas em aprendizado de máquina são geralmente desenvolvidas e validadas a partir de um conjunto de dados que carrega consigo informações de

afinidade dos conjuntos de proteínas-ligantes as quais foram medidas e registradas experimentalmente [72]. Além disto, são utilizados outros tipos de dados para treinar e avaliar a docagem molecular, são os chamados *decoys*, os quais são estruturas artificiais de complexos proteína-ligante geradas artificialmente por um computador [51]. Essas bases de dados são extraídas do PDBbind e todas as funções escolhidas tem seus conjuntos de dados disponíveis de forma pública. Isso possibilita a geração de conjuntos de dados de teste com base nos atributos esperados por cada modelo, evitando o sobreajuste ao testar dados que foram utilizados no treinamento das funções. Garantir que os dados de teste sejam independentes dos dados de treinamento é crucial para uma avaliação justa e confiável do desempenho dos modelos.

Neste trabalho, o conjunto de dados de treinamento considerado foi o PDBBind em sua versão 2020.

3.2.1 PDBbind *general set* 2020

O conjunto de dados PDBbind *general set* inclui arquivos resumindo as informações básicas e arquivos processados para os complexos proteína-ligante. Cada complexo se encontra em uma pasta separada pelo seu código de quatro caracteres, na qual contém o arquivo PDB da proteína, arquivos nos formatos '.mol2' e '.sdf' referente ao ligante. Além dessas pastas, cada um dos complexos tem um arquivo que descreve os ligantes e o valor de energia de interação determinada experimentalmente pK_d . Este pacote de dados inclui 14.127 complexos no total. Pela razão do seu maior escopo de complexos disponíveis, esse é o conjunto de dados escolhido para elaboração do presente trabalho.

A Figura mostra a organização e estrutura do PDBBind utilizado. Nesta figura de exemplo tem-se 2 pastas, que corresponde a dois diferentes complexos, os quais é fornecido o valor do pK_d , que corresponde a energia de interação proteína-ligante medida experimentalmente. Para cada complexo, tem-se os arquivos da proteína, do ligante e o respectivo pocket (ou sítio) de ligação.

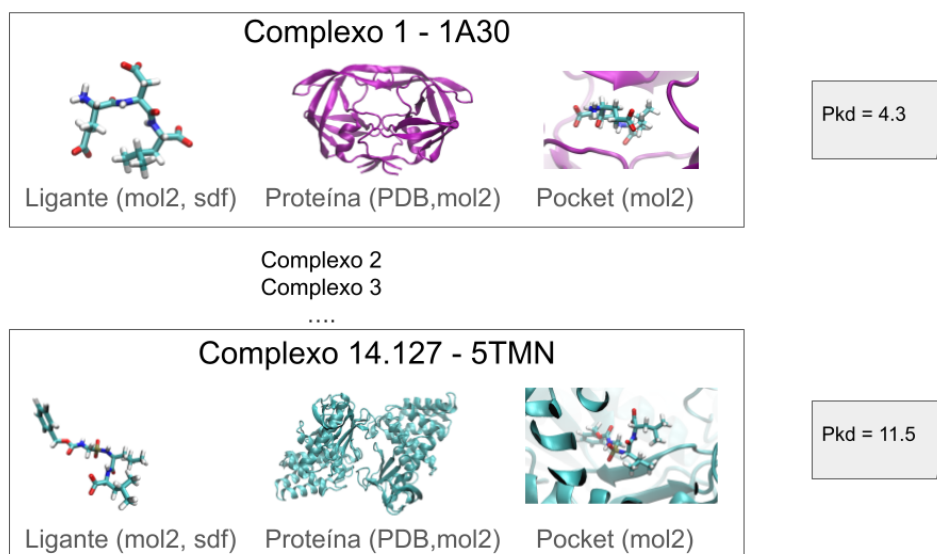


Figura 9: Organização e exemplo de dados do PDBBind 2020.

3.2.2 PDBbind *refined set* 2020

Para reduzir o tamanho do pacote de dados, o PDBbind *refined set* é elaborado, sendo ele um subconjunto do *general set*. Este pacote de dados inclui 5.316 complexos no total.

Essa base de dados foi usada para retrainar as funções de escore baseadas em machine learning. É fundamental que essa base seja distinta daquela empregada nas previsões dos modelos de ensemble. Isso evita o vazamento de dados, garantindo a integridade e a eficácia das análises.

O vazamento de dados ocorre quando informações do conjunto de dados de teste são incorporadas ao modelo durante o treinamento. Isso pode levar a uma superestimação do desempenho do modelo, uma vez que o modelo pode estar aprendendo padrões específicos dos dados de teste, o que dificulta a generalização do mesmo.

Portanto, é importante garantir que os dados de treinamento e teste sejam independentes e representativos, garantindo assim a generalização adequada do modelo para novos dados não vistos. Isso significa que a base de dados usada para treinar as funções de escore deve ser completamente separada daquela usada para avaliar o desempenho dos modelos de ensemble.

3.3 Geração do *dataset*

3.3.1 Preparo da proteína e do ligante

Na abordagem de docagem molecular o formato dos dados de entrada desempenham um papel fundamental no desempenho, na precisão dos resultados e no seu correto funcionamento. Para a preparação da proteína e do ligante, diferentes métodos exigem diferentes formatos. O PDBbind, por exemplo, adota os padrões de `pdb_protein.pdb`, `pdb_ligand.sdf`

e `pdb_ligand.mol2`.

Para funções de escore específicas, como o Autodock Vina, que requer o formato PDBQT, é essencial realizar a conversão adequada uma vez que o ligante é fornecido no formato mol2 e sdf. O Autodock Tools [68] oferece ferramentas integradas para essa tarefa, permitindo a conversão eficiente dos dados para o formato necessário.

No entanto, em funções de escore baseadas em machine learning, dependem de atributos extraídos dos dados moleculares. Nesse contexto, a etapa de ETC (Extração, Transformação e Carregamento) desempenha um papel crucial.

Na etapa de extração, as informações são retiradas dos dados brutos. No contexto de docagem molecular, isso pode envolver a extração das coordenadas tridimensionais dos átomos, identificação de características específicas da proteína e do ligante, bem como qualquer informação adicional crucial para a análise.

Após a extração, os dados passam por um processo de transformação. Isso inclui a conversão dos dados para atributos utilizados no modelo, a criação de representações numéricas ou categóricas relevantes, visto que dado um modelo já treinado, ele só irá funcionar se os atributos extraídos forem os mesmo usados no treinamento. A transformação visa colocar os dados em um formato consistente e prepará-los para análises.

A última fase é o carregamento dos dados, que envolve a inserção dos dados transformados em uma estrutura adequada para análise. Em abordagens baseadas em machine learning, esses dados geralmente são organizados em um conjunto de treinamento, validação e teste, prontos para serem alimentados nos modelos. No caso das funções de escore já treinadas, apenas o dataset de teste é extraído.

Assim, a preparação dos dados para docagem molecular não se limita apenas à conversão de formatos, mas também à extração eficiente de atributos que alimentarão algoritmos de machine learning, garantindo uma abordagem abrangente e eficaz no processo de descoberta de fármacos. No presente trabalho, não foi aplicado nenhuma modificação nas características das funções de escore baseadas em machine learning

3.3.2 Algoritmo para geração dos resultados das FS

No algoritmo 1, podemos visualizar o pseudo-algoritmo de execução do método para a geração dos *datasets*. Nele é possível acompanhar os tipos de arquivos passados em cada função.

Algorithm 1: Pseudo-algoritmo de execução da geração do dataset

```

lista_pdb ← [PDBbind_dataset];
funcao_de_escore ← ['Lin_F9', 'Vina', 'rb_score_ET', 'rb_score_GBT',
'rb_score_RF', 'ECIF_GBT', 'ECIF_RF', 'KORP-PL', 'Convex-PL'];
foreach pdb in lista_pdb do
  prepare_ligand(ligand.mol2)
  prepare_protein(protein.pdb)
foreach funcao in funcao_de_escore do
  foreach pdb in lista_pdb do
    resultado ← funcao(pdb.get(ligand),pdb.get(protein))
    resultado_dataset ← salva_resultado(resultado);
Dataset_normalizado ← Normaliza_dado(resultado_dataset)

```

Conforme mostra o algoritmo 1, inicia-se a geração do dataset a partir do PDBBind 2020 general set e do conjunto de 6 FS: Lin_F9, Vina, GB_Score, ECIF, KORP_PL e Convex_PL. Para cada complexo proteína-ligante do PDBBind 2020, o mesmo tem seus arquivos da proteína e do ligante preparados (de acordo com cada FS). A seguir, para cada complexo proteína-ligante preparado (já no formato de arquivo adequado), é calculado o respectivo escore de ligação com cada uma das FS consideradas neste trabalho.

A Tabela 3.3.2 mostra o resultado da execução da função de escore RB_Score_RF para todos os complexos do PDBBind 2020. Essa lista de escores corresponde a uma das colunas do *dataset*. O mesmo é realizado para cada uma das FS consideradas neste trabalho.

Tabela 2: Resultado da FS RB_Score_RF, uma das colunas do *dataset*.

Complex	RB_Score_RF
3zzf	5.587917
3gww	6.024462
...	...
2avi	7.149385

Após a execução de todas as FS é gerado o dataset utilizado como entrada no próximo passo da metodologia proposta. O *dataset* gerado após todas as simulações do conjunto de dados *general set* pode ser visualizado na Figura 10. Neste *dataset*, a primeira coluna corresponde a identificação do complexo e as colunas seguintes, aos escores de ligação preditos por cada uma das funções de escore consideradas. A última coluna é o valor do escore de ligação experimental, o *pkd*, que corresponde ao atributo-alvo neste *dataset*. .

	pdb	rb_score_RF_x	score_KORP_PL	Saída ECIF_x	Saída ECIF_y	rb_score_RF_y	rb_score_RF	score_vina_x	score_vina_y	score_Convex-PL	valor real
0	3zzf	5.587917	-200.6497307733	7.941933	7.49126	4.0361	4.21180	-5.93209	-5.32469	6.15493263	0.40
1	3gww	6.024462	-171.3181115193	7.389059	6.88552	4.9302	3.76538	-6.92595	-6.25654	6.21974764	0.45
2	1w8l	5.458314	-68.94232968277	4.342268	3.87772	3.3530	3.33720	-3.82426	-3.64529	5.44893644	0.49
3	3fqa	5.687460	-97.71893492873	8.093161	7.95266	3.7357	3.90904	-3.38655	-4.12970	5.51399615	0.49
4	1zsb	6.069729	-892.0679590824	6.187905	5.71062	5.8589	5.94554	-5.67050	-6.02468	5.98615588	0.60
...	...	...	...	...	...	...	...	...	...	...	...
14121	7cpe	6.571095	-358.9212759298	6.799138	6.48438	7.8038	7.74190	-7.87375	-9.37592	6.99581087	13.96
14122	2xuf	9.008545	-262.6237430866	7.217983	6.88706	12.8067	11.01574	-12.38646	-15.77266	7.52070361	14.30
14123	1avd	8.458464	-610.2412124039	7.069903	7.15486	11.8259	11.34978	-6.53097	-6.35137	6.32085176	15.00
14124	2xui	8.197966	-391.6148145889	4.304542	4.40604	11.1380	10.40784	-11.36048	-15.34914	7.3656818	15.00
14125	2avi	7.149385	-609.6760686712	7.652412	7.01300	7.4389	7.39296	-7.78368	-6.67350	6.77918153	15.22

Figura 10: *Datasets* gerados sem a limpeza de dados.

3.3.3 Pré-processamento dos dados

A etapa de pré-processamento de dados é uma fase crucial no desenvolvimento de modelos de AM e pode influenciar significativamente o desempenho final do modelo.

A normalização dos dados é um passo essencial para garantir que as diferentes funções de escore, que podem ter intervalos e escalas distintas, contribuam de forma igual para o modelo. A normalização foi realizada usando os limites inferior e superior do resultado que está sendo previsto, para que tenhamos um visualização mais eficiente. A fórmula utilizada para normalizar é encontrada a seguir:

$$\text{Valor Normalizado} = \frac{\text{Valor Original} - \text{Limite Inferior}}{\text{Limite Superior} - \text{Limite Inferior}} \quad (7)$$

Mesmo que modelos como XGBoost e *Random Forest* possam ser robustos em relação à escala dos dados e não necessitar de uma normalização para ter bons desempenho, normalizar o conjunto de dados é uma boa prática para garantir que todas as funções e pesos estejam na mesma amplitude. Isso ajuda a evitar problemas de otimização e facilita a interpretação dos pesos atribuídos a cada função no modelo de *ensemble*. O pré-processamento consistente ajuda a garantir uma comparação justa entre diferentes funções de escore, visto que o presente trabalho busca além de mostrar resultados, apresentar uma análise exploratória dos resultados das funções.

Podemos observar a necessidade de um tratamento nos dados, pois diferentes funções apresentam uma faixa de resultados distintos. Após o tratamento necessário, o conjunto de dados de resultados é apresentado na figura 11 e o mesmo é utilizado para o *ensemble* das funções de escore.

	ECIF_GBT	rb_score_ET	Lin_F9	ECIF_RF	vina	KORP_PL	rb_score_GBT	rb_score_RF	Convex-PL	valor real
0	0.590215	0.183096	0.204415	0.548812	0.104933	0.101661	0.069806	0.122756	0.616798	0.000000
1	0.533100	0.135759	0.240258	0.474936	0.122514	0.086786	0.188518	0.212185	0.623318	0.003374
2	0.218349	0.090356	0.139819	0.108109	0.067648	0.034868	0.034562	0.054431	0.545779	0.006073
3	0.605837	0.150993	0.158451	0.605083	0.059905	0.049462	0.096875	0.092709	0.552324	0.006073
4	0.409014	0.366937	0.231339	0.331647	0.100306	0.452303	0.200828	0.305074	0.599820	0.013495
...	...	...	...	...	...	...	...	...	...	...
13814	0.472158	0.557418	0.360241	0.426014	0.139280	0.181926	0.337168	0.499605	0.701386	0.914980
13815	0.515427	1.000000	0.606284	0.475124	0.219106	0.133091	1.000000	1.000000	0.755092	0.943995
13816	0.500129	0.939987	0.243905	0.507785	0.115527	0.309379	0.850413	0.901899	0.633499	0.985155
13817	0.214452	0.840107	0.589993	0.172542	0.200957	0.198506	0.779574	0.833095	0.738682	0.985155
13818	0.560306	0.520417	0.256295	0.490484	0.137687	0.309092	0.494426	0.463107	0.679594	1.000000

Figura 11: *Datasets* após tratamento dos dados.

3.4 Geração dos modelos preditivos

As funções de *ensemble* propostas foram geradas com o auxílio de bibliotecas de aprendizado de máquina, como o Scikit-Learn [56]. No presente trabalho foi utilizado o *Random Forest*, XGBoost e uma função com média ponderada. As escolhas desses algoritmos foram feitas para representar os ensembles de voting, boosting e bagging.

A média ponderada proposta consiste em combinar os escores das funções consideradas ponderados pela correlação obtida por cada função calculada a partir do valor predito pela função e o valor experimental da interação do complexo

3.5 Avaliação da função proposta

Neste trabalho a avaliação das FS propostas baseadas em ensemble foi realizada a partir do *Root Mean Squared Error* (RMSE) e da correlação de Pearson (*Scoring Power*). Esses valores foram calculados considerando o valor predito por cada função de escore proposta para cada complexo proteína-ligante do PDBBind 2020 *general set* e o valor experimental deste mesmo complexo.

É importante mencionar que todos os códigos relacionados a esta pesquisa estão acessíveis no GitHub de forma pública, facilitando a replicação do estudo. Dessa forma, torna-se simples ajustar o código para diferentes conjuntos de dados ou explorar diversas configurações experimentais. O link direto para o repositório pode ser encontrado em `git@github.com:combilab-furg/dissertacao-balboni.git`. Adicionalmente, a documentação abrangente e instruções presentes no repositório foram elaboradas para simplificar a compreensão e a utilização eficiente dos códigos. Junto ao link, são disponibilizados os conjuntos de dados empregados no treinamento das funções de escore, os dados utilizados no *ensemble*, bem como um script automatizado para a geração de todos os gráficos presentes neste estudo.

4 RESULTADOS

4.1 Análise exploratória de dados

Na Figura 12, temos a relação entre o valor da afinidade de ligação experimental p_{kd} normalizado (eixo Y) e o valor da afinidade de ligação predito pelo função de escore Convex-PL (eixo X) para todas as instâncias da base de dados de entrada (PDBBind 2020 general set). Analogamente, nas Figuras 13, 14, 15, 16, 17, 18, 19 e 20, apresentam-se as distribuições de dados das predições das respectivas funções de escore utilizadas. A linha vermelha em cada gráfico indica o resultado ideal, onde o valor experimental e o valor predito são idênticos.

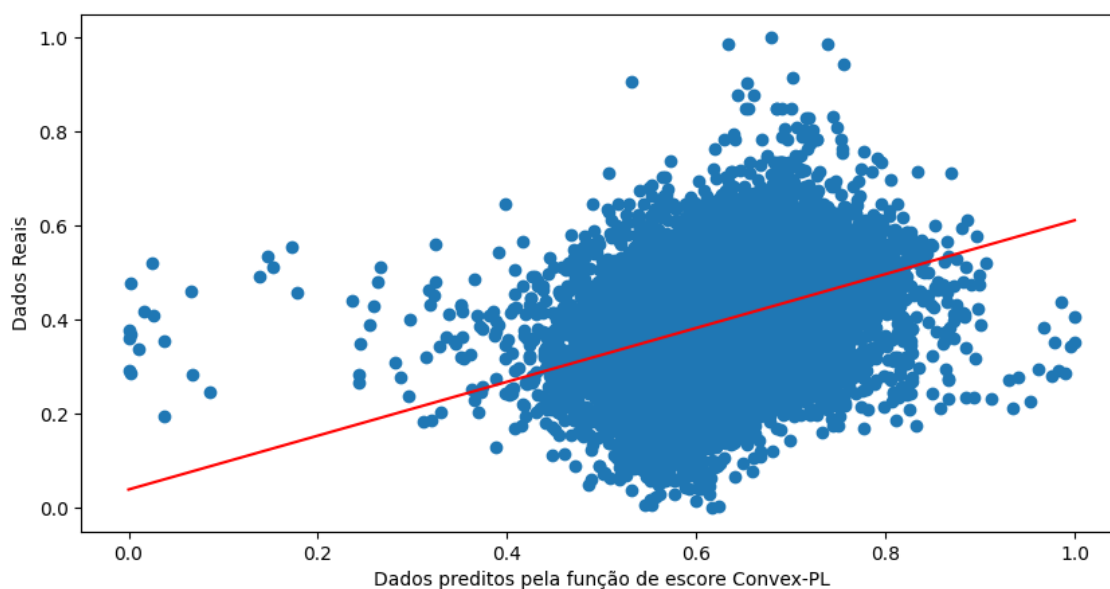


Figura 12: Distribuição dos dados preditos pela função de escore Convex-PL.

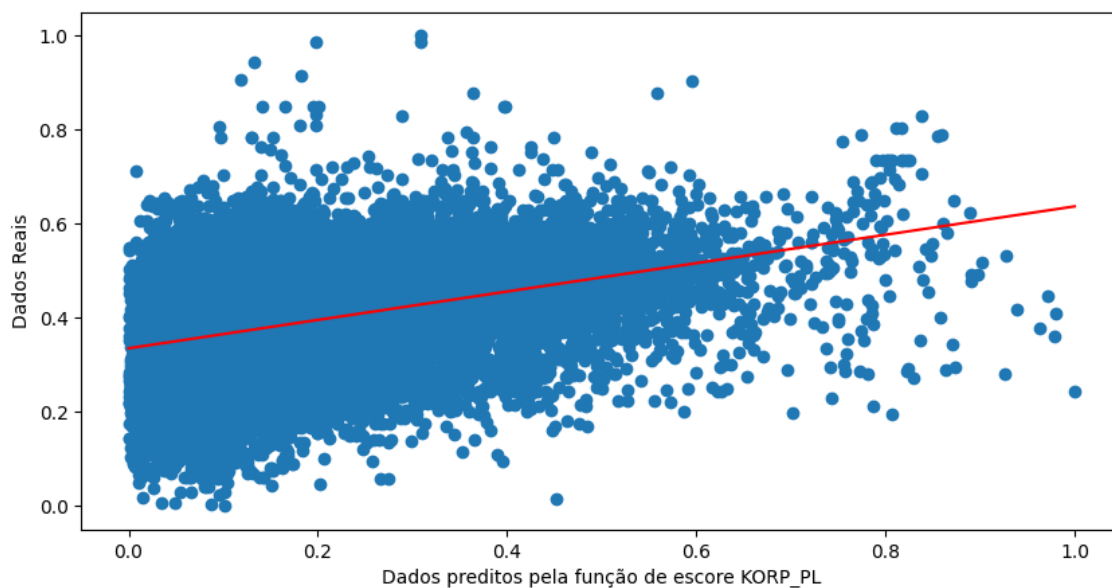


Figura 13: Distribuição dos dados preditos pela função de escore Korp-PL.

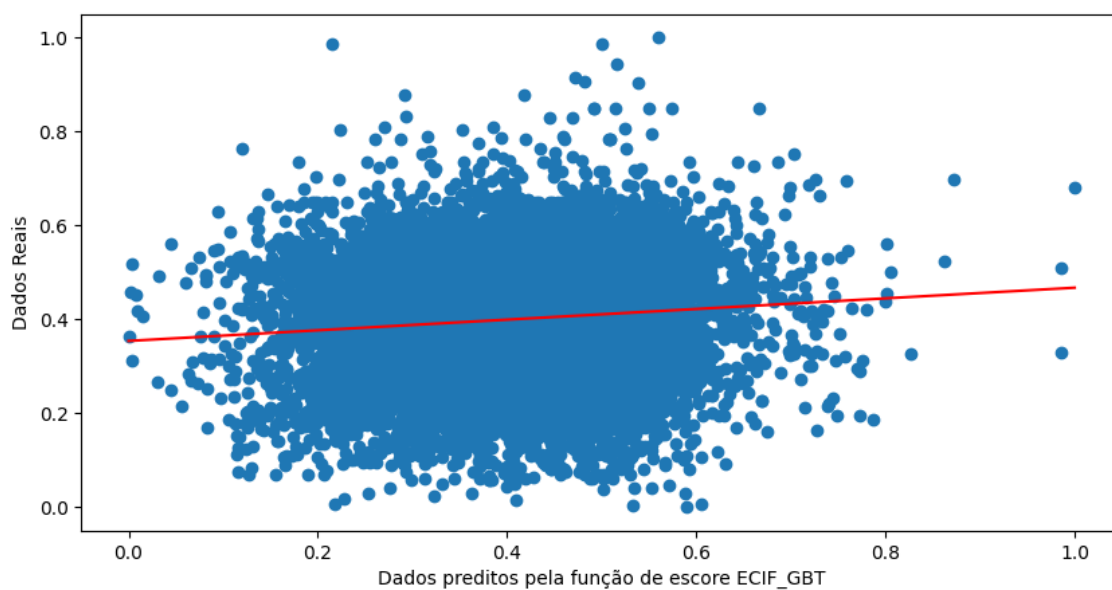


Figura 14: Distribuição dos dados preditos pela função de escore ECIF-GBT.

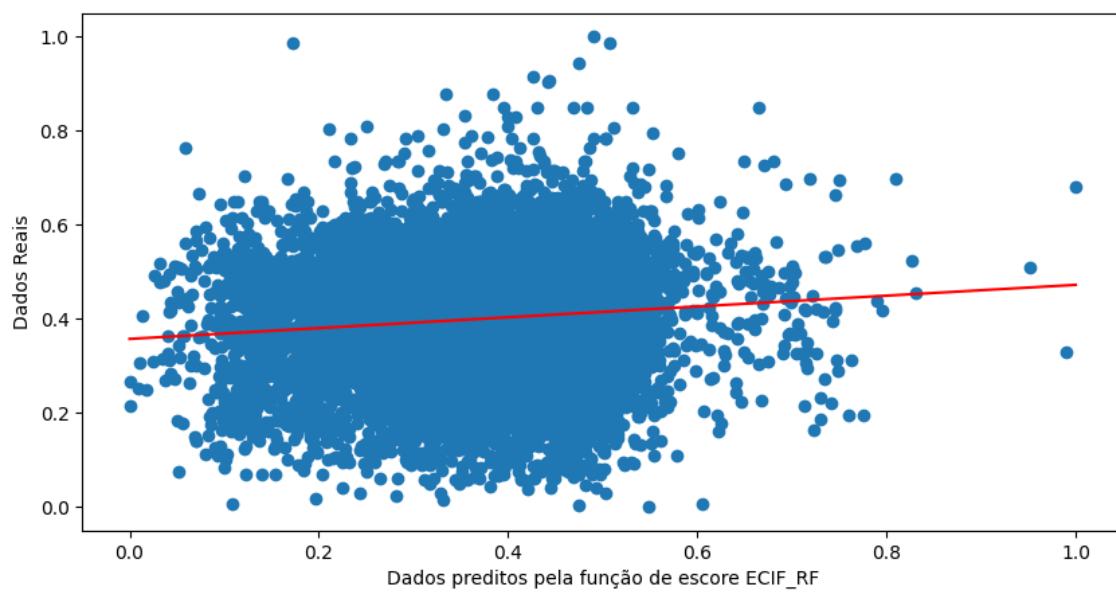


Figura 15: Distribuição dos dados preditos pela função de escore ECIF-RF.

Nas Figuras 14 e 15, observa-se uma notável semelhança nas distribuições de dados. Inclusive, a inclinação da linha vermelha nas duas figuras é praticamente idêntica. Essa semelhança sugere que os atributos extraídos por esse conjunto não são generalistas, pois, mesmo ao utilizar dois modelos distintos para gerar os atributos e os resultados, as distribuições resultantes são muito semelhantes.

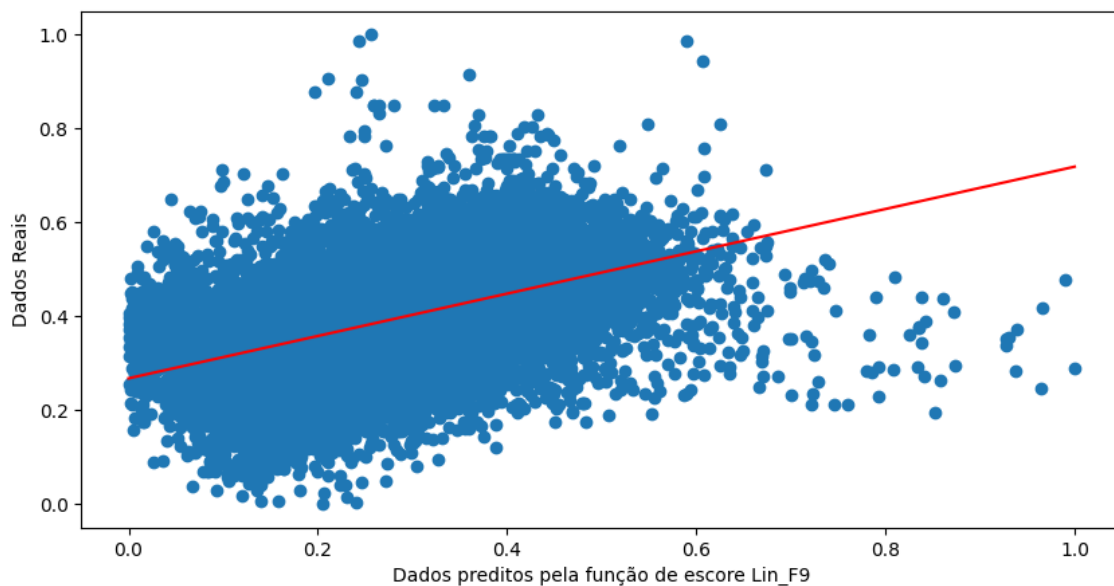


Figura 16: Distribuição dos dados preditos pela função de escore Lin_F9.

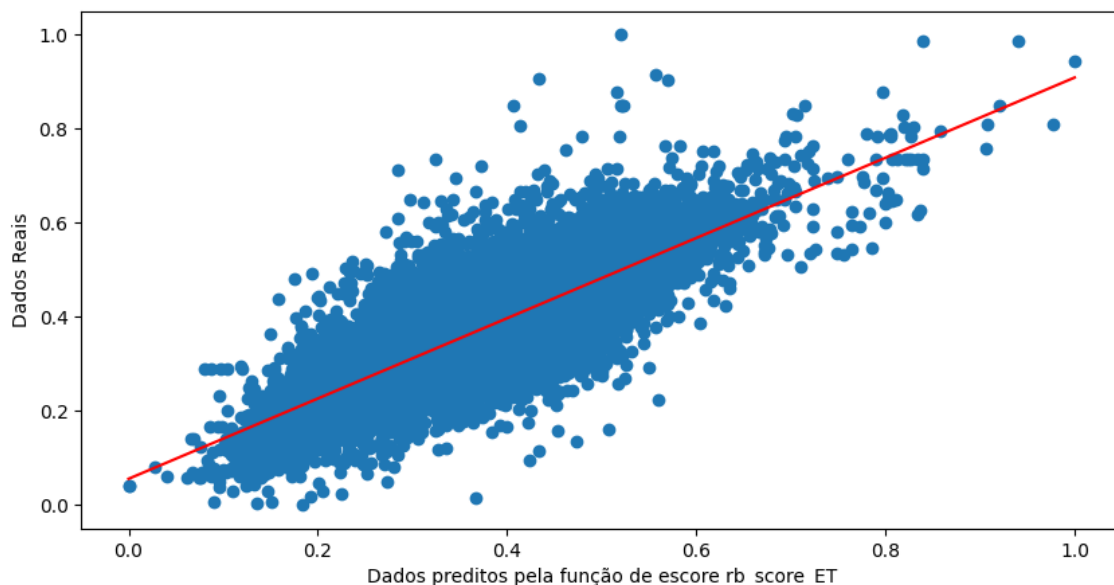


Figura 17: Distribuição dos dados preditos pela função de escore rb_score_ET.

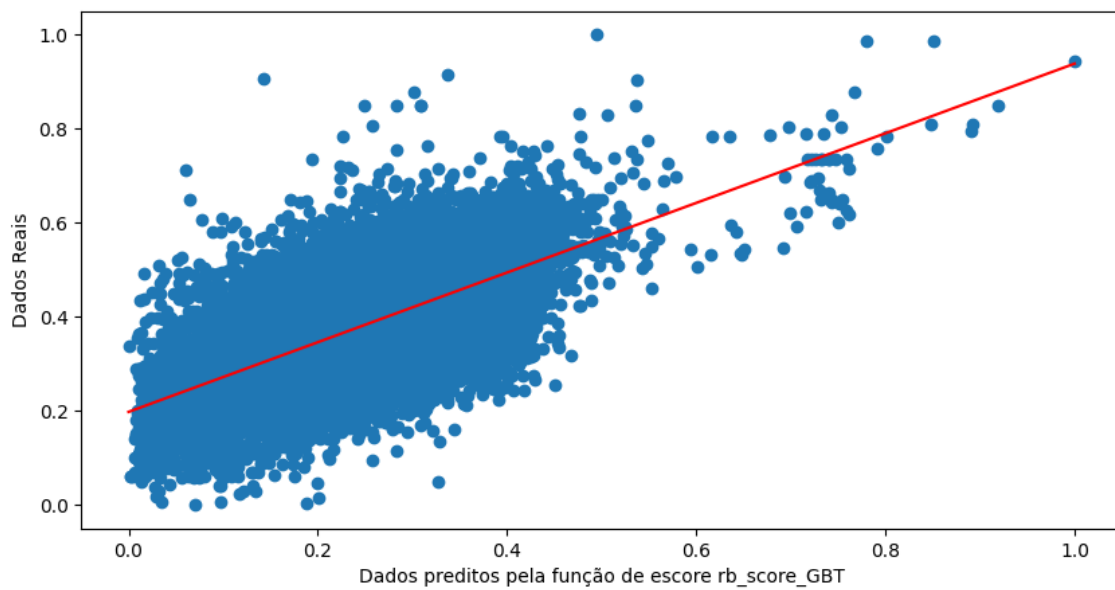


Figura 18: Distribuição dos dados preditos pela função de escore `rb_score_GBT`.

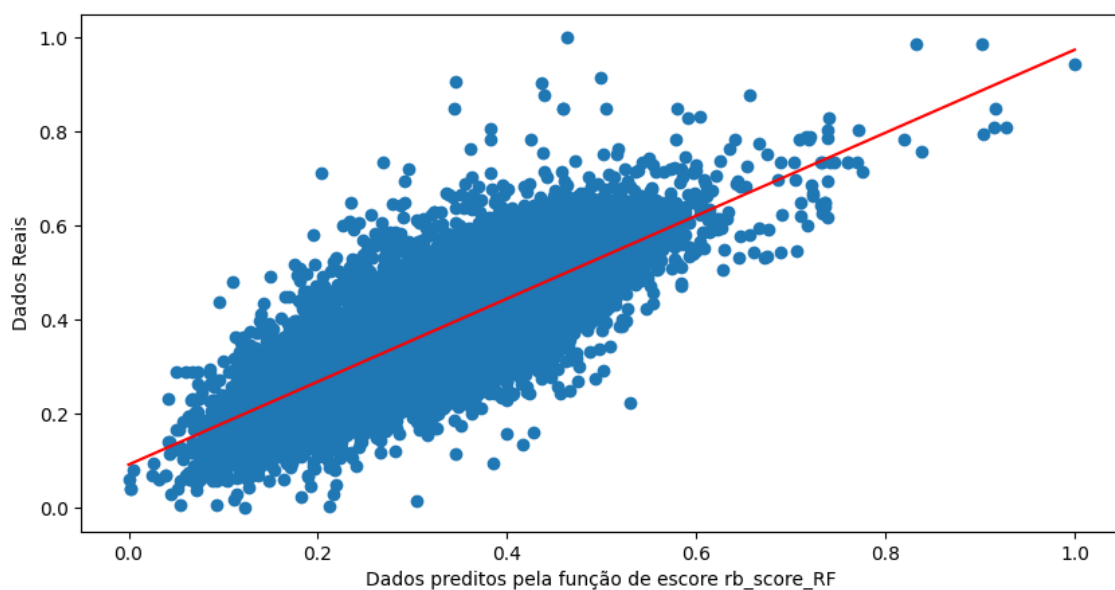


Figura 19: Distribuição dos dados preditos pela função de escore `rb_score_RF`.

Nas Figuras 17, 18 e 19, observamos que o modelo pode influenciar nos resultados, especialmente quando seus atributos e o conjunto de dados utilizados são mais generalistas. Isso contrasta com a função ECIF, visto que nesse caso temos três distribuições distintas, evidenciando a variação nos modelos de Random Forest e Extra Tree ao GBT.

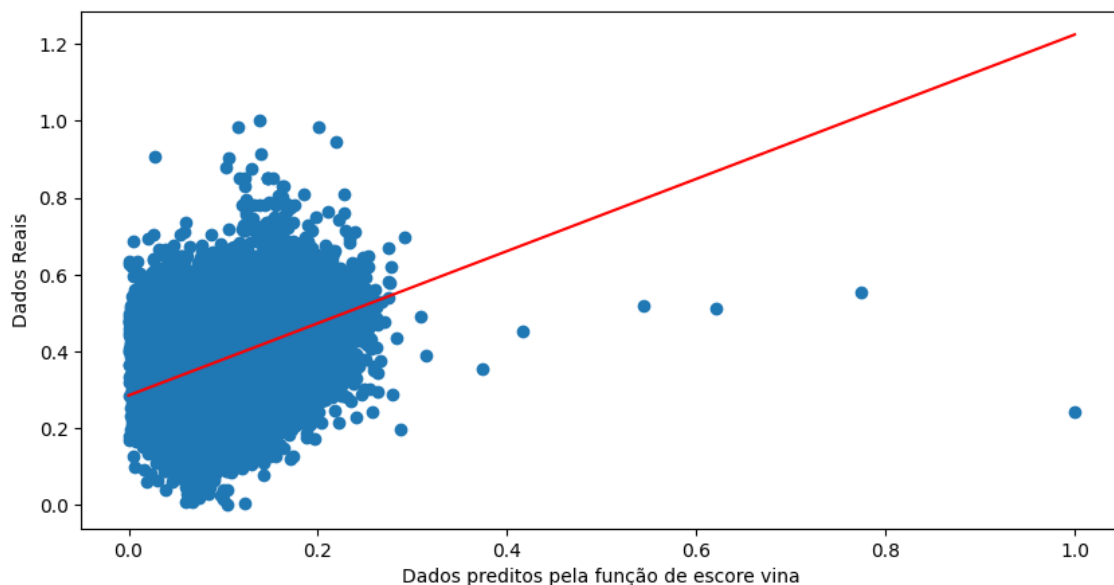


Figura 20: Distribuição dos dados preditos pela função de escore vina.

Na Figura 20, podemos observar uma função que apresentou um comportamento significativamente diferente das outras. Isso ocorre devido à sua sensibilidade a dados discrepantes, que impactam diretamente na distribuição das funções e na normalização aplicada. Nenhuma técnica de correção de valores discrepantes foi aplicada devido ao tamanho do conjunto de dados, que conta com aproximadamente 14.000 entradas. Esperava-se que esse volume de dados fosse suficiente para estabelecer um padrão de distribuição próprio para cada função.

É importante destacar o caráter investigativo dessa abordagem, uma vez que o objetivo principal é apresentar como os dados são dispersos pelas funções de escore. A partir dessa análise, teremos informações suficientes para obter *insights* sobre como proceder no tratamento individual de cada função, levando em consideração suas características específicas. Não foi realizada nenhuma limpeza de dados individual nas funções, preservando assim os dados brutos de cada função. Para obter resultados mais robustos, é recomendável realizar uma investigação individual e aplicar o tratamento adequado a cada conjunto de dados.

Nas Figuras 21 e 22, são apresentadas as métricas de RMSE (Erro Quadrático Médio) e a correlação de Pearson (métrica de avaliação de funções de escore chamada *scoring power*) para as funções de escore individuais. Essas métricas oferecem entendimento sobre o desempenho de cada função na predição dos resultados desejados. Observa-se que a função *gb_score* demonstrou um ganho significativo de desempenho em comparação com outras funções, destacando o potencial das abordagens baseadas em *machine learning*. No entanto, é importante notar que a função com o pior desempenho também é baseada em *machine learning*, evidenciando a importância de treinar modelos com dados sólidos e atributos relevantes para garantir resultados mais precisos.

É importante mencionar que o valor de Correlação de Pearson para a função de escore do vina de apenas 0.389 é causado pela normalização, uma vez que o escore do vina tem valores com maior variação (entre os valores mínimo e máximo do escore, com casos de escore muito positivo causado por proximidade de átomos e não tratado como outlier no pré-processamento). Neste caso, é necessário a remoção de outliers antes da normalização, um processo não avaliado no contexto deste trabalho.

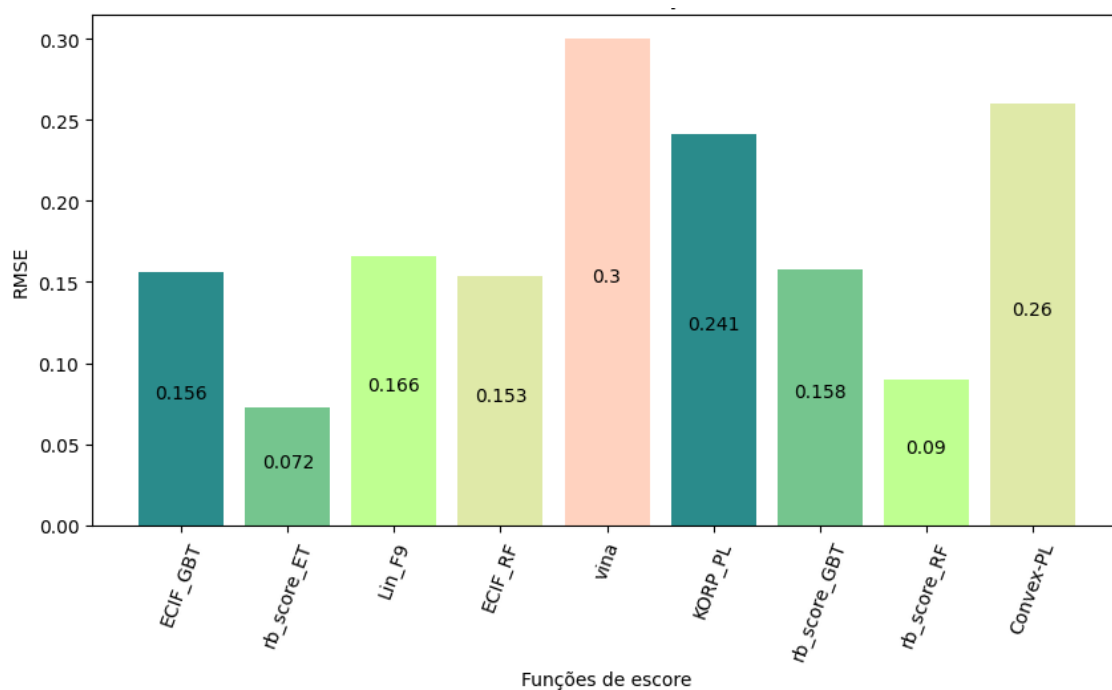


Figura 21: RMSE das funções de escore.

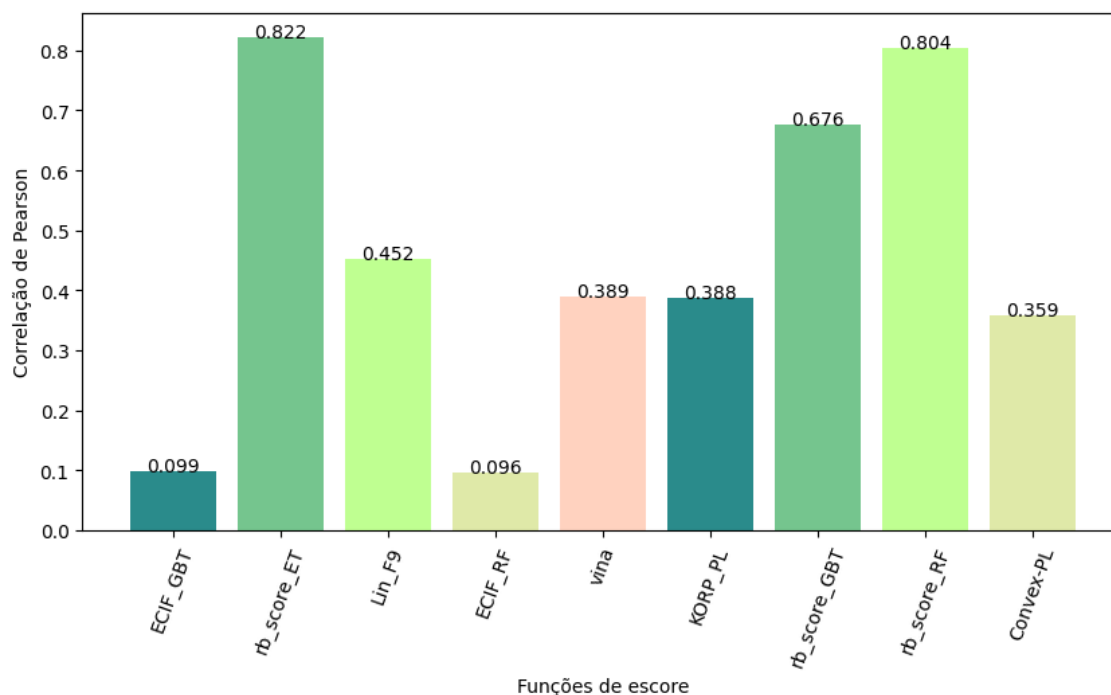


Figura 22: Correlação de Pearson das funções de escore.

Nas Figuras 23 e 24 são apresentadas as dez melhores e dez piores predições das funções de escore, ranqueadas a partir do valor de escore. Essa análise detalhada permite identificar padrões nos resultados obtidos por cada função. Percebe-se que nenhuma das funções apresentou resultados iguais, o que demonstra a capacidade individual de cada uma delas em lidar com diferentes cenários. Isso destaca a importância de uma abordagem diversificada ao selecionar ou combinar funções de escore, considerando a complexidade e características específicas de cada função.

ECIF_GBT	rb_score_ET	Lin_F9	ECIF_RF	vina	KORP_PL	rb_score_GBT	rb_score_RF	Convex-PL
2x4u	4xx4	3p2k	2wks	1f2p	4unq	3u1i	1o4r	5h1e
2hzi	4e5j	6bqj	4n00	1ro7	4w1	2jfh	6rfi	5tdi
2peh	5ybe	4cpx	3cn0	3i0n	4c0r	3snb	2opy	4hzx
4rgd	5xyz	3fk1	6gu6	4x8t	2xyu	6e7x	5lax	6ps0
5eta	5ubg	1tka	5wef	1o4l	6g92	4n3w	1xh6	1akv
3drs	5owt	1m5l	6azk	5ct1	4h3i	1aw1	3vg1	3jzo
5tnt	2zp0	6gg5	5vm6	2w97	2hzn	2vgo	4ny3	6ps4
5am1	6f29	6die	4zuo	6r9s	5e8w	6f0y	5w4e	3c7p
3fcb	2zlg	5or9	4p4b	3nm6	1i4l	1ms0	3v49	6pyc
2fwz	5db1	5mtx	5hu1	5vqe	5h0g	5b6c	6qrg	3rvv

Figura 23: Top 10 melhores predições das funções de escore.

ECIF_GBT	rb_score_ET	Lin_F9	ECIF_RF	vina	KORP_PL	rb_score_GBT	rb_score_RF	Convex-PL
2xui	2avi	2avi	2xui	1amn	2xuf	1amn	1amn	2xaj
5ief	1amn	1avd	4wr7	1avd	1amn	5nn0	2avi	2xah
4wr7	1a4m	4pri	5ief	2avi	2xui	1a4m	5nn0	2nyr
6cc9	5nn0	2nyr	3fqa	1eoj	2aoi	3u0d	1a4m	2xag
3fqa	1mau	1amn	6hib	2xui	7cpa	7cpa	1mau	5wlg
3zzf	2zx5	1eol	3o9g	1eol	2uyw	1stp	1eoj	5wkh
1stp	1stp	2uxx	4amy	7cpa	2zx5	2ada	1stp	3abt
3o9g	7cpa	1eoj	2xgs	3fck	5nn0	1swn	1ijk	3otx
2xgs	1zsb	4prh	1k9r	1stp	2avi	2zx5	2zx5	5wkf
2w6z	3u0d	1stp	2w6z	1a4m	6b5m	1mau	7cpa	4mnq

Figura 24: Top 10 piores previsões das funções de escore.

4.2 Função de Ensemble

Para os resultados de *ensemble*, foram exploradas três abordagens distintas. A primeira consiste no uso de uma função ponderada, incorporando os pesos atribuídos a cada uma das funções de escore. A segunda estratégia envolve a aplicação do algoritmo de *Random Forest* no *ensemble*. A terceira abordagem adota o algoritmo XGBoost para a construção do *ensemble*. Cada uma dessas técnicas busca otimizar a combinação das previsões das funções de escore individuais, proporcionando uma análise abrangente e comparativa do desempenho do *ensemble* em diferentes contextos.

Nas Figuras 25 e 26, são apresentados os parâmetros utilizados no treinamento dos modelos utilizando as abordagens de *Random Forest* (RF) e XGBoost, respectivamente. Essa transparência na configuração dos parâmetros permite uma compreensão detalhada do processo de treinamento, contribuindo para a replicação dos experimentos.

```
random_forest = RandomForestRegressor(n_estimators=100, random_state=42)
```

Figura 25: Parâmetros utilizados no modelo *Random Forest*

```
xgboost_model = xgb.XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
colsample_bynode=1, colsample_bytree=1, gamma=0.5,
importance_type='gain', interaction_constraints='', # num_class=2,
learning_rate=0.300000012, max_delta_step=0, max_depth=6,
min_child_weight=1,
n_estimators=100, n_jobs=0, num_parallel_tree=1,
objective='binary:logistic', random_state=0, reg_alpha=0,
reg_lambda=1, subsample=1, #scale_pos_weight="<class 'float'>",
tree_method='exact', validate_parameters=1)
```

Figura 26: Parâmetros utilizados no modelo XGBoost

Para a obtenção dos hiperparâmetros, foi utilizado uma abordagem de otimização probabilística. Esta estratégia foi selecionada devido à sua eficiência em explorar o espaço de busca de hiperparâmetros de forma mais rápida e eficaz, visto que todas as combinações de hiperparâmetros possíveis seja muito grande, tornando impraticável sua exploração completa devido a restrições de tempo e recursos computacionais. A otimização de hiperparâmetros probabilístico amostra aleatoriamente o espaço de busca, priorizando áreas com maior probabilidade de conter configurações promissoras. Essa abordagem permite uma exploração mais eficiente do espaço de hiperparâmetros, reduzindo o tempo de busca sem comprometer a qualidade da solução encontrada.

Para determinar o peso de cada função de escore em relação ao resultado final, foi realizada uma regressão linear. Essa abordagem permitiu estimar os coeficientes que ponderam a contribuição de cada função na predição final. Assim, cada função teve seu peso atribuído com base em sua influência nos resultados do modelo de ensemble. O resultado da predição foi adquirida com base na fórmula matemática encontrada na Equação 8. Os valores específicos desses pesos estão apresentados na tabela 3.

$$\text{Ponderação} = \sum_{i=1}^9 \text{valores}[i] \times \text{pesos_normalizados}[i] \quad (8)$$

Tabela 3: Pesos utilizados na função de ponderação.

Funções de Escore	Pesos
ECIF-GBT	0.113992
rb_score_ET	0.125305
Lin_F9	0.112628
ECIF_RF	0.114355
vina	0.094570
KOPR_PL	0.102516
rb_score_GBT	0.113732
rb_score_RF	0.122959
Convex-PL	0.999402

Na Figura 27, temos a distribuição de dados das predições geradas pela função de es-

core RF. Da mesma forma, nas Figuras 28 e 29, são apresentadas as distribuições de dados das predições geradas pelas funções de escore XGBoost e ponderada, respectivamente.

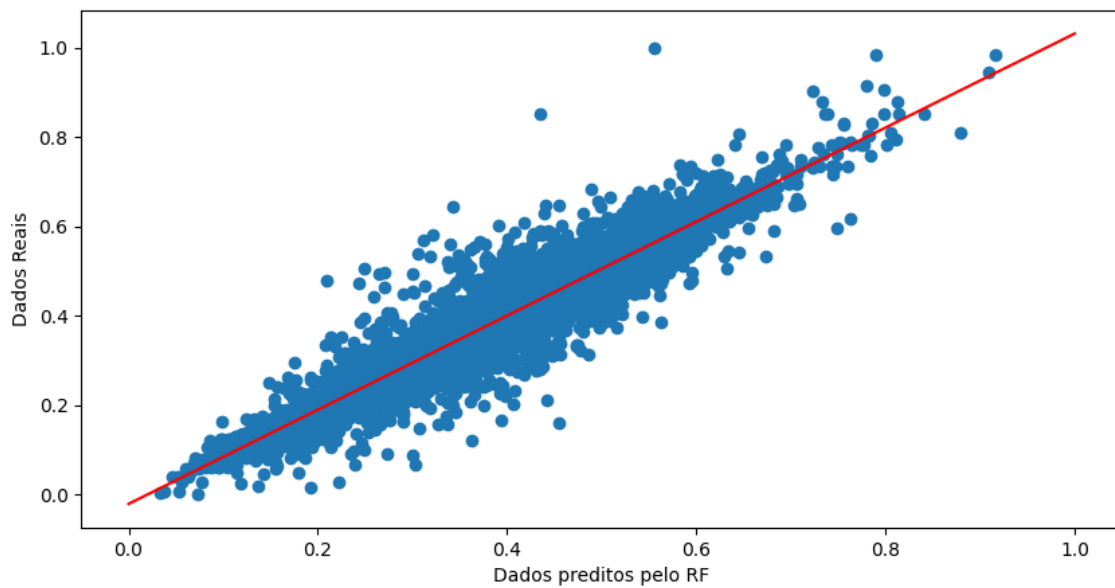


Figura 27: Distribuição dos dados preditos pelo *ensemble* utilizando *Random Forest*

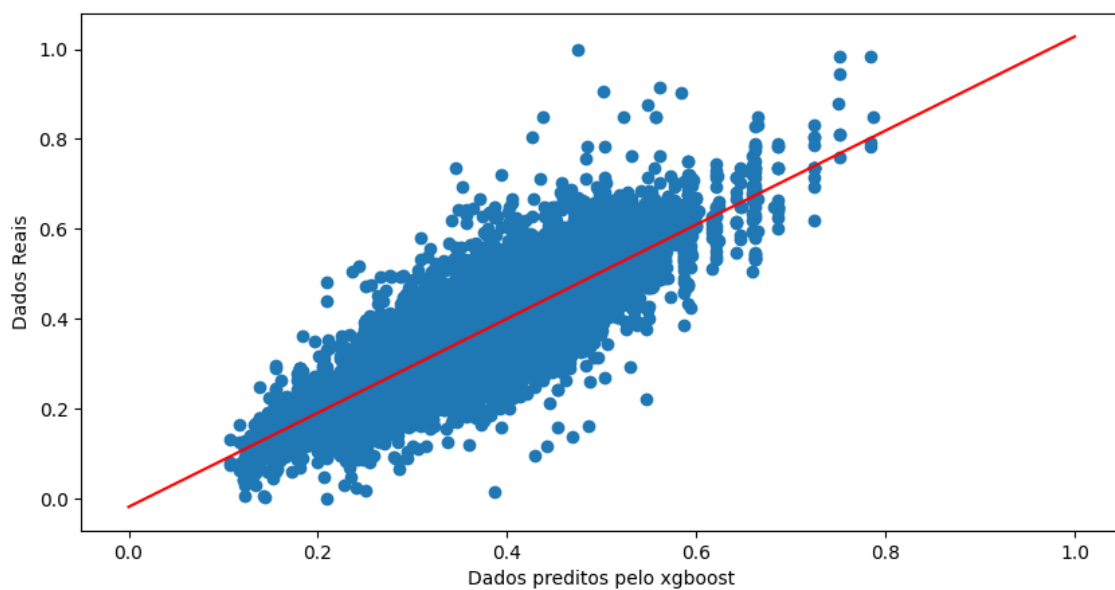


Figura 28: Distribuição dos dados preditos pelo *ensemble* utilizando XGBoost

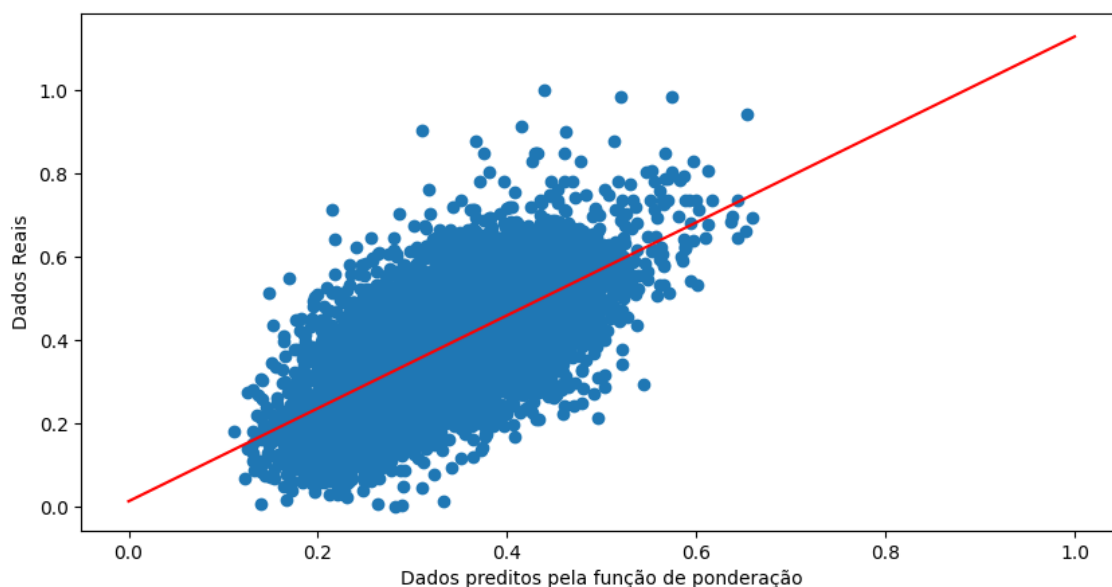


Figura 29: Distribuição dos dados preditos pela função de ponderação

A partir dos gráficos, podemos observar uma distribuição mais favorável para o *Random Forest*, seguido pelo XGBoost. Entretanto, é possível identificar alguns pontos retos nas predições do XGBoost, sugerindo a ocorrência de sobreajuste, principalmente no intervalo entre 0,6 e 0,8.

As análises de qualidade do modelo podem ser encontradas nas Figuras 30 e 31. Nelas, podemos observar que ambos os métodos de *ensemble* trouxeram resultados superiores às outras funções de escore. O melhor resultado apresentado em todos os testes realizados foi obtido com o algoritmo *Random Forest*, o que pode indicar uma boa capacidade de generalização do modelo para diferentes conjuntos de dados. Isso é crucial em aplicações de docagem molecular, onde a diversidade dos complexos proteína-ligante pode variar amplamente. A falta de generalização em outras funções de escore destaca a importância de abordagens como o *ensemble*, que pode ser uma estratégia eficaz para mitigar as limitações individuais de cada função.

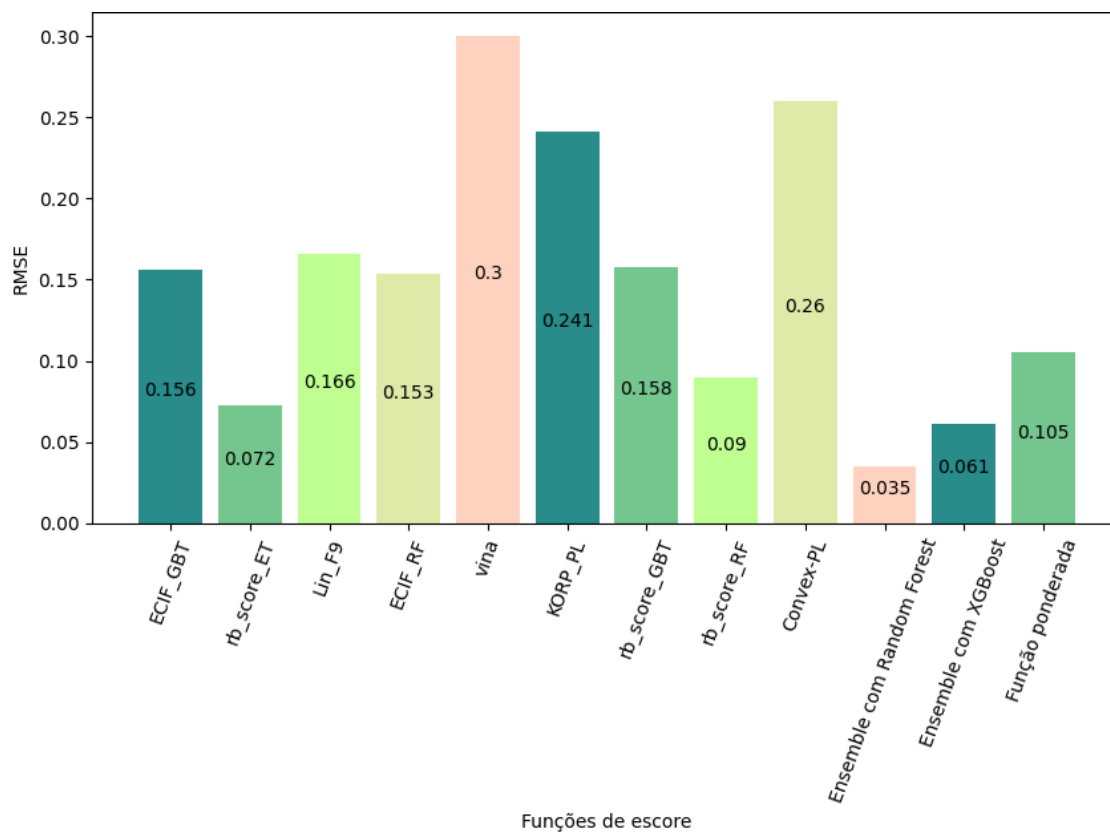


Figura 30: Comparativo do RMSE dos modelos e das funções de escore.

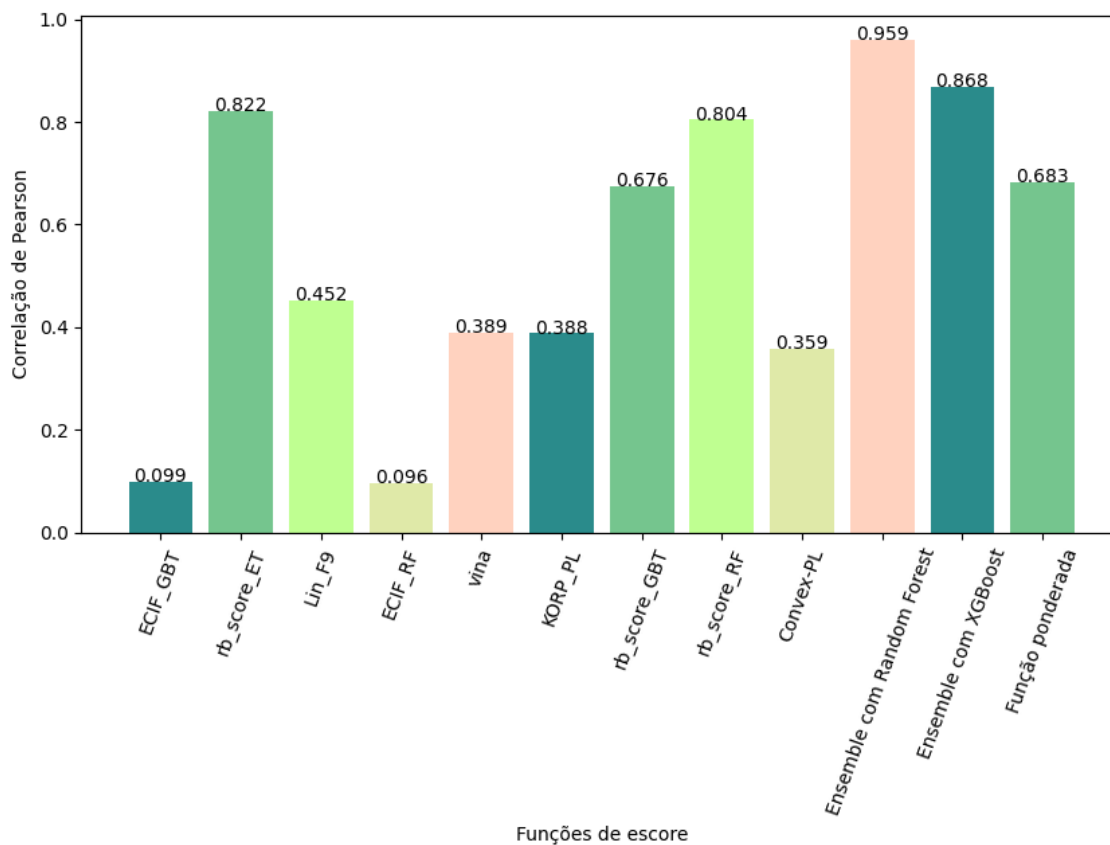


Figura 31: Comparativo da correlação de Pearson dos modelos e das funções de escore

Nas tabelas 4 e 5, são apresentadas as dez melhores e dez piores predições dos métodos de *ensemble*, organizadas por ordem. Nesse cenário, ao contrário das outras funções, algumas proteínas são consistentes nos casos de piores predições. Por exemplo, a proteína '2avi' ocupa a primeira posição tanto no *Random Forest* quanto no XGBoost, e a segunda posição na função de ponderação. Esse padrão é repetido para outras proteínas também. A recorrência dessas proteínas nos casos de piores predições pode indicar desafios específicos associados a essas moléculas ou à características das interações que não são bem capturadas pelos modelos de *ensemble* utilizados. Além disso, essa recorrência pode indicar uma possível obtenção errada do valor experimental dessas proteínas. Essa análise auxilia na identificação de possíveis fontes de erro ou imprecisões tanto na avaliação dos modelos gerados como também no resultado obtido experimentalmente.

Tabela 4: Top 10 melhores predições dos modelos gerados

Ensemble Random Forest	Ensemble XGBoost	Ensemble função ponderada
5m7m	3ws9	2y76
6pxc	2wti	6ao7
4qmv	6gfy	5qat
5ngb	4qw4	5ajp
5nat	1jbd	3adt
6oip	4jik	2vww
5d1w	1jif	4k2y
4bci	2y06	4tyl
4nks	5qaw	2csm
5q0n	2ohk	3fqh

Tabela 5: Top 10 piores predições dos modelos gerados

Ensemble Random Forest	Ensemble XGBoost	Ensemble função ponderada
2avi	2avi	1amn
1a4m	1a4m	2avi
6me4	1amn	1stp
6aan	1mau	7cpa
4zph	2zx5	5nn0
2on3	1zsb	1a4m
1yz3	7cpa	2xui
6n55	2wva	4wr7
3l7d	6aaq	1eoj
6k2n	3eqb	6e4c

5 CONCLUSÃO

No contexto da análise, os métodos de *ensemble* revelaram-se promissores, mostrando que a combinação de diferentes funções de escore pode efetivamente melhorar a precisão das predições. A utilização de *ensemble*, especialmente com o algoritmo *Random Forest*, apresentou consistentemente resultados superiores em comparação com as funções de escore individuais.

Ao observar as métricas de avaliação, como RMSE e correlação de Pearson, notou-se que o *ensemble* proporcionou uma performance mais robusta e confiável. Esse aprimoramento pode ser crucial em aplicações práticas, como na docagem molecular, onde a precisão na previsão da afinidade de ligação entre moléculas é de extrema importância.

A análise detalhada das distribuições de dados e a comparação entre as funções de escore individuais e o *ensemble* destacaram a capacidade do *ensemble* em lidar com diferentes cenários e melhorar a generalização. Esses resultados sugerem que a abordagem de *ensemble* pode ser uma estratégia valiosa para otimizar a precisão das predições em estudos de interações moleculares.

A abordagem de realizar o pré-processamento individualmente com base na análise exploratória dos resultados pode proporcionar benefícios significativos no contexto da docagem molecular. Ao examinar cuidadosamente as distribuições de dados, identificar padrões nas predições e compreender as particularidades de cada função de escore, é possível adaptar estratégias de pré-processamento que maximizem a qualidade das previsões. Por exemplo, ao observar variações nas distribuições de dados entre diferentes funções de escore, pode-se ajustar estratégias de normalização, tratamento de *outliers* e outras técnicas de pré-processamento para atender às características específicas de cada função. Isso permite a personalização do pré-processamento para otimizar a performance de cada função de escore individual.

Contudo, a análise exploratória pode revelar *insights* sobre padrões de erro ou limitações específicas de cada função. Ao incorporar essas informações no pré-processamento, é possível mitigar potenciais fontes de viés ou imprecisão, melhorando assim a confiabilidade das predições.

A disponibilização dos códigos e dados no GitHub proporciona uma contribuição vali-

osa para a comunidade científica, permitindo a replicação dos experimentos, a exploração de diferentes conjuntos de dados e a adaptação dos métodos para diversas aplicações.

Como trabalhos futuros, podemos aplicar técnicas mais avançadas de pré-processamento, como a exclusão de *outliers* dos resultados nas funções de escore. Isso pode melhorar a normalização dos dados, reduzindo a influência de valores extremos que podem distorcer a distribuição dos dados e afetar a qualidade das predições. Além disso, podemos analisar os melhores resultados de cada uma das funções de escore para compreender as razões por trás de cada predição. é viável identificar as vantagens e desvantagens de cada função e utilizar esse entendimento para elaborar estratégias mais eficazes de *ensemble*.

REFERÊNCIAS

- [1] Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Citeseer.
- [2] Ain, Q. U., Aleksandrova, A., Roessler, F. D., and Ballester, P. J. (2015). Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(6):405–424.
- [3] Arce, O. E. A. (2020). Função de escore baseada em machine learning para docagem molecular proteína-ligante. Master’s thesis.
- [4] Bank, P. D. (1971). Protein data bank.
- [5] BARROS, L. (2010). *Estudos de Modelagem Molecular Visando a Síntese de um Polímero de Impressão Molecular para a Determinação de Fenitrotona em Tomate. 2010. p.* PhD thesis, Dissertação (Mestrado) Universidade Estadual de Campinas. SP.
- [6] Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.
- [7] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.
- [8] Borgelt, C. and Kruse, R. (2002). Induction of association rules: Apriori implementation. In *Compstat*, pages 395–400. Springer.
- [9] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- [10] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- [11] Caruana, R., Niculescu-Mizil, A., Crew, G., and Ksikes, A. (2004). Ensemble selection from libraries of models. page 18.

- [12] Cavasotto, C. N. (2015). *In silico drug discovery and design: theory, methods, challenges, and applications*. CRC Press.
- [13] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.
- [14] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.
- [15] de Albuquerque, N. L. S. (2020). Planejamento operacional durante a pandemia de covid-19: comparação entre recomendações da organização mundial da saúde e o plano de contingência nacional. *Cogitare Enfermagem*, 25.
- [16] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.
- [17] Eldridge, M. D., Murray, C. W., Auton, T. R., Paolini, G. V., and Mee, R. P. (1997). Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design*, 11(5):425–445.
- [18] Gabel, J., Desaphy, J., and Rognan, D. (2014). Beware of machine learning-based scoring functions on the danger of developing black boxes. *Journal of chemical information and modeling*, 54(10):2807–2815.
- [19] Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc."
- [20] Graves, A. P., Brenk, R., and Shoichet, B. K. (2005). Decoys for docking. *Journal of medicinal chemistry*, 48(11):3714–3728.
- [21] Guo, J., Hurley, M. M., Wright, J. B., and Lushington, G. H. (2004). A docking score function for estimating ligand- protein interactions: Application to acetylcholinesterase inhibition. *Journal of medicinal chemistry*, 47(22):5492–5500.
- [22] Hagen, J. B. (2000). The origins of bioinformatics. *Nature Reviews Genetics*, 1(3):231–236.
- [23] Huang, S.-Y., Grinter, S. Z., and Zou, X. (2010). Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Physical Chemistry Chemical Physics*, 12(40):12899–12908.

- [24] Huang, S.-Y. and Zou, X. (2006). An iterative knowledge-based scoring function to predict protein–ligand interactions: I. derivation of interaction potentials. *Journal of computational chemistry*, 27(15):1866–1875.
- [25] Imai, H., Iri, M., and Murota, K. (1985). Voronoi diagram in the laguerre geometry and its applications. *SIAM Journal on Computing*, 14(1):93–105.
- [26] Kadukova, M. and Grudinin, S. (2017). Convex-pl: a novel knowledge-based potential for protein-ligand interactions deduced from structural databases using convex optimization. *Journal of computer-aided molecular design*, 31:943–958.
- [27] Kadukova, M., Machado, K. d. S., Chacón, P., and Grudinin, S. (2021). Korp-pl: a coarse-grained knowledge-based scoring function for protein–ligand interactions. *Bioinformatics*, 37(7):943–950.
- [28] Khamis, M. A., Gomaa, W., and Ahmed, W. F. (2015). Machine learning in computational docking. *Artificial intelligence in medicine*, 63(3):135–152.
- [29] Kitchen, D. B., Decornez, H., Furr, J. R., and Bajorath, J. (2004). Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949.
- [30] Krishna, K. and Murty, M. N. (1999). Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.
- [31] Kundu, I., Paul, G., and Banerjee, R. (2018). A machine learning approach towards the prediction of protein–ligand binding affinity based on fundamental molecular properties. *RSC Advances*, 8(22):12127–12137.
- [32] Kuntz, I. D. (1992). Structure-based strategies for drug design and discovery. *Science*, 257(5073):1078–1082.
- [33] Langley, P. and Simon, H. A. (1995). Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64.
- [34] Li, Y., Liu, Z., Li, J., Han, L., Liu, J., Zhao, Z., and Wang, R. (2014). Comparative assessment of scoring functions on an updated benchmark: 1. compilation of the test set. *Journal of chemical information and modeling*, 54(6):1700–1716.
- [35] Li, Y., Rezaei, M. A., Li, C., and Li, X. (2019). Deepatom: a framework for protein-ligand binding affinity prediction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 303–310. IEEE.

- [36] Lise, S., Archambeau, C., Pontil, M., and Jones, D. T. (2009). Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. *BMC bioinformatics*, 10(1):1–17.
- [37] Liu, J. and Wang, R. (2015). Classification of current scoring functions. *Journal of chemical information and modeling*, 55(3):475–482.
- [38] Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., and Wang, R. (2015). Pdb-wide collection of binding data: current status of the pdbname database. *Bioinformatics*, 31(3):405–412.
- [39] Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. (2017). Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50(2):302–309.
- [40] Luscombe, N. M., Greenbaum, D., and Gerstein, M. (2001). What is bioinformatics? a proposed definition and overview of the field. *Methods of information in medicine*, 40(04):346–358.
- [41] Lybrand, T. P. (1995). Ligand—protein docking and rational drug design. *Current opinion in structural biology*, 5(2):224–228.
- [42] McConkey, B. J., Sobolev, V., and Edelman, M. (2002). The performance of current methods in ligand–protein docking. *Current Science*, pages 845–856.
- [43] Meng, E. C., Shoichet, B. K., and Kuntz, I. D. (1992). Automated docking with grid-based energy evaluation. *Journal of computational chemistry*, 13(4):505–524.
- [44] Mitchell, T. M., Mabadevan, S., and Steinberg, L. I. (1990). Leap: A learning apprentice for vlsi design. In *Machine learning*, pages 271–289. Elsevier.
- [45] Mohs, R. C. and Greig, N. H. (2017). Drug discovery and development: Role of basic biological research. *Alzheimer’s & Dementia: Translational Research & Clinical Interventions*, 3(4):651–657.
- [46] Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.
- [47] Morris, G. M. and Lim-Wilby, M. (2008). Molecular docking. In *Molecular modeling of proteins*, pages 365–382. Springer.
- [48] Muegge, I. (2006). Pmf scoring revisited. *Journal of medicinal chemistry*, 49(20):5895–5902.

- [49] Mukesh, B. and Rakesh, K. (2011). Molecular docking: a review. *Int J Res Ayurveda Pharm*, 2(6):1746–51.
- [50] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT press.
- [51] Mysinger, M. M., Carchia, M., Irwin, J. J., and Shoichet, B. K. (2012). Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594.
- [52] O’Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., and Hutchison, G. R. (2011). Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):1–14.
- [53] Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.
- [54] Pagadala, N. S., Syed, K., and Tuszynski, J. (2017). Software for molecular docking: a review. *Biophysical reviews*, 9(2):91–102.
- [55] Pason, L. P. and Sotriffer, C. A. (2016). Empirical scoring functions for affinity prediction of protein-ligand complexes. *Molecular informatics*, 35(11-12):541–548.
- [56] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- [57] Piñero, J., Furlong, L. I., and Sanz, F. (2018). In silico models in drug development: where we are. *Current opinion in pharmacology*, 42:111–121.
- [58] Raha, K. and Merz, K. M. (2004). A quantum mechanics-based scoring function: study of zinc ion-mediated ligand binding. *Journal of the American Chemical Society*, 126(4):1020–1021.
- [59] Rajamani, R. and Good, A. C. (2007). Ranking poses in structure-based lead discovery and optimization: current trends in scoring function development. *Current opinion in drug discovery & development*, 10(3):308–315.
- [60] Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.
- [61] Rayka, M. and Firouzi, R. (2023). Gb-score: Minimally designed machine learning scoring function based on distance-weighted interatomic contact features. *Molecular Informatics*, 42(3):2200135.
- [62] Reddy, M. R. and Parrill, A. L. (1999). Overview of rational drug design. ACS Publications.

- [63] Sánchez-Cruz, N., Medina-Franco, J. L., Mestres, J., and Barril, X. (2021). Extended connectivity interaction features: improving binding affinity prediction through chemical description. *Bioinformatics*, 37(10):1376–1382.
- [64] Singh, N., Shah, P., Dwivedi, H., Mishra, S., Tripathi, R., Sahasrabuddhe, A. A., and Siddiqi, M. I. (2016). Integrated machine learning, molecular docking and 3d-qsar based approach for identification of potential inhibitors of trypanosomal n-myristoyltransferase. *Molecular biosystems*, 12(12):3711–3723.
- [65] Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., and Wang, R. (2018). Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913.
- [66] Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introdução ao data mining: mineração de dados*. Addison-Wesley.
- [67] Thomas, P. D. and Dill, K. A. (1996). Statistical potentials extracted from protein structures: how accurate are they? *Journal of molecular biology*, 257(2):457–469.
- [68] Trott, O. and Olson, A. J. (2010). Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461.
- [69] Velec, H. F., Gohlke, H., and Klebe, G. (2005). Drugscorecsd knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction. *Journal of medicinal chemistry*, 48(20):6296–6303.
- [70] Verli, H. (2014). Bioinformática: da biologia à flexibilidade molecular.
- [71] Wang, C. and Zhang, Y. (2017). Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *Journal of computational chemistry*, 38(3):169–177.
- [72] Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The pdbind database: Collection of binding affinities for protein–ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980.
- [73] Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming*, pages 81–100. Springer.
- [74] Withers, L. A. (1991). 4. in-vitro conservation. *Biological Journal of the Linnean Society*, 43(1):31–42.

- [75] Yang, C. and Zhang, Y. (2021). Lin_f9: a linear empirical scoring function for protein–ligand docking. *Journal of chemical information and modeling*, 61(9):4630–4644.