Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação

Getúlio Rodrigues Silva

Redução de Dimensionalidade em Problemas de Múltiplos Alvos: uma Comparação entre PCA e Autoencoder

Monografia de Graduação em Engenharia de Controle e Automação

Ouro Preto, 2023

Getúlio Rodrigues Silva

Redução de Dimensionalidade em Problemas de Múltiplos Alvos: uma Comparação entre PCA e Autoencoder

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Jadson Castro Gertrudes

Coorientadora: Profa. Dra. Adrielle de Carvalho Santana

Ouro Preto

2023

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Getúlio Rodrigues Silva

Redução de Dimensionalidade em Problemas de Múltiplos Alvos:

uma Comparação entre PCA e Autoencoder

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia de Controle e Automação

Aprovada em 19 de Julho de 2023

Membros da banca

Dr. Jadson Castro Gertrudes - Orientador (DECOM - Universidade Federal de Ouro Preto)
Dra. Adrielle de Carvalho Santana - Coorientadora (DECAT - Universidade Federal de Ouro Preto)
Me. Philipe de Oliveira Fernandes - Convidado (Faculdade de Farmácia da Universidade Federal de Minas Gerais)
Leonardo Macedo Freire - Convidado (PPGCC - Universidade Federal de Ouro Preto)

Jadson Castro Gertrudes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em XX/07/2023

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 21/07/2023, às 21:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0559380** e o código CRC **D2B702A1**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Agradeço primeiramente a Deus, por me proporcionar oportunidades para testar minhas competências cada vez mais e me tornar capaz para realizar este grande feito e todas as conquistas até o momento.

Agradeço aos meus pais, Elizabeth e Gilmar por me guiarem durante meus anos de estudo na cidade de Ouro Preto e serem grandes fontes de inspiração e de apoio.

Aos meus orientadores Jadson e Adrielle, os grandes tutores desta etapa e que puderam me ensinar, sempre estando disponíveis para qualquer dúvida e ajuda. Foi uma honra ser aluno de vocês e poder concluir este trabalho de conclusão de curso.

Aos meus amigos pelo apoio, conselhos e suporte durante a graduação. Em especial a República Taturrodano pelos momentos de alegria e ser minha moradia durante todos esses anos.

Agradeço a Universidade Federal de Ouro Preto, por proporcionar ensino de qualidade com ótimos profissionais da educação pública. E a todos os professores, técnicos e responsáveis pelo seu funcionamento.

Resumo

As pesquisas na área de Química Medicinal têm utilizado cada vez mais técnicas para acelerar o processo de criação e desenvolvimento de medicamentos para combater novas doenças, especialmente por conta do crescimento exponencial da tecnologia em várias áreas do conhecimento. No entanto, esse avanço também leva ao aumento da quantidade de dados e variáveis a serem analisadas, tornando as tarefas atuais longas e complexas, um problema conhecido como "maldição da dimensionalidade". As ferramentas da inteligência artificial, as técnicas de análise de dados e os modelos de predição são cada vez mais comuns na área da Química Medicinal. Diante disso, o objetivo deste trabalho é criar uma ferramenta para auxiliar os pesquisadores na redução do tempo de análise de dados de compostos bioativos. Para isso, foram utilizadas técnicas de redução de dimensionalidade de dados, como a Análise de Componentes Principais e uma Rede Neural *Autoencoder*, juntamente com algoritmos de aprendizado de máquina para classificação *multioutput*. Os resultados obtidos foram comparados, o que mostra que a Rede Neural *Autoencoder* consegue proporcionar uma melhoria na classificação dos descritores obtendo-se coeficientes de F-Measure superiores aos testes com Análise de Componentes Principais principalmente em conjuntos de dados com maiores observações obtendo *score* de 0,610 nos testes de descritores KR para Autoencoder e 0,553 para Análise de Componentes Principais. É possível concluir que as técnicas de redução de dimensionalidade são efetivas, pois conseguem condensar informações de conjuntos extensos de dados, melhorando assim os modelos de predição.

Palavras-chave: Multioutiput, PCA, Autoencoder, Classificação, Redução de dimensionalidade.

Abstract

Research in the field of Medicinal Chemistry has increasingly utilized techniques to accelerate the process of creating and developing drugs to combat new diseases, especially due to the exponential growth of technology in various areas of knowledge. However, this advancement also leads to an increase in the amount of data and variables to be analyzed, making current tasks long and complex, a problem known as the ”curse of dimensionality”. Artificial intelligence tools, data analysis techniques, and prediction models are becoming more common in the field of medicinal chemistry. In light of this, the objective of this work is to create a tool to assist researchers of new drugs in reducing the time required for analyzing data from bioactive compounds tested on biological targets. For this purpose, data dimensionality reduction techniques such as Principal Component Analysis and an Autoencoder Neural Network, along with machine learning algorithms for multioutput classification, were utilized. The obtained results were compared, showing that the Autoencoder Neural Network can improve the classification of descriptors, obtaining F-Measure coefficients superior to Principal Component Analysis, especially in datasets with larger observations, achieving a score of 0.610 for Autoencoder and 0.553 for Principal Component Analysis in KR descriptor tests. It can be concluded that dimensionality reduction techniques are effective as they can condense information from extensive datasets, thus improving prediction models.

Key-words: Multioutiput, PCA, Autoencoder, Classification, dimensionality reduction.

Lista de ilustrações

Figura 1 – Representação de neurônio artificial Fonte: (FACELI et al., 2011) 17
Figura 2 – Rede Neural com duas camadas ocultas Fonte: (FACELI et al., 2011). . 18
Figura 3 – Representação da arquitetura de um *Autoencoder* 19
Figura 4 – Representação de um *Multioutput Classifier*. Fonte: Próprio autor . . . 20
Figura 5 – Representação de um *Classifier Chain*. Fonte: (SANTOS; ROSSI; SUL–MS–BRAZIL, s.d.) 21
Figura 6 – Árvore de Decisão e Espaço de Decisões. Fonte: (FACELI et al., 2011) . 21
Figura 7 – Representação de um *K-Nearest Neighbors*. Fonte: (FACELI et al., 2011) 22
Figura 8 – Gráficos das componentes principais para cada conjunto de atributos. . 28
Figura 9 – Erro Quadrático Médio em função das épocas para cada conjunto de atributos. 29
Figura 10 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *substructure* para treino e teste. 31
Figura 11 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *pubchem* para treino e teste. 32
Figura 12 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *KR* para treino e teste. 33
Figura 13 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *atompair* para treino e teste. 34

Lista de tabelas

Tabela 1 – Quantidade de atributos em cada conjunto de dados 25
Tabela 2 – Comparativo da medida F-measure entre os resultados dos quatro subconjuntos de atributos 35

Sumário

1 INTRODUÇÃO 9
1.1 Introdução 9
1.2 Objetivos 10
1.3 Justificativa 11
1.4 Estrutura do Trabalho 11

2 REFERENCIAL TEÓRICO 12
2.1 Análise de relações entre estrutura química e atividade biológica . . 12
2.2 Redução de dimensionalidade 13
2.2.1 Análise de Componentes Principais 14
2.3 Autocodificadores 16
2.3.1 O que de fato é uma Rede Neural Autocodificadora? 18
2.4 Classificação de múltiplos rótulos 20
2.5 Métricas de avaliação 22

3 METODOLOGIA 24
3.1 Linguagem de Programação e Ambiente de execução 24
3.2 Conjunto de dados 24
3.3 Tratamento dos dados 25
3.3.1 Eliminação de Atributos e Normalização 25
3.4 Redução de Dimensionalidade 26
3.5 Classificação 26

4 RESULTADOS 28

5 CONSIDERAÇÕES FINAIS 36
5.1 Trabalhos Futuros 36

Referências 37

1 Introdução

1.1 Introdução

Com o crescente aumento no número de casos de doenças crônicas e novas variações epidemiológicas, muito se discute acerca da evolução de métodos para criação de novos medicamentos e tratamento de doenças. Entretanto, a conclusão desse processo leva muito tempo e as subetapas do mesmo são de suma importância na criação de novos candidatos a fármacos.

A descoberta de medicamentos começa com o diagnóstico de uma doença com sintomas bem definidos que reduzem a qualidade de vida. Convencionalmente, uma droga desejável é um produto químico ou uma combinação de produtos químicos que reduz os sintomas sem causar efeitos colaterais graves ao paciente (XIA, 2017). Após a identificação da doença, são buscados alvos biológicos, como bactérias, proteínas, DNA e RNA, e assim, identificar moléculas candidatas que possuam afinidade com o alvo biológico para estudo. As técnicas computacionais são o instrumento fundamental no auxílio das análises moleculares utilizando bases de dados já conhecidas e revisões bibliográficas de estudos semelhantes (GERTRUDES, 2013).

Para a utilização desses produtos químicos, o processo de análise de dados torna-se necessário, tendo em vista que, nos testes, inúmeros compostos moleculares multivariados são utilizados com milhares de descritores químicos que podem ser capazes de produzir uma resposta biológica em testes laboratoriais, ou seja, uma molécula é alterada em certas características para produzir diferentes respostas no alvo, por exemplo, uma bactéria. Essa técnica é conhecida como o estudo da relação estrutura-atividade, SAR (do inglês, *Structure Activity-Relationship*) (GERTRUDES et al., 2012).

A grande quantidade de dados que é analisada pode gerar muitas informações acerca da molécula estudada, mas o número excessivo de características pode se tornar um problema gerando longas informações desnecessárias. Além disso, as tarefas de análise levariam muito tempo, já que o crescimento linear de características aumenta exponencialmente o volume dos dados. Esse crescimento é conhecido como a maldição da dimensionalidade (CAMARGO, 2010). Uma das técnicas utilizadas neste processo é a técnica de redução de dimensionalidade, Análise de Componentes Principais, PCA (do inglês, *Principal Component Analysis*), em que um espaço de dados correlacionados é reduzido para um conjunto substancialmente menor de variáveis não correlacionadas que contém a maior parte da informação do conjunto original (HONGYU; SANDANIELO; OLIVEIRA JUNIOR, 2016).

Dessa forma métodos computacionais são utilizados para análise dos dados químicos buscando padrões, reduzindo número de dados redundantes e facilitando o reconhecimento dos melhores candidatos . Outra forma de se fazer isso é a aplicação de modelos de inteligência artificial, bem como Aprendizado de Máquina e Redes Neurais que podem permitir uma análise ampla dos dados, que podem realizar classificações prévias com base em padrões de características (GERTRUDES, 2013).

As classificações na inteligência artificial geralmente funcionam de forma que, ao se conhecer certos padrões em dados sabe-se classificar a que classe pertence uma variável alvo. Em alguns problemas, a variável de resposta deixa de ser única, dando origem aos problemas de classificação *multioutput*. Por exemplo, em vez de prever se uma imagem contém um gato ou um cachorro, um modelo de classificação *multioutput* pode prever se a imagem contém um gato e sua raça, e se contém um cachorro e sua raça (WANG et al., 2019).

Dentro dos métodos de aprendizado de máquina, também pode-se contar com as ferramentas de aprendizado profundo bem como as Redes Neurais Artificiais (RNA), que podem ser definidas como sistemas computacionais que processam informações baseadas no funcionamento do cérebro e nas conexões de seus neurônios. A rede adquire conhecimento por meio de um processo de aprendizado, e as forças das conexões entre os neurônios artificiais da rede, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido (CERRI, 2020).

1.2 Objetivos

A proposta deste trabalho de conclusão de curso consiste na criação de um modelo de ferramenta para redução no tempo de análise de dados laboratoriais para detecção de características químicas que produzem respostas biológicas, utilizando técnicas de redução de dimensionalidade de dados e modelos de aprendizado de máquina para classificação *multioutput*.

Os objetivos específicos este trabalho são:

- Avaliar o desempenho da predição antes e depois de empregar algoritmos de redução de dimensionalidade;
- Avaliar o desempenho das técnicas de redução de dimensionalidade aplicadas nos conjuntos de atributos;

1.3 Justificativa

Na atualidade, a inserção de processos para automatizar e reduzir o esforço humano e a criação de modelos de inteligência artificial como forma de evitar o trabalho manual de tarefas repetitivas e desgastantes são cada vez mais frequentes.

Portanto, este trabalho se justifica em dois principais pensamentos. O primeiro é a criação de um protótipo de modelo de previsão de dados, que consiga ajudar pesquisadores da Química Medicinal na etapa de análise dos teste feitos em laboratório podendo assim usar este mecanismo para prever quais moléculas terão as melhores respostas biológicas de acordo com sua características químicas. O segundo está relacionado ao aprendizado que este trabalho traz ao discente sobre algoritmos para análise de dados, que é um ramo crescente no campo das tecnologias de informação.

1.4 Estrutura do Trabalho

O restante do trabalho encontra-se organizado da seguinte forma: o Capítulo 2 traz uma revisão bibliográfica sobre as análises químicas e respostas biológicas empregadas nos conjuntos de dados estudados, as técnicas de redução de dimensionalidade e algoritmos de classificação *multioutput* utilizados. O Capítulo 3 apresenta as metodologias utilizadas nas análises e como o trabalho foi desenvolvido. No Capítulo 4 são apresentados os resultados de todas as técnicas empregadas neste trabalho, bem como algumas discussões relevantes acerca dos resultados obtidos. Por fim, o Capítulo 5 traz uma conclusão sobre o trabalho e sugestões para trabalhos futuros.

2 Referencial Teórico

O presente capítulo descreve, de forma geral, os algoritmos e métodos utilizados na condução do trabalho, além da apresentação de trabalhos relacionados ao tema de estudo.

2.1 Análise de relações entre estrutura química e atividade biológica

O campo da Química Medicinal abrange o processo de planejamento racional de novas substâncias bioativas, o qual envolve diversas etapas cruciais para o desenvolvimento de fármacos eficazes. Estas etapas são geralmente descritas por (i) caracterização de moléculas, compreendendo sua estrutura atômica e características estruturais fundamentais, que permitem a interação com um alvo biológico específico; (ii) modificação molecular de compostos existentes para otimizar sua atividade biológica. (iii) compreensão, em nível molecular, dos processos químicos envolvidos, bem como o isolamento de princípios ativos naturais; (iv) determinação e elucidação da estrutura química das moléculas em estudo; (v) validação de modelos SAR (Relações Estrutura-Atividade) por meio do uso de técnicas matemáticas ou estatísticas, permitindo a proposição de novos compostos (GERTRUDES, 2013).

As etapas do processo de análise na Química Medicinal, apresentadas no parágrafo anterior, demandam tempo e esforço, uma vez que as moléculas tendem a se tornar cada vez mais complexas, resultando na geração de novos compostos. Cada molécula, pode possuir um efeito distinto no corpo humano. Nesse contexto, as análises de SAR têm como objetivo estabelecer relações entre a estrutura química de compostos semelhantes, por meio da construção de modelos matemáticos.

Geralmente, as diferenças entre esses compostos são mínimas e envolvem pequenas alterações em sua estrutura química, como substituição de átomos. Essas modificações geram substâncias muito similares, porém com reatividades distintas em diferentes meios (FERREIRA; MONTANARI; GAUDIO, 2002).

Além das análises de SAR, existem também as Relações Quantitativas entre estrutura química e atividade biológica QSAR (do inglês *Quantitative Structure-Activity Relationship*)(YOUNG, 2009). As QSAR são ferramentas úteis para a compreensão e explicação do mecanismo de ação de fármacos em nível molecular, permitindo o projeto e desenvolvimento de novos compostos com propriedades biológicas desejáveis. Cada modificação em um composto pode levar a uma resposta específica em um determinado alvo biológico,

possibilitando a ativação ou inibição de processos bioquímicos em alvos como carboidratos, lipídios, proteínas, ácidos nucleicos, entre outros (MARTINS; FERREIRA, 2013).

Além dos descritores químicos, é fundamental estudar em detalhes o alvo biológico, a fim de compreender sua interação com a molécula em questão e obter uma resposta adequada no organismo. Para quantificar o nível de interação entre uma molécula e o meio em que ela será testada, os pesquisadores utilizam índices de atividade biológica, como a EC_{50} (concentração do composto que ativa 50% do alvo biológico) e a IC_{50} (concentração da substância que inibe 50% da atividade biológica). Esses índices são frequentemente convertidos para uma escala logarítmica (pEC_{50} e pIC_{50}), e quanto maior o valor logarítmico, melhor a resposta biológica da molécula em questão (GERTRUDES, 2013).

2.2 Redução de dimensionalidade

Atualmente, os experimentos têm a capacidade de gerar uma quantidade enorme de dados para análise. Geralmente, acredita-se que adicionar mais informações a um problema é melhor do que lidar com uma quantidade reduzida. No entanto, surge um desafio na área de ciência de dados conhecido como "maldição da dimensionalidade" (ALTMAN; KRZYWINSKI, 2018). Ao adicionar mais características a um conjunto de dados, aumentando o número de observações ou recursos (*features*), resulta em um crescimento exponencial na quantidade de dados, o que por sua vez aumenta o número de dimensões no espaço matemático do problema (CAMARGO, 2010).

Em outras palavras, a maldição da dimensionalidade ocorre quando a alta dimensionalidade dos dados torna a análise e interpretação deles desafiadoras. Isso se deve ao fato de que o aumento nas dimensões pode levar à dispersão dos dados e dificultar a identificação de padrões significativos. Além disso, o aumento na dimensionalidade também pode levar a problemas computacionais, tornando os cálculos mais complexos e exigindo mais recursos computacionais.

A representação adequada de conjuntos de dados químicos é uma questão crucial para os pesquisadores em Química Medicinal. A fim de aplicar técnicas de aprendizado de máquina, é necessário realizar transformações nos compostos por meio do cálculo de descritores físico-químicos que os representem de forma adequada. Essas representações podem incluir propriedades físico-químicas, características topológicas e descritores farmacológicos relevantes (YOUNG, 2009).

Durante o processo de transformação dos compostos químicos em um conjunto de dados, os pesquisadores podem utilizar diversos pacotes computacionais capazes de calcular centenas e até milhares de descritores químicos. Diante desse problema, a redução de dimensionalidade torna-se uma abordagem essencial na análise de dados químicos.

Em geral, as técnicas de redução de dimensionalidade têm o objetivo de encontrar uma representação compacta dos dados originais, preservando as informações essenciais e reduzindo o número de dimensões. A redução facilita a visualização dos dados, simplifica a análise estatística e pode até mesmo melhorar o desempenho de algoritmos de aprendizado de máquina.

Nesta seção descrevem-se duas técnicas que serão utilizadas no trabalho para a redução de dimensionalidade: A Análise de Componentes Principais (PCA) e os Autocodificadores (*Autoencoders*). Essas técnicas foram escolhidas por serem técnicas de redução de dimensionalidade citadas na literatura.

2.2.1 Análise de Componentes Principais

A análise de componentes principais (PCA) é amplamente utilizada na análise de dados multivariados, sendo considerada uma das técnicas mais comuns nesse contexto. O principal objetivo do método é reduzir a dimensionalidade do conjunto de dados original, preservando ao máximo a variabilidade presente. Essa redução é alcançada por meio de uma transformação linear que resulta em um novo conjunto de variáveis, conhecidas como componentes principais (PC), que não são correlacionadas entre si. Além disso, as componentes principais são ordenadas de forma que as primeiras expliquem a maior parte da variabilidade do conjunto de dados original (JOLLIFFE, 2002).

De acordo Gertrudes (2013), o algoritmo PCA é descrito da seguinte forma: a partir de um conjunto de dados $\mathbf{X}$, combinações lineares de suas características (*features*) $\mathbf{x}_i$ são geradas para produzirem PC's representadas por $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$, tal que $var(\mathbf{z}_1) \geq var(\mathbf{z}_2) \geq ... \geq var(\mathbf{z}_n)$. Para que essa transformação linear ocorra, determina-se suas bases ortogonais que são os autovetores $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n]$ obtidos a partir da matriz de correlação dos dados. Esses autovetores estão ordenados de acordo com seus respectivos autovalores $\Delta = \{\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n\}$. Com base nos autovetores, cada PC é obtida por meio da seguinte equação:

$$\mathbf{z}_i = \mathbf{X}\mathbf{a}_i, \tag{2.1}$$

em que cada $\mathbf{z}_i$ representa a i-ésima componente principal e $\mathbf{a}_i$ é o i-ésimo autovetor classificado por ordem decrescente dos autovalores.

Uma característica especial da técnica PCA é que cada autovalor representa a variância contida em cada componente principal ($\lambda_i = var(\mathbf{z}_i)$). Isso permite que os autovalores sejam organizados em ordem decrescente, de modo a preservar a máxima variabilidade nas primeiras componentes. A formação da primeira componente principal é denotada por:

$$\mathbf{z}_1 = \mathbf{x}_1 a_{11} + \mathbf{x}_2 a_{12} + ... + \mathbf{x}_n a_{1n}, \tag{2.2}$$

que possuirá a máxima variação, com a condição de que

$$a_{11}^2 + a_{12}^2 + ... + a_{1n}^2 = 1 \tag{2.3}$$

e que $var(\mathbf{z}_1)$ seja a maior possível. A segunda componente principal será representada por:

$$\mathbf{z}_2 = \mathbf{x}_1 a_{21} + \mathbf{x}_2 a_{22} + ... + \mathbf{x}_n a_{2n}, \tag{2.4}$$

que irá possui a segunda maior variação, com a condição de que

$$a_{21}^2 + a_{22}^2 + ... + a_{2n}^2 = 1, \tag{2.5}$$

e que $\mathbf{z}_2$ não tenha correlação com $\mathbf{z}_1$. A terceira componente principal será representada por:

$$\mathbf{z}_3 = \mathbf{x}_1 a_{31} + \mathbf{x}_2 a_{32} + ... + \mathbf{x}_n a_{3n} \tag{2.6}$$

que irá possuir a terceira maior variação, com a condição de que

$$a_{31}^2 + a_{32}^2 + ... + a_{3n}^2 = 1, \tag{2.7}$$

e que $\mathbf{z}_3$ não tenha correlação com $\mathbf{z}_1$ e $\mathbf{z}_2$. Este cálculo é realizado até que sejam geradas as n componentes principais.

A redução de dimensionalidade por meio do método PCA envolve a análise dos autovalores (que representam a variância) de cada componente. Nesse contexto, selecionam-se como bases de transformação os autovetores correspondentes aos m maiores autovalores, onde m é menor que o número de dimensões original ($m < n$). A escolha de m está diretamente relacionada com a quantidade de variância que se pretende preservar no novo espaço vetorial resultante.

Além disso, a técnica PCA também é aplicada na área de Química Medicinal para avaliar a relevância de cada variável do conjunto de dados original. Isso é feito por meio da análise dos coeficientes presentes na matriz de autovetores utilizada na transformação dos dados (GERTRUDES, 2013).

A aplicação da PCA nas análises SAR tenta propor, para o conjunto de dados como um todo quais características moleculares podem ter influência na resposta biológica do alvo. A PCA pode então separar estatisticamente os compostos para que se possa classificá-los entre os ativos e inativos se busque a relevância de suas características. Entretanto, características que possuem grande poder descritivos podem perder relevância nas componentes principais (CASTRO et al., 2009).

No entanto, é importante mencionar que a técnica de Análise de Componentes Principais (PCA) apresenta algumas desvantagens a serem consideradas. Primeiramente, ela é sensível a *outliers*, ou seja, dados que se encontram significativamente afastados dos demais pontos do conjunto. A presença desses *outliers* pode distorcer a análise e comprometer os resultados obtidos (HONGYU; SANDANIELO; OLIVEIRA JUNIOR, 2016).

Além disso, a PCA pode ter um desempenho insatisfatório quando a base de dados contém muitos valores nulos ou faltantes. A presença desses valores faltantes pode afetar a precisão das estimativas dos componentes principais e introduzir viés nos resultados. Outra limitação da PCA está relacionada à situação em que o número de variáveis é maior do que o número de amostras disponíveis. Essa condição pode dificultar a aplicação da técnica, uma vez que a estimativa dos componentes principais requer um número suficiente de amostras para uma representação adequada do espaço de dados (HONGYU; SANDANIELO; OLIVEIRA JUNIOR, 2016).

2.3 Autocodificadores

Antes da definição do que é um autocodificador e das suas possibilidades de aplicação, serão apresentados conceitos básicos sobre redes neurais artificiais (RNA), que são base para construção desses tipos de métodos.

As redes neurais artificiais são um tipo de aprendizado de máquina, que funcionam como uma rede de processamento densa e que pode aprender com a experiência. Elas foram projetadas para funcionar utilizando camadas de nós (conhecidos como neurônios artificiais) interconectados e podem ser usadas para reconhecer padrões, aprender conceitos e tomar decisões. O conhecimento é adquirido por um ambiente por meio de processo de aprendizado e assim as forças de conexão entre neurônios são utilizadas para armazenar o conhecimento adquirido. Sendo assim, seu funcionamento se assemelha ao de um cérebro humano (HAYKIN, 2001).

Um neurônio artificial pode ser representado por um nó, que processa as informações com base em uma ou mais entradas e produz uma saída. O funcionamento deste neurônio acontece de forma que os valores de entrada são combinados ponderadamente e somados por uma função $\mathbf{f}$, como mostra a Figura 1. Suponha um objeto com $\mathbf{d}$ atributos que pode ser representado por $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_d}]$ e um neurônio com $\mathbf{d}$ terminais de entrada. As entradas são ajustadas por pesos $\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_d}$, que podem ser representados na forma vetorial como $\mathbf{w} = [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_d}]$ os pesos representam um valor de ajuste do aprendizado do neurônio que se corrige conforme as iterações (FACELI et al., 2011).

Sendo assim, a função de entrada recebida por um neurônio u pode ser representada pela equação 2.8:

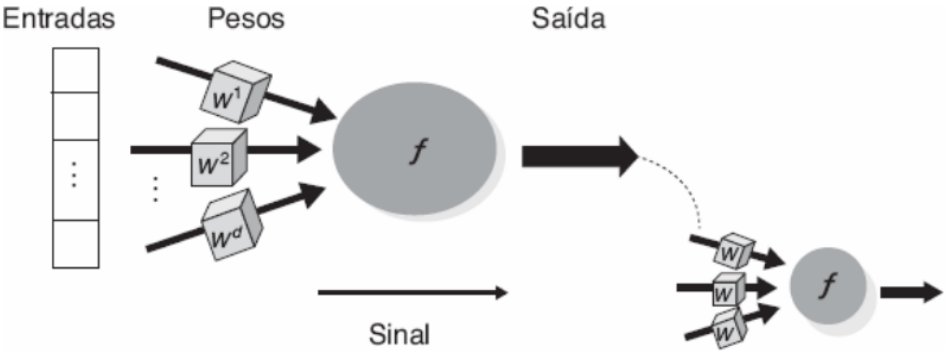


Figura 1 – Representação de neurônio artificial Fonte: (FACELI et al., 2011)

$$u = \sum_{j}^{d} (x_j \times w_j) \tag{2.8}$$

Os pesos dos neurônios $\mathbf{w}_j$ podem assumir valores positivos ou negativos. Quando o peso de um neurônio é igual a 0, o mesmo indica que a conexão de entrada do neurônio não existe. A saída do neurônio é configurada de acordo com uma função de ativação aplicada na entrada total do neurônio. Existem diversos tipos de funções utilizadas para ativação de um neurônio. Dentre elas, pode-se citar a função linear, a função limiar e a função sigmoide (SIQUEIRA et al., 2019).

As RNAs podem ser caracterizadas por dois aspectos básicos: (i) a arquitetura, que se refere ao número e tipo de conexões das unidades de processamento (neurônios) e (ii) o aprendizado, que são as regras utilizadas para ajustar os pesos da rede (SIQUEIRA et al., 2019).

De forma geral, uma rede neural pode ser configurada com o número de neurônios que for necessário para a tarefa em questão. Esses neurônios podem ser organizados de diferentes maneiras para criar a arquitetura desejada. Uma forma comum de organização é a criação de camadas de neurônios, e quando uma rede neural possui mais de uma camada entre a entrada e a saída, é chamada de rede neural multicamadas. As camadas intermediárias entre a entrada e a saída, também conhecidas como camadas ocultas, desempenham um papel fundamental no processamento dos dados (HAYKIN, 2001). Na Figura 2, é apresentado um exemplo de uma rede neural multicamadas com duas camadas ocultas.

É importante destacar que a quantidade e o tamanho das camadas ocultas podem variar dependendo do problema e da complexidade da tarefa. A arquitetura da rede neural, incluindo o número de camadas ocultas e o número de neurônios em cada camada, é uma escolha crucial que afeta o desempenho e a capacidade de aprendizado da rede. Portanto, é importante realizar ajustes e experimentações para encontrar a configuração mais adequada para cada aplicação (FACELI et al., 2011).

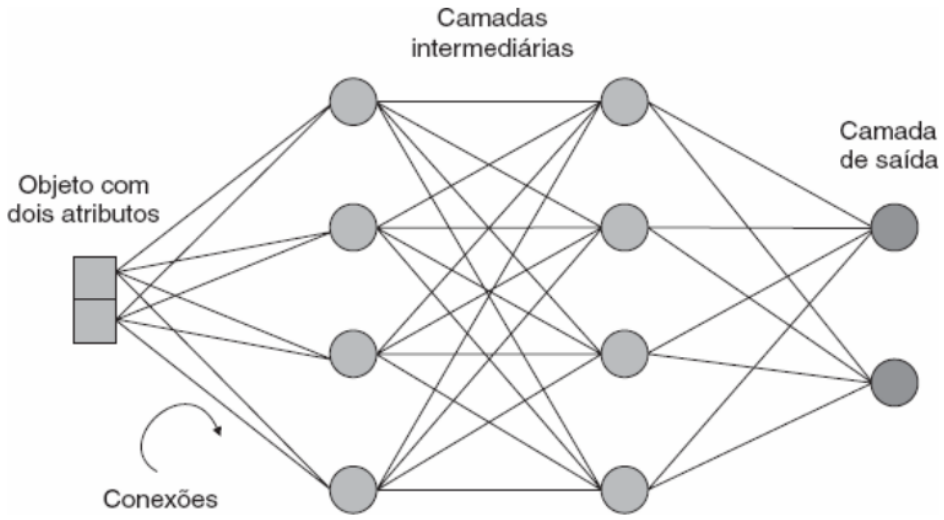


Figura 2 – Rede Neural com duas camadas ocultas Fonte: (FACELI et al., 2011).

Os neurônios de uma rede também podem apresentar duas diferentes arquiteturas básicas. As redes em que as informações fluem da camada de entrada para a camada de saída de neurônios são chamadas de redes *feedfoward*, pois, não possuem conexões de retroalimentação entre os neurônios. A retroalimentação consiste em um neurônio poder receber saídas de neurônios de sua mesma camada ou de uma camada posterior, podendo até mesmo receber a sua própria saída. As redes que possuem essa topologia são chamadas redes *feedback* ou recorrentes (FACELI et al., 2011).

2.3.1 O que de fato é uma Rede Neural Autocodificadora?

Uma Rede Neural Autocodificadora, também conhecida como *Autoencoder*, é um tipo de Rede Neural Artificial (RNA) frequentemente utilizada para tarefas como redução de ruídos, redução de dimensionalidade e classificação de padrões. A principal característica de um *Autoencoder* é sua capacidade de copiar as entradas para as saídas, buscando assim aprender representações mais compactas e generalizadas dos dados. Geralmente, um *Autoencoder* é composto por três camadas de neurônios: a camada de codificação (*encoder*), o gargalo (*bottleneck*) e a camada de decodificação (*decoder*) (SIQUEIRA et al., 2019). A Figura 3 ilustra a arquitetura típica de um *Autoencoder*.

A camada de codificação tem a função de comprimir os dados em um espaço latente de menor dimensão do que a entrada original, enquanto a camada de decodificação é responsável por reconstruir a entrada a partir da representação latente. O gargalo é a representação reduzida dos dados, contendo informações generalizadas.

Uma característica importante a ser observada em um *Autoencoder* com múltiplos pares de camadas ocultas, conforme mostrado na Figura 3, é a simetria estrutural. Essa

[1]Fonte: http://bit.ly/3JmLDT5

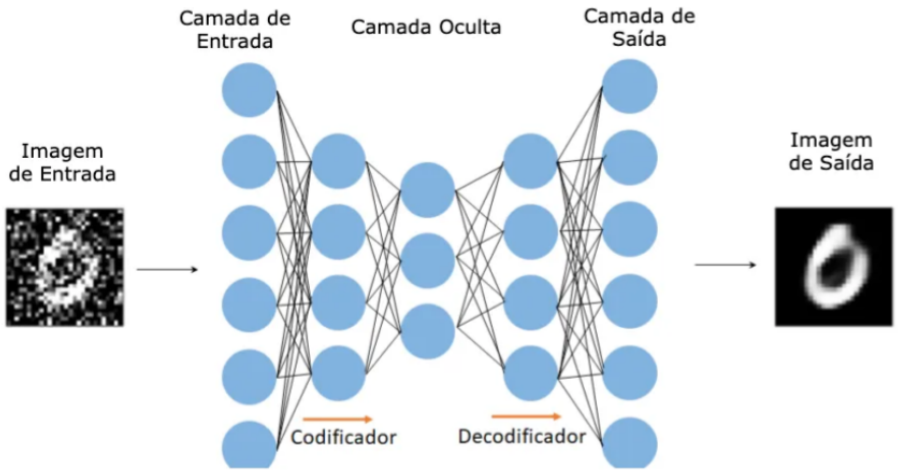


Figura 3 – Representação da arquitetura de um *Autoencoder*
Fonte: Medium[1]

simetria garante que as camadas ocultas possuam uma dimensão menor do que as camadas de entrada e saída, com as camadas de saída sendo simétricas às camadas de entrada. As camadas ocultas contêm menos neurônios do que as camadas de entrada e saída, permitindo que a rede seja capaz de codificar as informações de entrada por meio da função de codificação e reconstruir as entradas nas saídas utilizando a função de decodificação (HAYKIN, 2001).

A função de codificação pode ser representada pela função $\mathbf{f}$, dada por:

$$\mathbf{Z} = \mathbf{f}(\mathbf{x}) = \mathbf{s}(\mathbf{W}\mathbf{x} + \mathbf{B}), \tag{2.9}$$

onde $\mathbf{x}$ representa uma matriz de dados $\mathbf{m} \times \mathbf{n}$, $\mathbf{W}$ representa a matriz de pesos, $\mathbf{B}$ é o vetor de viés (bias) dos neurônios e $\mathbf{s}$ é a função de ativação escolhida para o sistema.

Assim é aplicada a função de ativação $\mathbf{s}$ no produto dos elementos das linhas da matriz de dados $\mathbf{x}$ pelas colunas da matriz $\mathbf{W}$ e somados ao respectivo viés $\mathbf{B}$. Obtém-se tem o resultado Z pode ser escrita como $\mathbf{z_{ij}}$ que é produzida na saída do gargalo, utilizada para extração de características de um dado problema e formando a representação latente de um conjunto de dados (SIQUEIRA et al., 2019).

De forma análoga, a decodificação utiliza os dados de saída do gargalo para copiá-los voltando ao espaço latente original aplicando a função $\mathbf{g}(\mathbf{z})$. A reconstrução da base $\mathbf{X}'$ e dada por:

$$\mathbf{X}' = \mathbf{g}(\mathbf{z}) = \mathbf{s}(\mathbf{Z}\mathbf{w}' + \mathbf{B}'), \tag{2.10}$$

Em que $\mathbf{Z}$ é uma matriz em que cada linha é uma representação de um exemplo $\mathbf{x}$ e cada linha é atribuída a $\mathbf{m}$ valores de $\mathbf{n}$ dos atributos originais. $\mathbf{w}'$ é a matriz de pesos

reconstruída para **m** atributos de entrada por **n** neurônios. E $\mathbf{B}'$ é o vetor de viés de cada neurônio da camada de saída (SIQUEIRA et al., 2019).

2.4 Classificação de múltiplos rótulos

Nesta seção, são descritos os algoritmos utilizados para a classificação de múltiplos rótulos, também conhecidos como algoritmos *multioutput*, empregados neste trabalho de aprendizado supervisionado, no qual as saídas esperadas são conhecidas. Esses algoritmos se destacam pela capacidade de lidar com múltiplas saídas, ou seja, em vez de prever apenas um rótulo em um determinado problema, eles são capazes de prever vários rótulos simultaneamente (CHERMAN, 2013). Entre os algoritmos utilizados, citam-se o *Multioutput Classifier* e o *Classifier Chain*, que utilizam por base nos modelos de classificação os algoritmos *K-Nearest Neighbors* e *Decision Tree*.

O *Multioutput Classifier* é uma estratégia que consiste em ajustar um classificador para cada alvo, permitindo assim várias classificações de variáveis de destino. O objetivo dessa classe é estender os estimadores para classificar mais de um alvo de destino, diferentemente do que é feito normalmente em um único alvo (BORCHANI et al., 2015). A Figura 4 mostra um esquema do algoritmo *Multioutput Classifier*. Nela pode-se ver que dados em que as variáveis alvo (representadas pelas cores diferente no conjunto de dados) são atribuídas ao modelo estimador que é estendido por um classificador de árvore de decisão.

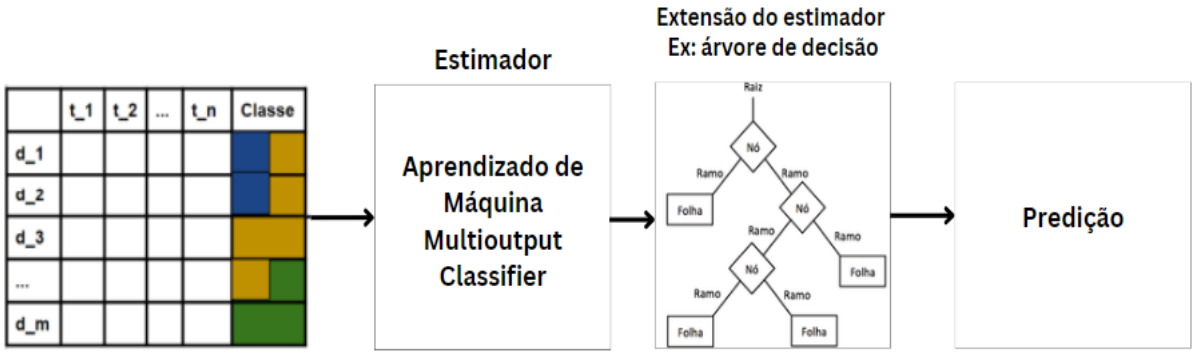


Figura 4 – Representação de um *Multioutput Classifier*. Fonte: Próprio autor

.

O algoritmo *Classifier Chain* utiliza uma abordagem de classificação em cadeia de classificadores binários para lidar com problemas de classificação com múltiplas saídas. Cada saída é modelada por um classificador separado, e cada saída subsequente leva em consideração a saída anterior como uma das entradas (PEDREGOSA et al., 2011). A Figura 5 mostra as cadeias de classificadores. O início da classificação é marcado pelo primeiro classificador, C_1, e segue até o último classificador, com a troca de informações sobre rótulos por meio do espaço de destaque. Portanto, há uma preservação da dependência entre os rótulos. No entanto, é importante notar que os resultados podem ser diferentes

dependendo da ordem em que as cadeias são consideradas (SANTOS; ROSSI; SUL–MS–BRAZIL, s.d.).

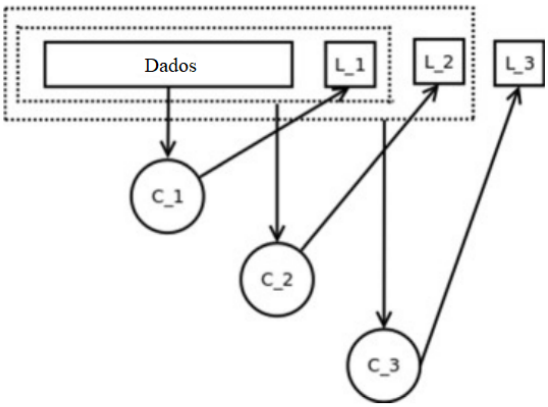


Figura 5 – Representação de um *Classifier Chain*. Fonte: (SANTOS; ROSSI; SUL–MS–BRAZIL, s.d.)

Na estratégia de classificação de múltiplos alvos, vários algoritmos clássicos podem ser utilizados, incluindo as Árvores de Decisão (DT) e o *K-Nearest Neighbors* (KNN). As Árvores de Decisão são modelos de aprendizado caracterizados por uma estrutura em forma de "nó-folha", na qual o modelo toma decisões dividindo a base de dados em conjuntos menores. Elas recebem esse nome porque, a cada iteração, o problema é dividido, formando uma estrutura semelhante a uma árvore. Nos modelos de Classificação por Árvores de Decisão, o objetivo é prever o valor da variável alvo com base em suas características. A Figura 6 ilustra uma Árvore de Decisão e o espaço de decisões correspondente.

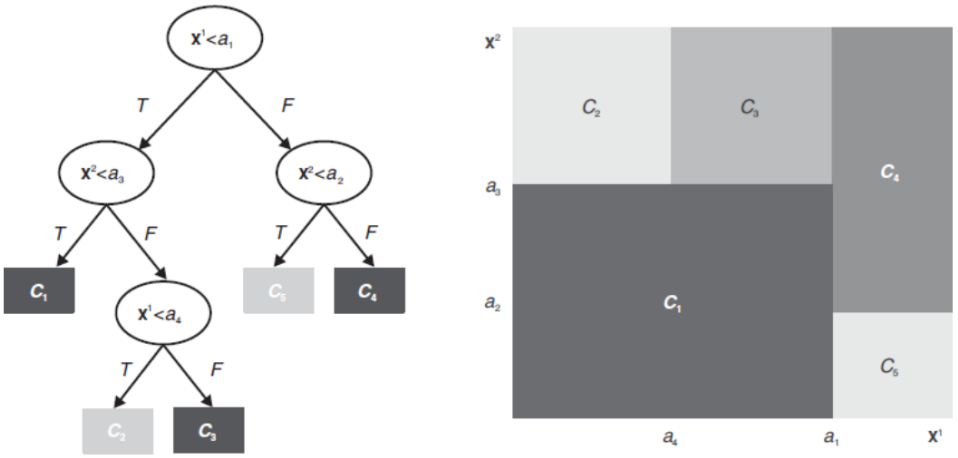


Figura 6 – Árvore de Decisão e Espaço de Decisões. Fonte: (FACELI et al., 2011)

A árvore apresenta os atributos x^1 e x^2, bem como o espaço no qual uma variável pode ser classificada. Cada nó da árvore representa uma condição para satisfazer uma região específica no espaço, enquanto cada folha corresponde a uma região específica. Dessa

forma, é possível classificar com precisão qualquer exemplo que possua esses atributos (FACELI et al., 2011).

O algoritmo *K-Nearest Neighbors* é um modelo de classificação baseado no agrupamento dos dados com base na distância entre seus vizinhos. A forma mais utilizada nas literaturas para calcular a distância entre dois pontos é chamada de distância euclideana, representada pela Equação 2.11. Durante a fase de treinamento, as distâncias calculadas pelos pontos **p** e **q** em **n** dimensões são usadas para atribuir classes conhecidas aos dados rotulados mais próximos. Posteriormente, durante a fase de teste, as previsões são feitas para instâncias desconhecidas utilizando os dados de treinamento mais próximos das instâncias conhecidas (MARTINS, 2011). Porém, a quantidade de vizinhos a ser analisada pode alterar a classificação do alvo na fase de teste. A Figura 7 exemplifica o impacto dos **k** vizinhos na classificação de indivíduos saudáveis e doentes com base em 2 exames (dimensões) feitos em pacientes. O ponto teste representado por "?" indica que o ponto pertenceria a classe doente caso analisados os 3 vizinhos mais próximos. Porém, ao observar os 5 vizinhos mais próximos o ponto teste pertence a classe saudável (FACELI et al., 2011).

$$D = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.11}$$

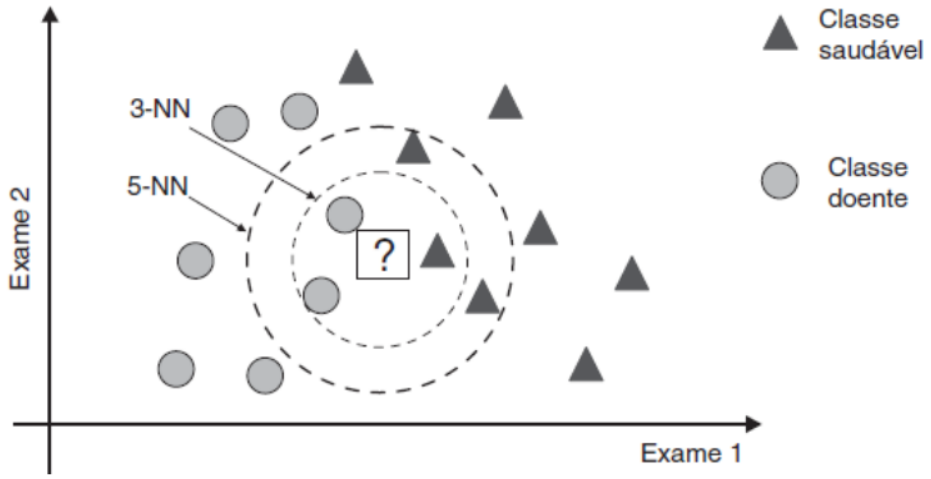


Figura 7 – Representação de um *K-Nearest Neighbors*. Fonte: (FACELI et al., 2011)

2.5 Métricas de avaliação

O Erro Quadrático Médio (MSE, do inglês *Mean Squared Error*) é uma das funções de perda mais utilizadas na aprendizagem de *Autoencoders* para redução de dimensionalidade. Ele é calculado como a média dos quadrados dos erros entre os sinais de saída

reconstruídos ($\mathbf{x}'$) e os sinais de entrada originais ($\mathbf{x}$) para os exemplos de treinamento. A equação que caracteriza o Erro Quadrático Médio é a seguinte:

$$MSE = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n}(x_{ij} - x'_{ij})^2 \tag{2.12}$$

Nessa equação, **r** é a quantidade de exemplos de treinamento no conjunto de dados, **n** é a quantidade de atributos de treinamento, $\mathbf{x_{ij}}$ representa um atributo de entrada e $\mathbf{x'_{ij}}$ representa o correspondente atributo de saída reconstruído.

Durante o treinamento do *Autoencoder*, o objetivo é minimizar o MSE a cada época (ciclo de iteração), buscando aproximar os sinais de saída dos sinais de entrada. Quanto mais próximo de zero for o valor do MSE, melhor será o desempenho do modelo na tarefa de reconstrução dos dados (SIQUEIRA et al., 2019).

Quanto à métrica de avaliação dos modelos de classificação, utilizou-se o *F-Measure*. O *F-Measure* combina dois conceitos importantes: precisão e sensibilidade. A precisão mede a proporção de previsões positivas corretas em relação a todas as previsões positivas feitas. A sensibilidade mede a proporção de verdadeiros positivos corretamente classificados em relação a todos os exemplos verdadeiramente positivos. O *F-Measure* é uma média ponderada dessas duas medidas, representando uma medida geral do desempenho do modelo de classificação. A fórmula para o cálculo da *F-Measure* é a seguinte:

$$F = \frac{2 \times \text{precisão} \times \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}} \tag{2.13}$$

onde a precisão e a sensibilidade são calculadas de acordo com as definições padrão (FACELI et al., 2011).

3 Metodologia

Neste capítulo, descreve-se como foi realizada a estruturação e condução dos experimentos envolvendo a análise de dados da área de Química Medicinal.

3.1 Linguagem de Programação e Ambiente de execução

Durante a execução do trabalho, utilizou-se a linguagem de programação *Python*, na versão 3.8.10, para realizar toda a análise de dados. Em relação ao Ambiente de Desenvolvimento Integrado IDE (do original, em Inglês, *Integrated Development Environment*), utilizou-se a base de implementação do *Google Colaboratory*, que permite a execução de códigos escritos na linguagem de programação *Python* na nuvem de forma simples e eficiente.

3.2 Conjunto de dados

O conjunto de dados utilizado possui 38 moléculas (observações) estudadas por Xue et al. (2020) no Jornal Europeu de Química Medicinal que testaram agentes antibacterianos contra *Staphylococcus aureus*. Em seguida estudos feitos pelos autores Fernandes e Maltarollo (2022), membros da Faculdade de Farmácia da Universidade Federal de Minas Gerais. Os dados contém informações sobre infecções causadas pelo *Staphylococcus aureus*, que representam um sério problema de saúde, especialmente devido à prevalência de cepas resistentes. O S. aureus é resistente à meticilina foi classificado pela Organização Mundial da Saúde como uma das maiores prioridades para o desenvolvimento de novos antibióticos (WORLD HEALTH ORGANIZATION, 2017).

Dos estudos realizados por Fernandes e Maltarollo (2022), tomamos por bases dois alvos biológicos de uma molécula, para a classificação por meio dos algoritmos de *multioutput*: (i) atividade inibitória da girase do DNA e (ii) classificação da atividade antibacteriana. Para os dois casos, tem-se o valor 1 que indica que a molécula apresenta resposta positiva contra o alvo biológico, enquanto o valor 0 indica o oposto (a molécula não contém resposta contra o alvo biológico). Sendo assim, pode-se obter quatro combinações possíveis de rótulo para uma molécula, sendo eles:

- $[1, 1]$, representando moléculas ativas para os alvos (i) e (ii);
- $[1, 0]$, representando moléculas ativas para o alvo (i) e inativas para o alvo (ii);
- $[0, 1]$, representando moléculas inativas para o alvo (i) e ativas para o alvo (ii);

- $[0, 0]$, representando moléculas inativas para os alvos (i) e (ii).

A representação das moléculas no conjunto de dados é feita por meio de códigos SMILES (*Simplified Molecular Input Line Entry System*), que utilizam uma notação textual para descrever as relações entre os átomos em uma molécula. O código SMILES foi introduzido por Arthur and Morgan em 1965 (MORGAN, 1965) e é amplamente utilizado na área de Química Medicinal.

Para utilização dos dados por parte dos algoritmos de *classificação multioutput*, é necessário que esta informação textual seja transformada em um conjunto na forma matricial. Para isso, são utilizadas diversas formas de cálculo de descritores químicos a partir de uma representação SMILES. No presente trabalho, utilizou-se quatro formas de cálculo de descritores químicos: (i) *Substructure*, (ii) *Pubchem*, (iii) *descritores KR* e (iv) *Atompair* (MAURI; CONSONNI; TODESCHINI, 2017). A Tabela 1 apresenta a descrição de cada um dos subconjunto de atributos, além da quantidade de atributos calculados para cada subconjunto.

Tabela 1 – Quantidade de atributos em cada conjunto de dados

Descrição	**Quantidade**
Substructure	308
Pubchem	882
KR	4861
Atompair	781

Ao final da composição do banco de dados, obteve-se um conjunto de dados com quatro diferentes configurações de atributos. Neste trabalho analisou-se cada um dos subconjunto de atributos separadamente no momento de apresentação aos algoritmos de aprendizado de múltiplos rótulos.

3.3 Tratamento dos dados

3.3.1 Eliminação de Atributos e Normalização

Em todos os subconjuntos foram excluídos apenas um atributo correspondente ao nome das moléculas, pois, este não carrega nenhuma informação de relevância para a análise de dados.

Como mencionado anteriormente, os dados somente assumem valores booleanos (1 ou 0). Sendo assim não se fez necessário a etapa de normalização dos dados.

3.4 Redução de Dimensionalidade

Nesta etapa do trabalho utilizou-se a técnica PCA e a Rede Neural *Autoencoder* para a redução de dimensionalidade dos dados. Na aplicação da PCA, considerou-se como critério para redução as quatro componentes principais reportadas pela técnica, uma vez que, em geral, as quatro componentes mantinham ao menos 60% de explicação da variabilidade dos dados. Também vale ressaltar que para a PCA foram utilizados todos os 38 exemplos do conjunto de dados. A partir da quinta componente, as subsequentes explicavam uma porcentagem muito baixa dos conjuntos e portanto não foram necessárias a adição de novas componentes

A Rede Neural *Autoencoder* também foi utilizada para redução de dimensionalidade dos dados. De forma resumida, ela funciona de maneira que tenta copiar suas entradas para suas saídas transformando os dados de entrada de um neurônio para um espaço latente de dimensão menor.

A Rede Neural *Autoencoder* possui muitos parâmetros e foi configurada para que após os dados passarem pela camada de entrada fossem reduzindo a quantidade de atributos. Assim, na camada de codificação eles foram divididos por 2, 5 e 7 vezes até que se atingisse o gargalo, o qual foi configurado para somente 30 dimensões. Em seguida na camada de decodificação os dados voltam ao tamanho original respeitando a mesma divisão mencionada.

Foram produzidas 9 camadas de neurônios, sendo que a quinta camada produz o gargalo, sendo as 4 primeiras camadas de entrada e as 4 últimas as camadas de saída. Para ajuste dos pesos da rede utilizou-se o otimizador *Adam* e como função de ativação utilizou-se a função sigmoide. Essas escolhas foram feitas por serem funções mais comuns na literatura e proporcionam resultados satisfatórios.

O modelo foi treinado por 50 épocas no conjunto de treinamento utilizando um tamanho de lote com 16 exemplos. A camada de codificação foi salva para apresentação dos dados na dimensão reduzida.

3.5 Classificação

Após a etapa de redução de dimensionalidade os dados foram divididos em conjuntos de treino e teste para criação dos modelos de classificação e assim treiná-los da forma apropriada. A divisão dos conjuntos foi feita de forma que 80% dos dados foram utilizados no conjunto de treino e 20% no conjunto de teste.

Depois da divisão, passou-se para a etapa de classificação utilizando classificadores *multioutput*. Os algoritmos utilizados foram o *Multioutput Classifier* (MC) e o *Classifier Chain* (CC). Esses estimadores exigem um classificador base que seja treinado para prever

cada uma das saídas.

No caso do estudo realizado, foram utilizados o *K-Nearest Neighbors Classifier* (KNN) e o *Decision Tree Classifier* (DT) como algoritmos base para os classificadores *multioutput*. Esses classificadores são responsáveis por prever as classes das diferentes saídas.

Durante a implementação dos algoritmos, os hiperparâmetro do *K-Nearest Neighbors Classifier* foram configurado para analisar os 5 vizinhos mais próximos dos dados. E para o *Decision Tree Classifier* foi utilizada a configuração de profundidade máxima da árvore igual a 8 ramos.

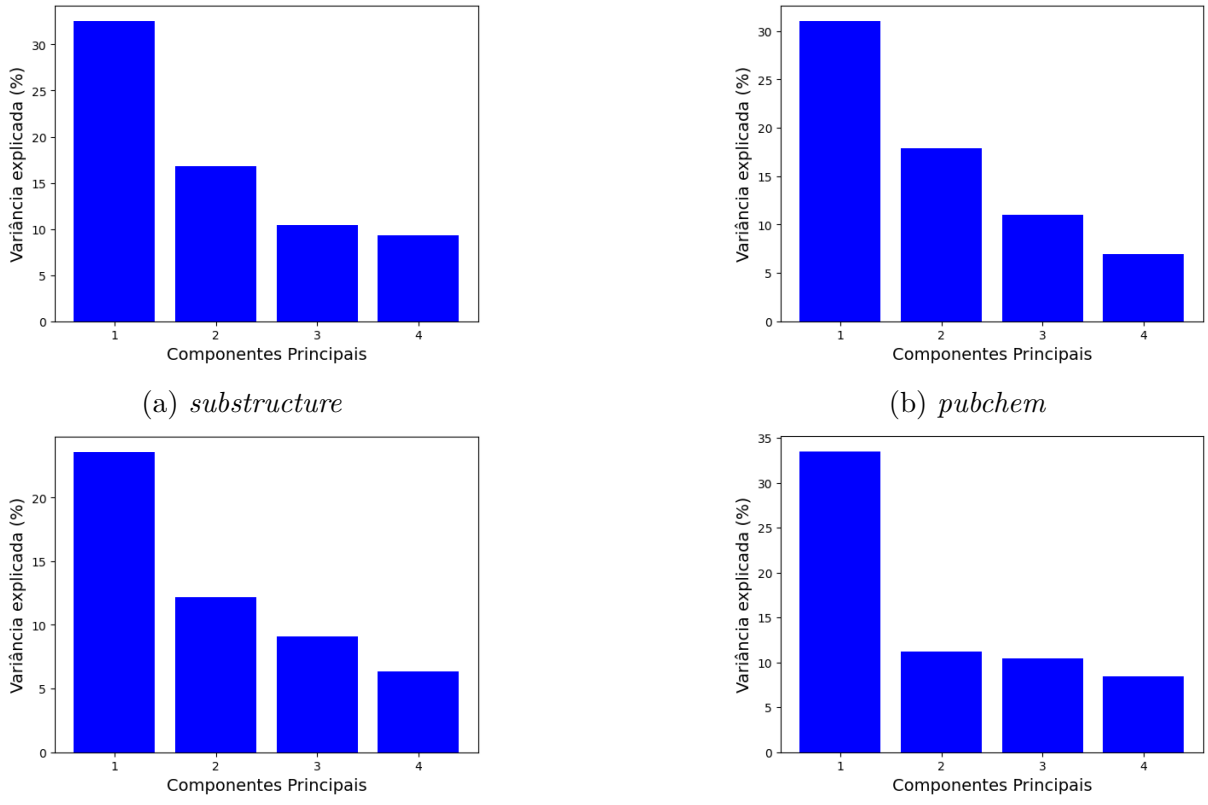
Como métrica de avaliação, foram calculados os coeficientes F-Measure para cada conjunto de treino e teste. Devido à aleatoriedade do modelo, o algoritmo foi repetido 50 vezes e os resultados foram armazenados para análise. Para uma melhor visualização e interpretação, foram calculadas a média e o desvio padrão dos resultados, permitindo avaliar a precisão das respostas dos classificadores.

Essa análise estatística dos resultados é importante para entender a consistência e o desempenho dos classificadores *multioutput*, fornecendo informações sobre a capacidade de predição do modelo em relação às diferentes classes.

4 Resultados

Como descrito anteriormente, o conjunto de treinamento e de teste utilizado neste trabalho nos permite analisar de forma prática qual dos algoritmos testados conseguirá reduzir a dimensionalidade dos dados mostrando a melhor predição dentre os conjuntos de atributos selecionados. Este capítulo descreve quais os resultados foram obtidos neste trabalho.

Durante a aplicação da Análise de Componentes Principais (PCA), foram extraídas 4 componentes para transformar a base de dados em uma matriz de 4 colunas não correlatas. Cada componente explicando uma porcentagem do conjunto de atributos estudado. A Figura 8 mostra a extração de cada componente para cada um dos conjuntos de atributos.



(a) *substructure* (b) *pubchem*

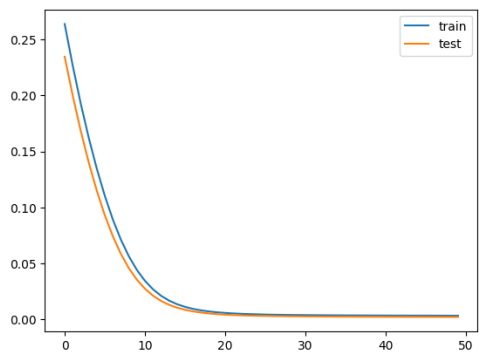
(c) *descritores KR* (d) *atompair*

Figura 8 – Gráficos das componentes principais para cada conjunto de atributos.

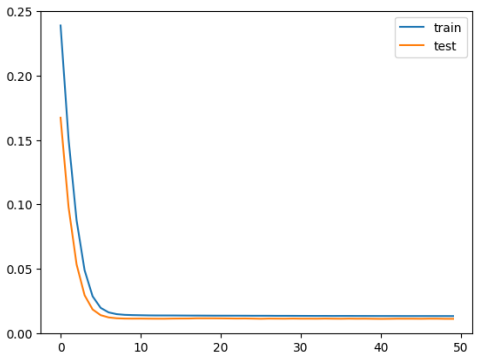
Como pode se observar a primeira componente principal dos conjuntos de atributos produziu 33% de variância explicada do conjunto de dados. O que indica que esta componente explica esta porcentagem dos dados e assim sucessivamente somando cerca de 60% dos conjuntos de atributos em cada conjunto para as 4 primeiras componentes.

Esta situação pode ocorrer devido a não linearidade dos dados, o que torna esta técnica pouco proveitosa.

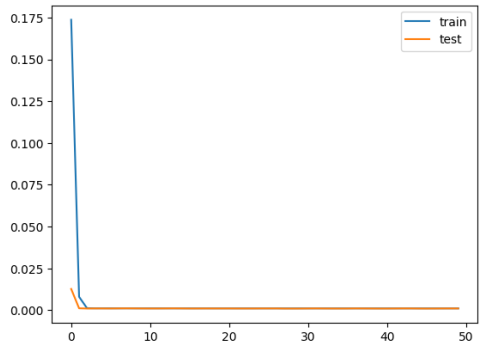
Para a redução de dimensionalidade utilizando a Rede Neural Autoencoder, o Erro Quadrático Médio foi minimizado de forma a permanecerem praticamente nulos com o passar das épocas, o que mostra que a redução de fato ocorreu e que a reconstrução das bases são próximas aos dados originais. A Figura 9 mostra gráfico do Erro Quadrático Médio em função das épocas durante a execução do algoritmo. Pode-se perceber que após a rede ser treinada, o erro leva menos tempo para atingir um valor nulo no conjunto de teste.



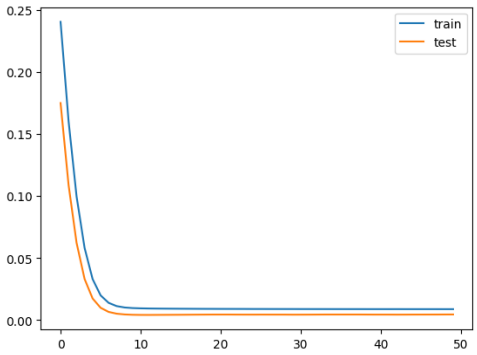
(a) *substructure*



(b) *pubchem*



(c) *descritores KR*

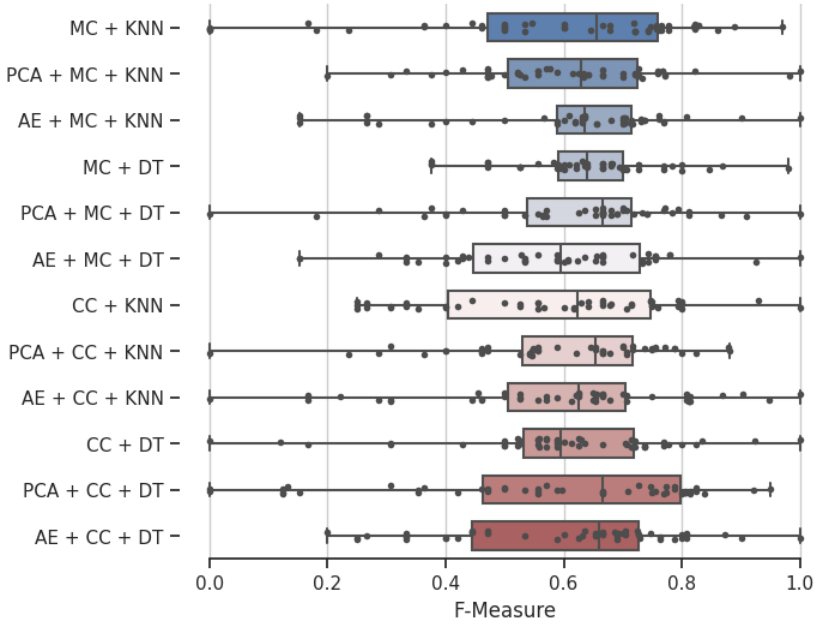


(d) *atompair*

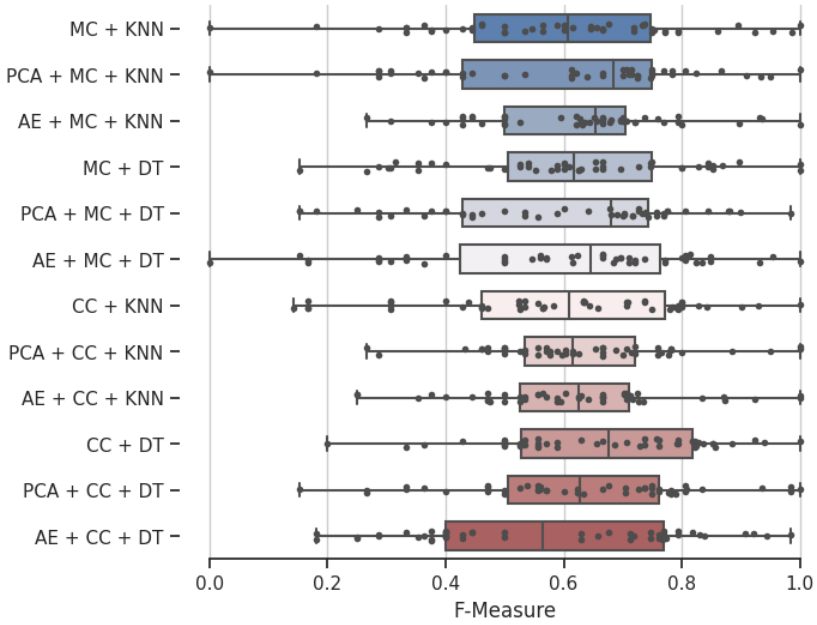
Figura 9 – Erro Quadrático Médio em função das épocas para cada conjunto de atributos.

Após as reduções de dimensionalidade, foram colocados em prática os algoritmos de classificação para prever as respostas biológicas das moléculas. Os códigos foram organizados de forma que, os dados puros, reduzidos pela PCA e pelo *Autoencoder* (AE) foram submetidos a 4 algoritmos de classificação sendo eles: *Multioutput Classifier* (MC) *Classifier Chain* (CC) combinados com *K-Nearest Neighbors Classifier* (KNN) e *Decision Tree Classifier* (DT) executados nos 4 conjuntos de atributos separados em treino e teste. Extraindo assim os coeficientes de F-Measure de cada um deles e sendo iterado um total de 50 vezes para para extração das médias e desvios padrão para análises estatísticas. Totalizando assim um total de 48 saídas para cada conjunto de atributos.

No primeiro conjunto que apresenta 307 atributos pode-se observar em destaque na Figura 10a a maior média dos coeficientes de F-Measure no algoritmo *Multioutput Classifier* combinado com *Decision Tree Classifier* apresentando um score de 0,642 para o conjunto de treino com um baixo desvio padrão de 0,121. Indicando que a medida de fato possui grande assertividade dentre as iterações. Porém no conjunto de teste este valor não corresponde ao melhor valor mas também atinge um bom score de 0,613 com desvio de 0,199. Um valor de desvio não muito baixo mostrando que existem muitos valores distantes da média, ou seja, a precisão do modelo é boa, porém há certa dispersão na medida. Podemos observar também que o maior score no conjunto de testes de 0,664 nos algoritmos *Classifier Chain* e *Decision Tree Classifier* na Figura 10b, entretanto este valor é superior ao conjunto de treinamento de 0,601 o que pode indicar *underfitting*, um problema que ocorre quando o modelo não consegue aprender as informações mais importantes entre as classes.



(a) Treino

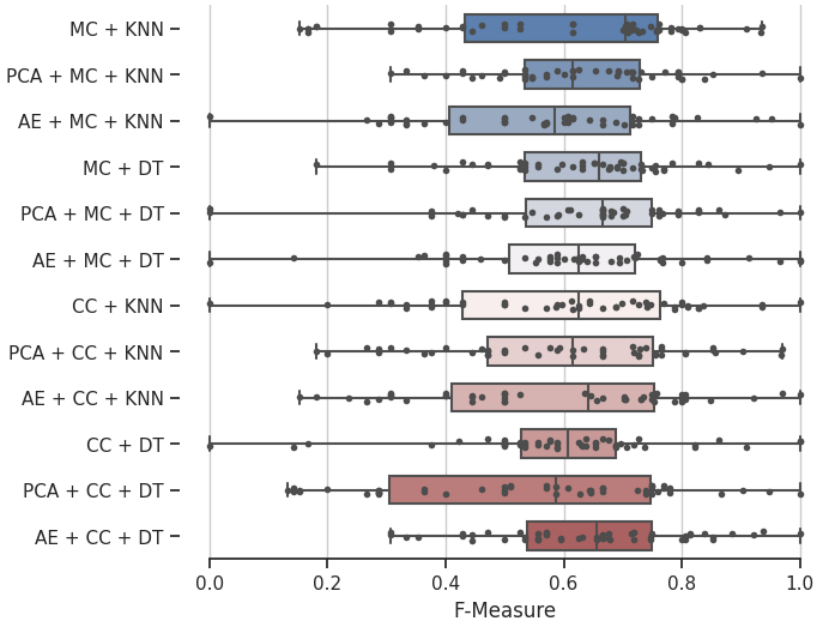


(b) Teste

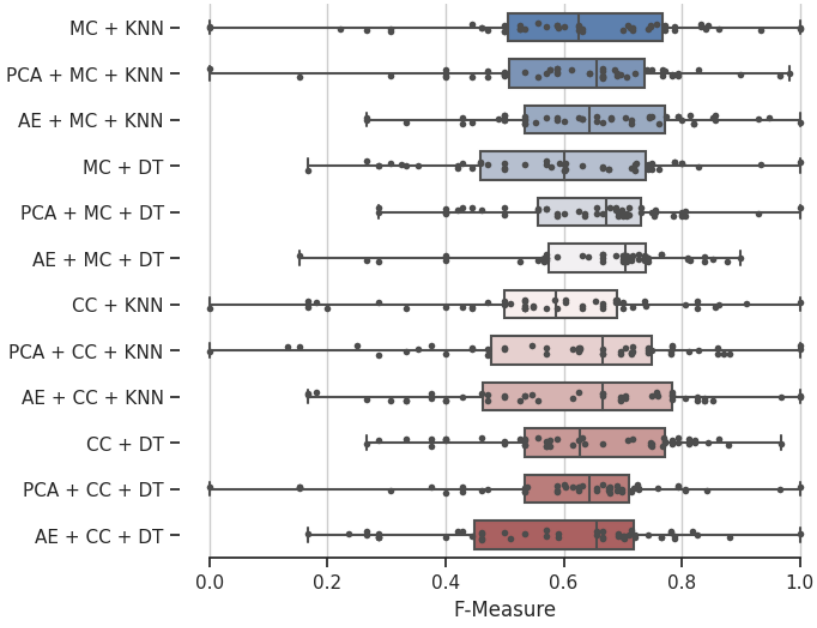
Figura 10 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *substructure* para treino e teste.

No segundo conjunto que apresenta 881 atributos podemos observar em destaque na Figura 11a a maior media no conjunto conjunto de treino utilizando a redução de dimensionalidade pela Rede Neural *Autoencoder* e os classificador *Classifier Chain* com métricas de *Decision Tree Classifier* com score de média 0,641 e desvio padrão de 0,168 e para seu respectivo teste uma média muito inferior de 0,582 com desvio 0,198. Um bom score para treino, mas muito baixo para teste e com desvio padrão alto, indicando baixa precisão do modelo. Entretanto, a aplicação do *Autoencoder* e utilização do algoritmo

Multioutput Classifier e *Decision Tree Classifier* na Figura 11b que mostraram melhor resultado nos conjuntos de teste com média de 0,664 e desvio padrão de 0,153 e treino respectivo de média 0,610 e desvio padrão 0,196. Porém com desvio elevado, mostrando pouca precisão.



(a) Treino

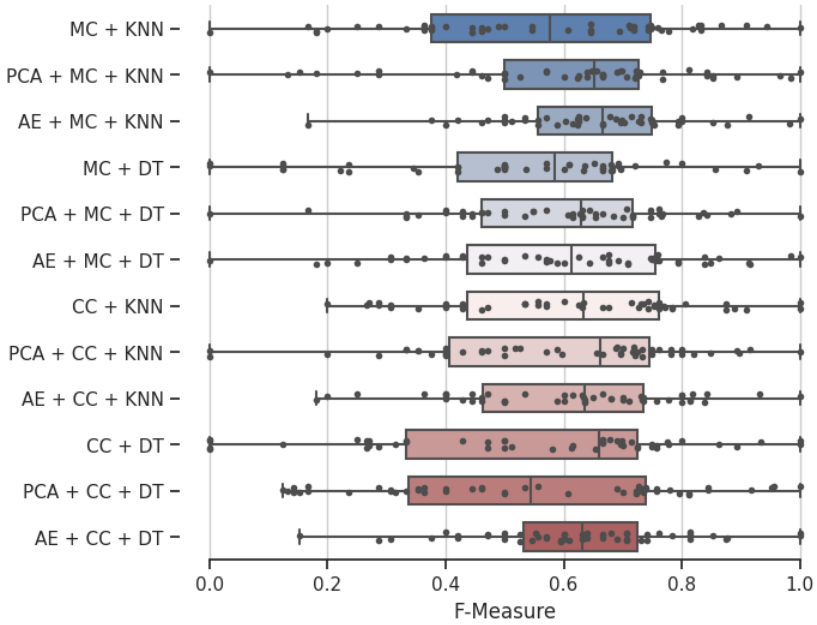


(b) Teste

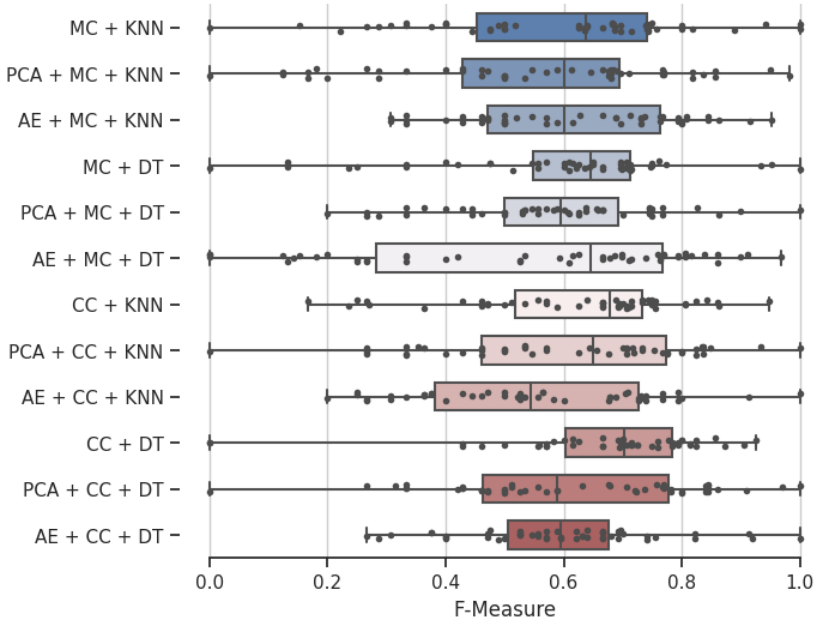
Figura 11 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *pubchem* para treino e teste.

O terceiro conjunto conta com 4860 atributos, podemos observar na Figura 12a o algoritmo que se destacou foi a utilização do *Autoencoder* utilizando classificador *Multioput Classifier* em conjunto com *K-Nearest Neighbors* com uma média no conjunto de

treino 0,654 e desvio padrão de 0,158. E para o conjunto de testes a Figura 12buma média de 0,610 e desvio padrão de 0,177, mostrando valores realmente próximos e de baixa variação. Sendo assim, a quantidade relativamente alta de atributos pode ser reduzida pela Rede Neural e condensar mais informações.



(a) Treino

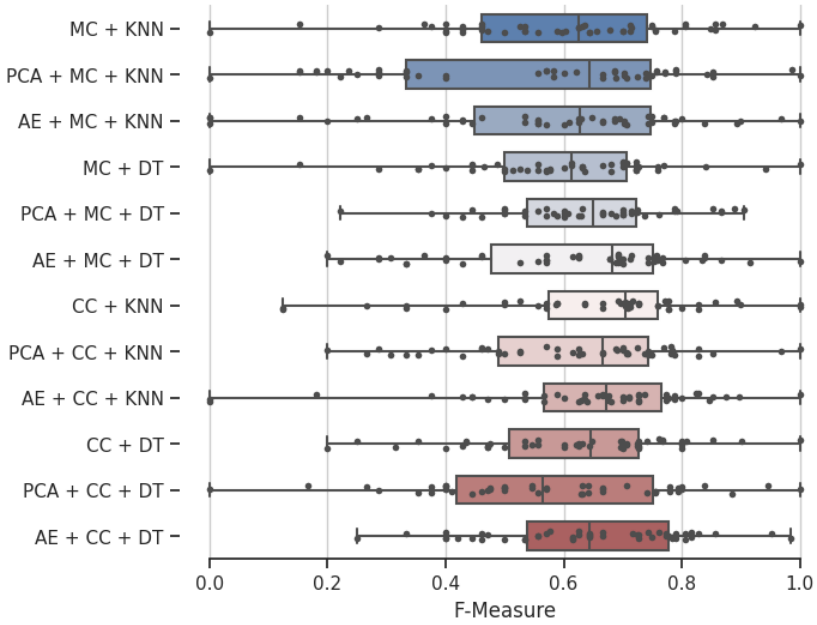


(b) Teste

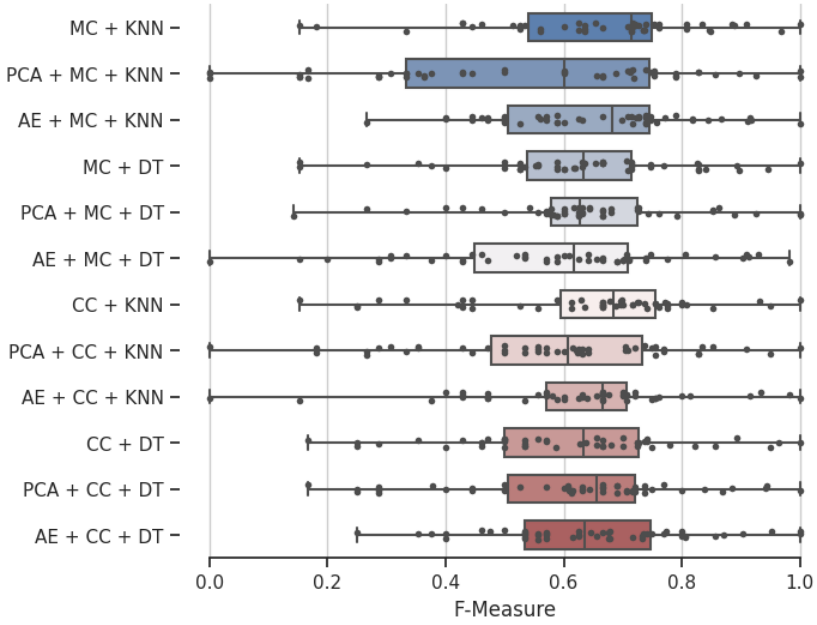
Figura 12 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *KR* para treino e teste.

Para o quarto conjunto com 780 atributos, podemos observar na Figura 13a que as combinações do modelo são muito semelhantes exibindo resultados próximos. O algoritmo com melhor média para *score* de *F-Measure* não utilizou técnica de redução de dimensi-

onalidade com os classificador *Classifier Chain* e *K-Nearest Neighbors*, mas podemos ver que há uma certa disparidade nos valores pois apresentam desvios padrão de 0,189 e 0,181 mesmo sem a redução de dimensionalidade. Porém ao se utilizar o *Autoencoder* com *Classifier Chain* e *Decision Tree Classifier* podemos ver que o modelo tem valores de precisão significativos e com desvios padrão reduzidos.



(a) Treino



(b) Teste

Figura 13 – Resultados dos algoritmos de classificação comparados por tipo de redução de dimensionalidade do conjunto de atributos *atompair* para treino e teste.

Para fins comparativos, os resultados obtidos podem ser vistos na Tabela 2, percebemos assim que o terceiro conjunto de atributos atingiu melhores resultados utilizando

a redução de dimensionalidade pela rede neural *Autoencoder* para seus 4861 atributos. Podemos perceber também que o primeiro conjunto de atributos por possuir uma quantidade menor de atributos, não se faz necessária a utilização das técnicas de redução. Tendo em vista as 38 observações dos conjuntos, ao comparar-se pela taxa de atributos o algoritmo *Autoencoder* é o mais indicado para reduções de dimensionalidade, pois conseguem capturar mais informações sobre dados não lineares.

Tabela 2 – Comparativo da medida F-measure entre os resultados dos quatro subconjuntos de atributos

Redução + Classificação	media treino	media teste	desvio treino	desvio teste
MC+DT	**0.642**	0.613	**0.121**	0.199
AE+CC+DT	**0.641**	0.582	**0.168**	0.198
AE+MC+KNN	**0.654**	0.610	**0.158**	0.177
AE+CC+DT	**0.647**	0.646	**0.162**	0.165

Podemos perceber também que a PCA não aparece nos conjuntos, isto pode ocorrer pelo fato dos conjuntos de dados apresentarem muitos valores nulos dos dados booleanos.

5 Considerações finais

O trabalho teve como objetivo avaliar a capacidade preditiva de modelos de classificação *multioutput* com dimensionalidade reduzida por PCA e *Autoencoder* para análises químicas , visando reduzir o tempo necessário para análise de dados laboratoriais.

A Rede Neural *Autoencoder* conseguiu proporcionar uma melhoria na classificação dos descritores e obteve-se coeficientes de F-Measure superiores aos testes com Análise de Componentes Principais principalmente em conjuntos de dados com maiores observações resultando em um *score* de 0,610 nos testes de descritores KR para *Autoencoder* e 0,553 para Análise de Componentes Principais.

Ao final deste trabalho, concluiu-se que métodos de redução de dimensionalidade e modelos de aprendizado de máquina podem proporcionar uma melhoria na qualidade dos modelos quando se há um grande número de atributos. No entanto, é necessário realizar estudos adicionais sobre as técnicas computacionais utilizadas, a fim de desenvolver e aprimorar algoritmos de predição de dados. O objetivo é identificar uma melhor forma de representar os dados, de forma a potencializar os modelos de classificação *multioutput*, que poderá prever com precisão quais compostos produzirão os melhores fármacos, levando em consideração suas características e interações bioquímicas.

Os algoritmos de redução de dimensionalidade também se mostram ferramentas excelentes para analisar grandes conjuntos de dados e transformá-los em conjuntos menores. Estudos recentes sobre essas técnicas são essenciais para a análise de dados e estão se tornando cada vez mais populares nessa área (GERTRUDES, 2013).

Portanto, este trabalho proporcionou ao estudante um melhor entendimento da teoria por trás da implementação de técnicas de redução de dimensionalidade, bem como sua aplicação em problemas de análise de dados, incluindo a implementação de algoritmos para criação de modelos de aprendizado de máquina e aprendizado profundo.

5.1 Trabalhos Futuros

Em trabalhos futuros, é possível realizar alterações nos hiperparâmetros dos algoritmos de redução de dimensionalidade, aprimorando as características da Rede Neural *Autoencoder* e dos classificadores *Decision Tree Classifier* e *K-Nearest Neighbors Classifier*, e avaliar as comparações entre os modelos. Além disso, outras técnicas mais robustas de redução de dimensionalidade, aprendizado de máquina e aprendizado profundo podem ser exploradas para criar modelos mais sólidos, levando em consideração a quantidade de atributos e um número maior de observações nos treinamentos dos modelos.

Referências

ALTMAN, Naomi; KRZYWINSKI, Martin. The curse (s) of dimensionality. *Nat Methods*, v. 15, n. 6, p. 399–400, 2018. Citado 1 vez na página 13.

BORCHANI, Hanen et al. A survey on multi-output regression. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, v. 5, n. 5, p. 216–233, 2015. DOI: 10.1002/widm.1157. Disponível em: https://doi.org/10.1002/widm.1157. Citado 1 vez na página 20.

CAMARGO, Sandro da Silva. Um modelo neural de aprimoramento progressivo para redução de dimensionalidade, 2010. Citado 2 vezes nas páginas 9, 13.

CASTRO, Lílian dos Santos et al. Estudo teórico da relaÇão estrutura atividade de derivados análogos da cafeína contra o câncer epitelial. Pontificia Universidade Católica de Goiás, 2009. Citado 1 vez na página 15.

CERRI, Ricardo. Introdução às Redes Neurais Artificiais com Implementações em R. In: SBC. ANAIS da I Escola Regional de Aprendizado de Máquina e Inteligência Artificial de São Paulo. 2020. P. 47–50. Citado 1 vez na página 10.

CHERMAN, Everton Alvares. *Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo*. 2013. Tese (Doutorado) – Universidade de São Paulo. Citado 1 vez na página 20.

FACELI, Katti et al. Inteligência artificial: uma abordagem de aprendizado de máquina, 2011. Citado 6 vezes nas páginas 16–18, 21–23.

FERNANDES, Philipe Oliveira; MALTAROLLO, Vinicius Gonçalves. Structure-Activity Relationship Studies of Staphylococcus aureus DNA Gyrase B Inhibitors as Antibacterial Agents Employing Random Forest Models. *International Journal of Quantitative Structure-Property Relationships (IJQSPR)*, IGI Global, v. 7, n. 1, p. 1–16, 2022. Citado 2 vezes na página 24.

FERREIRA, Márcia Miguel Castro; MONTANARI, Carlos Alberto; GAUDIO, Anderson Coser. Seleção de variáveis em QSAR. *Química Nova*, SciELO Brasil, v. 25, p. 439–448, 2002. Citado 1 vez na página 12.

GERTRUDES, J Castro et al. Machine learning techniques and drug design. *Current medicinal chemistry*, Bentham Science Publishers, v. 19, n. 25, p. 4289–4297, 2012. Citado 1 vez na página 9.

GERTRUDES, Jadson Castro. *Emprego de técnicas de análise exploratória de dados utilizados em Química Medicinal*. 2013. Tese (Doutorado) – Universidade de São Paulo. Citado 7 vezes nas páginas 9, 10, 12–15, 36.

HAYKIN, Simon. *Redes neurais: princípios e prática*. Bookman Editora, 2001. Citado 3 vezes nas páginas 16, 17, 19.

HONGYU, Kuang; SANDANIELO, Vera Lúcia Martins; OLIVEIRA JUNIOR, Gilmar Jorge de. Análise de componentes principais: resumo teórico, aplicação e interpretação. *ES Engineering and Science*, v. 5, n. 1, p. 83–90, 2016. Citado 3 vezes nas páginas 9, 16.

JOLLIFFE, I. T. *Principal Component Analysis*. Edição: Springer. Springer, 2002. v. XXIX, p. 487. Citado 1 vez na página 14.

MARTINS, Inês Filipa dos Santos. *Machine learning algorithms to predict blood-brain barrier permeability of drug molecules*. 2011. Tese (Doutorado). Citado 1 vez na página 22.

MARTINS, João Paulo A; FERREIRA, Márcia. QSAR modeling: um novo pacote computacional open source para gerar e validar modelos QSAR. *Química Nova*, SciELO Brasil, v. 36, p. 554–560, 2013. Citado 1 vez na página 13.

MAURI, Andrea; CONSONNI, Viviana; TODESCHINI, Roberto. Molecular descriptors. In: HANDBOOK of computational chemistry. Springer, 2017. P. 2065–2093. Citado 1 vez na página 25.

MORGAN, Harry L. The generation of a unique machine description for chemical structures - a technique developed at chemical abstracts service. *Journal of chemical documentation*, ACS Publications, v. 5, n. 2, p. 107–113, 1965. Citado 1 vez na página 25.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 1 vez na página 20.

SANTOS, Luiz Henrique A dos; ROSSI, Rafael G; SUL–MS–BRAZIL, Mato Grosso do. Aprendizado Multirrótulo para a Classificaçao Automática de Textos. Citado 1 vez na página 21.

SIQUEIRA, Rafael Fernandes et al. *Redução de dimensionalidade em bases de dados de classificação hierárquica multirrótulo usando autoencoders*. 2019. Diss. (Mestrado) – Universidade Tecnológica Federal do Paraná. Citado 6 vezes nas páginas 17–20, 23.

WANG, Xiaoyan et al. Consistent classification with generalized metrics. *arXiv preprint arXiv:1908.09057*, 2019. Citado 1 vez na página 10.

WORLD HEALTH ORGANIZATION. *Global Priority List of Antibiotic-Resistant Bacteria to Guide Research, Discovery, and Development of New Antibiotics*. 2017. Disponível em: https://www.who.int/news/item/27-02-2017-who-publishes-list-of-bacteria-for-which-new-antibiotics-are-urgently-needed. Citado 1 vez na página 24.

XIA, Xuhua. Bioinformatics and drug discovery. *Current topics in medicinal chemistry*, Bentham Science Publishers, v. 17, n. 15, p. 1709–1726, 2017. Citado 1 vez na página 9.

XUE, Wenjie et al. N-thiadiazole-4-hydroxy-2-quinolone-3-carboxamides bearing heteroaromatic rings as novel antibacterial agents: Design, synthesis, biological evaluation and target identification. *European Journal of Medicinal Chemistry*, Elsevier, v. 188, p. 112022, 2020. Citado 1 vez na página 24.

YOUNG, David C. *Computational drug design - A Guide for Computational and Medicinal Chemists*. Wiley, 2009. Citado 2 vezes nas páginas 12, 13.