Uma Análise Comparativa de Modelos para Solubilidade de Substâncias Químicas

1st Victor Santos
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
victor.eas19@gmail.com

2nd Guilherme Sales de Andrade
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
guilhermeslandrade@gmail.com

3nd Samuel Nicolau Alves Ribeiro da Silva
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
samuel.labjt@gmail.com

***Abstract*—This article aims to present, analyze, and compare different approaches for making predictions using linear regression models based on a dataset related to the solubility of chemical compounds. This document will outline the predictive methods applied, their differences, and observations made throughout the executed processes, such as predictive analysis, preprocessing, training, and testing. The methods applied include linear regression using the ordinary least squares (OLS) criterion, Ridge regression, Lasso regression, principal component regression (PCR), partial least squares regression (PLS), and a neural network for linear regression. The content of this document will also address potential differences in model validation when implementing the training methods manually compared to using the native methods provided by programming language libraries.**

***Index Terms*—Ordinary Least Squares (OLS),Ridge regression, Lasso regression, Principal Component Regression (PCR), Partial Least Squares (PLS), neural network, Training, Testing.**

I. Introdução

Assim como em diversas áreas do conhecimento, a estatística e a ciência de dados desempenham papéis fundamentais na análise e predição de fenômenos complexos. A modelagem por regressão linear, por exemplo, tem sido amplamente utilizada em diferentes campos para descrever e prever relações entre variáveis. Na química, a previsão de propriedades físico-químicas, como a solubilidade de compostos, é de grande interesse, visto que essa característica influencia diretamente no desenvolvimento de novos materiais e fármacos.

Entre os métodos disponíveis para executar previsões por meio de regressão linear, destacam-se os Mínimos Quadrados Ordinários (OLS), que minimizam a soma dos quadrados dos resíduos ($RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$), sendo um método base para ajustes precisos entre os valores observados (y_i) e previstos ($\hat{y}_i$). A Regressão Linear Ridge adiciona uma penalização à magnitude dos coeficientes através de uma regularização L^2 ($\lambda \sum_{j=1}^{p} \beta_j^2$), o que reduz problemas de multicolinearidade em dados de alta dimensionalidade. A Regressão Linear Lasso, por sua vez, aplica uma penalização ($\lambda \sum_{j=1}^{p} |\beta_j|$), promovendo a seleção automática de variáveis ao zerar coeficientes menos relevantes.

Além disso, a Regressão com Componentes Principais (PCR) combina análise estatística e redução de dimensionalidade, projetando os preditores em componentes principais e ajustando o modelo apenas nos componentes mais relevantes. Por outro lado, a Regressão por Mínimos Quadrados Parciais (PLS) vai além ao buscar combinações lineares de variáveis que maximizam a covariância entre os preditores e a variável resposta, proporcionando modelos que equilibram dimensionalidade e interpretabilidade.

Mais recentemente, a utilização de Redes Neurais como alternativa não-linear tem ganhado destaque. Esses modelos, através da minimização de uma função de perda como o erro quadrático médio ($MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$), conseguem capturar relações complexas nos dados, indo além das limitações impostas pelas suposições lineares dos métodos tradicionais. Cada um desses métodos apresenta características distintas que os tornam mais ou menos adequados para diferentes problemas, considerando fatores como multicolinearidade, dimensionalidade dos dados e os objetivos específicos do modelo.

Para além da escolha do método, a implementação também desempenha um papel crucial. Diferenças podem surgir entre o uso de bibliotecas pré-existentes, que oferecem métodos nativos otimizados, e a implementação manual dos algoritmos, que pode proporcionar maior controle sobre os detalhes do processo. Essa variação pode influenciar os resultados finais e até mesmo a interpretação das análises realizadas.

Sendo assim, o objetivo deste artigo é comparar e analisar o desempenho dos métodos citados em um estudo de caso sobre a previsão da solubilidade de compostos químicos, abordando tanto as características teóricas de cada abordagem quanto as implicações práticas de suas implementações. A intenção é fornecer insights sobre as condições em que cada método pode ser mais vantajoso, contribuindo para um entendimento mais profundo de suas aplicações e limitações.

II. Métodos

A. Análise exploratória e Pré-processamento

O conjunto de dados denominado data2 é composto por N = 1267 observações de compostos químicos. Para cada observação, existem D = 228 variáveis preditoras, sendo: 208 fingerprints binários (FP), que indicam a presença ou ausência de uma determinada subestrutura química, 16 descritores quantitativos, que indicam o número de ligações ou o número de átomos de bromo, e 4 descritores continuos, que indicam o

peso molecular e a área de superfície. As observações das variáveis preditoras foram divididas entre os conjuntos de treino (951 amostras) e teste (316 amostras).

Nesta etapa calculou-se a média e o desvio padrão e assimetria de cada preditor não binário, tanto do conjunto de dados não transformado como do conjunto transformado. Após o cálculo do resumo estatístico também verificou-se qual seria a média da assimetria de todos os preditos, para o conjunto não transformado e para o conjunto transformado. Esse processo foi aplicado para verificar os efeitos esperados da transformação dos dados uma vez que normalmente esses processos são utilizados para centralizar os dados, escalonar os dados e reduzir assimetrias. Portanto a média da assimetria dos preditores não transformados foi de 1.649. A média do desvio padrão dos preditores não transformados foi de 12.341. A média da assimetria dos preditores não transformados foi de 0.248. E por fim, a média do desvio padrão dos preditores não transformados foi de 1.052. Verificando os resultados obtidos é possível verificar que a média das médias de cada preditor, após a transformação ficou em torno de zero e o desvio padrão médio, muito próximo de um. É possível observar estes aspectos da transformação comparando alguns dos histogramas de antes e depois da transformação dos dados.

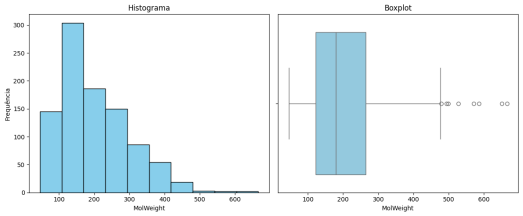

Fig. 1. Histograma e Boxplot para MolWeight não transformado

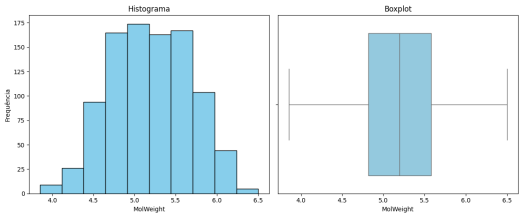

Fig. 2. Histograma e Boxplot para MolWeight transformado

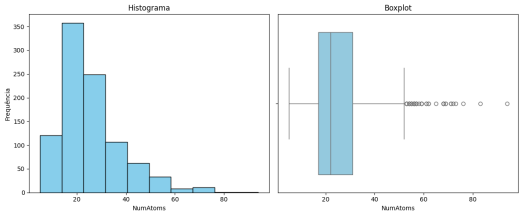

Fig. 3. Histograma e Boxplot para NumAtoms não transformado

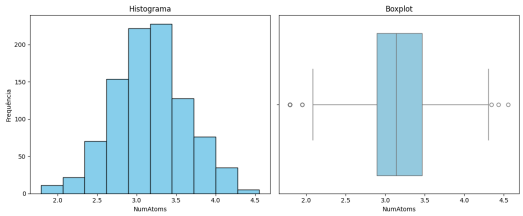

Fig. 4. Histograma e Boxplot para NumAtoms transformado

Ainda nesta etapa foi realizado um cálculo da correlação de cada preditor não binário com cada outro preditor não binário assim como foi calculada a correlação de cada preditor não binário com a variável alvo, que neste caso é a solubilidade do composto. Ao calcular a matriz de correlação obteve-se o seguinte heatmap:

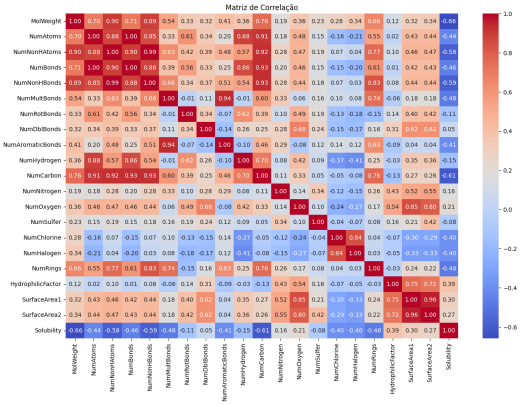

Fig. 5. Heatmap da matriz de correlação para todos os preditos não binários.

Através desse heatmap foi possível verificar que existem alguns preditores com alta correlação o que pode acabar afetando negativamente o modelo. Os preditores com muita correlação entre si e baixa correlação com a variável alvo são os principais candidatos a não serem utilizados para o treino e teste do modelo posteriormente. É possível escolher alguns preditores que têm alta correlação entre si para retirar do dataset e manter outros que capturam características relevantes e que tem alta correlação com a variável alvo.

Preditores selecionados: MolWeight, que, apesar da alta correlação com outras variáveis (NumNonHAtoms e NumNonHBonds), a forte correlação com o alvo (-0.643) justifica sua inclusão; NumCarbon possui a segunda maior correlação negativa com o alvo (-0.599) e captura informações distintas em relação ao número de átomos de carbono; NumNonHBonds conta com correlação de -0.575, é mais relevante que NumBonds e NumAtoms e representa ligações específicas; HydrophilicFactor apresenta uma correlação positiva (0.379) com o alvo sugere que é um preditor interessante e não redundante com outros; SurfaceArea1 mantém-se por capturar informações geométricas e ter uma correlação moderada (0.295) com o alvo.

Ao incluir na análise de correlação os preditores binários é possível identificar alguns fingerprints que apresentam uma

boa correlação com a variável alvo e portanto seriam relevantes ao modelo a ser desenvolvido. Após a análise da correlação destes outros preditores foi então feita uma nova seleção de preditores de interesse para o modelo: MolWeight (-0.643): Melhor relação com o alvo; NumCarbon (-0.599): Informação complementar ao MolWeight; NumNonHBonds (-0.575): Alta relevância com menor redundância; FP076 (-0.533): Melhor binário com forte impacto no alvo; FP089 (-0.499): Segundo melhor binário; NumMultBonds (-0.478): Informação estrutural relevante; HydrophilicFactor (0.379): Informação química diferenciada; SurfaceArea1 (0.295): Captura dimensões geométricas; FP044 (-0.459): Outro binário com impacto moderado. FP092 (-0.393): Complementar entre os binários.

TABLE I
RESUMO ESTATÍSTICO - PREDITORES NÃO BINÁRIOS

Nome	**Média**	**Desvio Padrão**	**Média (transformado)**	**Desvio Padrão (transformado)**
MolWeight	201.6	97.9	5.1	0.4
NumAtoms	25.5	12.6	3.1	0.4
NumNonHAtoms	13.1	6.4	2.5	0.4
NumBonds	25.9	13.4	3.1	0.4
NumNonHBonds	13.5	7.5	3.3	0.8
NumMultBonds	6.1	5.1	2.5	1.7
NumRotBonds	2.2	2.4	0.9	0.7
NumDblBonds	1.0	1.2	0.3	0.3
NumAromaticBonds	5.1	5.2	1.2	1.1
NumHydrogen	12.3	7.3	3.6	1.1
NumCarbon	9.8	5.2	3.3	0.9
NumNitrogen	0.8	1.1	0.2	0.2
NumOxygen	1.5	1.7	0.7	0.6
NumSulfer	0.1	0.4	0.04	0.1
NumChlorine	0.5	1.4	0.09	0.1
NumHalogen	0.6	1.4	0.1	0.1
NumRings	1.4	1.2	0.7	0.5
HydrophilicFactor	-0.0	1.1	-0.4	1.0
SurfaceArea1	36.4	35.2	6.7	4.5
SurfaceArea2	1.4	1.2	7.0	4.6

B. Regressão Linear Simples - Mínimos Quadrados (OLS)

Nesta etapa, após a seleção dos preditores mais relevantes para o modelo, foi realizado o treino de um modelo de regressão linear simples utilizando o critério dos mínimos quadrados para estimar os coeficientes que minimizam o erro quadrático do modelo. Foi feita uma abordagem utilizando a função nativa do Python e uma implementação manual que aplica a expressão do OLS. Após a implementação do treino utilizando as duas abordagens descritas foi possível verificar a acurácia do modelo a partir dos coeficientes estimados.

Ao aplicar a biblioteca statsmodel.api para treinar o modelo com os preditores selecionados: MolWeight, NumCarbon, FP076, FP089, NumMultBonds, HydrophilicFactor, SurfaceArea1, FP044 e FP092. Obteve-se os seguintes valores para os coeficientes de cada preditor incluindo o coeficiente do intercepto:

TABLE II
COEFICIENTES DO MODELO

Variável	**Coeficiente**
Intercepto	8.746952
MolWeight	-1.864008
NumCarbon	-0.329860
NumNonHBonds	-0.662208
FP076	0.433472
FP089	-0.365588
NumMultBonds	-0.031436
HydrophilicFactor	0.004388
SurfaceArea1	0.250729
FP044	-1.070942
FP092	-0.155747

Cada coeficiente reflete a influência de seu respectivo preditor na solubilidade do composto químico. Neste caso, o peso molecular (MolWeight) tem uma relação negativa (-1.864008), indicando que, conforme aumenta o peso molecular, a solubilidade tende a diminuir. Já o fator hidrofílico (HydrophilicFactor) tem um coeficiente positivo (0.004388), sugerindo que compostos mais hidrofílicos podem ter solubilidade ligeiramente maior. Para essa primeira abordagem o erro quadrático ou acurácia obtida foi de 0.788 e o RMS foi de 0.941. O modelo treinado apresentou um desempenho razoável com os preditores selecionados, explicando grande parte da variação da solubilidade ($R^2 = 0.788$).

Após isso foi necessário implementar uma versão explícita da estimação dos coeficientes que permitiu comparar tanto os coeficientes estimados quanto o desempenho do modelo.

TABLE III
COEFICIENTES DO MODELO

Variável	**Coeficiente**
Intercepto	8.746952
MolWeight	-1.86400758
NumCarbon	-0.329859527
NumNonHBonds	-0.662207730
FP076	0.433471915
FP089	-0.365588123
NumMultBonds	-0.0314363271
HydrophilicFactor	0.00438799799
SurfaceArea1	0.250729189
FP044	-1.07094165
FP092	-1.55746724

Para essa segunda abordagem o erro quadrático ou acurácia obtida foi de 0.788 e o RMS foi de 0.941, ambos idênticos aos valores obtidos através da abordagem utilizando diretamente a biblioteca statsmodel.api. Conclui-se que para uma regressão linear simples a aplicação da função fornecida pela biblioteca statsmodel.api do Python é tão eficaz quanto a implementação manual do modelo de treino. Ainda nesta etapa foi necessário aplicar um método de validação do modelo chamado de k-fold-cross-validation. Consiste em dividir o conjunto de dados de treino em 5, se k = 5, ou 10, se k = 10, partes. O processo de cross-validation consiste em utilizar k-1 conjuntos de dados para treinar o modelo e utilizar o conjunto (fold) restante para teste. Isso deve ser feito k vezes de forma que todos os folds

tenham sido utilizados para validar o modelo. Em cada rodada de validação cruzada é realizada a verificação da acurácia (R^2) e do RMS e no fim uma média destes dois parâmetros é obtida e definida como o R^2 e RMS do modelo. Foi feita a implementação do k-fold-cross-validation para k = 5 e os seguintes resultados foram obtidos:

TABLE IV
RESULTADOS DO K-FOLD CROSS-VALIDATION (K=5)

Fold	R^2	RMS
Fold 1/5	0.7862	0.9352
Fold 2/5	0.7740	0.9634
Fold 3/5	0.7913	0.9295
Fold 4/5	0.7961	0.9297
Fold 5/5	0.7963	0.9403
Média	**0.7888**	**0.9396**

Após o cálculo do R^2 e RMS para cada rodada da k-fold-cross-validation obteve-se as médias do R^2 e RMS: R^2 médio: 0.7888, RMS médio: 0.9396. Ambos muito próximos dos valores obtidos na validação convencional. Isso nos permite inferir que esse conjunto de dados selecionado para treino não possuí regiões em cujos valores são particularmente muito diferentes das amostras do conjunto de treino como um todo.

C. Regressão Linear Penalizada - L1 e L2

Modelos de regressão linear penalizados são variações da regressão linear que inclui termos de penalização no processo de ajuste dos coeficientes. A penalização é usada para reduzir o overfitting e melhorar a generalização do modelo, especialmente em situações com muitas variáveis explicativas ou quando existe multicolinearidade.

Existem vários modelos de penalização, mas os mais famosos deles são o Ridge e o Lasso. O Ridge penaliza a regressão linear com uma penalização ao da somatória dos coeficientes da regressão ao quadrado (ver figura 6). Nesse método os coeficientes são reduzidos mas não chegam a zero. No Lasso a penalização é a somatória da norma dos coeficientes. Nesse método os coeficientes podem ser zeros, logo, ele exclui da regressão preditores que não agregam muito a saída da regressão.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad (1)$$

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad (2)$$

Fig. 6. Função de penalização do Ridge

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j \quad (3)$$

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad (4)$$

Fig. 7. Função de penalização do Ridge

Como método de penalização utilizamos o Ridge, e para obtermos os coeficientes precisamos minimizar a função de custo que é função da figura 6. Para minimizar precisamos encontrar onde a derivada da função é zero. Logo a resolução para encontrar a função que vai minimizar os coeficientes é esta:

$$RSS_{ridge} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad (1)$$

$$RSS_{ridge} = ||Y - X\beta||^2 + \lambda||\beta||^2 \quad (2)$$

$$\nabla RSS_{ridge} = -2X^T(Y - X\beta) + 2\lambda I\beta \quad (3)$$

$$-2(X^T(Y - X\beta) - \lambda I\beta) = 0 \quad (4)$$

$$X^T(Y - X\beta) - \lambda I\beta = 0 \quad (5)$$

$$X^TY - X^TX\beta + \lambda I\beta = 0 \quad (6)$$

$$(X^TX + \lambda I)\beta = X^TY \quad (7)$$

$$\boldsymbol{\beta} = (X^TX + \lambda I)^{-1}X^T\mathbf{y} \quad (8)$$

Fig. 8. Equação para encontrar os coeficientes do modelo Ridge

Com essa fórmula aplicamos todas as colunas do nosso conjunto de dados de treino e teste para modelar nossa regressão linear Ridge para acharmos os coeficientes. A primeira parte do procedimento foi para encontrar qual é o λ que minimiza o RMSE por meio de um cross-validation com k = 5, o resultado pode ser visto na figura 10.

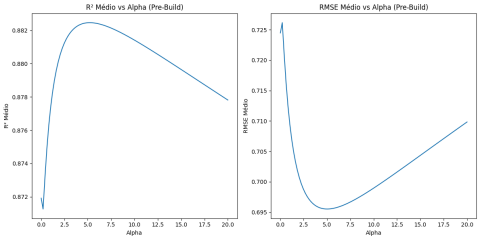

Fig. 9. Resultado da média do cross-validation para nossa classe Ridge. (Pre-Build)

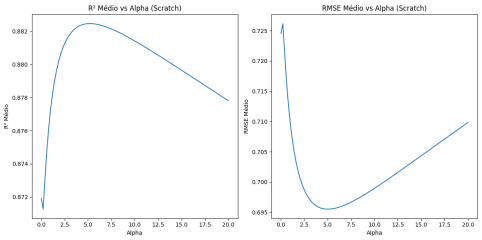

Fig. 10. Resultado da média do cross-validation para nossa classe Ridge. (Scratch)

Como podemos ver pelos gráficos, começando com um λ em zero obtemos um R^2 muito baixo e um RMSE muito alto, e logicamente isso deveria acontecer, visto que aplicar um λ zero significa não fazer nenhuma penalidade, logo, é como se estivéssemos fazendo uma regressão linear simples. A medida que vamos aumentando o λ, vamos aumentando o R^2 e o RMSE, até encontrar um valor ótimo, onde o R^2 é maximizado e o RMSE é minimizado, após esses valores o R^2 volta a diminuir e o RMSE volta a aumentar, mostrando que não é viável utilizar valores de λ maiores. O valor ótimo de λ encontrado foi aproximadamente 4. Utilizando o λ para realizar os ajustes e previsões no modelo foi obtido esses resultados:

TABLE V
RESULTADOS DO MODELO RIDGE COM $\lambda \approx 4$

Modelo	Previsão com Conjunto de Teste	Cross-validation scratch (k=5)	Cross-validation sklearn (k=5)
Nosso Ridge	RMSE: 0.7305, R²: 0.8761	RMSE: 0.7483, R²: 0.4082	RMSE: 0.7575, R²: 0.4061
Ridge do sklearn	RMSE: 0.7120, R²: 0.8823	RMSE: 0.7346, R²: 0.4061	RMSE: 0.7421, R²: 0.4038

Podemos notar pelos resultados obtidos pelo nosso modelo Ridge e pelo modelo da biblioteca do scikit-learn que eles estão bem parecidos e obtiveram bons resultados com a previsão do conjunto de teste, contudo quando aplicamos o cross-validation obtivemos RMSE muito parecidos com o RMSE ao aplicar o conjunto de teste, embora 0.74 ainda é considerado um RMSE um pouco alto, mas o R² foi muito abaixo em relação a previsão com o conjunto de teste. Esses resultados nos mostraram que embora nosso modelo esteja conseguindo prever valores próximos do que o esperado, ele não está se adaptando muito bem para dados que ele não viu antes, ou seja, ele ainda tem uma variância um pouco elevada, logo, pode ser necessário um outro método de regressão linear caso queríamos encontrar um modelo mais assertivo para nossa problema de solubilidade.

D. Mínimos Quadrados Parciais (PLS) e Regressão por Componentes Principais (PCR)

Nesta etapa foi necessário utilizar o mesmo conjunto de dados já transformados que foram utilizados nas etapas anteriores. Aqui comparou-se a performance destes dois modelos que compartilham muitas semelhanças entre si, no entanto possuem diferenças conceituais que resultam em diferenças práticas em seus resultados.

Tanto o PLS quanto o PCR são modelos que aplicam os mínimos quadrados ordinários (OLS), porém esta regressão é aplicada sobre as componentes principais adquiridas a partir do conjunto de dados. Esta é a semelhança entre estes dois modelos. As diferenças se dão no método utilizado para determinar essas componentes principais. Enquanto no PCR o meio para encontrar tais componentes é através da maximização de variância entre os preditores (ou seja: no PCA se escolhem os preditores de tal forma que eles tenham mínima correlação entre si). Já no PLS, o meio e os critérios para definir as componentes principais se dá através da maximização da correlação entre os preditores e a variável alvo.

Foi necessário treinar e testar, ambos, para diferentes quantidades de componentes principais a fim de escolher a quantidade que melhor se saísse em termos de acurácia (R²) e RMS. Para adquirir tais parâmetros de comparação foi necessário aplicar uma k-cross-validation para k = 5 em cada quantidade de componentes principais escolhidos.

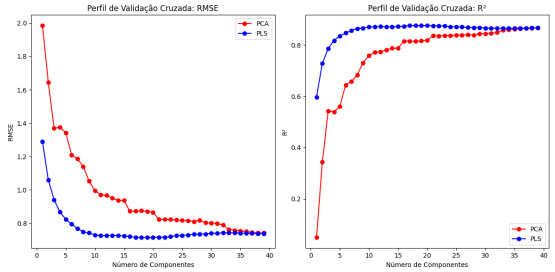

Fig. 11. Comparação entre a performance dos modelos PCR e PLS em termos RMS e R²

Os resultados obtidos vistos nas figura 11 revelam que para uma quantidade significativamente menor de componentes o modelo do PLS se ajusta mais do que em comparação ao PCR. Para atingir performance máxima (melhor R² e melhor RMS) com PCR foi necessário se utilizar de 39 componentes principais, enquanto para atingir a melhor R² com o PLS foi necessário utilizar 18 componentes principais e para adquirir melhor RMS foram necessários 19 componentes principais.

TABLE VI
RESULTADOS OBTIDOS PARA PLS E PCR

Métrica	PLS	PCR
Número de Componentes Principais	18	39
RMS	0.7296	0.8048
R²	0.8764	0.8496
RMS (Cross-Validation)	0.7149	0.7415
R² (Cross-Validation)	0.8761	0.8669

Apesar de parecer que a diferença em termos de acurácia e erro quadrático médio serem relativamente pequenas, existe um diferença implícita e significativa em termos de custo

computacional, uma vez que que é menos custoso computacionalmente treinar e testar um modelo que se utiliza de 19 componentes principais do que um modelo que necessita de 39.

A partir desses resultados é possível notar que, no geral, PLS costuma performar melhor que PCR em cenários onde a relação entre as variáveis preditoras (X) e a variável resposta (Y) é forte e direta. Isso ocorre porque o PLS constrói os componentes latentes priorizando a correlação com Y, tornando-o mais adequado para prever a variável alvo.

Por outro lado, o PCR pode ser mais robusto em situações onde o foco está apenas na redução da dimensionalidade de X, sem uma forte relação com Y, já que ele não considera a variável resposta na construção dos componentes.

E. Regressão com Rede Neural

Modelos computacionais inspirados no cérebro humano, com o objetivo de identificar padrões complexos de dados, esses chamados de Redes Neurais, consistem de nós, organizados de uma forma hierárquica, cada nó com um peso que é ajustado para diminuir a diferença dos valores reais e os previstos pelo modelo. Consistindo de três camadas: entrada, ocultas, saída, onde respectivamente os dados são recebidos, passam por um processamento utilizando funções matemáticas, como o ReLU (Rectified Linear Unit) ou Sigmoid, aonde são introduzidos não linearidades no modelo para que a rede capture informações complexas dos preditores e por fim através da saída forneça a previsão ou classificação final. No contexto atual, foi utilizado uma rede neural para modelar relações entre os preditores e as variáveis resposta. A rede foi configurada com uma camada de entrada correspondente ao número de preditores, duas camadas ocultas de 258 e 128 neurônios respectivamente, na camada de saída um único neurônio para prever os valores contínuos. A função de ativação ReLU foi empregada para encontrar complexidades nos dados, já a função de perda foi o MSE (Mean squared error). O treinamento foi realizado por 100 épocas com o otimizador Adam, que ajusta dinamicamente a taxa de aprendizado para acelerar a convergência. Após o treinamento, o modelo foi testado em um conjunto de teste independente, a rede apresentou um desempenho melhor que os modelos lineares com um RMSE de 0.6896 e um R2 de 0.8896, o que indica maior precisão e capacidade de explicar a variabilidade de dados, isso pode ser explicado pelo fato que os preditores possuem relações não-lineares complexas com a variável resposta, que não foram capturadas eficientemente pelos modelos lineares.

TABLE VII
COMPARAÇÃO DE DESEMPENHO DOS MODELOS

Modelo	RMSE	R^2
OLS	0.7456	0.8709
Ridge	0.7300	0.8762
PCA	0.9137	0.8061
PLS	0.7296	0.8764
Rede Neural	0.6896	0.8896

III. RESULTADOS

Os modelos de regressão analisados, incluindo regressão linear, Ridge, Lasso, PCR, PLS e redes neurais, mostraram bom desempenho na previsão da solubilidade dos compostos químicos. O modelo linear simples obteve um R^2 de 0.788 e RMSE de 0.941, enquanto as versões com penalização (Ridge e Lasso) ajudaram a reduzir o overfitting sem grandes melhorias no R^2. A análise indicou que variáveis como peso molecular e número de átomos de carbono foram as mais influentes.

Embora a rede neural tenha sido mais complexa, seu desempenho não superou significativamente os modelos lineares, sugerindo que, neste caso, abordagens mais simples são eficazes. A regressão Ridge mostrou-se útil para mitigar problemas de multicolinearidade, enquanto o PLS superou o PCR ao exigir um menor número de componentes principais para atingir bons resultados. Isso reforça que modelos lineares bem ajustados são uma solução eficiente e interpretável para esse problema específico.

Portanto, a escolha do modelo ideal depende do equilíbrio entre desempenho, interpretabilidade e custo computacional. Em aplicações práticas, abordagens como Ridge e PLS podem ser preferidas por oferecerem um bom compromisso entre precisão e simplicidade, enquanto redes neurais podem ser consideradas quando há forte suspeita de relações não lineares mais complexas.

REFERÊNCIAS

[1] G. Andrade, S. Silva e V. Santos, "homework2," *Colab Notebook*, disponível em: https://colab.research.google.com/drive/Homework2. [Acessado em: Jan. 25, 2025].

[2] S. Silva, "homework2," *Colab Notebook*, disponível em: https://colab.research.google.com/drive/WH2SAM. [Acessado em: Jan. 26, 2025].

[3] J. Gregório, "Regularização, Lasso (L1) e Ridge (L2)," *Medium*, Setembro, 2022. [Online]. Disponível em: https://medium.com/@jackelinegleme/regularização-lasso-l1-e-ridge-l2. [Acessado em: Jan. 21, 2025].

[4] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Chapter 6, Springer, 2013.