Métodos para Predição da Concentração de Álcool em Vinhos Brancos

1st Gabriel Pinheiro Palitot Pereira
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
gabrielppalitot@alu.ufc.br

2nd Igor Ferreira Vasconcelos
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
igorfervasc@alu.ufc.br

3rd Eliabe Bastos Dias
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
eliabebastosdias@alu.ufc.br

***Resumo*—Modelos preditivos são técnicas que aprendem certos padrões com base em dados anteriores e está em constante uso nos dias atuais. Dessa forma, este artigo visa entender o funcionamento de diversos modelos de predição, sendo utilizados para predir a concentração de álcool em vinhos portugueses. Foi utilizado conhecimentos de pré-processamento dos dados e análise exploratória. Foi usado desde métodos lineares simples, como o método dos mínimos quadrados ordinários (OLS), passando por métodos penalizados, como a regressão de Ridge indo até o uso do PCR e redes neurais simples. Além disso, os resultados foram todos analisados com métricas como RMSE, R^2 e auxílio de gráficos para ser possível a comparação dos modelos.**

***Index Terms*—Regressão, OLS, Ridge, PCR , Vinho, Predição, Rede Neural, álcool.**

I. Introdução

Primeiramente, este artigo é uma continuação direta do que fora feito em [1], no entanto, qualidade é uma variável qualitativa e não pode ser prevista por métodos de regressão, partiremos para aquela que tem maior correlação com ela: a concentração de álcool.

Como o álcool é derivado de um processo de fermentação, uma reação química, é esperado que somente os componentes químicos não trarão uma relação linear. Então além da variável de saída ser o álcool, estamos colocando a qualidade como um preditor, já que caso contrário, a linearidade seria muito fraca.

Dito isso, a maioria dos métodos de regressão funcionam melhor quando utilizamos dados padronizados e com distribuição gaussiana (média 0, desvio padrão 1 e o mais simétrico possível), assim, precisamos transformá-los, etapa conhecida como pré-processamento.

Para isso, utilizamos o método de Centralização e Dimensionamento, que de acordo com [2] é o método mais comum e direto, que consiste em subtrair a média de cada preditor por cada um de seus valores individuais, em seguida dividir todos pelo desvio padrão, como mostra a equação 1, em que x é o valor original, x' o valor transformado, m e sd correspondem às médias e ao desvio padrão do grupo de preditores de x.

$$x' = \frac{(x - m)}{sd} \tag{1}$$

Isso faz com que todos os valores tenham média 0 e desvio padrão 1.

Para melhorar a simetria, utilizamos o método de Yeo-Johnson. Que, de acordo com [6], é da família das transformações de potência, semelhante ao box-cox, mas que funciona com valores negativos.

Ela é definida matematicamente pela equação 2.

$$y(x, \lambda) = \begin{cases} \frac{((x+1)^{\lambda}-1)}{\lambda}, & \text{se } \lambda \neq 0 \text{ e } x \geq 0, \\ -\ln(x+1), & \text{se } \lambda = 0 \text{ e } x \geq 0, \\ -\frac{((1-x)^{2-\lambda}-1)}{(2-\lambda)}, & \text{se } \lambda \neq 2 \text{ e } x < 0, \\ \ln(1-x), & \text{se } \lambda = 2 \text{ e } x < 0. \end{cases} \tag{2}$$

O lambda é definido por meio da maximização da estimativa de máxima verossimilhança. Em casos em que x é zero, será transformado para zero independentemente do λ.

Com o conjunto de dados dessa maneira, os métodos propostos nas seções seguintes serão mais eficazes, trazendo um erro mínimo desejado referente aos valores reais, comparando com o conjunto de dados bruto.

Ademais, serão utilizados métodos de regressão linear, como a Regressão OLS, regressão penalizada de Ridge e PCR (Principal Components Regression) que se utiliza de um método usado em [1] que é o PCA, muito importante para a redução de dimensionalidade. Por fim, será testada uma rede neural para ver como os dados pré-processados se saem em um modelo não linear.

II. MÉTODOS

A. Análise Exploratória

Nessa etapa, a nossa visão sobre [1] foi expandida para o preditor álcool. E é notório que ele tem uma relação com os demais preditores muito semelhante com a qualidade, então a grande maioria das informações serão reaproveitadas.

B. Regressão Linear (OLS)

A regressão linear é um dos métodos mais simples de aprendizado supervisionado, sendo muito útil para a predição de variáveis quantitativas [2]. Há essencialmente dois tipos de regressão linear, a simples e a múltipla. A primeira é utilizada quando tem-se apenas de uma variável preditora, mas, geralmente, temos mais de um preditor, portanto a segunda vai ser mais útil. Ela segue o mesmo formato da primeira, mas todos os preditores possuem seus coeficientes de regressão (β) [3]. Abaixo, imaginando que temos p preditores, podemos formular a seguinte forma para a regressão linear múltipla:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{3}$$

Onde X_i é a matriz de amostras do i-ésimo preditor e β_i é o i-ésimo coeficiente de regressão associado a ele. $\hat{Y}$ é a saída estimada. Já ϵ é o resíduo devido a aleatoriedade, sendo assim, um erro irredutível

Como essa saída é estimada, então vamos ter a saída real que vai servir para descobrir o erro do nosso modelo. O cálculo desse erro é dado por [2]:

$$SSE = e_1^2 + e_2^2 + \cdots + e_n^2 = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{4}$$

Percebe-se então que minimizar o erro que obtém-se ao diminuir a saída real pela estimada do nosso modelo é a melhor forma para melhorar a precisão. É nisso que se baseia o método de regressão Linear OLS (*Ordinary least Squares*) em que se busca diminuir a soma dos erros quadráticos (SSE). É importante destacar que a regressão linear assume que há uma relação linear entre a resposta e os preditores [3].

Para isso, há formas de estimar esses coeficientes de regressão, como a vista abaixo que se obtém ao tentar minimizar a soma dos erros quadráticos (Utilizando a Fórmula acima, derivando parcialmente em relação aos coeficientes de regressão e igualando a zero):

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{5}$$

A equação 5 está escrita na forma matricial. Sendo X a matriz de preditores, Y é o vetor de resposta. É importante notar que a matriz $(X^T X)^{-1}$ deve ser invertível.

Há formas de avaliar o desempenho do modelo, como o RMSE (*root mean squared error*) e o coeficiente de determinação (chamado de R^2). Geralmente quanto menor o RMSE, melhor a capacidade de predição do modelo, que é interpretado como a distância média entre os valores observados e as previsões do modelo. Já para o R^2 indica a proporção da variabilidade que é explicada pelo modelo e que varia de 0 a 1, quanto mais próximo de 1 melhor. De acordo com [2] há várias formas de calcular o R^2, sendo a mais comum calcular o coeficiente de correlação entre os valores observados e os da predição e elevar ao quadrado. Já o RMSE tem a fórmula abaixo (sendo $\hat{y}_i$ o valor predido, n o número de amostras e y_i os valores reais):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{6}$$

Para formular um modelo de regressão linear é interessante separar uma parte dos dados apenas para treino e outra parte para teste. Uma estratégia comum e simples é utilizar 75% para treinamento (estimar os betas) e os outros 25% para os testes (verificar o desempenho do modelo). Um porém dessa estratégia é que podem haver problemas de acordo com a escolha da divisão dos dados, além da possibilidade de overfitting, já que o modelo pode se ajustar demais ao treinamento e não generalizar bem.

Uma alternativa para a questão citada é a K-fold Cross Validation (Validação cruzada). A sua utilização é simples e consiste em separar os dados em K grupos iguais [3]. O primeiro fold ou grupo é utilizado como conjunto de teste e os outros $k-1$ são usados como conjunto de treinamento, na próxima iteração o segundo fold é utilizado como teste e os $k-1$ são usados como conjunto de treinamento. Após k iterações são obtidas k métricas de desempenho (geralmente RMSE) em que fazemos a média dessas métricas. Que é o resultado do nosso modelo.

Agora, com a explicação dos conceitos chave, pode-se prosseguir para aplicação da predição. Foi decidido nesse trabalho realizar a predição do álcool do dataset utilizado em [1] e que pode ser encontrado em [4]. Foi primeiramente realizada uma regressão linear múltipla com o método dos mínimos quadrados. Foram testadas as capacidades das funções das bibliotecas prontas do Python e também uma implementação feita pelos autores. Além disso, 75% dos dados foram divididos (de forma pseudo-aleatória) para o conjunto de treino e 25% para o teste para essa tentativa sem validação cruzada. Na Tabela I, pode-se ver o resultado do modelo de Regressão Linear (OLS).

Tabela I
COMPARAÇÃO DE METRICAS OLS SEM VALIDAÇÃO CRUZADA

Métricas	Lib	Autoral
RMSE	0.542	0.542
R²	0.805	0.805

Nota-se que os resultados tanto para as funções implementadas na biblioteca do Python quando as feitas pelos autores são idênticas. Os valores obtidos do RMSE significam que, na média, o modelo está cometendo um erro de 0.542 em relação aos valores reais. Para o R^2, o valor de 0.805 significa que o modelo consegue explicar 80.5% de variância de Y (álcool). O próximo passo é ver como o modelo se sai ao aplicar validação cruzada, que ,novamente, foi implementada pelos autores e comparada com uma função nativa da linguagem de

programação Python. Foi utilizado 5-fold e 10-fold. Pode-se ver os resultados na Tabela II.

Tabela II
COMPARAÇÃO DE METRICAS OLS COM VALIDAÇÃO CRUZADA

Métricas	Lib	Autoral
RMSE(5-FOLD)	0.442	0.442
R²(5-FOLD)	0.852	0.852
RMSE(10-FOLD)	0.429	0.429
R²(10-FOLD)	0.857	0.857

Vemos que com a validação cruzada o RMSE e o R^2 foram melhores (RMSE diminuiu e o R^2 aumentou), isso pode ser porque o modelo está demonstrando um desempenho consistente e robusto. Conseguindo generalizar os dados para diferentes conjuntos de testes diferentes. Mais resultados serão discutidos na seção: *Discussão dos Resultados*.

C. Regressão Linear Penalizada

Há outros métodos de regressão linear que podem melhorar o modelo que já foi feito, os métodos são a Regressão de Ridge e Regressão de Lasso. Em [3] é possível ver que esses métodos podem ser usados para penalizar coeficientes de regressão de preditores ao invés de selecionarmos manualmente. Também é comum associar esses dois métodos para evitar o problema do *Overfitting* dos dados já que introduzem uma penalização, resultando em um menor ajuste ao conjunto de treino e uma consequente melhor generalização para o conjunto de teste.

Esses dois métodos são parecidos, nesse trabalho foi utilizada a regressão Ridge, pois visa fazer os coeficientes de regressão chegarem próximos de zero, mas não totalmente zero como é o caso do Lasso, que, dessa forma, pode ser visto como um selecionador de preditores. Para Ridge, calculamos a penalização de acordo com a equação 7.

$$SSE^r = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{7}$$

No qual o λ é um hiperparâmetro positivo que pode ser determinado via validação cruzada. O intuito da regressão de Ridge também é minimizar o SSE. O termo adicional: $\lambda \sum_{j=1}^{p} \beta_j^2$ é a penalização feita pelo método de Ridge. Agora, para determinar os valores de β para Ridge basta aplicar a equação 8:

$$\beta^r = (X^T X + \lambda I)^{-1} X^T y \tag{8}$$

Sendo I a matriz identidade com mesma dimensão de $X^T X$. De acordo com [2], com esse termo de penalização é possível inserir um pouco de bias diminuindo a variância, ou seja, o ajuste excessivo dos dados.

Com a explicação do modelo que foi utilizado, pode-se prosseguir para verificar o método por meio das métricas RMSE e R^2. Primeiramente, foi escolhido um valor de λ qualquer e depois foi realizada uma validação cruzada para verificar um lambda ótimo. Foi utilizado um espaço de procura arbitrário de 10 valores, além disso, foi utilizada uma escala logaritmica para pegar valores de 10^{-3} até 10^3. Para conseguir encontrar esse λ foi utilizada validação cruzada nos dados de treino (ou seja, primeiramente se separou 75% dos dados em conjunto de treino e os outros 25% em teste), em que se foi testando cada um desses 10 valores de λ do espaço de procura, verificando qual forneceria um menor RMSE, e o escolhemos. Na validação cruzada foi usada 5-fold e 10-fold. Na tabela III estão os resultados obtidos.

Tabela III
COMPARAÇÃO DE METRICAS DO CONJUNTO DE TESTE DA REGRESSÃO DE RIDGE COM E SEM λ ÓTIMO

Métricas	λ	Lib	Autoral
RMSE (Lambda Qualquer)	1	0.542	0.542
R² (Lambda Qualquer)	1	0.805	0.805
RMSE(5-FOLD)	0.001	0.542	0.542
R²(5-FOLD)	0.001	0.805	0.805
RMSE(10-FOLD)	0.001	0.542	0.542
R²(10-FOLD)	0.001	0.805	0.805

Percebe-se que os valores encontrados pela biblioteca e pela implementação manual foram os mesmos. Além disso, vê-se que os valores são praticamente os mesmos no λ escolhido aleatoriamente quanto o feito por meio de cross-validation naqueles que são o conjunto de treino. O λ ser mínimo indica que a regressão de Ridge pode não ser necessária, já que quando o λ é pequeno, ela se aproxima de uma OLS e que a penalização aos coeficientes de regressão é mínima. Para 5 folds e 10 folds, vê-se que o resultado é o mesmo, o que é notório já que foi encontrado o mesmo valor de λ.

Outra coisa que se pode avaliar é os valores de RMSE e R^2 para o conjunto de treino, disponível na Tabela IV e comparar com o de teste da Tabela III.

Tabela IV
COMPARAÇÃO DE METRICAS RIDGE PARA CONJUNTO DE TREINO PARA A ESCOLHA DO λ

Métricas	λ	Lib	Autoral
RMSE(5-FOLD)	0.001	0.385	0.384
R²(5-FOLD)	0.001	0.901	0.902
RMSE(10-FOLD)	0.001	0.385	0.384
R²(10-FOLD)	0.001	0.901	0.902

É notável que há uma diferença para o conjunto de teste, mas é válido lembrar que esses são os valores de RMSE e R^2 que foram achados ao selecionar esse λ (0.001), portanto faz sentido esses valores serem muito bons (RMSE baixo e R^2 próximo de 1). Claro que há uma preocupação com overfitting, mas isso se descarta com o fato da diferença para o conjunto de teste ser próxima (cerca de 0.10).

D. PLS ou PCR

A regressão de Componentes Principais (PCR) é uma técnica estatística para estimar a relação entre uma variável dependente e um conjunto de variáveis independentes. Isso porque podemos representar as variáveis independentes por um conjunto de componentes principais, que são linearmente independentes e capturam a maior parte da variância das variáveis originais. Em outras palavras, é como se as variáveis

independentes são projetadas em um espaço de menor dimensão, definido pelos componentes principais. O modelo PCR então é estimado neste espaço de menor dimensão.

O PLS (Método alternativo à PCR) não foi utilizada, pois nosso conjunto de dados possui bastante variância e pouca correlação, de acordo com os box-plots de [1], que, de acordo com [7], são os campos que podem gerar inconsistências na PCR. Além do fato de já existir maior familiarização com a análise de componentes principais (PCA).

Podemos então usar a OLS como forma de fazer a regressão. Os componentes principais são obtidos a partir das variáveis independentes originais usando um procedimento de análise fatorial. O procedimento de análise fatorial busca encontrar um conjunto de vetores ortogonais que maximizem a variância da matriz de dados. Para fazer esse processo, precisamos seguir esse conjunto de passos.

1) Começamos com uma matriz de dados com n observações e p variáveis.
2) Calculamos a matriz de covariância dos dados.
3) Encontramos os autovetores da matriz de covariância.
4) Ordenamos os autovetores em ordem decrescente de autovalores.
5) Os autovetores com os maiores autovalores são os componentes principais.

A partir desse cálculo, conseguimos diminuir o número de componentes, obtendo os preditores independentes. A relação entre a PCR e o PCA é que a PCR é um modelo de regressão linear que usa os componentes principais do PCA como variáveis independentes.

O PCR temas seguintes vantagens:

- Pode ser usada para reduzir o número de variáveis independentes do modelo, o que pode simplificar a interpretação do modelo.
- Pode ser usada para melhorar a precisão da estimativa do modelo.

Algumas desvantagens são:

- Podemos adicionar um viés na escolha das componentes principais.

E. Rede Neural

Uma rede neural é uma técnica de aprendizado de máquina que se tornou muito popular nos últimos anos, principalmente pela sua facilidade de implementação e eficiência. Ela é composta por uma entrada, por camadas intermediárias que podem ser chamadas de camadas escondidas (hidden layers) e a saída. A saída no caso da rede neural de regressão é composta por apenas um único neurônio, visto que há apenas um valor que é o de predição.

O funcionamento da rede neural pode ser explicado por partes, a primeira é a realização de uma combinação linear entre os pesos associados a cada entrada e a entrada, de acordo com [5], pode-se indicar como: $u_k = \sum_{j=1}^{m} w_{kj} x_j$, sendo x_1,x_2,$\cdots$,x_m os sinais de entrada e w_{kj} são os pesos sinápticos (eles podem representar o quão importante é uma informação) para o neurônio k.

Sendo u_k a combinação linear para o neurônio k, o próximo passo é somar com um termo chamado Bias que, pode estar ligada ao próximo passo que é usar uma função de ativação que é utilizada para trazer uma não linearidade ao modelo, o bias(ou viés) pode ajudar a trazer essas funções de ativação para o lado negativo ou lado positivo, modificando a sua origem. Resumidamente, possuimos o seguinte para a saída do neurônio, de modo geral: $y_k = \phi(u_k + b_k)$, sendo ϕ a função de ativação e b_k o viés. Essa saída é utilizada como entrada para a próxima camada. Ao chegar na saída é considerada uma passagem completa, também chamado de Propagação Direta. Após isso o que ocorre é que com a geração de uma saída, pode-se comparar a saída predida com a saída esperada e é calculado uma função de perda (nesse artigo foi utilizado o RMSE).

Após isso, é feito a etapa de retropropagação. Essa etapa é de forma simples tentar minimizar essa função de perda ajustando os pesos (é um algoritmo de otimização). A primeira coisa é o cálculo do gradiente do erro em relação aos pesos. Após o cálculo do gradiente pode-se fazer a atualização dos pesos: $w(n+1) = w(n) + \eta \nabla E(w) n$, sendo $E(w)$ a função custo, n representa a iteração, η é um hiperparâmetro chamado de taxa de aprendizado que é fundamental, pois determina o tamanho dos passos em direção oposta ao gradiente, não pode nem ser muito pequeno, pois pode demorar para convergir e nem muito grande, pois pode nunca convergir. $w(n)$ e $w(n+1)$ são os pesos antigos e novos, respectivamente. Dessa forma, uma rede neural vai sempre aprendendo e ajustando seus pesos para tentar performar melhor na próxima época.

Há outro hiperparâmetro que pode ser interessante, mas geralmente é escolhido por tentativa e erro (valores típicos são: 32, 64 e assim por diante), é o "lote"ou "batch". Ele basicamente escolhe a quantidade de amostras que será utilizada em cada iteração de uma época. Época é uma passagem completa por todo o conjunto de treinamento.

Com a teoria explicada, pode-se partir para a aplicação da rede neural na predição do álcool. A primeira coisa a se fazer é o mesmo pré-processamento para o conjunto de preditores e separar em treino e teste, novamente foi escolhida a proporção 75%/25%. Como já foi visto em [1] muitos dados não possuem uma relação linear com o álcool. Com isso, métodos como a rede neural podem ser benéficos por se tratar de um modelo que consegue capturar não linearidades. Ademais, foi realizado a construção de uma rede neural simples, com apenas uma camada densa com 128 neurônios que com certeza é maior que a quantidade de preditores que é o requisito mínimo e também não se possue um cálculo certo para indicar sua quantidade. Foram realizados dois testes, um com a função de ativação sigmoide e outro com a Relu (Rectified Linear Unit) cada uma tem suas vantagens, a Relu geralmente é mais rápida, pois é mais simples (ela retorna o próprio valor da entrada se for positivo e se for negativo retorna 0), a sigmoide tem uma forma de S e coloca qualquer valor entre 0 e 1. Vamos ver as métricas em termos de R^2 e RMSE para essas duas funções de ativação na Tabela V.

Pode-se ver que os resultados das duas funções foram muito

Tabela V
COMPARAÇÃO DE METRICAS REDE NEURAL

Métricas	ReLU	Sigmoide
RMSE	0.400	0.446
R²	0.894	0.868

próximos, mas mesmo assim percebe-se uma diferença. A ReLU conseguiu um menor RMSE e um maior R^2 para um mesmo número de épocas (40), quantidade de neurônios (128) e tamanho de batch (32). Isso pode ocorrer devido a uma captura maior de não linearidades.

Outro ponto a se comentar é que a rede neural sempre pode mudar essas métricas dependendo da quantidade de vezes que ela é treinada e testada, isso se deve pela própria natureza do modelo de redes neurais. Ou seja, possa ser que ao rodar o algoritmo ele nos dê outro valor de RMSE, já que o limite dele é a quantidade de épocas, isso foi escolhido devido a simplicidade e praticidade, mas o valor de RMSE que vai ser colocado não vai variar muito dos valores observados na Tabela V.

III. DISCUSSÃO DOS RESULTADOS

A. *Pré-processamento*

Primeiramente, os métodos propostos na introdução foram aplicados em todo o nosso conjunto de dados. Tomamos um único preditor como exemplo por questões de visualização, como pode-se ver na figura 1. Fica claro que os dados estão mais simétricos.

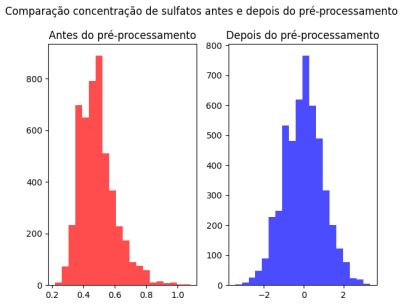

Figura 1. Comparativo dos dados antes e depois do pré-processamento

Seguindo, Após calcular a média e o desvio padrão, observa-se na tabela VI que com os dados brutos tínhamos um valor significativo para média e o desvio padrão era diferente de 1, com os pré-processados observa-se o contrário, Como fora dito na seção 1.

Tabela VI
COMPARAÇÃO MÉDIA E DESVIO-PADRÃO

Métrica	Brutos	Pré-processados
Média	7.978	0
Desvio Padrão	0.114	1

B. *Regressão Linear (OLS) e Penalizada L2*

Para realização da análise dos resultados desses dois modelos, vamos utilizar gráficos e as Tabelas I, II e III. Começando com o modelo OLS, vamos analisar um gráfico da Figura 2 que compara os valores de predição da biblioteca com as funções implementadas manualmente, além disso verifica a comparação entre valores esperados com a previsões feitas pelo nosso modelo OLS.

Na Figura 2. A reta em vermelho faz a comparação entre os valores obtidos por meio da biblioteca do Python e os obtidos por meio da implementação feita pelos autores, vê-se que é uma reta contínua, o que indica que foi obtido valores parecidos tanto para a biblioteca do Python quanto para a implementação manual. Além disso é comparado os valores predidos e os valores reais em azul (saída de teste) e vê que não ficaram dispersos, o que pode significar que as previsões foram parecidas com os valores esperados.

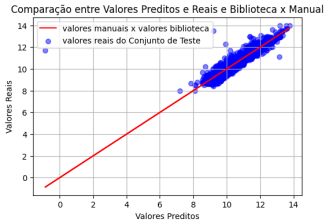

Figura 2. Comparação de dados da biblioteca e Comparação entre Valores Reais e Previsões para OLS

Ademais, na Figura 3 e com base na Tabela I e II, vamos ver uma comparação do valor RMSE, ao usar e não usar Cross-Validation. Vemos que a avaliação do modelo ficou com um RMSE melhor (menor) ao usar o cross validation, isso pode ser explicado porque o modelo pode ter usado diferentes tipos de dados para treino e teste, podendo até evitar possíveis overfittings.

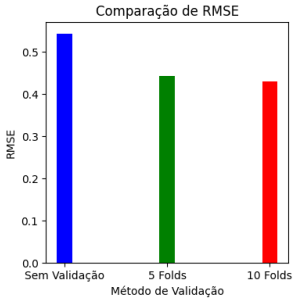

Figura 3. Comparando os resultados ao usar e não usar Validação Cruzada

Agora, vamos ver como ficaram os resultados da Regressão Penalizada, nota-se na Figura 4 que conforme aumentamos o λ, o RMSE aumenta e o R^2 diminui, esse resultado é esperado, pois é visto na Tabela IV que o λ escolhido foi o menor (0.001), tanto pela biblioteca quanto pela implementação dos autores. Esse valor de λ foi escolhido porque o nosso modelo está preferindo usar os preditores sem penalizar, pois a sua

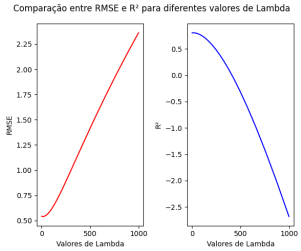

Figura 4. Comparando os Valores de λ e o RMSE e R^2

capacidade de predição ficaria pior. É possível notar que os λ da Figura 4 quanto maior, pior está sendo as métricas do modelo, isso acaba indo de encontro com o que foi encontrado como λ ideal, que foi de λ = 0.001.

Por fim, podemos perceber que os resultados obtidos para esses dois modelos foram bem parecidos, até porque a regressão de Ridge ficou praticamente como a regressão OLS. Portanto, para o nosso conjunto de dados não se tornou necessário o uso da regressão de Ridge, já que um modelo mais simples como o OLS conseguiu performar com um menor RMSE e R^2 maior em situações que podem ser vistas na Tabela II.

C. PCR

Após diversos testes, percebe-se que quanto mais preditores são removidos da regressão pior fica o RMSE e a pontuação R^2. Com este resultado incomum, é esperado que o PCR com qualquer redução de dimensionalidade tenha um resultado pior do que uma regressão linear ordinária.

Como podemos ver na tabela VII, se compararmos com a tabela I as métricas vem piorando à medida que removemos preditores.

Tabela VII
MÉTRICAS RELATIVAS A QUANTIDADE DE PREDITORES

Métricas	11 Pred. (Todos)	8 Pred.	5 Pred.	2 Pred.
RMSE	0.512	0.742	0.751	0.883
R²	0.823	0.628	0.619	0.474

Para confirmar, foi realizado a validação cruzada. Que nos mostrou que o número ótimo de preditores é 11 (O conjunto de preditores completo). Com isso, como mostra a figura 5 conclui-se que devido à pequena dimensionalidade do nosso conjunto, devemos manter todos os preditores que inclusive obteve resultados melhores que em relação a validação cruzada sem Cross-validation.

D. Rede Neural

No desenvolvimento da rede neural foi percebido que os valores foram favoráveis já que o RMSE diminuiu e o R^2 aumentou mais que o OLS como pode ser visto na Tabela V em comparação com a I e II. Pela Tabela V pode-se ver que houve uma diferença dependendo da função de ativação. Uma das possíveis causas é pela natureza da função de ativação e pelo fato da ReLU conseguir captar mais dados não lineares.

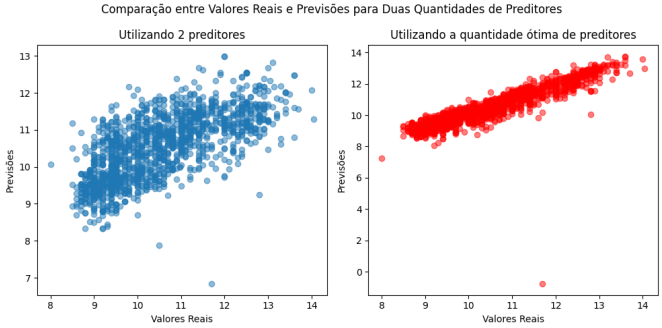

Figura 5. Predição com 11 preditores comparado à Predição com 2 preditores

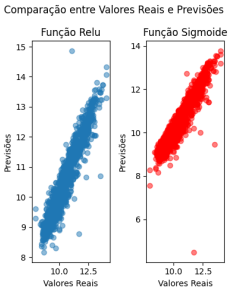

Figura 6. Comparando os Resultados das Funções de Ativação

É interessante ver como os modelos se saíram com dados de teste. Podemos ver isso facilmente por meio de um plot. Olhe a Figura 6.

Pode-se ver na Figura 6 que forma uma relação linear e que há muitos dados sobrepostos, o que pode dizer que o modelo conseguiu acertar muitas previsões. Mesmo a Relu tendo se saído melhor, conforme pôde ser visto, o resultado das duas são muito parecidos.

O modelo de rede neural é muito interessante para dados que não possuem uma natureza linear, conforme foi visto em [1] isso de fato aconteceu para muitos preditores. E dentro todas os modelos testados no presente artigo, o que conseguiu melhor predir o resultado do álcool foi o uso de redes neurais (usando as métricas RMSE e R^2 como avaliação), mesmo com um modelo simples de uma camada escondida e 128 neurônios.

REFERÊNCIAS

[1] Pinheiro, Gabriel P. P., Vasconcelos, Igor F., Dias, Eliabe B., (2023). Estatística Descritiva Como Ferramenta Para Análise Exploratória de Qualidade de Vinho Português.

[2] Kuhn, Max, and Kjell Johnson. Applied Predictive Modeling. Springer, 2013.

[3] Gareth James, Danie$_l aWitten, Trevor Hastie, Robert Tibshirani. An Introductio$
$with Applications in Python, 1sted., New York : Springer, 2023$

[4] Cortez, P. et al. “Modeling wine preferences by data mining from physico-chemical properties.” Decis. Support Syst. 47 (2009): 547-553.

[5] Haykin, Simon S.. Neural networks and learning machines. Third Upper Saddle River, NJ: Pearson Education, 2009.

[6] In-Kwon Yeo, Richard A. Johnson. A new family of power transformations to improve normality or symmetry. Biometrika , Dec., 2000,

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.