Predição das Emissões de Gases de Efeito Estufa em Edifícios de Seattle: Comparação de Modelos de Regressão e Redes Neurais

Homework 2, Applied Computational Intelligence

1st Yago Castro dos Reis
Departamento de Engenharia Elétrica
Universidade Federal do Ceará
Fortaleza, Ceará
yago.castro@alu.ufc.br

***Abstract*—O objetivo deste trabalho é analisar o desempenho de diferentes algoritmos de regressão, incluindo modelos lineares como regressão linear simples e múltipla, utilizando um conjunto de dados previamente explorado. O pré-processamento inclui técnicas como centralização, escalonamento, tratamento de assimetrias, transformação de variáveis categóricas, e seleção de features. Diferentes abordagens de construção de conjuntos de dados foram testadas para avaliar seu impacto nos modelos.**

Na modelagem, foram aplicados algoritmos como regressão linear simples, múltipla e penalizada, além de redes neurais para modelos não lineares. Os resultados foram avaliados com as métricas R^2 e RMSE para identificar as abordagens mais eficazes.

***Index Terms*—pré-processamento, regressão linear, redes neurais TotalGHGEmissions, eficiência energética, Seattle**

I. Introdução

A. Objetivos

1) **Pré-processamento dos Dados**
   - Converter variáveis categóricas em numéricas.
   - Normalizar os dados e analisar correlações.
   - Criar quatro conjuntos com diferentes pré-processamentos.
2) **Escolha da Variável Alvo**
   - Definir *TotalGHGEmissions* e verificar sua relevância.
3) **Regressão Linear Simples**
   - Aplicar e comparar o modelo com base na variável mais correlacionada.
4) **Regressão Linear Múltipla**
   - Utilizar múltiplas variáveis e validar o modelo.
5) **Regressão Penalizada**
   - Aplicar Ridge e LASSO para evitar overfitting.
6) **Regressão Transformada**
   - Implementar PCR e PLS e analisar impactos.
7) **Redes Neurais**
   - Criar modelos simples e complexos com regularização.

Identify applicable funding agency here. If none, delete this.

B. Regressão Linear

A regressão linear é uma ferramenta simples e eficaz para prever variáveis quantitativas. Embora seja limitada a relações lineares, ela serve como base para técnicas mais avançadas, como as Redes Neurais. A equação da regressão linear para uma única variável é dada pela fórmula (1).

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

Na equação da regressão linear, Y é a saída esperada, $_0$ é a interceptação no eixo das ordenadas,$_1$ é o coeficiente angular, X é o valor do preditor e é o erro. Quando há múltiplos preditores, a equação é generalizada, como mostrado na equação (2).

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon \tag{2}$$

Na equação generalizada, $_p$ é o coeficiente do preditor, e x_ip é o valor da amostra do i-ésimo preditor. Em vez de um único coeficiente angular, todas as variáveis contribuem com um peso para determinar a saída. Os valores ótimos de $\hat{\beta}$, que equilibram viés e variância, são encontrados resolvendo a equação (3).

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y \tag{3}$$

$\hat{\beta}$ é o vetor de coeficientes, X a matriz de preditores e Y o vetor de respostas. A regressão linear ordinária busca minimizar o erro da soma dos quadrados (ESS), conforme a equação (4):

$$\text{ESS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4}$$

O ESS é a soma dos quadrados do erro, onde y_i é o valor real e $\hat{y}_i$ é a saída estimada. O objetivo é minimizar a diferença entre a saída prevista e a real.

C. Regressão PCR e PLS

Em conjuntos de dados grandes, a inclusão de todos os preditores pode ser desnecessária. A Principal Component Regression (PCR) reduz a dimensionalidade, explicando a variância sem considerar a relação com a variável resposta. Já o Partial Least Squares (PLS) é supervisionado e foca na relação entre preditores e saída, aprimorando a regressão linear.

D. Regressão Ridge e Lasso

A regressão de Ridge adiciona uma penalização ao erro, com um fator λ multiplicado por β_j, para reduzir a variância e melhorar o desempenho do modelo.

$$\hat{\beta}_R = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{5}$$

A penalização com diferentes valores de λ ajuda a resolver problemas em matrizes não invertíveis, mas na regressão de Ridge ela aplica penalização a todos os preditores, o que pode dificultar a interpretação. A regressão LASSO resolve isso, pois grandes valores de λ reduzem os coeficientes a zero, diminuindo a dimensionalidade.

$$\hat{\beta}_L = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{6}$$

E. Métricas de avaliação

Até agora, foram apresentados os métodos de regressão linear, mas para validar os coeficientes, é necessário dividir os dados em treino e teste. Os coeficientes β_i são ajustados no conjunto de treino e aplicados no conjunto de teste para avaliar a precisão do modelo. A equação (7), conhecida como raiz da média das somas dos quadrados, indica o erro entre o valor predito e o real, sendo que quanto menor o valor, menor o erro.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{7}$$

Já a equação (8) é um indicador que varia de 0 a 1 e revela a proporção de variância explicada pela regressão. Ou seja, quanto mais próximo de 1, melhor é o modelo.

$$R^2 = 1 - \frac{\text{ESS}}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \tag{8}$$

O valor $\bar{y}$ é a média do preditor. A validação cruzada k-fold divide o conjunto de treino em k subconjuntos, utilizando cada um para validação e os outros para treino, repetindo o processo até que todos os subconjuntos sejam validados. O LOOCV, semelhante ao k-fold, usa uma amostra para validação e o restante para treino, mas é mais custoso e gera resultados parecidos com o 10-fold.

F. Redes Neurais

As redes neurais realizam processamento paralelo massivo de processos simples, armazenando experiências e ajustando pesos de acordo com cada variável para aprender e gerar saídas. As equações (9) e (10) representam matematicamente o modelo de redes neurais.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{9}$$

u_k é o resultado da combinação linear, w_{kj} é o peso atribuído à entrada, e x_j é a entrada.

$$y_k = \phi(u_k + b_k) \tag{10}$$

y_k é a saída, ϕ é a função de ativação (como Sigmoid ou ReLU), e b_k é o bias. As redes neurais são baseadas na regressão linear, com a principal diferença sendo a função de ativação nos modelos.

II. Metodologia

A. Estrutura Geral de Organização e Desenvolvimento do Trabalho

A estrutura organizacional do projeto foi baseada na metodologia CRISP-DM, que, conforme descrito por Caio Roberto [2], possui seis etapas bem definidas. Neste trabalho, serão exploradas principalmente as etapas de preparação dos dados, modelagem e avaliação. O desenvolvimento foi realizado utilizando a linguagem Python, com as bibliotecas listadas no arquivo **requirements.txt**, disponível no **repositório do projeto**.

B. Pré-processamento dos Dados

No pré-processamento dos dados, foram aplicadas técnicas para melhorar os resultados dos modelos de regressão. As variáveis categóricas foram convertidas para formato numérico por meio de Codificação Label, Codificação de Frequência e One Hot Encoding. A variável alvo foi separada para evitar a aplicação de técnicas como centralização e escalonamento nela. A centralização foi realizada nas variáveis para ajustar suas distribuições, conforme a figura 1,trazendo a média para zero. Também foi aplicado o escalonamento com MinMaxScaler para ajustar os valores entre 0 e 1, o que auxilia no desempenho de alguns algoritmos. Para tratar a assimetria dos dados, foi feita uma transformação logarítmica. A matriz de correlação foi analisada para identificar variáveis com maior correlação com a variável alvo, possibilitando a redução do conjunto de dados e a adição de novas features.

- Relações de proporcionalidade
- Interações entre as variáveis
- Transformações matemáticas

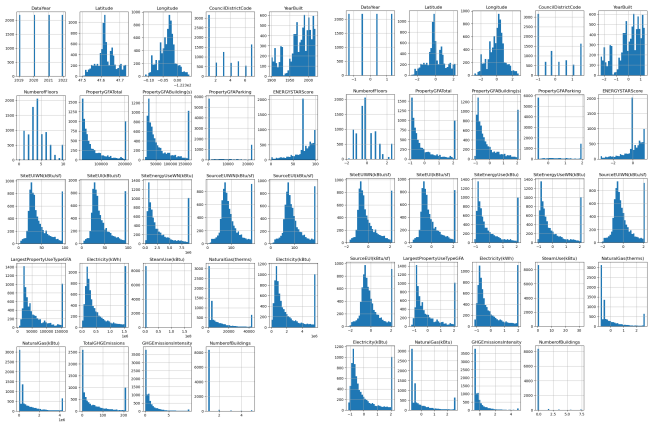

Fig. 1: Centralização das variáveis. Fonte: O próprio autor.

1) Escolha e verificação da variável alvo: A variável TotalGHGEmissions representa as emissões totais de gases de efeito estufa (GHG), como dióxido de carbono, metano e óxidos de nitrogênio, geradas pelo consumo de energia nos edifícios. É medida em toneladas métricas de CO equivalente (tCOe) e calculada com base nos fatores de emissão da Seattle City Light. Ela é crucial para avaliar a pegada de carbono, a eficiência energética e apoiar práticas sustentáveis, além de ser um indicador de conformidade com políticas ambientais. O objetivo do trabalho é prever essas emissões com base em dados anuais e analisar a correlação entre a variável TotalGHGEmissions e as variáveis explicativas no conjunto de dados, utilizando coeficientes de correlação para entender o grau de explicação das variáveis sobre a emissão total.

```
Ranking das variáveis como possíveis alvos:
SiteEnergyUse(kBtu): 8.3115
SiteEnergyUseWN(kBtu): 8.3110
Electricity(kBtu): 7.0398
Electricity(kWh): 7.0173
TotalGHGEmissions: 6.8668
PropertyGFATotal: 6.1265
PropertyGFABuilding(s): 5.9378
LargestPropertyUseTypeGFA: 5.7787
SiteEUI(kBtu/sf): 5.6642
SiteEUIWN(kBtu/sf): 5.6500
NaturalGas(kBtu): 5.2797
NaturalGas(therms): 5.2681
SourceEUIWN(kBtu/sf): 5.0212
SourceEUI(kBtu/sf): 5.0212
GHGEmissionsIntensity: 3.7478
PropertyGFAParking: 1.8005
NumberofFloors: 1.6683
ENERGYSTARScore: 1.2340
YearBuilt: 0.7278
NumberofBuildings: 0.4320
CouncilDistrictCode: 0.3870
SteamUse(kBtu): 0.3444
Latitude: 0.2340
Longitude: 0.0949
DataYear: 0.0250
```

Fig. 2: Escolha e verificação da variável alvo. Fonte: O próprio autor.

Como podemos verificar na Figura 2, a variável TotalGHGEmissions apresenta um somatório igual a 6.8668, sendo esta a quinta variável no ranking de somatórios das correlações com as demais variáveis. Logo, ela é uma boa escolha para realizarmos os estudos e as aplicações dos modelos de regressão.

2) Divisão dos conjunto de dados: Foram construídos 4 modelos para os testes e aplicações no conjunto de dados, utilizando diferentes técnicas de pré-processamento. O objetivo é testar esses modelos e compará-los com base nas métricas de avaliação geradas.

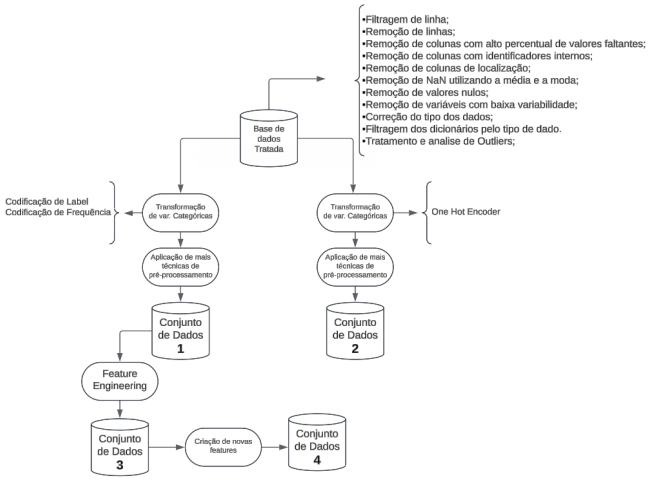

Fig. 3: Fluxograma dos procedimentos que foram realizados para cada conjunto de dados. Fonte: O próprio autor.

- **Conjunto de dados 1**: Criado com base nos procedimentos de limpeza e transformação das variáveis categóricas (label encoding e codificação por frequência), seguido de centralização, escalonamento e tratamento de assimetria dos dados.
- **Conjunto de dados 2**: Também derivado dos procedimentos de limpeza, mas com a aplicação de one hot encoding nas variáveis categóricas, gerando mais variáveis. Centralização, escalonamento e tratamento de assimetria também foram aplicados.
- **Conjunto de dados 3**: Derivado do conjunto 1, com foco em armazenar as principais variáveis, utilizando feature engineering a partir da correlação das variáveis com a variável alvo.
- **Conjunto de dados 4**: Derivado do conjunto 3, com a geração de novas features a partir das variáveis existentes.

C. Regressão Linear Simples

Na aplicação da regressão linear simples, o primeiro passo foi definir a variável independente, com base na análise da matriz de correlação de cada conjunto de dados. Variáveis com alta correlação com a variável alvo foram selecionadas, como mostrado na Figura 3 para o Conjunto 4. Além disso, o modelo foi construído e comparado conceitualmente, utilizando também o modelo da biblioteca Scikit-learn. Em seguida, as variáveis independentes foram escolhidas para cada conjunto de dados.

- Conjunto 1: NaturalGas(therms)

- Conjunto 2: NaturalGas(kBtu)
- Conjunto 3: SiteEnergyUse(kBtu)
- Conjunto 4: LogNaturalGas

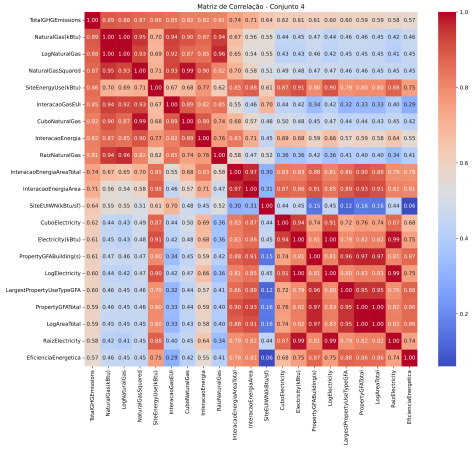

Fig. 4: Matriz de correlação gerada para o conjunto de dados 4, utilizada para definir a variável independente do modelo OLS. Fonte: O próprio autor.

D. Regressão Linear Múltipla

Nessa etapa, será construído um modelo conceitual e um modelo utilizando a biblioteca Scikit-learn para avaliar os resultados nos quatro conjuntos de dados gerados. Em seguida, será aplicada a validação cruzada com k = 5 e k = 10 no conjunto com a melhor avaliação inicial, para garantir que o modelo seja capaz de generalizar bem, e não apenas memorizar os dados de treinamento.

E. Regressão Penalizada

Nesta seção, adota-se uma abordagem penalizada para reduzir o overfitting e melhorar a generalização do modelo, implementando as técnicas LASSO e Ridge separadamente. O melhor valor de lambda será testado com validação cruzada antes de aplicar os modelos. O Ridge será implementado manualmente, devido ao desempenho esperado em conjuntos com correlação entre variáveis. Observou-se uma divergência nos coeficientes da regressão Ridge quando calculada com gradiente descendente em comparação ao método de Cholesky, sendo os resultados ajustados iguais aos do Scikit-learn.

F. Regressão Transformada

Nessa etapa foram aplicados tanto o modelo PCR e o PLS verificando o número de componentes.

G. Redes Neurais

Na aplicação das redes neurais, serão utilizados dois modelos. O primeiro é mais simples, com camadas densas e funções de ativação ReLU para capturar relações não lineares entre as variáveis de entrada e a variável alvo. A rede é treinada com o otimizador Adam e a função de perda RMSE. O segundo modelo é mais complexo, adicionando regularização com múltiplas camadas ocultas, BatchNormalization e Dropout para evitar overfitting e melhorar a generalização, visando capturar padrões mais complexos e melhorar a precisão da previsão.

III. Resultados e Discussão

A. Regressão Linear Simples

O modelo foi treinado de forma independente, com base na equação de definição dos parâmetros, usando os mesmos conjuntos X e y, e as mesmas features e variável alvo. Depois, os resultados obtidos foram comparados com a implementação do modelo utilizando a biblioteca Scikit-learn.

TABLE I: Tabela de resultados para a aplicação da Regressão Linear Simples nos conjuntos de dados de forma conceitual e utilizando a biblioteca Scikit-learn.

Modelo	b0	b1	R^2	RMSE
OLS_conceitual (Conj1)	9,1908	292,1798	0,8076	31,4654
OLS_conceitual (Conj2)	9,0599	292,5643	0,8128	31,0401
OLS_conceitual (Conj3)	-14,8288	283,0472	0,7536	35,6055
OLS_conceitual (Conj4)	5,5348	372,283	0,8014	32,8488
OLS_Scikit (Conj1)	9,1908	292,1798	0,8076	31,4654
OLS_Scikit (Conj2)	9,0599	292,5643	0,8128	31,0401
OLS_Scikit (Conj3)	-14,8288	283,0472	0,7536	35,6055
OLS_Scikit (Conj4)	5,5348	372,283	0,8014	32,8488

A Tabela 1 mostra os resultados da aplicação da regressão linear simples, tanto de forma conceitual (usando as equações descritas no início) quanto com a biblioteca Scikit-learn. Os resultados obtidos de ambas as abordagens foram iguais para todos os conjuntos de dados.

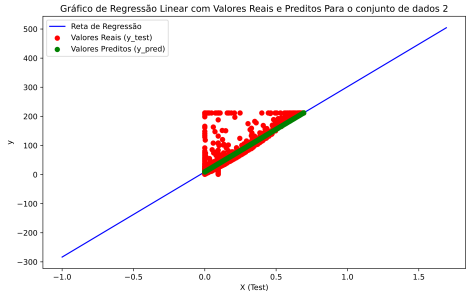

Fig. 5: Resultado da Regressão Linear Simples aplicada no conjunto de dados 2. Fonte: O próprio autor.

Na figura, é apresentado o comparativo da reta de regressão obtida a partir do modelo de regressão linear aplicado ao conjunto de dados 2, mostrando que ainda é necessário explorar outros modelos.

B. Regressão Linear Múltipla

Na aplicação da regressão linear múltipla, a implementação manual não convergiu para o resultado esperado, enquanto a biblioteca obteve o resultado correto, possivelmente devido ao número elevado de features. No entanto, os demais resultados foram consistentes em ambas as abordagens, e os valores da validação cruzada foram próximos nas duas metodologias.

TABLE II: Resultados da regressão linear múltipla de forma conceitual e utilizando a biblioteca do Scikit-learn para cada conjunto de dados

Modelo	R^2	RMSE	Tempo de Execução [s]
MLR_conceitual(Conj1)	0,9482	16,3229	1,6413
MLR_conceitual(Conj2)	-0,4057	85,048	9,9447
MLR_conceitual(Conj3)	0,9371	17,9888	0,3192
MLR_conceitual(Conj4)	0,9471	16,4233	1,4458
MLR_scikit(Conj1)	0,9482	16,3229	0,0830
MLR_scikit(Conj2)	0,9489	16,2125	0,2792
MLR_scikit(Conj3)	0,9371	17,9888	0,0197
MLR_scikit(Conj4)	0,9471	16,4233	0,9831

TABLE III: Resultados de Modelos de Regressão

Modelo	k = 5		k = 10	
	R^2	RMSE	R^2	RMSE
MLR__scikit(Conj1) [manual]	0.9445	16.4559	0.9449	15.8956
MLR__scikit(Conj1)	0.9451	16.3873	0.9451	16.3920
MLR__scikit(Conj4) [manual]	0.9498	15.6334	0.9508	14.9861
MLR__scikit(Conj4)	0.9498	15.6780	0.95001	15.6583

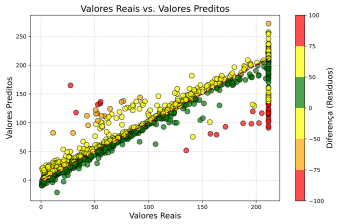

Fig. 6: Gráfico comparativo dos resultados obtidos (valor real e valor predito) para o modelo $\text{MLR}_scikit(Conj4).Fonte : Opróprioautor$.

O gráfico da figura acima ilustra que os dados estão se ajustando ainda mais em comparação à regressão linear simples e destaca os pontos considerando o erro residual. Quanto maior o erro, mais próximo do vermelho estará o ponto.

C. Modelo Penalizado (L1 ou L2)

TABLE IV: Resultados dos métodos Ridge e LASSO para regressão linear

Método	Melhor λ	R^2	RMSE
Ridge	0,1	0,9480	16,4244
LASSO	0,01	0,9476	16,4843

TABLE V: Resultados do modelo Ridge

Modelo	R^2	RMSE
Ridge_conceitual[Gradiente]_(Conj1)	0,8687	25,9892
Ridge_conceitual[Cholesky]_(Conj1)	0,9477	16,4034
Ridge_scikit_(Conj1)	0,9477	16,4034
Ridge_conceitual[Gradiente]_(Conj2)	0,8679	26,0677
Ridge_conceitual[Cholesky]_(Conj2)	0,9485	16,2764
Ridge_scikit_(Conj2)	0,9485	16,2764
Ridge_conceitual[Gradiente]_(Conj3)	0,7556	35,4606
Ridge_conceitual[Cholesky]_(Conj3)	0,9368	18,0213
Ridge_scikit_(Conj3)	0,9368	18,0213
Ridge_conceitual[Gradiente]_(Conj4)	0,8638	26,3601
Ridge_conceitual[Cholesky]_(Conj4)	0,9423	17,1538
Ridge_scikit_(Conj4)	0,9423	17,1538

Inicialmente, analisamos o melhor valor de lambda e aplicamos a regressão Ridge nos dados. O gradiente descendente encontrou os parâmetros, mas não convergiu de forma eficiente. Ao utilizar o algoritmo de Cholesky, conseguimos obter os resultados desejados, sendo que o conjunto de dados 2 apresentou o melhor desempenho.

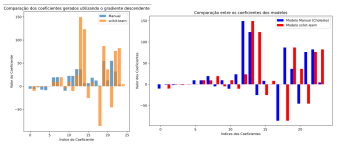

Fig. 7: Gráfico comparativo entre os parâmetros obtidos a partir do gradiente descendente e do método de Cholesky para o modelo $\text{MLR}_scikit(Conj4).Fonte : Opróprioautor$.

Na Figura 0, o gradiente descendente mostrou valores distintos para alguns coeficientes, o que não aconteceu com o método de Cholesky.

D. Modelo de Regressão PLS ou PCR

TABLE VI: Resultados das métricas de avaliação obtidas em cada conjunto aplicado o PCR e o PLS

Modelo	R^2	RMSE
PCR_(Conj1)	0,9485	16,2854
PCR_(Conj2)	0,9487	16,2439
PCR_(Conj3)	0,9371	17,9888
PCR_(Conj4)	0,9464	16,5427
PLS_(Conj1)	0,9215	20,1021
PLS_(Conj2)	0,9188	20,4464
PLS_(Conj3)	0,8746	25,4061
PLS_(Conj4)	0,8774	25,0140

Conforme a tabela 0, o PCR se destaca em termos de desempenho em relação ao PLS devido à sua maior capacidade de explicar a variabilidade dos dados e ao menor erro de previsão (RMSE mais baixo)

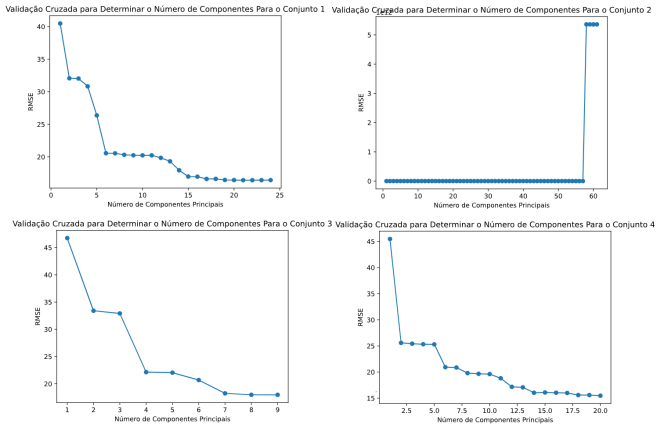

Fig. 8: Gráficos da determinação do número de componentes principais para cada conjunto de dados. Fonte: O próprio autor.

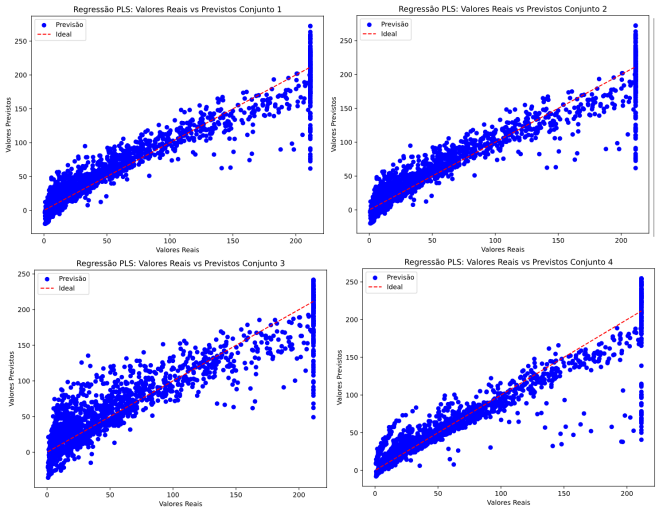

Fig. 9: Comparativo dos resultados obtidos na aplicação do PLS para os dados de previsão e os dados reais. Fonte: O próprio autor.

A figura mostra a seleção de componentes para o PCR, com alguns casos apresentando muitos componentes principais. Também é exibido o comportamento da previsão com PLS, onde, em áreas com valores altos da variável alvo, a correlação com as variáveis independentes diminui, dificultando a previsão. Isso pode ser devido à concentração de valores em um quartil após o tratamento de outliers, afetando a correlação.

E. Rede Neural

TABLE VII: Resultados das métricas de avaliação obtidas em cada conjunto aplicado o modelo de Rede Neural 1 e 2

Modelo	R^2	RMSE	Tempo de Execução (s)
RedeNeural_1(Conj1)	0.9778	10.7183	107.1915
RedeNeural_1(Conj2)	0.9850	8.8115	88.0186
RedeNeural_1(Conj3)	0.9545	15.3588	104.1874
RedeNeural_1(Conj4)	0.9567	14.9724	96.1138
RedeNeural_2(Conj1)	0.9810	9.9212	154.4361
RedeNeural_2(Conj2)	0.9801	10.1371	152.9070
RedeNeural_2(Conj3)	0.9589	14.5902	150.5189
RedeNeural_2(Conj4)	0.9589	14.5902	150.5189

O modelo de rede neural 2 teve o melhor desempenho na maioria dos conjuntos de dados, embora o conjunto de dados 2 tenha mostrado melhores resultados com uma rede mais simples. A técnica de one hot encoding foi eficaz, e adicionar mais camadas nem sempre melhorou os resultados. O tempo de execução também foi destacado como um fator importante, considerando o custo computacional em relação à melhoria do modelo.

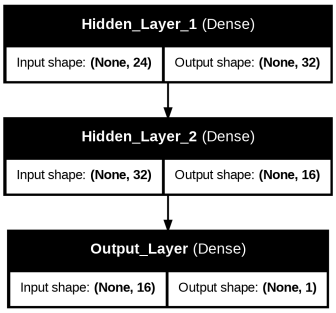

Fig. 10: Apresentação da estrutura da rede neural 1. Fonte: O próprio autor.

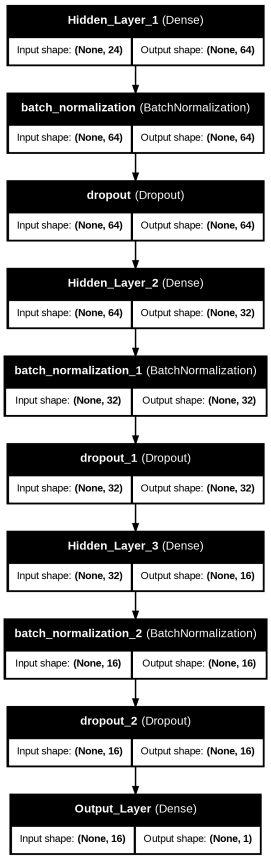

Fig. 11: Apresentação da estrutura da rede neural 2. Fonte: O próprio autor.

As Figuras acima mostram as estruturas das redes neurais, destacando a rede mais complexa na figura 11.

IV. Conclusão

Este estudo aplicou modelagem preditiva para estimar as emissões de GEE em edifícios, utilizando o benchmarking energético de Seattle e a metodologia CRISP-DM. O pré-processamento incluiu codificação, normalização, análise de correlação e feature engineering. Modelos como Ridge e Redes Neurais tiveram melhor desempenho, mas a correlação entre variáveis diminuiu em áreas com altas emissões, dificultando

a previsão. A avaliação das métricas revelou a necessidade de reduzir o RMSE, pois seus valores representam toneladas emitidas. Com o estudo, foi possível verificar a importância da modelagem na aplicação de políticas sustentáveis e na otimização da eficiência energética. Para aprimorar os resultados, também pode-se explorar novas técnicas para tratar os outliers, além de combinar modelos preditivos e incorporar variáveis, como as climáticas.

REFERENCES

[1] Seattle.gov, "2019 Building Energy Benchmarking", City of Seattle. [Online]. Disponível: https://data.seattle.gov/Built-Environment/2019-Building-Energy-Benchmarking/3th6-ticf/about_data. [Acessado: 10 de novembro de 2024].

[2] Seattle.gov, "2020 Building Energy Benchmarking", City of Seattle. [Online]. Disponível: https://data.seattle.gov/Built-Environment/2020-Building-Energy-Benchmarking/auez-gz8p/about_data. [Acessado: 10 de novembro de 2024].

[3] Seattle.gov, "2021 Building Energy Benchmarking", City of Seattle. [Online]. Disponível: https://data.seattle.gov/Built-Environment/2021-Building-Energy-Benchmarking/bfsh-nrm6/about_data. [Acessado: 10 de novembro de 2024].

[4] Seattle.gov, "2022 Building Energy Benchmarking", City of Seattle. [Online]. Disponível: https://data.seattle.gov/Built-Environment/2022-Building-Energy-Benchmarking/5sxi-iyiy/about_data. [Acessado: 10 de novembro de 2024].

[5] Seattle.gov, "Office of Sustainability Environment", City of Seattle. [Online]. Disponível: https://www.seattle.gov/environment/climate-change/buildings-and-energy/energy-benchmarking. [Acessado: 12 de novembro de 2024].

[6] Seattle.gov, "Seattle Energy Benchmarking Ordinance", City of Seattle. [Online]. Disponível: https://energyfave.com/energy-benchmarking-compliance/seattle-energy-benchmarking/#:~:text=Through%20benchmarking%2C%20building%20owners%20and,emissions%2C%20and%20generate%20cost%20savings. [Acessado: 12 de novembro de 2024].

[7] BUSSAB, Wilton de O.; MORETTIN, Pedro A. *Estatística Básica*. 5ª edição. São Paulo: Saraiva, 2002.

[8] KUHN, Max et al. Applied predictive modeling. New York: Springer, 2013.

[9] MORAES, Lailson B.; CAVALCANTI, George DC; TSANG, I. R. Analise do PCA Assim ´ etrico para Detecc¸ao de Objetos em Imagens, ´ 2023