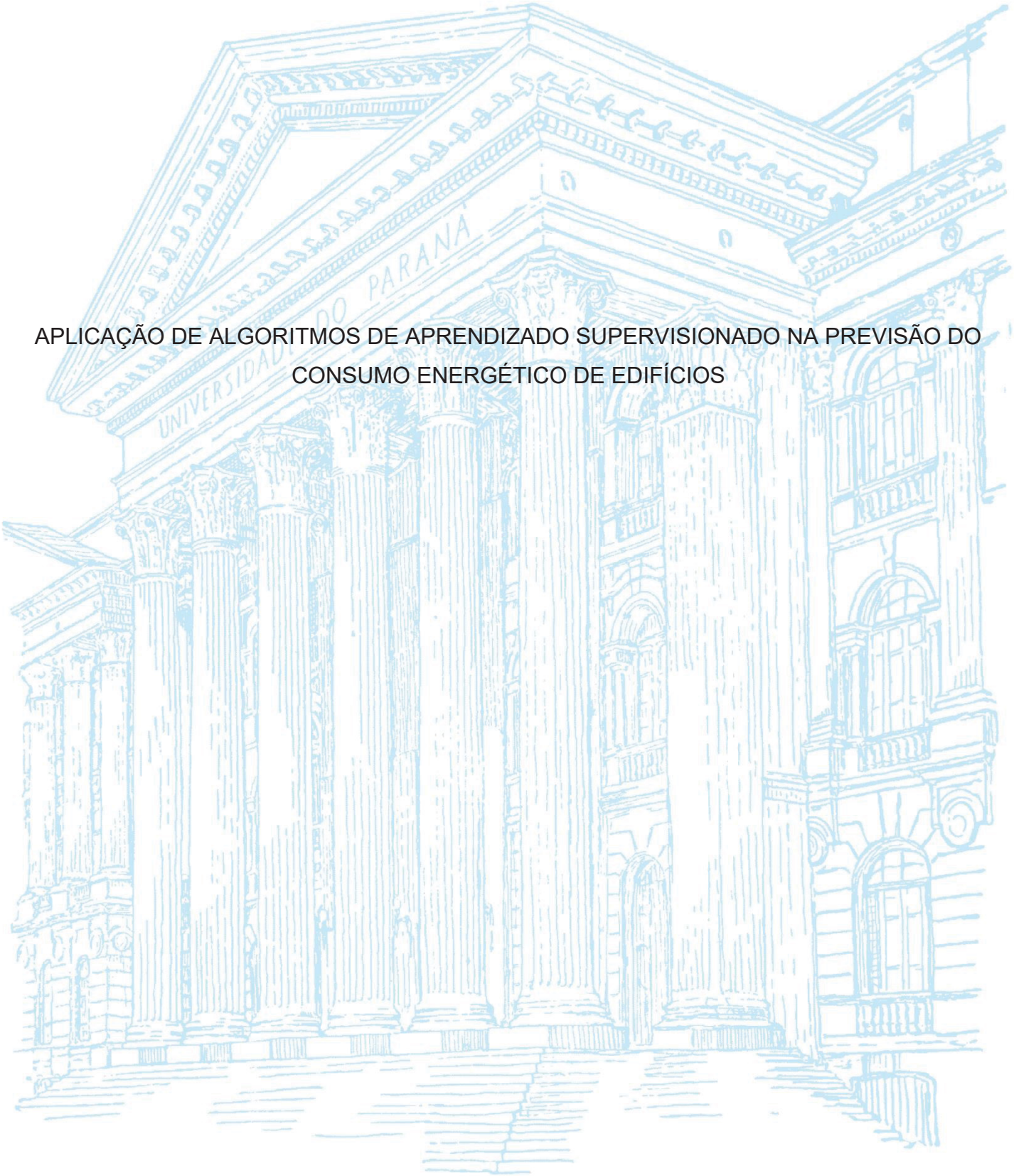
UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS DE SOUZA ALMEIDA



APLICAÇÃO DE ALGORITMOS DE APRENDIZADO SUPERVISIONADO NA PREVISÃO DO CONSUMO ENERGÉTICO DE EDIFÍCIOS

CURITIBA

2022

LUCAS DE SOUZA ALMEIDA

APLICAÇÃO DE ALGORITMOS DE APRENDIZADO SUPERVISIONADO NA PREVISÃO DO CONSUMO ENERGÉTICO DE EDIFÍCIOS

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador(a): Prof(a). Dr(a). Jaime Wojciechowski

CURITIBA

2022

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **LUCAS DE SOUZA ALMEIDA** intitulada: **APLICACAO DE ALGORITMOS DE APRENDIZADO SUPERVISIONADO NA PREVISAO DO CONSUMO ENERGETICO DE EDIFICIOS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 09 de Dezembro de 2022.

JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Aplicação de Algoritmos de Aprendizado Supervisionado na Previsão do Consumo Energético de Edifícios

Lucas de Souza Almeida
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
lucas.almeida1606@gmail.com

Dr. Jaime Wojciechowski
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jaimewo@ufpr.br

***Resumo*— Este estudo retrata a aplicação de métodos de aprendizado supervisionado na predição do consumo energético em edifícios. Dada a morfologia dos dados coletados, foi realizada uma etapa de pré-processamento, e na sequência foram testados todos os modelos de regressão disponíveis na biblioteca scikit-learn da linguagem python. Por fim o modelo ExtraTreeRegressor que apresentou maior coeficiente de determinação R^2 passa por uma otimização de hiperparâmetros e tem seu desempenho determinado em relação às métricas MAE, RMSE e R^2. Percebe-se que o desempenho do modelo é inferior em algumas categorias de consumo energético, e para essas é realizada a seleção e otimização de um novo modelo, o HistGradientBoostingRegressor, o qual tem seu desempenho em relação às métricas MAE, RMSE e R^2 comparado com o modelo treinado na primeira etapa. Percebe-se uma elevação de 12,22% no coeficiente de determinação R^2, e uma redução de 24,55 e 22,23 respectivamente nas métricas de erro MAE e RMSE.**

Palavras-chave—consumo energético, regressão, previsão.

Abstract*— *This study portrays the application of supervised learning methods in the prediction of energy consumption in buildings. Given the morphology of the collected data, a pre-processing step was carried out, all regression models available in the scikit-learn library of the python language were tested. Finally, the model* ExtraTreeRegressor *that presented the highest coefficient of determination R^2 undergoes a hyperparameter optimization and its performance is determined in relation to the MAE, RMSE and R^2 metrics. It is noticed that the performance of the model is inferior in some categories of energy consumption, and for these a new model* HistGradientBoostingRegressor *is selected and optimized, which has its performance in relation to the MAE, RMSE and R^2 metrics compared to the trained model. In the first stage. An increase of 12.22% can be seen in the coefficient of determination R^2, and a reduction of 24.55 and 22.23 respectively in the MAE and RMSE error metrics.

Keywords—energy consumption, regression, prediction.

I. DESENVOLVIMENTO

Como consequência do crescimento populacional observado nos grandes centros urbanos, há uma preocupação relacionada à crescente demanda por consumo energético de diversas fontes [1]. Por esse motivo a eficiência energética, sobretudo nos edifícios desses centros, é um tema que tem atraído atenção de pesquisadores e de instituições de administração pública em todo o mundo [2].

Uma das abordagens mais adotadas para solução desse problema é a execução de projetos de modernização em edifícios, os quais buscam, através da modernização, elevar a eficiência energética destes [3]. Porém, um dos problemas existentes para a viabilização desses projetos é a falta de modelos preditivos que quantifiquem assertivamente a economia de energia que o edifício obterá após sua conclusão. Isso ocorre pois prever o consumo energético de qualquer sistema com multivariáveis é extremamente complexo [3].

Desse modo, este trabalho consiste em comparar o desempenho de diferentes modelos supervisionados de regressão aplicados na previsão de séries temporais multivariadas relacionadas ao consumo energético de edifícios. Os dados utilizados pertencem a uma base pública fornecida pelo site Kaggle[1] em que estão disponíveis diversos dados climáticos, geográficos e morfológicos, bem como medições do consumo de energia de centenas de edifícios em 15 localidades ao redor do mundo. Foram realizadas medições de quatro formas de consumo de energia, as quais são: energia elétrica, aquecimento de água, resfriamento de água e energia consumida na geração de vapor. Segundo os mantenedores da base de dados, as medições foram realizadas com sensores físicos reais e não passaram por tratamento antes de sua publicação, portanto foi necessário realizar uma análise criteriosa buscando expurgar da base de dados as medições com possíveis erros ou omissões na medição.

A. Descrição dos dados

A base de dados analisada apresenta medições de consumo de energia em kWh de edifícios, realizadas a cada hora pelo período de um ano. Essas medições foram feitas em 1448 edifícios localizados em 16 cidades ao redor do mundo. As medições de consumo foram categorizadas por finalidade de consumo, sendo essas: energia elétrica, aquecimento de água, resfriamento de água e energia consumida na geração de vapor. Alguns edifícios não apresentam medições de todos os grupos de finalidade de consumo, e, em outros edifícios, é possível perceber inconsistências de medição por picos extremos no consumo, longos períodos com medição constante, ou períodos com medições nulas.

[1] https://www.kaggle.com/competitions/ashrae-energy-prediction/overview

A base de dados é dividida em três tabelas, são elas: *building_meta*, *weather_train* e *train_data*, as quais demonstram respectivamente: dados relativos à morfologia dos edifícios onde foram realizadas as medições, dados climáticos medidos em uma estação meteorológica mais próxima ao edifício, e medições de consumo de energia propriamente dita. As tabelas: Tabela I, Tabela II e Tabela III apresentam as descrições dos campos de dados que compõem a base. A tabela *building_meta* contém o total de 1.449 linhas e seis colunas, a tabela *weather_train* possui 139.773 linhas e nove colunas, e a tabela *train_data* o total de 20.216.100 linhas e quatro colunas.

B. Métodos

O método adotado para obtenção do modelo de previsão para o consumo de energia dos edifícios é ilustrado na Figura 1. Este tem como objetivo preparar os dados para treinamento e selecionar o melhor modelo para realização de predições, e é dividido em seis etapas: 1) preparação da base de dados; 2) pré-processamento; 3) seleção do melhor modelo; 4) otimização de hiperparâmetros; 5) predição; 6) análise de resultados e métricas de performance. Também é importante destacar o uso da biblioteca scikit-learn [2] e pandas [3] da linguagem Python. A biblioteca pandas apresenta a implementação de vários métodos e ferramentas de manipulação e transformação de dados, já a biblioteca scikit-learn fornece diversos métodos de pré-processamento de dados, treinamento de modelos de aprendizado de máquina para regressão, classificação e agrupamento, bem como seleção de testes e validação de modelos. Portanto, são ferramentas fundamentais para a construção deste trabalho, e a utilização de seus métodos é citada diversas vezes durante o texto.

TABELA I

TRAIN_DATA – TABELA CONTENDO AS MEDIÇÕES DE CONSUMO DE ENERGIA DOS EDIFÍCIOS

Atributo	Descrição
building_id	Chave estrangeira para a base de dados *building*_meta
meter	Tipo de consumo energetico 0: eletricidade, 1: resfriamento de água, 2: vapor, 3: aquecimento de água
timestamp	Medida de tempo em que ocorreu a medição no formato: "*ano-data-mês hora:minuto:Segundo*"
meter_reading	Variável correspondente às medidas de consumo de eletricidade em kWh

TABELA II

BUILDING_META – TABELA CONTENDO OS DADOS PREDIAIS E MORFOLÓGICOS DOS EDIFÍCIOS

Atributo	Descrição
site_id	Chave Estrangeira para a base de dados *wather_train*
building_id	Chave estrangeira para a base de dados *building*_meta
primary_use	Indica a principal atividade para qual o edifício é utilizado
square_feet	Área bruta total do edifício em ft^2
year_built	Ano de construção do edificiof
floor_count	Número de pavimentos do edificiof

TABELA III

WEATHER_TRAIN – TABELA CONTENDO OS DADOS CLIMÁTICOS DAS LOCALIDADES DOS EDIFÍCIOS

Atributo	***Descrição***
site_id	Chave Estrangeira para a base de dados *wather_train*
air_temperature	Temperatura do Ar °C
cloud_coverage	Proporção do céu encoberto por nuvens, escala de 1 a 0
dew temperature	Temperatura do ponto de orvalho em °C
precip_depth_1_hr	Precipitação em mm
sea_level_pressure	Pressão a nível do mar mBar/10^4Pa
wind_direction	Direção do vento, medida angural de 1 a 360 graus
timestamp	Medida de tempo em que ocorreu a medição no formato: "*ano-data-mês hora:minuto:Segundo*"
wind_speed	Velocidade do vento em m/s

[2] https://scikit-learn.org/stable/

[3] https://pandas.pydata.org/docs/index.html

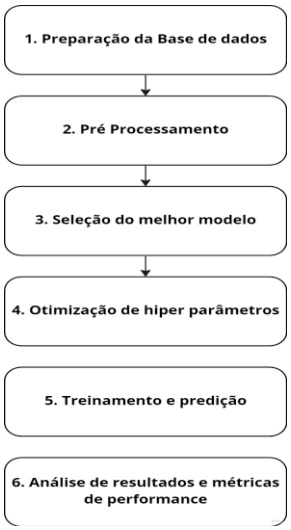


Fig. 1. Etapas do desenvolvimento do estudo

1) Composição da base de dados: visto que os dados coletados são divididos em três tabelas relacionadas cardinalmente entre si, realizou-se a junção destas para a formação de uma única tabela. As tabelas *building_meta* e *train_data* foram unidas por meio da junção esquerda com a coluna *'building_id'*, em seguida a tabela resultante dessa primeira junção também é submetida à junção esquerda com a tabela weather_train pelas colunas *'site_id'*, e *'time_stamp'*. Para realizar as junções, foi utilizado o método *merge*[4] da biblioteca sci-kit learn da linguagem Python. A tabela resultante apresentou 20.216.100 linhas e 16 colunas, totalizando 2.6 GB de memória.

2) Pré-processamento: a tabela resultante da preparação de dados inicialmente foi submetida a uma etapa de limpeza de dados. Primeiramente desprezaram-se as colunas com proporção de valores nulos superior a 40%, essas foram: *'year_built'*, *'floor_count'*, e *'coloud_coverage'*, com respectivamente 59.90%, 82.65 e 43.65% de valores nulos. O valor limite de 40% foi arbitrado pois a coluna com a quarta maior proporção de valores nulos *'precip_depth_1_hr'* apresentou apenas 18,54% de valores nulos, valor muito inferior à terceira colocada 'cloud_coverage', portanto tomou-se a decisão de excluir as três colunas com maior proporção de valores nulos.

Depois a coluna '*timestamp*' é decomposta e substituída por outras três novas colunas, sendo essas: *"dia"*, *"hora"* e *"mês"*. Para isso, inicialmente a coluna '*timestamp*' é convertida para o formato datetime64 com o emprego do método "to_datetime" da biblioteca, depois são extraídos os atributos: hour, day, e month da coluna '*timestamp*' para a criação das demais colunas citadas. Por último, a coluna '*timestamp*' é excluída.

Em seguida transformou-se a coluna categórica *'primary_use'* em numérica com emprego do método *LabelEncoder* [5] da biblioteca scikit-learn. Esse método substitui os valores categóricos de uma coluna por um número inteiro correspondente.

[4] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

Percebe-se que a coluna *'precip_depth_1_hr'*, a qual traz dados referentes à precipitação pluviométrica da localidade em milímetros, apresentam-se medições com valor -1. Tendo em vista que valores negativos são incoerentes com as medições apresentadas nesse campo, para estes foram atribuídos valor 0.

De acordo com [6], uma das condições para que uma variável seja empregada de maneira segura na realização de testes paramétricos é que ela apresente o princípio da normalidade, ou seja, que seu comportamento se assemelhe a uma distribuição gaussiana normal, porém com algum nível de variância. Visto que, segundo [7], a combinação da análise de assimetria e curtose pode ser útil para estimar a normalidade de uma série temporal; na Tabela IV são listados os valores de assimetria e Curtose de cada uma das colunas numéricas da base de dados. Evidencia-se que as colunas *'meter_reading'*, *'precip_depth_1_hr'* e *'square_feet'* apresentam valores bastante acentuados.

TABELA IV

CURTOSE E ASSIMETRIA DAS VARIÁVEIS NUMÉRICAS

Coluna	Assimetria	Curtose
meter_reading	104,81	11671,87
precip_depth_1_hr	19,01	510,52
square_feet	2,67	9,99
meter	1,18	0,21
wind_speed	0,82	1,16
primary_use	0,79	-0,19
dia	0,01	-1,19
hora	0,00	-1,20
mes	-0,3	-1,19
site_id	-0,04	-1,53
wind_direction	-0,07	-1,24
sea_level_pressure	-0,10	1,12
building_id	-0,31	-1,24
air_temperature	-0,37	-0,04
dew_temperature	-0,43	-0,33

Para essas foi aplicada a função logarítmica com o uso da função *log*[6] da biblioteca numpy[7]. A Tabela V retrata os valores de assimetria e curtose das colunas *'meter_reading'*, *'precip_depth_1_hr'* e *'square_feet'* após a aplicação da função.

TABELA V

ASSIMETRIA E CURTOSE DAS COLUNAS COM BAIXA NORMALIDADE

Coluna	*Assimetria*	*Curtose*
meter_reading	-0,28	-0,14
precip_depth_1_hr	5,22	29,24
square_feet	-0,78	1,10

As Figuras 2 e 3 demonstram os histogramas dessas três colunas antes e após a aplicação de função logarítmica. Nelas é possível perceber redução da curtose e assimetria das amostras elevando a sua normalidade. No eixo x é

[6] https://numpy.org/doc/stable/reference/generated/numpy.log.html

[7] https://numpy.org/

plotado o valor assumido em cada coluna, e no eixo y o número de ocorrências multiplicado por 10^7.

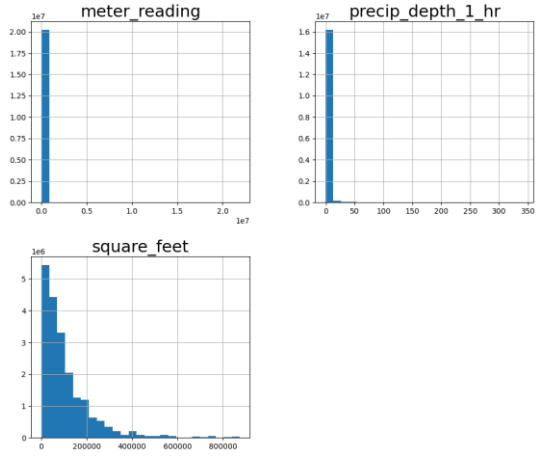


Fig. 2. Histograma das colunas antes da aplicação da função logarítimica

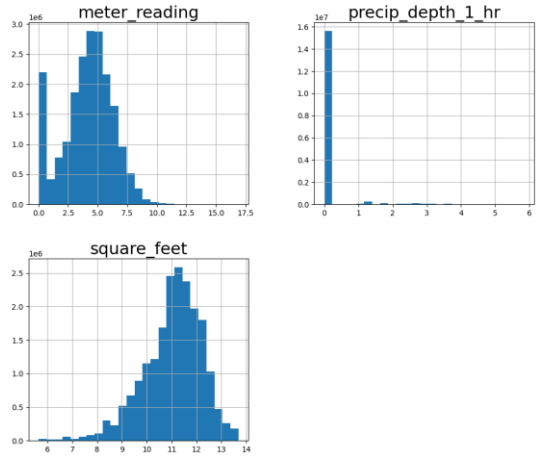


Fig. 3. Histograma da colunas após aplicação da função logarítimica

Em seguida foram tratados os valores nulos das colunas remanescentes. Para as colunas: *'air_temperature'*, *'dew_temperature'*, *'precip_depth_1_hr'*, *'wind_direction'*, *'wind_speed'* e *'sea_level_pressure'*, foram atribuídos seus respectivos valores médios, e para a variável objetivo *'meter_reading'*, considera-se a hipótese de que não são admitidos valores nulos, e portanto as linhas que os contêm são desprezadas para evitar influência destes no treinamento do modelo.

Depois é aplicado o método *SelectFromModel* [8] da biblioteca scikit-lean para selecionar as melhores variáveis para utilização no treinamento dos modelos. Esse método seleciona as variáveis que têm a maior importância durante o treinamento. A execução do método *SelectFromModel* retornou as colunas: *'building_id'*, *'meter'*, *'site_id'*, *'primary_use'* e *'square_feet'* como as de maior importância.

Foi também realizada uma análise da correlação de Pearson entre todas as variáveis da base de dados. O coeficiente de correlação de Pearson mede a proporção que variações entre uma variável interferem na outra, e é dado pela Equação 1:

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

Onde r é o coeficiente de correlação de Pearson, x_i é o valor de uma variável da qual se deseja calcular o coeficiente de correlação com uma segunda variável, $\bar{x}$ é a média da coluna, y_i é o valor da segunda variável, $\bar{y}$ é o valor da média da coluna y.

Para medir a correlação de Pearson, foi utilizada a função corr[9] da biblioteca pandas da linguagem python. A Figura 4 ilustra o coeficiente de correlação de Pearson que cada coluna tem com a variável objetivo *'meter_reading'*.

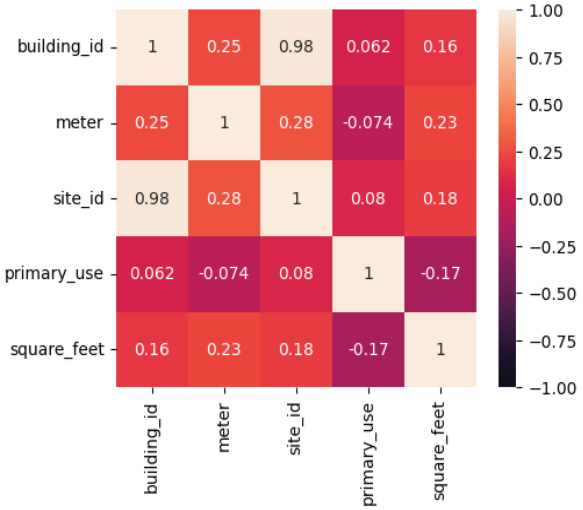


Fig. 4. Coeficiente de correlação de pearson das colunas selecinadas da base de dados entre si

De acordo com [8], quando duas variáveis apresentam uma correlação forte entre si, superior a 75%, uma delas deve ser desprezada durante o treinamento do modelo. Por isso, após analisar a correlação das colunas entre si, percebe-se que as colunas *'building_id'* e *'site_id'* possuem uma correlação de 0.98 entre si, portanto optou-se por desprezar a coluna '*building_id'*.

Na última fase da etapa de pré-processamento, aplicou-se a normalização da base de dados com a classe StandartScaler [10] da biblioteca scikit-learn, método de normalização que é realizado pela aplicação da Equação 2:

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

Onde z é o valor da variável normalizada, x é o valor da mesma variável anterior à normalização, μ é o valor da média de todos os valores da base de dados, e σ é o valor do desvio padrão da base de dados.

3) Seleção do melhor modelo: nessa etapa realizou-se uma avaliação de todos os modelos de aprendizado supervisionado por regressão disponíveis na biblioteca scikit-learn, cuja lista é obtida através do método

[8] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html

[9] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

all_estimator[11] da referida biblioteca. Esse método retorna uma lista de tuplas contendo os nomes e os métodos de todos os modelos de aprendizado de máquina da biblioteca; passando o argumento "regressor" na função *all_estimators*, são retornados apenas os métodos de regressão. Todos os modelos da lista retornada foram treinados com dados pré-processados hiperparâmetros em estado padrão e seus desempenhos avaliados utilizando a técnica de validação cruzada também da biblioteca scikit-learn *cross_val_score*[12] com três repartições. Nas etapas subsequentes do trabalho, os testes de validação cruzada são realizados com dez repartições, nesta, porém, buscando reduzir o tempo de execução dos algoritmos e evitar a ocorrência de falhas de execução por falta de memória ram, os testes são realizados em um número menor de repartições.

A métrica empregada na avaliação dos modelos é o coeficiente de determinação R^2 o qual é representado pela Equação 3:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \quad (3)$$

Onde x_i é o valor da variável a ser predita, $\hat{x}_i$ o valor da predição a cada ponto e $\bar{x}_i$ a média de todos os valores da coluna. Para o coeficiente de determinação, valores mais próximos de 1 retratam melhor ajuste do modelo em relação à variável de interesse.

A Tabela VI apresenta os valores obtidos durantes os testes de validação cruzada. Nessa são apresentados o R^2 médio das três medições, o máximo, o mínimo e também o tempo de execução de cada teste em segundos.

TABELA VI

COMPARAÇÃO DOS MODELOS DE REGRESSÃO

Algoritmo	*R^2 médio*	*R^2 Máximo*	*R^2 Mínimo*	*Tempo de Execução(s)*
ExtraTreesRegressor	0,7553	0,7983	0,7093	1303,57
ExtraTreeRegressor	0,7552	0,7979	0,7093	18,49
DecisionTreeRegressor	0,7548	0,7971	0,7093	27,66
BaggingRegressor	0,7548	0,7972	0,7094	203,61
HistGradientBoostingRegressor	0,6451	0,6817	0,5903	75,55
MLPRegressor	0,5748	0,6170	0,4970	1924,68
KNeighborsRegressor	0,5494	0,7280	0,3378	1269,69
GradientBoostingRegressor	0,5440	0,5707	0,4934	715,00
ARDRegression	0,3811	0,4196	0,3485	8,82
LarsCV	0,3809	0,4191	0,3485	23,61
LassoLarsCV	0,3809	0,4191	0,3485	18,77
BayesianRidge	0,3809	0,4190	0,3485	7,03
Lars	0,3809	0,4190	0,3485	5,68
LassoLarsIC	0,3809	0,4190	0,3485	7,90
LinearRegression	0,3809	0,4190	0,3485	6,30
HuberRegressor	0,3777	0,4166	0,3436	20,16
LinearSVR	0,3744	0,4143	0,3388	281,90
AdaBoostRegressor	0,3380	0,3807	0,2728	892,85
GammaRegressor	0,3044	0,3296	0,2789	6,71
ElasticNet	0,2160	0,2163	0,2154	5,31
Lasso	0,0503	0,0575	0,0362	6,43
DummyRegressor	-0,0011	0,0000	-0,0018	4,16
LassoLars	-0,0011	0,0000	-0,0018	5,29
DummyRegressor	-0,0010	0,0000	-0,0020	6,39
LassoLars	-0,0010	0,0000	-0,0020	218,12

[11] https://scikit-learn.org/0.21/modules/generated/sklearn.utils.testing.all_estimators.html

[12] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

O algoritmo *ExtraTreeRegressor*[13] obteve maior R^2 médio juntamente com os modelos *ExtraTreesRegressor*, *BaggingRegressor*. Tendo em vista que este obteve o menor tempo de execução durante o treinamento, ele foi escolhido para prosseguir para próxima etapa, em que é realizada otimização de seus hiperparâmetros.

4) Otimização de hiperparâmetros: nessa etapa o algoritmo *ExtraTreeRegressor* da biblioteca scikit-learn teve seus hiperparâmetros otimizados com o emprego da técnica *gridSearchCV*[14], também da biblioteca scikit-learn. Os parâmetros investigados foram:

- max_depth: [10, 25, 100, 250, 500];
- min_samples_split: [5, 15, 25];
- criterion: ['squared_error', 'friedman_mse'];
- splitter: ['random', 'best'];

A escolha desse conjunto de hiperparâmetros foi arbitrária, porém buscou-se manter uma amplitude de valores entre 200% acima e 200% abaixo dos valores padrão desse hiperparâmetro na documentação da biblioteca scikit-learn.

Empregando o método *best_params* da classe *gridSearchCV*[8] foi verificado que os melhores valores para os hiperparâmetros investigados foram: *max_depth* = 10; *min_samples_split* = 25; *criterion* = 'squared_error'; *spliter* = 'best'. Portanto, para a seção de análise de resultados, o modelo final foi treinado com a aplicação desses hiperparâmetros.

Por fim o modelo treinado foi avaliado por validação cruzada com outras métricas de erro além do coeficiente de determinação R^2. As métricas escolhidas foram RMSE (raiz quadrada do erro quadrático médio) e MAE (erro absoluto médio), as quais são dadas pelas Equações 4 e 5.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \quad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}} \quad (5)$$

Onde x_i é o valor da variável a ser predita, $\hat{x}_i$ o valor da predição a cada ponto e n o número de valores não nulos presentes na coluna.

C. Tecnologias

O código empregado na implementação deste estudo foi desenvolvido em um computador pessoal utilizando um processador Ryzen 5 3600 de 12 núcleos a 4.5 GHz, com sistema operacional Windows 10 Home. Todo código foi desenvolvido com linguagem Python versão 3.10.5 utilizando Jupyter Notebook versão 2022.9.1202862440. Nele foram empregadas as seguintes bibliotecas:

- Numpy, versão 1.23.2: empregado na realização de operações com vetores;

[13] https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html

[14] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

- Pandas, versão 1.4.3: empregado na manipulação de dados dentro de *dataframes*;
- *Scikit-learn*, versão 1.1.2: empregado no pré-processamento, na seleção, no treinamento e na validação dos modelos;
- *Sea Born*, versão 0.12.0: empregado na criação de gráficos.

II. Resultados e Discussões

Nesta seção são apresentadas as análises dos resultados obtidos com o método retratado na seção anterior.

A. *Métricas de Qualidade dos Modelos*

O modelo em sua forma final foi submetido à validação cruzada com dez repartições. Em cada uma dessas validações são medidas as métricas de erro: MAE, RMSE e o coeficiente de determinação R^2. A Tabela VII demonstra o resultado dos valores médio, máximo e mínimo para cada uma das métricas e para o coeficiente de determinação. Para a obtenção dos valores máximo, médio e mínimo de cada uma das métricas, são utilizadas as funções mean, max e min da biblioteca numpy, as quais retornam respectivamente o valor da média simples, maior e menor valor dos resultados obtidos pela validação cruzada em dez repartições.

TABELA VII

VALORES MÉDIO, MÁXIMO E MÍNIMO PARA CADA UMA DAS MÉTRICAS DE ERRO

Erro	Médio	Mínimo	Máximo
R^2	0,8037	0,7484	0,8498
RMSE	0,7821	0,6882	0,9194
MAE	0,5038	0,4350	0,5998

B. *Investigando Erro por Tipo de Consumo de Energia*

Visto que as medições de consumo de energia foram realizadas com sensores físicos reais, e foram aferidas quatro categorias distintas de consumo de energia sendo elas: "0 – Eletricidade", "1 – Refrigeração", "2- Vapor", e "3 - aquecimento de Água", considera-se a hipótese de que os dados coletados das diferentes categorias de consumo de energia podem apresentar comportamentos diferentes em relação aos resultados de suas predições.

Para iniciar a avaliação da influência da categoria de consumo de energia no desempenho do estimador, avalia-se a distribuição das medições contidas na base de dados em relação ao grupo de consumo. A Tabela VIII apresenta a proporção e o número de medições realizadas em cada categoria de consumo.

TABELA VIII

VALORES MÉDIO, MÁXIMO E MÍNIMO PARA CADA UMA DAS MÉTRICAS DE ERRO

Categoria de Consumo	N° de Medições	Proporção %
0 - Eletricidade	11530741	62,86%
1 – Resfriamento de água	3525936	19,22%
2 - Vapor	2361753	12,88%
3 – Aquecimento de água	923694	5,04%

Verifica-se que a proporção de medidas de consumo relacionadas à eletricidade, com 62,86% das medições, é substancialmente maior à das outras categorias; também é notório que o aquecimento de água, que corresponde a apenas 5,04% das medições, é proporcionalmente muito menor. Em razão dessa distribuição desproporcional, foi realizada validação do modelo em sua parametrização final, porém avaliando os resultados das predições exclusivamente para cada categoria de consumo de energia. Essa avaliação ocorrerá por validação cruzada com dez repartições, realizando treinamento do modelo com a base de dados segregada com apenas uma categoria de consumo por vez, avaliando as métricas de erro MAE, RMSE e coeficiente de determinação R^2.

As Figuras 5, 6 e 7 retratam gráficos com os resultados das métricas avaliadas:

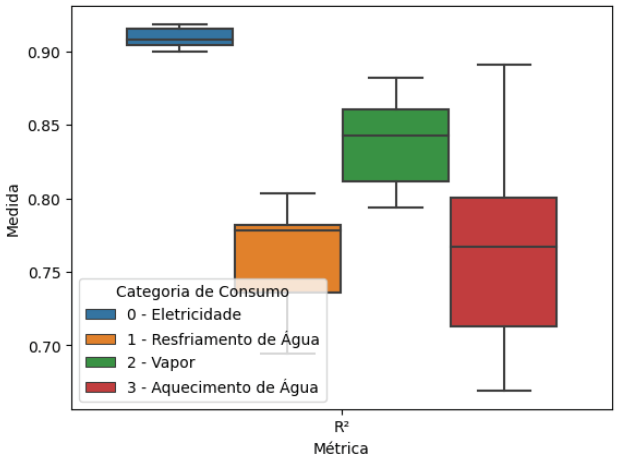


Fig. 5. Gráfico Boxplot de métrica R^2 das medições das categorias de consumo

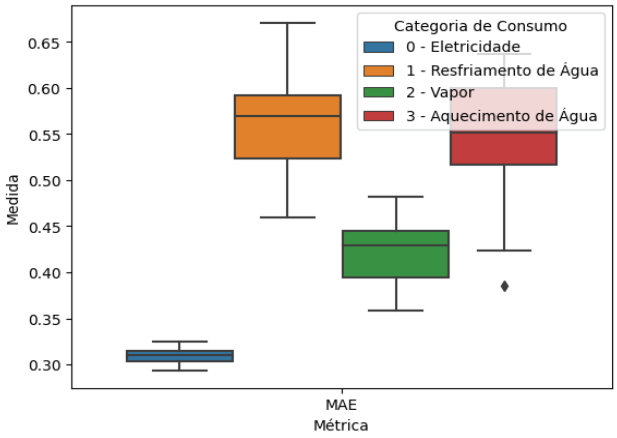


Fig. 6. Gráfico Boxplot de métrica MAE das medições das categorias de consumo.

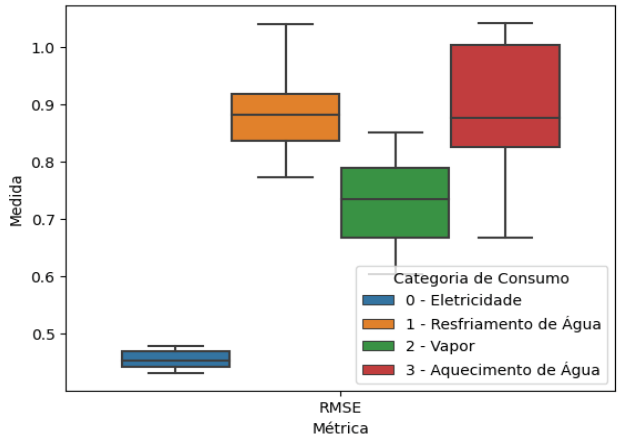


Fig. 7. Gráfico *Boxplot* de métrica RMSE das medições das categorias de consuimo.

Através dos gráficos é possível perceber que o modelo apresentou um desempenho substancialmente superior na predição das medições do grupo de consumo "0 – Eletricidade". Para os demais grupos de consumo, o desempenho verificado em todas as métricas encontra-se em um patamar parecido, e por mais que possa ser considerado satisfatório para realização de predições na aplicação desejada, com o desempenho substancialmente mais baixo. A Tabela IX demonstra o resultado dos erros médio, máximo e mínimo para as métricas de erro: MAE, RMSE e coeficiente de determinação R^2.

TABELA IX

VALORES MÉDIO MÁXIMO E MÍNIMO PARA CADA UMA DAS MÉTRICAS DE ERRO

Categoria de Consumo	*Métrica*	*MAE*	*RMSE*	R^2
0 - Eletricidade	Mínimo	0,2935	0,4308	0,8997
	Máximo	0,3245	0,4771	0,9183
	Médio	0,3089	0,4549	0,9090
1 - Resfriamento de água	Mínimo	0,4592	0,7732	0,6945
	Máximo	0,6706	1,0400	0,8032
	Médio	0,5625	0,8872	0,7614
2 - Vapor	Mínimo	0,3584	0,6034	0,7934
	Máximo	0,4821	0,8513	0,8817
	Médio	0,4246	0,7290	0,8376
3 - Aquecimento de água	Mínimo	0,3850	0,6668	0,6693
	Máximo	0,6371	1,0422	0,8912
	Médio	0,5392	0,8858	0,7676

Verifica-se que os valores médios dos Erros MAE e RMSE determinados são até 1,56 e 1.79 vezes maiores nas amostras de "1 - Resfriamento de água", "2 - Vapor", e "3 - Aquecimento de água" em relação ao grupo de consumo: "0 – Eletricidade". Também, analisando a variação entre os valores mínimo e máximo das métricas dentro de um mesmo grupo de consumo, é possível perceber que, para o grupo de consumo '0 – Eletricidade', a variação é extremamente pequena, na ordem de 2% para o coeficiente de determinação e de 5% para as métricas de erro RMSE e MAE. Porém, nas demais amostras, há uma variação de até 72,85%, como no caso do erro MAE no grupo de consumo "1 – Resfriamento de Água", e 141,84% para o coeficiente de determinação R^2 também no grupo de consumo "1 – Resfriamento de água".

Contudo, em casos em que os dados das medições disponíveis se encontrem corretamente categorizados de acordo com o tipo de consumo, ou qualquer outra categorização em que as predições demonstrarem comportamentos bastante diferentes a despeito de métricas de erros e coeficiente de determinação, aventa-se a possibilidade de que o treinamento e a otimização de modelos distintos para bases de dados segregadas por essas categorias podem entregar um melhor desempenho relacionado às métricas e ao coeficiente citado.

Por esse motivo realizou-se novamente o experimento da seção B3 (seleção do melhor modelo), onde é realizada validação cruzada com três repartições para todos os modelos de regressão da biblioteca scikit-learn para a base de dados segregada pelas categorias de consumo 1, 2 e 3.

Nas Tabelas X, XI e XII são apresentados os valores obtidos durante os testes de validação cruzada dos três modelos que obtiveram maior coeficiente de determinação médio em cada base de dados. Nessa são apresentados os valores R^2 médio, máximo e mínimo das três medições, além do tempo de execução de cada teste em segundos.

TABELA X

MELHORES MODELOS DE REGRESSÃO PARA GRUPO DE CONSUMO 1

Algoritmo	*R^2 médio*	*R^2 Máximo*	*R^2 Mínimo*	*Tempo de Execução(s)*
ExtraTreesRegressor	0,5822	0,6938	0,4738	58,83
DecisionTreeRegressor	0,5815	0,6919	0,4738	1,56
BaggingRegressor	0,5812	0,6921	0,4740	10,88

TABELA XI

MELHORES MODELOS DE REGRESSÃO PARA GRUPO DE CONSUMO 2

Algoritmo	*R^2 Médio*	*R^2 Máximo*	*R^2 Mínimo*	*Tempo de Execução(s)*
ExtraTreeRegressor	0,4907	0,6108	0,3509	2,13
BaggingRegressor	0,4912	0,6119	0,3511	22,11
HistGradientBoostingRegressor	0,4707	0,6052	0,3344	15,17

TABELA XII

MELHORES MODELOS DE REGRESSÃO PARA GRUPO DE CONSUMO 3

Algoritmo	*R^2 Médio*	*R^2 Máximo*	*R^2 Mínimo*	*Tempo de Execução(s)*
HistGradientBoostingRegressor	0,4753	0,6079	0,3599	4,321
BaggingRegressor	0,4744	0,6069	0,3552	5,475
DecisionTreeRegressor	0,4744	0,6070	0,3548	0,932

Verifica-se que, para o grupo de consumo 3, o modelo de regressão que apresentou maior coeficiente de

determinação foi *HistGradientBostingRegressor* [15]. Este também aparece na terceira colocação da base de dados segregada com as medições do grupo de consumo 1. Dessa maneira, além de apresentar um resultado interessante em todas as bases de dados, o modelo apresenta também o menor tempo de execução entre os algoritmos testados; portanto decide-se pelo prosseguimento da otimização dos seus hiperparâmetros e posteriormente a comparação dos seus resultados com os resultados obtidos pela validação cruzada do modelo *ExtraTreeRegressor* em sua parametrização final, apresentados na seção II A.

Nessa etapa são repetidos os passos da etapa de otimização de hiperparâmetros; em que o modelo *HistGradientBostingRegressor* teve seus hiperparâmetros otimizados com o emprego do método *gridSearchCV* da biblioteca scikit-learn. Os parâmetros investigados foram:

- max_iter: [250, 1000];
- max_leaf_nodes: [100 ,150, 200];
- min_samples_leaf: [1, 2, 5];
- max_bins: [150, 255, 300].

A escolha desse conjunto de hiperparâmetros foi arbitrária, porém buscou-se manter uma amplitude de valores entre 200% acima e 200% abaixo dos valores padrão desse hiperparâmetro na documentação da biblioteca scikit-learn. Empregando o método *best_params* da classe *gridSearchCV*, foi verificado que os melhores valores para os hiperparâmetros investigados foram: *max_iter* = 1000; *min_leaf_nodes* = 200; *min_samples_leaf* = 1; *max_bins* = 255. Portanto esta será a parametrização do modelo p*HistGradientBostingRegressor* durante a realização da avaliação do seu desempenho; foi realizada a mesma avaliação apresentada na seção II.B para o modelo *ExtraTreeRegressor*, porém com uma pequena modificação. Percebeu-se que o modelo *HistGradientBostingRegressor* não é compatível com o método *SelectFomModel* da biblioteca scikit-learn, pois ele não é compatível com a análise de importância de variáveis durante o treinamento. Por esse motivo a função de seleção de variáveis *SelectFromModel* foi substituída pela *SequentialSelector* [16] da biblioteca scikit-learn na direção foward. Após a etapa de seleção de melhores variáveis, foi realizada a validação cruzada com dez repartições, realizando treinamento do modelo com a base de dados segregada com apenas uma das categorias de consumo 1, 2 e 3 por vez, avaliando as métricas de erro MAE, RMSE e coeficiente de determinação R^2. A tabela XIII demonstra os resultados das métricas de erro e coeficiente de determinação do experimento.

TABELA XIII

VALORES MÉDIO, MÁXIMO E MÍNIMO PARA CADA UMA DAS MÉTRICAS DE ERRO

Categoria de Consumo	***Métrica***	***MAE***	***RMSE***	***R^2***
1 - Resfriamento de água	Mínimo	0,3417	0,5709	0,7825
	Máximo	0,5367	0,8734	0,8944
	Médio	0,4244	0,6900	0,8545
2 - Vapor	Mínimo	0,3368	0,5614	0,8151
	Máximo	0,4457	0,8042	0,9040
	Médio	0,3874	0,6799	0,8587
3 - Aquecimento de Água	Mínimo	0,3206	0,6048	0,7607
	Máximo	0,5652	0,9082	0,9070
	Médio	0,4449	0,7571	0,8309

As tabelas XIV, XV e XVI demonstram a comparação entre os valores mínimo, máximo e médio das métricas de erro e coeficiente de determinação entre os modelos *ExtraTreeRegressor* e *HistGradientBostingRegressor* obtidas através da validação cruzada em dez repartições. As colunas **Δ%** apresentam a variação percentual entre os valores obtidos entre os dois modelos.

Os resultados obtidos denotam que houve uma expressiva elevação do coeficiente de determinação, e uma redução substancial das métricas de erro em cada uma das categorias de consumo e energia. Para a métrica de erro médio absoluto MAE, foi obtida uma redução média de -15,32% entre todas as medidas das categorias de consumo, para a raiz do erro quadrático médio RMSE houve uma redução média de -13,37, e o coeficiente de determinação R^2 apresentou uma elevação média de 7,52%. Portanto conclui-se que a seleção do modelo *HistGradientBostingRegressor* e otimização de seus hiperparâmetros e seleção de variáveis de treinamento para realização de predições dos dados das categorias de consumo 1, 2 e 3 apresentou um desempenho melhorado em relação ao modelo *ExtraTreeRegressor* em termos das métricas de erro e coeficiente de determinação avaliados.

TABELA XIV

COMPARAÇÃO DOS VALORES MÉDIO, MÁXIMO E MÍNIMO DA MÉTRICA DE ERRO MAE OBTIDOS COM OS DOIS MODELOS

Categoria de Consumo	***Métrica***	***MAE ExtraTrees***	***MAE HistBoost***	***Δ%***
1 - Resfriamento de água	Mínimo	0,4592	0,3417	-25,58%
	Máximo	0,6706	0,5367	-19,96%
	Médio	0,5625	0,4244	-24,55%
2 - Vapor	Mínimo	0,3584	0,3368	-6,03%
	Máximo	0,4821	0,4457	-7,55%
	Médio	0,4246	0,3874	-8,77%
3 - Aquecimento de Água	Mínimo	0,3850	0,3206	-16,72%
	Máximo	0,6371	0,5652	-11,28%
	Médio	0,5392	0,4449	-17,48%

[15] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html

[16] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

TABELA XV

COMPARAÇÃO DOS VALORES MÉDIO, MÁXIMO E MÍNIMO DOS COEFICIENTES DE DETERMINAÇÃO R^2 OBTIDOS COM OS DOIS MODELOS

Categoria de Consumo	***Métrica***	***R^2 ExtraTrees***	***R^2 HistGradient***	*Δ%*
1 - Resfriamento de água	Mínimo	0,6945	0,7825	12,67%
	Máximo	0,8032	0,8944	11,35%
	Médio	0,7614	0,8545	12,22%
2 - Vapor	Mínimo	0,7934	0,8151	2,74%
	Máximo	0,8817	0,9040	2,52%
	Médio	0,8376	0,8587	2,51%
3 - Aquecimento de Água	Mínimo	0,6693	0,7607	13,66%
	Máximo	0,8912	0,9070	1,77%
	Médio	0,7676	0,8309	8,25%

TABELA XVI

COMPARAÇÃO DOS VALORES MÉDIO, MÁXIMO E MÍNIMO DOS VALORES DE ERRO RMSE OBTIDOS COM OS DOIS MODELOS

Categoria de Consumo	***Métrica***	***RMSE ExtraTrees***	***RMSE HistBoost***	*Δ%*
1 - Resfriamento de água	Mínimo	0,7732	0,5709	-26,17%
	Máximo	1,0400	0,8734	-16,02%
	Médio	0,8872	0,6900	22,23%
2 - Vapor	Mínimo	0,6034	0,5614	-6,96%
	Máximo	0,8513	0,8042	-5,53%
	Médio	0,7290	0,6799	-6,74%
3 - Aquecimento de água	Mínimo	0,6668	0,6048	-9,30%
	Máximo	1,0422	0,9082	-12,86%
	Médio	0,8858	0,7571	-14,53%

C. Considerações Finais

Analisando as métricas de erro e coeficiente de determinação obtidas por validação cruzada com a base de dados completa e pré-processada, obteve-se a percepção de que o modelo *ExtraTreeRegressor*[7] em sua parametrização final apresenta um desempenho satisfatório para realização das predições na aplicação pretendida. Porém, após análise específica para cada categoria de consumo de energia, nota-se um desempenho muito superior para o grupo de consumo '0 – Eletricidade', o qual também tem um maior número de medições na base de dados. Nos demais grupos de consumo, o coeficiente de determinação médio R^2 obtido demonstra um ajuste menos aderente das previsões à validação, e os erros MAE e RMSE são substancialmente mais elevados. Por esse motivo, buscou-se validar a hipótese de que a seleção e otimização de um novo modelo, treinado especificamente com os dados das medições associadas às categorias de consumo em que o modelo inicial obteve um desempenho inferior, poderia prover predições mais acuradas. Realiza-se a seleção de variáveis para treinamento, e a otimização de hiperparâmetros de um novo modelo *HistGradientBostingRegressor*[9]. Neste, após a aplicação do mesmo método de seleção, otimização, e validação a qual o primeiro modelo foi submetido, foi percebida uma melhoria bastante representativa das métricas de erro e coeficiente de determinação investigados.

Por esse motivo foi entendido que os modelos de aprendizado de máquina, sobretudo quando aplicados na realização de regressões, como é o caso do estudo, podem apresentar diferentes desempenho em métricas de erros e coeficiente de determinação quando comparados a diferentes categorias de uma mesma base de dados. Em casos em que esse comportamento for detectado no modelo de aprendizado de máquina, percebe-se que, com o treinamento e a otimização de modelos especificamente para cada categoria que apresente um desempenho inferior, é possível obter uma melhoria.

D. Trabalhos Futuros

A pretensão inicial do trabalho consistiu em realizar a seleção e otimização de um único algoritmo de regressão para obter as previsões de consumo de energia. Porém ao longo do desenvolvimento percebeu-se que o modelo otimizado apresentou um desempenho expressivamente desigual entre as categorias de consumo de energia; além do desempenho desigual, verificou-se que a base estava desbalanceada em suas medições. Portanto utilizamos a abordagem de realização do treinamento de um novo modelo e avaliação de seu desempenho para as classes em que o primeiro modelo teve pior desempenho. Para trabalhos futuros sugerimos que sejam realizadas abordagens diferentes para lidar com o desbalanceamento da base de dados e para mitigar uma possível diferença expressiva no desempenho do modelo nas diferentes classes de uma base de dados.

III. Referências

[1] K. Saidi, and S. Hammami, The impact of CO2 emissions and economic growth on energy consumption in 58 countries, *Energy Reports*, vol. 1, pp. 62-70, 2015.

[2] J. C. Lam, K. K, Wan, C. L. Tsang, and L. Yang, Building energy efficiency in different climates, *Energy Conversion and Management*, vol. 49, n. 8, pp.2354-2366, 2008.

[3] L. Costa-Carrapiço, R. Raslan, and J. N. González, A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency, *Energy and Buildings*, vol. 210, p.109690, 2020.

[4] S. Chan, I. Oktavianti, and V. Puspita, A deep learning cnn and ai-tuned svm for electricity consumption forecasting: multivariate time series data, in IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) pp. 0488-0494, 2019.

[5] S. Papadopoulos, E. Azar, W. L. Woon, and C. E. Kontokosta, Evaluation of tree-based ensemble learning algorithms for building energy performance estimation, *Journal of Building Performance Simulation*, vol. 11, no. 3, pp.322-332, 2018.

[6] S. García, J. Luengo, and F. Herrera, Data preprocessing in data mining, Cham, Switzerland: Springer International Publishing, vol. 72, pp. 59-139, 2015.

[7] J. Bai, and S. Ng, Tests for skewness, kurtosis, and normality for time series data, *Journal of Business & Economic Statistics*, vol. 23, no. 1, pp.49-60, 2005.

[8] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to linear regression analysis. *John Wiley & Sons*, 2021.