SEMANA ACADÊMICA
REVISTA CIENTÍFICA
ISSN 2236-6717

USO DA TÉCNICA DE BOOSTING PARA PREVISÃO DE CONSUMO DE COMBUSTÍVEL DE UMA FROTA DE NAVIOS

[ver artigo online]

Jorge Luiz do Carmo[1]

RESUMO

A previsão do consumo de combustível é de grande importância para o transporte marítimo, pois permite uma gestão de combustível mais eficiente, sendo uma excelente ferramenta para auxiliar na tomada de decisões no que se refere a medidas operacionais para melhoria da eficiência energética da frota. As empresas ligadas ao transporte marítimo estão cada vez mais pressionadas a reduzir o custo das viagens e ao mesmo tempo aumentar a eficiência energética e reduzir as quantidades de emissões de gases do efeito estufa. O objetivo deste artigo é apresentar modelos para prever o consumo de óleo combustível de uma frota de navios, com baixos erros de previsão, usando a técnica de aprendizado de máquina (*machine learning*) *boosting*.

Palavras-chave: *Boosting, Adaboost*, XGBoost, previsão de consumo de combustível, aprendizado de máquina

USE OF BOOSTING TECHNIQUE TO FORECASTING FUEL CONSUMPTION OF A VESSELS FLEET

ABSTRACT

Forecasting fuel consumption is of great importance for shipping, as it allows for more efficient fuel management, being an excellent tool to assist in decision making regarding operational measures to improve the energy efficiency of the fleet. Companies linked to shipping are increasingly under pressure to reduce the cost of travel while increasing energy efficiency and reducing the amount of greenhouse gas emissions. The purpose of this paper is to present models to predict the fuel oil consumption of a fleet of vessels, with low forecast errors, using the machine learning technique boosting.

Keywords: Boosting, Adaboost, XGBoost, forecasting of fuel oil consumption, machine learning

1 Analista de Transporte Marítimo da Petrobras. Graduado em Engenharia Mecânica pela Pontifícia Universidade Católica de Minas Gerais – PUC/MG. Mestrando em Engineering Management pela International University of Applied Sciences. Mestre em Comércio Internacional pela Universidad Isabel I de Castilla. Pós-graduado em Gerenciamento de Projetos pela Universidade Cândido Mendes. MBA Executivo em Administração de Negócios pelo Ibmec. E-mail: jorgedocarmo58@yahoo.com.br

http://dx.doi.org/10.35265/2236-6717-212-9357
FORTALEZA-CE. EDIÇÃO 212. V.9. ANO 2021.

INTRODUÇÃO

Os gastos com combustível marítimo, também conhecido como *bunker*, representam uma parcela considerável dos custos operacionais de uma frota de navios. Vários autores divergem sobre a representatividade destes custos com relação ao total dos custos operacionais. Para Stopford (2009), os custos com bunker representam cerca de 47% do total dos custos operacionais. Conforme Rodrigue (2021), o transporte marítimo é muito sensível aos custos dos combustíveis, que representam entre 45 e 50% dos custos operacionais. Um estudo do *The Oxford Institute for Energy Studies* (2018) aponta que o custo do bunker é um fator chave para os operadores de navios e pode representar entre 60 e 80% dos custos operacionais. De acordo com *World Shipping Council* (2008), os custos com combustíveis em uma empresa de navegação estão estimados entre 50% e 60% dos custos operacionais.

Prever o consumo de combustível de um navio não é uma tarefa fácil. O consumo de combustível depende do tipo do navio, características do projeto, performance do sistema de propulsão, velocidade de navegação, peso total da carga e condições de navegação como ventos, alturas das ondas, marés, correntes marítimas e correntezas. A previsão do consumo é de grande importância para o transporte marítimo, pois permite uma gestão de combustível mais eficiente. Pode ser uma excelente ferramenta para auxiliar na tomada de decisões no que se refere a medidas operacionais para melhoria da eficiência energética da frota. As empresas ligadas ao transporte marítimo estão cada vez mais pressionadas a reduzir o custo das viagens e ao mesmo tempo aumentar a eficiência energética e reduzir as quantidades de emissões de gases do efeito estufa.

São vários os combustíveis marítimos. Os principais são: óleo combustível, óleo diesel marítimo, gás liquefeito de petróleo e gás natural liquefeito. O objetivo deste artigo é apresentar modelos de aprendizado de máquina supervisionados, baseados nos algoritmos de *boosting*, para prever o consumo de óleo combustível de uma frota de navios petroleiros, com baixos erros de previsão.

1. BOOSTING

Ensemble é o nome dado à técnica de aprendizado de máquina (*machine learning*, em inglês) que combina o resultado de múltiplos modelos para obter um melhor modelo de predição.

É chamado de *boosting* qualquer método *ensemble* que use aprendizes fracos para transformá-los em um aprendiz forte (Géron, 2017). Como descrevem Marques et al. (2012), um classificador *ensemble* combina decisões de um conjunto de classificadores de base treinados individualmente.

Algoritmos de *boosting* se tornaram muito populares por atuarem como classificadores e regressores. Segundo Freund e Schapire (1996), o *boosting* visa melhorar a performance de qualquer algoritmo de aprendizado de máquina.

O *boosting* parte de um modelo básico, chamado de aprendiz fraco. A cada iteração, o algoritmo de *boosting* se aprimora, filtrando os resultados corretos e focando naqueles em que o modelo não conseguiu prever corretamente. A ideia é treinar os preditores em sequência para que eles corrijam os predecessores. Vários classificadores de baixa performance são combinados para criar um classificador melhor.

1.1. O ALGORITMO ADABOOST

O Adaboost (de *Adaptive Boosting*, em inglês) é um algoritmo meta-heurístico criado por Yoav Freund e Robert Schapire em 1995, que usa algoritmos base para comparar dados, reconhecer padrões e realizar classificações. Atualmente o Adaboost pode ser usado em classificação ou regressão. O algoritmo é treinado usando classificadores fracos em repetidas iterações. Em cada iteração ele identifica as fraquezas da anterior para gerar resultados satisfatórios no final. A abordagem adaptativa é usada para combinar os classificadores fracos, usando os erros destes, com base nas iterações anteriores, para fazer ajustes e obter um classificador final muito mais forte do que qualquer um dos classificadores isoladamente. Ao se concentrar nos casos difíceis, o classificador aprende com os erros e melhora as futuras análises. O classificador final é chamado de classificador forte, com melhor acuracidade e capacidade de predição.

O algoritmo precisa ser treinado para fazer as predições. Um algoritmo base será analisado repetidamente com base em pesos atribuídos. Inicialmente, cada classificador terá um peso associado distribuído de forma uniforme. Para minimizar o erro e aprender, ao final de cada rodada o algoritmo base estabelecerá uma hipótese de acordo com os pesos determinados. Os pesos dos registros classificados de forma incorreta são alterados, sendo identificados mais facilmente na próxima iteração. Quanto maior for o erro, maior será o peso associado ao classificador, ou seja, os dados classificados incorretamente terão peso maior do que aqueles classificados corretamente. Antes de realizar uma nova iteração, os dados que foram classificados incorretamente são reavaliados e ponderados por seus pesos aumentados. A cada rodada a rotina se repete até que o Adaboost possa combinar todas as hipóteses intermediárias para formular uma hipótese final. Concluído o treinamento do algoritmo, os parâmetros e coeficientes usados são armazenados para classificar novos dados.

1.2. O ALGORITMO GRADIENT BOOSTING

O *Gradient Boosting* usa os erros residuais do preditor anterior para ajustar o novo preditor, podendo ser usado em classificações ou regressões. Segundo Kuhn e Johnson (2013), o principal objetivo do algoritmo é minimizar a função de perda. Como explica Gong et al. (2017) trata-se de uma excelente ferramenta para construir modelos de predição que usa comumente árvores de regressão como aprendiz fraco.

Diferentemente do Adaboost, que usa os pesos a cada iteração para ajustar o aprendiz fraco, o *Gradient Boosting* usa repetidamente os padrões dos resíduos para fortalecê-lo, baseando-se no conceito de que ao melhorar o próximo modelo, combinado com os modelos anteriores, irá minimizar o erro geral. O *Gradient Boosting* minimiza a função de perda dos aprendizes fracos.

Em termos de funcionamento, o *Gradient Boosting* é parecido com o Adaboost. Inicialmente, o primeiro modelo faz o melhor palpite para a resposta, mas de forma simples. Ao comparar o valor observado frente ao previsto, tem-se o erro residual. Após treinar os próximos modelos com base nesse erro residual, o modelo tenta prever o erro encontrado no primeiro modelo, corrigindo-o. A cada iteração o algoritmo minimiza os erros residuais. As funções de perdas são otimizadas para redução dos erros.

2. MÉTODO

Para o estudo foram escolhidos 300 navios petroleiros transportando petróleo e seus derivados na cabotagem no Brasil, entre 66 locais de carga e 57 locais de descarga.

Foram usados 11 arquivos, em formato xls, relativos a viagens no período de 01/01/2011 a 31/12/2020. Os arquivos foram unidos em um só para formar a base de dados. A base de dados inicial era composta de 18 colunas correspondendo a diferentes informações sobre as viagens e características dos navios, sendo que cada linha representa uma viagem.

Como a qualidade dos atributos (*features,* em inglês) interfere no resultado da predição, o banco de dados foi previamente tratado para melhoria do desempenho. Foram escolhidas as *features* que melhor se adequam ao problema, com o objetivo de melhorar a precisão do modelo. As features que não faziam sentido foram removidas. Cada *feature* adicionada ao modelo pode aumentar sua complexidade, fazendo com que se ajuste muito aos dados de treinamento, fornecendo resultados insatisfatórios quando usando os dados de teste.

Os dados faltantes foram completados ou excluídos. Foi usada a biblioteca Pandas (2021) para limpar o banco de dados, identificar e remover valores faltantes, remover *features* e ajustar todas as colunas ao formato correto. A situação da base de dados antes e após o pré-processamento é mostrada na figura 1.

Figura 1 – Situação da base de dados antes e após o pré-processamento

BASE DE DADOS	ANTES DO PRÉ-PROCESSAMENTO	APÓS O PRÉ-PROCESSAMENTO
Número de linhas	37.439	33.423
Total de features	18	12
Número de variáveis categóricas	4	4
Número de variáveis numéricas	11	5
Número de variáveis de tempo	2	2
Número de features target	0	1
Percentual de valores faltantes	3,1%	0%

Fonte: Elaborado pelo autor.

ISSN 2236-6717

Como os dados estão em formato de tabela, cada *feature* corresponde a uma coluna. As seguintes *features* foram selecionadas para compor a base de dados:

- Navio: é o nome da embarcação, tal qual foi registrada.
- Porte bruto: é a soma dos pesos, em toneladas, que o navio é capaz de embarcar com segurança, considerando todos os pesos variáveis como carga, combustíveis, suprimentos, tripulação, água, etc.
- Classe: classificação do navio de acordo com o produto transportado e o porte bruto. As classes dos navios considerados são: handy, LR1, LR2, MR1, MR2, panamax, aframax, suezmax, VLCC, SGC, MGC e VLGC.
- Potência: medida da potência do motor principal do navio em hp (horse-power, em inglês).
- Local de origem: Local onde o navio iniciou a viagem, podendo ser porto, plataforma, monoboia, quadro de boias, FSO, FPSO, fundeadouro ou área de STS.
- Local de destino: Local onde o navio terminou a viagem, podendo ser porto, plataforma, monoboia, quadro de boias, FSO, FPSO, fundeadouro ou área de STS.
- Data e horário de saída: data e horário nos quais o navio deixou o local de origem.
- Data e horário de chegada: data e horário nos quais o navio chegou ao local de destino.
- Combustível na saída: quantidade de óleo combustível disponível para propulsão, em toneladas métricas, no momento da saída do navio do local de origem.
- Combustível na chegada: quantidade de óleo combustível disponível para propulsão, em toneladas métricas, no momento da chegada do navio ao local de destino.
- Milhas viajadas: milhas náuticas percorridas pelo navio na viagem entre os locais de origem e destino.
- Combustível consumido: quantidade de óleo combustível consumido pelo navio, em toneladas métricas, na viagem entre os locais de origem e destino.

As *features* foram selecionadas como dados de entrada, ou seja, como *features* preditoras. A *feature target* é a informação que queremos prever, no caso, o consumo de combustível.

A validação cruzada (*cross validation*, em inglês) é uma técnica de reamonstragem usada para avaliar a capacidade de generalização de determinado modelo. Em suma, ela serve para medir a precisão de um modelo para predição em um conjunto de dados nunca usado anteriormente. Para tal, o conjunto de dados é dividido em uma parte de treinamento e outra de teste. A parte de treinamento é dividida em k subconjuntos, onde k-1 representam os dados de treinamento para estimar os parâmetros, e o restante para avaliar a performance. A repetição do processo garante que todos os subconjuntos participem do treinamento e validação do modelo. A divisão do conjunto garante que o modelo treine, diminua o seu erro e generalize de forma satisfatória.

O treinamento dos modelos serve para ajustá-los, com objetivo de minimizar as funções de perda. A cada ajuste espera-se que o modelo melhore seu desempenho. Entretanto, há um ponto em que ajustar o modelo não melhorará sua performance. O excesso de ajuste, ou sobreajuste (*overfitting*, em inglês) fará com que o modelo performe bem melhor nos dados de treinamento do que quando fazendo previsões com os dados de teste. Um modelo com *overfitting* será ineficaz quando apresentados a dados nunca antes vistos.

"*Overfitting* acontece quando o modelo é muito complexo em relação à quantidade e ruído dos dados de treinamento." (Géron, 2017, p. 27, tradução nossa).

Ainda segundo Géron (2017), o subajuste (*underfitting*) é o oposto do sobreajuste, e acontece quando o modelo é muito simples para ter boa performance com os dados de treinamento ou com os dados de teste.

A representação gráfica do *underfitting, overfitting* e do melhor ajuste é mostrada na figura 2.

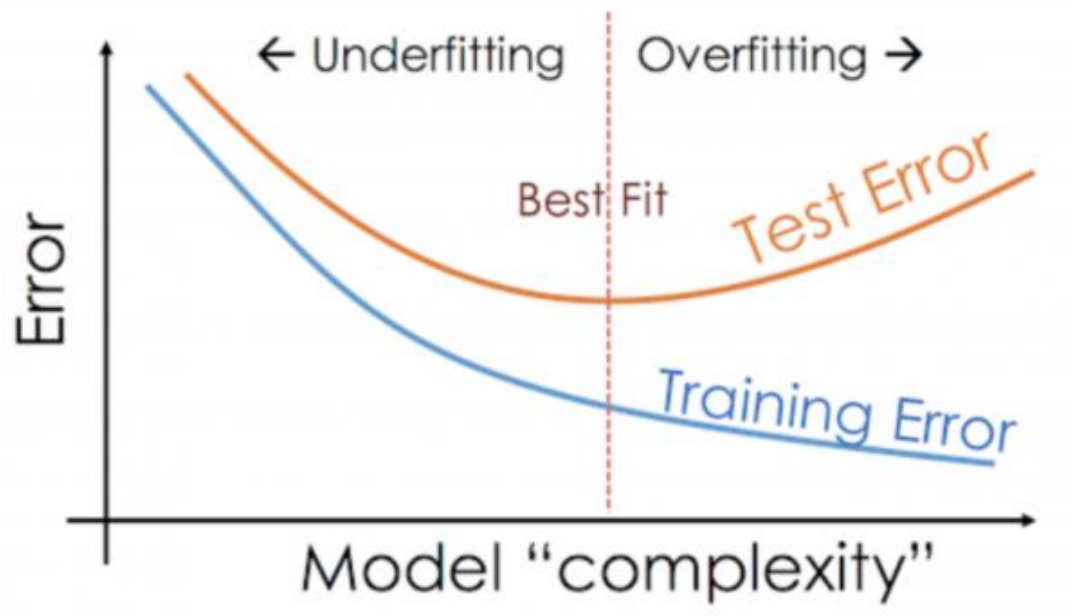
Através da validação podemos verificar se o modelo não está sobreajustado e estimar a performance do modelo.

Nesse estudo foi usada validação cruzada k-fold, com k=10, para dividir os dados de treinamento e teste em 10 partes de tamanhos iguais, ou próximos. A cada iteração, 9 partes foram usadas no treinamento do modelo, com diferentes configurações – os hiperparâmetros. A outra parte restante foi usada para estimativa da performance.

Cada algoritmo tem um conjunto de parâmetros que precisam ser ajustados antes do treinamento para maximizar o potencial de predição. Eles são chamados de hiperparâmetros. A

escolha deles foi feita usando a biblioteca Scikit-Learn (2021). A otimização é feita para minimizar os erros dos modelos entre os dados de treinamento e de teste. Antes dos ajustes ambos os modelos foram rodados com as configurações básicas para posterior comparação dos modelos com hiperparâmetros ajustados.

Figura 2 – *Underfitting* x *Overfitting*



Fonte: Analytic Vidhya (2021)

Os hiperparâmetros foram ajustados individualmente, pois ajustar todos ao mesmo tempo acarretaria um altíssimo custo computacional.

3. MÉTRICAS DE AVALIAÇÕES DOS MODELOS

A variável que se pretende prever, o consumo de óleo combustível, é contínua. Sendo assim, as performances dos modelos foram avaliadas pelas métricas Erro Absoluto Médio, MAE (*mean absolute error*, em inglês), Raiz do Erro Quadrático Médio, RMSE (*root mean squared error*, em inglês) e Coeficiente de Determinação, R2, (*R-Squared*, em inglês).

ISSN 2236-6717

$$\mathbf{MAE} = \frac{\mathbf{1}}{\mathbf{n}}\sum_{\mathbf{i=1}}^{\mathbf{n}}|\mathbf{y_i} - \mathbf{x_i}| \quad (1)$$

$$\boldsymbol{RMSE} = \frac{\mathbf{1}}{\mathbf{n}}\sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}}|\mathbf{y_i} - \mathbf{x_i}|^{\mathbf{2}}} \quad (2)$$

Onde:

n = número de observações;

y_i = valor real;

x_i = valor previsto.

O MAE mede a média aritmética dos desvios entre os valores reais e os previstos. O RMSE mede a raiz quadrada dos erros entre eles. Ambas métricas são calculadas na variável de interesse, no caso, em toneladas métricas de óleo combustível consumido. Quanto menores forem os valores, menores serão as diferenças entre os consumos previstos pelos modelos em comparação com os consumos observados.

Coeficiente de Determinação, R^2, (R-Squared, em inglês) é calculado através da equação 3.

$$\boldsymbol{R^2} = \mathbf{1} - \frac{\sum(\boldsymbol{y_i} - \boldsymbol{x_i})}{\sum(\boldsymbol{y_i} - \boldsymbol{\bar{x}})} \quad (3)$$

Onde:

y_i = valor real;

x_i = valor previsto;

$\bar{x}$ = média dos valores previstos.

O R^2 varia entre 0 e 1 e é uma indicação de quanto o modelo explica os dados. Qualquer valor maior que zero mostra o quanto de variabilidade dos dados que o modelo consegue explicar. Ou seja, é uma medida de quanto o modelo avaliado se ajusta aos dados e prevê corretamente.

4. RESULTADOS

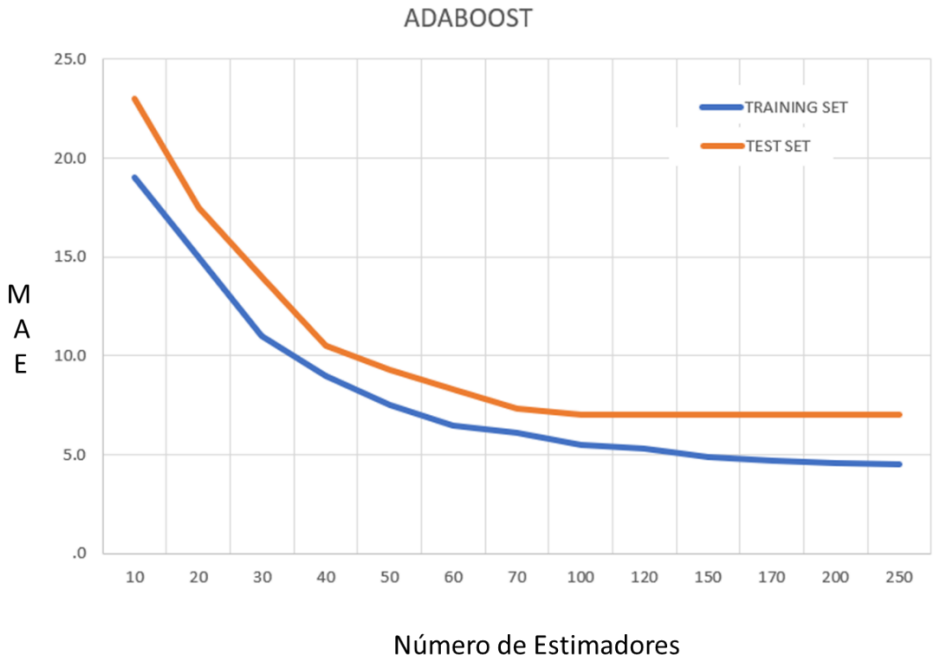
4.1. RESULTADOS USANDO O ALGORITMO ADABOOST

O modelo Adaboost foi implementado em Python, usando a biblioteca do Scikit-learn Sklearn Adaboost Regressor (2021). Os hiperparâmetros usados no ajuste foram:

- Número de estimadores (n_estimator): é o número máximo de aprendizes fracos que serão treinados. O valor padrão é 50. Foram usados os valores 30, 50, 70, 100, 120, 150, 200 e 250.
- Taxa de aprendizagem (learning_rate): é o peso aplicado a cada classificador em cada iteração. Determina em que medida as informações antigas serão substituídas pelas novas. O valor padrão é 1. Foram usadas as taxas 0,3, 0,5, 0,7 e 1.
- Função de perda (loss): é a função de perda usada pelo modelo para ponderar as iterações. A função padrão é a linear. Foram usadas as funções linear e exponencial.

A performance do modelo aplicado aos dados de treinamento, em termos de MAE, é mostrada na figura 3.

Figura 3 – Performance do modelo Adaboost – métrica MAE



Fonte: Elaborado pelo autor.

Como a performance do modelo aplicado aos dados de treinamento não supera significantemente a performance quando aplicado aos dados de teste, conclui-se que o modelo não sofre de *overfitting*. Também está claro que acima de 100 estimadores não houve redução do erro de previsão nos dados de teste.

A tabela da figura 4 mostra o melhor arranjo de hiperparâmetros usados no modelo Adaboost.

Figura 4 – Melhor arranjo de hiperparâmetros - Modelo Adaboost

Hiperparâmetro	Valor
Número de estimadores	100
Taxa de aprendizagem	0,5
Função de perda	linear

Fonte: Elaborado pelo autor.

Os resultados do modelo ajustado em comparação com o modelo usado com todos os valores padrões são mostrados na tabela da figura 5.

Figura 5 – Comparação entre modelos com parâmetros padrões e ajustados do modelo Adaboost

MODELO	RMSE	MAE	R^2
ADABOOST PADRÃO	23,17	6,54	0,857
ADABOOST AJUSTADO	22,54	6,35	0,865

Fonte: Elaborado pelo autor.

4.2. RESULTADOS USANDO O ALGORITMO GRADIENT BOOSTING

O modelo *Gradient Boosting* foi implementado em Python, usando o pacote XGBoost (2021) para Python. Os hiperparâmetros usados no ajuste foram:

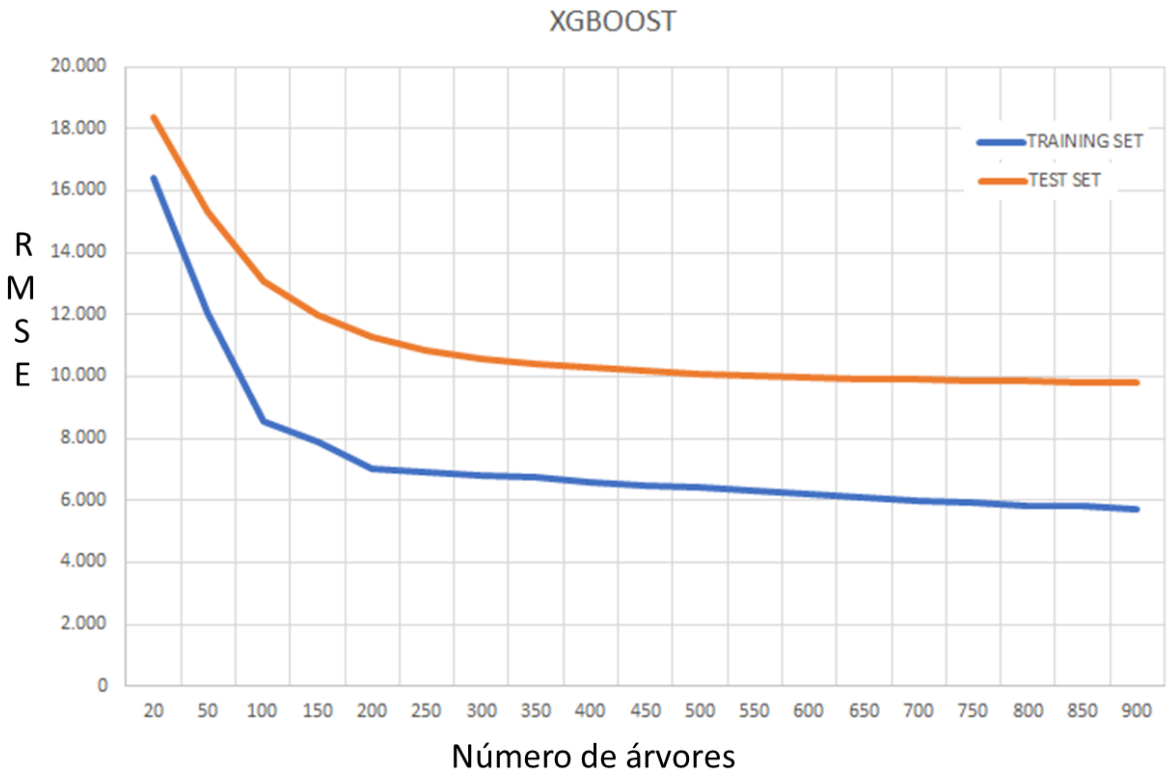
- Profundidade máxima (max_depth): é a profundidade máxima para cada estimador construir uma árvore. O valor padrão é 6. Foram usados os valores 6, 8 e 10.
- Taxa de aprendizagem (learning_rate): o valor padrão é 0,3. Foram usadas as taxas 0,3, 0,5 e 0,7.

ISSN 2236-6717

- Número de árvores (n_estimators): é o parâmetro que controla o número de árvores sequenciais que farão iterações para corrigir as árvores anteriores. O valor padrão é 100. Foram usados os valores 100, 200, 300, 400 e 500.
- Soma mínima dos pesos (min_child_weight): refere-se ao número de amostras necessárias para a formação de um nó de decisão na folha. Foram usados os valores 2, 4, 6 e 8.
- Subamostras de colunas por árvore (colsample_bytree): indica a fração de colunas usadas como amostras aleatórias de cada árvore. Foram usados os valores 0,3, 0,5, 0,7 e 0,9.

A figura 6 mostra que em termos de RMSE, as curvas possuem um longo achatamento estável. A performance do modelo XGBoost aplicado aos dados de treinamento apresenta erro ligeiramente menor do que a performance quando aplicado aos dados de teste. Logo, conclui-se que o modelo não possui *overfitting*.

Figura 6 – Performance do modelo XGBoost – métrica RMSE



Fonte: Elaborado pelo autor.

O modelo XGBoost foi rodado inúmeras vezes com diferentes configurações para decidir qual delas apresentava resultados mais estáveis. A tabela da figura 7 mostra o melhor arranjo de hiperparâmetros usados.

Figura 7 – Melhor arranjo de hiperparâmetros - Modelo XGBoost

Hiperparâmetro	Valor
Número de árvores	500
Taxa de aprendizagem	0,3
Profundidade máxima	6
Soma mínima dos pesos	4
Subamostras de colunas	0,7

Fonte: Elaborado pelo autor.

Os resultados do modelo ajustado em comparação com o modelo usado com todos os valores padrões são mostrados na tabela da figura 8.

Figura 8 – Comparação entre modelos com parâmetros padrões e ajustados do modelo XGBoost

MODELO	RMSE	MAE	R^2
XGBOOST PADRÃO	16,71	7,78	0,924
XGBOOST AJUSTADO	13,21	5,28	0,953

Fonte: Elaborado pelo autor.

4.3. CÁLCULO TEÓRICO DO CONSUMO DE COMBUSTÍVEL

Para comparar as performances dos modelos propostos em Adaboost e *Gradient Boosting*, foi calculado o consumo teórico de combustível (CT) para cada viagem, em toneladas métricas, através da equação 4.

$$\mathbf{CT = CN \frac{D}{24 .V}} \tag{4}$$

Onde:

CN = consumo teórico de óleo combustível do navio em viagem, em toneladas métricas por dia;

D = distância navegada, em milhas náuticas;

V = velocidade média do navio, em nós.

4.4. COMPARANDO OS MODELOS

Ao usar modelos de regressão, pretende-se que eles façam predições para valores de uma variável contínua e, sendo assim, a diferença entre os valores previstos e reais é inevitável. Na comparação dos modelos – mostrada na figura 9, o XGBoost apresentou melhor desempenho em todas as métricas. Em termos de erro absoluto médio, MAE, o XGBoost obteve resultado de 5,28, o que corresponde a dizer que o modelo conseguiu prever o consumo de óleo combustível do navio com erro médio de 5,28 toneladas métricas, para mais ou para menos.

Em comparação com o cálculo teórico do consumo de combustível, ambos modelos tiveram desempenho muito superior.

Figura 9 – Comparação das performances dos modelos com melhores arranjos de hiperparâmetros

MODELO	RMSE	MAE	R^2
CÁLCULO CONSUMO TEÓRICO	32,99	13,39	0,892
ADABOOST	22,54	6,35	0,865
XGBOOST	13,21	5,28	0,953

Fonte: Elaborado pelo autor.

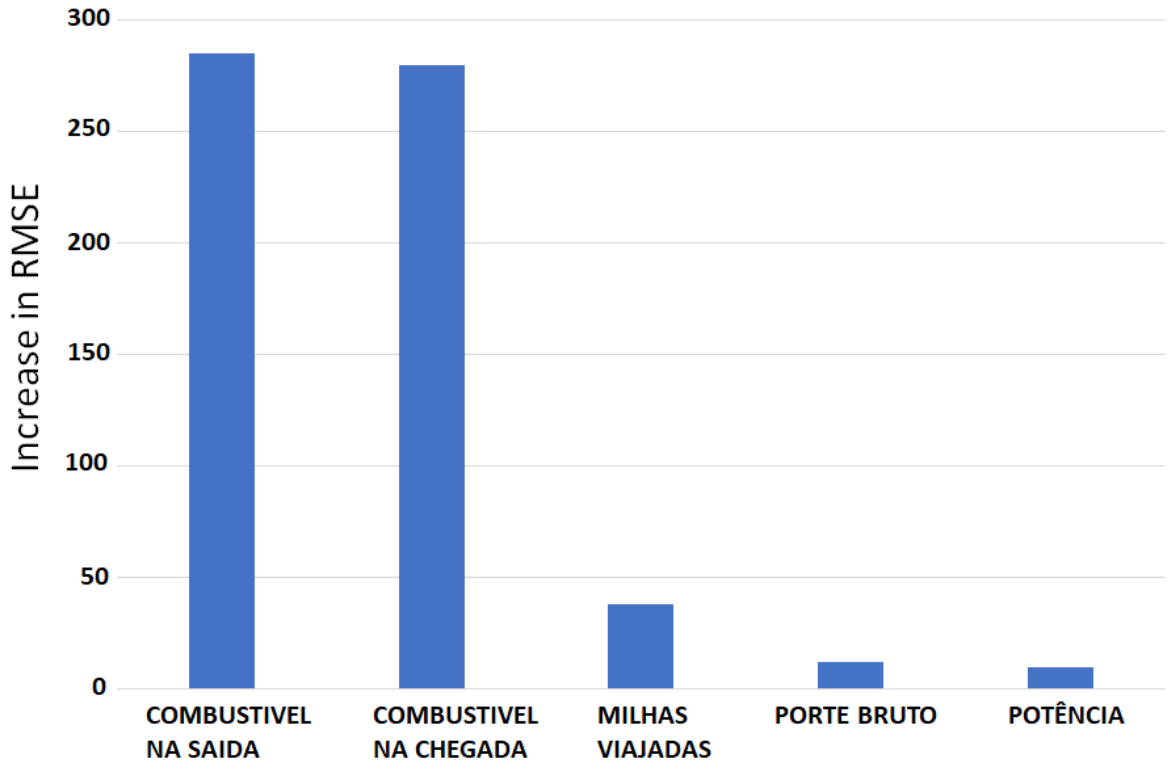
4.5. IMPORTÂNCIA DAS FEATURES

Há várias técnicas para avaliar a importância de cada *feature* e permitir descobrir quais possuem poder preditivo, independente do modelo que está sendo usado na predição. Foi usada a técnica *Permutation Feature Importance* (2021), do Scikit-learn, que atribui um score a cada preditor baseado na sua capacidade de melhorar as previsões. Assim podemos classificar os preditores com base em suas capacidades preditivas relativas.

A figura 10 mostra as *features* ordenadas baseados em suas importâncias. Pelo gráfico podemos ver que “Combustível na Saída”, “Combustível na Chegada”, “Milhas Viajadas”,

"Porte Bruto" e "Potência (HP)" possuem as maiores pontuações e foram os preditores que ofereceram as informações mais valiosas para prever o consumo de combustível. Os demais preditores – que não aparecem no gráfico – foram importantes, mas com pontuações menores.

Figura 10 – *Features* por ordem de importância



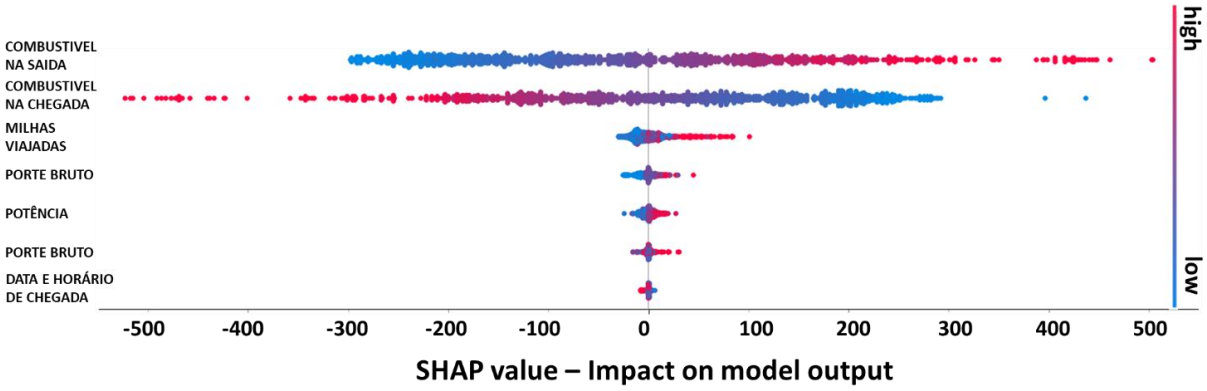
Fonte: Elaborado pelo autor.

O método SHAP (Shapley Additive explanations), criado em 1953 por Lloyd Shapley, possibilita medir como cada *feature* influi de forma positiva ou negativa na predição do modelo.

Foi usado o pacote SHAP (2021) para Python. Os valores com as sete *features* mais importantes são mostrados no gráfico da figura 11. Novamente as *features* "Combustível na Saída" e "Combustível na Chegada" são as mais importantes, ou seja, foram as que mais influenciaram nos valores previstos. O gráfico mostra que quanto maior for o valor da *feature*

"Combustível na Saída", maior será o impacto positivo no resultado do modelo, acontecendo o inverso com "Combustível na Chegada".

Figura 11 – *Features* por ordem de importância – valores SHAP



Fonte: Elaborado pelo autor.

CONCLUSÃO

O objetivo era predizer o consumo de óleo combustível de uma frota de navios, usando como base os dados de 300 navios em um intervalo de dez anos de viagens na cabotagem brasileira, usando modelos de *boosting* que tivessem uma performance com baixos erros de previsão, e que fossem melhor do que simplesmente fazer o cálculo teórico.

Neste estudo os modelos em Adaboost e *Gradient Boosting* foram construídos seguindo as etapas de escolha da base de dados, pré-processamento, aprendizado de máquina, predição e avaliação.

O modelo Adaboost tem suas vantagens. Ele é mais fácil de ajustar e não é propenso ao *overfitting*. Entretanto o modelo exigiu maiores tempos para execução e maior capacidade computacional à medida que aumentaram os números de estimadores.

Apesar de sua maior complexidade, com relação ao tempo de execução do modelo XGBoost, ele se mostrou muito mais rápido, mesmo com o número de árvores aumentando. Ambos os modelos apresentaram boas performances, com baixos erros de previsão, mas o modelo construído com o XGBoost provou ser capaz de fazer uma predição bem melhor do que o cálculo teórico do consumo de combustível, e mesmo melhor que o modelo Adaboost.

Com base nos resultados obtidos, pode-se afirmar que a técnica *boosting* de aprendizado de máquina pode ser usada com sucesso na predição do consumo de combustível de uma frota de navios.

Para estudos futuros, sugere-se a técnica de *boosting* aplicada a previsões de consumo de óleo diesel marítimo ou para prever o consumo de combustível em viagens de longo curso. As previsões podem se estender a outras frotas de navios, como de porta-contêineres, por exemplo. Outras técnicas também merecem ser testadas neste amplo campo do aprendizado de máquina.

REFERÊNCIAS BIBLIOGRÁFICAS

ANALYTICS VIDHYA (2021) Disponível em https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/. Acesso em 25 jun, 2021.

FREUND, Yoav; SCHAPIRE, Robert. *Experiments with a new boosting algorithm. Machine learning: proceedings of the thirteenth international conference*. 1996 (p. 148-156).

GÉRON, Aurélien. *Hands-on machine learning with scikit-learn and tensorflow*. Cap. 7. (p. 91). Sebastopol: O'Reilly Media, Inc., 2017.

GONG, Rong; FONSECA, Eduardo; BOGDANOV, Dmitry; SLIZOVSKAIA, Olga; GÓMEZ, Emilia; SERRA, Xavier. *Acoustic scene classification by fusing lightgbm and vgg-net multichannel predictions. Detection and classification of acoustic scenes and events* 2017. p. 1–4.

KUHN, Max; JOHNSON, Kjell. Applied predictive modeling. New York: Springer New York, 2013.

MARQUÉS, A.; GARCÍA, V.; SÁNCHEZ, J.. *Exploring the behaviour of base classifiers in credit scoring ensembles. Expert Systems with Applications*, vol. 39, issue 11 (p. 10244-10250), 2012. Disponível em https://doi.org/10.1016/j.eswa.2012.02.092. Acesso em 15 abr. 2021.

PANDAS (2021). Disponível em https://pandas.pydata.org/pandas-docs/stable/index.html. Acesso em 21 mar. 2021.

PERMUTATION FEATURE IMPORTANCE (2021) – SCIKIT-LEARN v. 0.24.2 – Disponível em https://scikit-learn.org/stable/modules/permutation_importance.html. Acesso em 25 jun. 2021.

RODRIGUE, Jean-Paul. *The Geography of Transport Systems* (2021). Disponível em https://transportgeography.org/contents/chapter5/maritime-transportation/containerships-operating-costs-panamax-post-panamax/. Acesso em: 21 jun. 2021.

SCIKIT-LEARN (2021) v. 0.24.2 – 3.2. *Tuning the hyper-parameters of an estimator.* Disponível em https://scikit-learn.org/stable/modules/grid_search.html. Acesso em 27 abr. 2021.

SHAP (2021) – *Tree based models* – Disponível em https://shap.readthedocs.io/en/latest/tabular_examples.html#tree-based-models. Acesso em 20 jun. 2021.

SKLEARN ADABOOST REGRESSOR (2021) - Sklearn.ensemble.AdaBoostRegressor. SCIKIT-LEARN v. 0.24.2 . Disponível em https://scikit-learn.org/stable/modules/generated/sklearn. ensemble.AdaBoost Regressor.html. Acesso em 12 mar. 2021.l

STOPFORD, Martin. *Maritime Economics*. Cap. 6. (p. 233). Ed. 3. London: Routledge, 2009.

THE OXFORD INSTITUTE FOR ENERGY STUDIES (2018). *A review of demand prospects for LNG as a marine transport fuel*. (p. 4). Disponível em https://www.oxfordenergy.org/wpcms/wp-content/uploads/2018/07/A-review-of-demand-prospects-for-LNG-as-a-marine-fuel-NG-133.pdf. Acesso em 15 mai. 2021.

WORLD SHIPPING COUNCIL (2008*). Record fuel prices place stress on ocean*. (p. 1). Disponível em http://www.worldshipping.org/pdf/wsc_fuel_statement_final.pdf. Acesso em 25 mai. 2021.

XGBOOST (2021). XGBoost *python package*. Disponível em https://xgboost.readthedocs.io/en/latest/python/index.html. Acesso em 19 mar. 2021.