

Previsão da duração de carregamentos de embarcações PLSV

Rachel Martins Ventriglia¹, Leonardo Bastos¹, Karla Figueiredo², Marley Vellasco³

¹Departamento de Engenharia Industrial - PUC-Rio / R. Marquês de São Vicente, 225, 22451-900 Rio de Janeiro/RJ, Brazil

²Departamento de Informática e Ciência da Computação / IME- UERJ - Rua São Francisco Xavier, 524, Maracanã, Rio de Janeiro/RJ - 20550-900

³Departamento de Engenharia Elétrica - PUC-Rio / R. Marquês de São Vicente, 225, 22451-900 Rio de Janeiro/RJ, Brazil

rachelventriglia@gmail.com, lsbastos@puc-rio.br,
karlafigueiredo@ime.uerj.br, marley@ele.puc-rio.br

Abstract. *Pipe-laying Support Vessels (PLSV) perform subsea interconnection tasks, which require various material resources. These resources are loaded onto ships, and currently the loading planning is resolved heuristically, with high error rates, around 84%. To assist in the operational planning of shipments, in this operational planning, this work proposed the investigation and selection of several machine learning models to predict the duration of loads. The models that presented the best performance in the test base were the Gradient Boosting, the Linear Regression, and the Stacking Regressor, with an absolute average percentage error of at most 36% in the test set.*

Resumo. *As embarcações Pipe-laying Support Vessel (PLSV) realizam tarefas de interligação submarinas, que necessitam de diversos recursos materiais. Estes recursos são carregados nos navios, e atualmente o planejamento dos carregamentos é resolvido de forma heurística, com taxas de erros altas, em torno de 84%. Com o objetivo de auxiliar neste planejamento operacional, este trabalho propôs a investigação e seleção de diversos modelos de aprendizado de máquina para prever a duração dos carregamentos. Os modelos que apresentaram melhor desempenho na base de teste foram o Gradient Boosting, Regressão Linear e o Stacking Regressor, com um erro percentual médio absoluto de no máximo 36% nos dados de teste.*

1. Introdução

A embarcação *Pipe-laying Support Vessel* (PLSV) é o recurso responsável pela interligação de poços submarinos à unidade estacionária de produção, sendo crítico para a exploração de *offshore* de petróleo. Além disso, este navio é um recurso limitado e de alto custo, com uma diária de aproximadamente 200 mil dólares [Bnamericas 2021].

A interligação entre poços e a unidade estacionária é a última etapa que deve ser feita para que um poço comece a produzir óleo, e diversos materiais são necessários para sua realização, como dutos submarinos (tramos), umbilicais (UEH), bobinas e

flutuadores [Ferreira 2013]. Eles são carregados em locais de atracação, chamados de berços, que estão localizados nos portos. Portanto, o carregamento destes materiais em embarcações PLSV é uma etapa de grande importância para a realização das atividades de interligação e cumprimento do cronograma.

Este trabalho é motivado por um problema de planejamento de carregamentos para embarcações PLSV de uma empresa de óleo e gás. Devido ao sigilo exigido pela empresa, esta não poderá ser revelada, e os dados usados foram anonimizados. Este planejamento é feito atualmente de forma manual e subjetiva aos conhecimentos de quem realiza esta tarefa. Por isso, cronogramas destas embarcações podem apresentar ociosidades ou atrasos, que são prejudiciais do ponto de vista financeiro, devido ao PLSV ser um recurso de alto custo e crítico para a produção de óleo.

Como estas embarcações são recursos limitados e de alto custo, é importante que o planejamento logístico seja feito de forma a evitar atrasos ou antecipações indesejadas no cronograma dos navios. Com isso, o principal objetivo deste trabalho é desenvolver um modelo preditivo capaz de prever a duração dos carregamentos das embarcações PLSV. Para isso, foi realizada uma investigação e seleção entre o desempenho de diversos modelos como a Regressão Linear, *Multilayer Perceptron*, *Decision Trees*, *Random Forest*, *Adaboost*, entre outros.

O restante do trabalho está distribuído em mais quatro seções: na Seção 2 é apresentada a revisão da literatura sobre o tema. A metodologia utilizada para solução do problema proposto é descrita na terceira seção, e os resultados estão apresentados na Seção 4. Por fim, a última seção encerra o trabalho apresentando as conclusões e perspectivas de novos trabalhos.

2. Revisão da Literatura

A aplicação de ciência de dados e do aprendizado de máquinas na indústria de óleo e gás tem ocorrido em diversos segmentos da engenharia de petróleo e geociências. Tariq et al. (2021) apresentaram uma revisão sistemática da literatura sobre estas aplicações em áreas como exploração de petróleo ou caracterização de reservatórios, utilizando técnicas de classificação, regressão e *clustering*.

Dentre as aplicações de técnicas de regressão na indústria de óleo e gás, estes autores trouxeram artigos cujo objetivo é prever permeabilidade, porosidade ou saturação da água. Em relação as técnicas mais utilizadas, percebeu-se uma grande quantidade de artigos que utilizavam Redes Neurais Artificiais (ANN), *Multilayer Perceptron*, *Radial Basis Functions*, *Support Vector Machines* e Árvores de Decisão [Tariq et al. 2021].

Analisando a aplicação de técnicas de regressão no contexto de embarcações marítimas, Mekkaoui et al. (2020) apresentam abordagens existentes para prever o tempo estimado de chegada (ETA) dos navios, utilizando modelos de Redes Neurais (NN) para realizar a previsão. O modelo que apresentou melhor desempenho foi uma rede neural recorrente *Long-Short Term Memory* (LSTM). Stepec et al. (2020) utilizaram o algoritmo *CatBoost* para prever o tempo de retorno do navio com dados padronizados de escala de porto, obtendo resultados melhores que uma regressão linear múltipla, com um erro percentual médio absoluto de 30%.

Alguns autores fizeram uso de aprendizado de máquinas para prever o consumo de combustível e energia de navios. No trabalho de Peng et al. (2020) observa-se um modelo para prever o consumo de energia de embarcações em portos sustentáveis, onde os melhores resultados foram obtidos com um modelo *Random Forest* e *Gradient Boosting Regression*.

Sobre a previsão do consumo de combustível de embarcações, Kim et al. (2021) utilizaram dados em serviço coletados de um navio porta-contêineres com modelos para fazer a previsão. Para implementar o modelo de predição, foram aplicadas uma rede neural artificial e uma regressão linear múltipla, onde a rede neural apresentou melhores resultados. Gkerekos et al. (2019), por sua vez, foca em realizar uma comparação de modelos de regressão para prever o consumo de óleo do motor principal do navio. Para isso, vários algoritmos de regressão múltipla foram empregados, e os modelos *Random Forest Regressors* e *Extra Trees Regressors* apresentaram melhor desempenho. Jeon et al. (2018) também apresentaram uma Rede Neural Artificial para prever o consumo de combustível de navios, comparando os resultados com modelos de Regressão Linear Múltipla e *Support Vector Machines*.

Apesar da aplicação de técnicas de regressão ser bastante utilizada para no contexto de previsão de atributos relevantes para embarcações marítimas, não há artigos específicos sobre a previsão da duração dos carregamentos dos navios PLSVs, para o melhor de nosso conhecimento. Por isso, este trabalho pretende ajudar a suprir esta lacuna na literatura.

3. Metodologia

A metodologia utilizada neste trabalho, apresentada na Figura 1, mostra as etapas seguidas para a realização e avaliação dos modelos preditivos propostos.

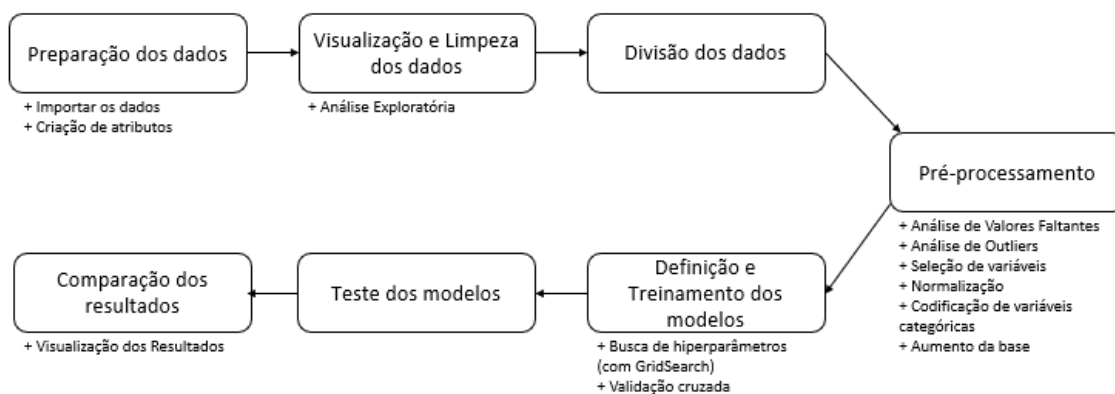


Figura 1: Metodologia (adaptado de [Dantas 2020])

O processo começa com a preparação dos dados, que envolve sua obtenção e criação de atributos (*feature engineering*), seguido pela visualização e limpeza da base de dados. Nesta etapa, é realizada uma análise exploratória da base. Com a base de treinamento é realizado o pré-processamento, que envolve uma análise de *outliers*, análise de valores faltantes, seleção de variáveis a serem utilizadas no modelo, normalização de variáveis numéricas e codificação das variáveis categóricas, além do aumento da base de treinamento, se necessário. Em seguida, define-se os modelos a serem utilizados e os melhores hiperparâmetros são definidos pelo *10-fold Cross-Validation*. Além disso, os melhores modelos da etapa anterior são combinados para

formar um *Stacking Regressor*, com o objetivo de melhorar os resultados. Posteriormente, os modelos são avaliados na base de teste por meio das métricas de avaliação e os melhores modelos são selecionados. Além disso, também é realizado um teste com os dados mais recentes do planejamento de carregamentos da empresa. Todos os experimentos foram realizados utilizando o *Python 3.9.7*, com auxílio do pacote *scikit-learn*.

3.1. Preparação dos Dados

Os dados utilizados neste trabalho são uma instância de uma empresa de óleo e gás. Ao todo são aproximadamente 510 registros, com 12 atributos referentes ao planejamento feito pela empresa: nome do berço, nome da base, nome da embarcação, data do carregamento, hora do carregamento, quantidade de tramos, comprimento dos tramos, quantidade de umbilicais (UEHs), comprimento das umbilicais (UEHs), quantidade de flutuadores, quantidade de bobinas e duração do carregamento.

No dado obtido, existem informações sobre as bases onde as embarcações podem realizar os carregamentos, os berços, que são os locais de atracação e qual embarcação PLSV realizará o carregamento. Sobre os recursos materiais carregados, tem-se tramos, umbilicais (UEHs), flutuadores e bobinas. Os tramos são dutos submarinos e UEHs são linhas submarinas rígidas. Já os flutuadores são boias responsáveis por minimizar a tensão vertical das linhas submarinas nas conexões da plataforma, e as bobinas são carretéis enrolados com as linhas submarinas.

3.2. Visualização e Limpeza dos dados

Foi realizada uma análise exploratória dos dados para entender melhor seu comportamento e uma análise de correlação entre os atributos da base.

3.3. Pré-processamento

Inicialmente foi realizada uma análise de valores faltantes dentro da base de dados, que mostrou que não existia nenhum valor faltante no dado utilizado. Como não havia valores faltantes na base de dados, iniciou-se a avaliação de possíveis *outliers*. O cálculo dos *outliers* foi feito utilizando o método dos quartis, com o critério moderado, indicado através de um *boxplot*. Após a identificação destes, decidiu-se avaliar os resultados com e sem o tratamento de valores considerados *outliers*.

Para entender quais atributos são mais relevantes na base de dados, foi realizada uma seleção de variáveis utilizando duas técnicas do tipo *filter*: *Correlation Based Feature Selection* [Hall 1999] e *Relief-F* [Kononenko et al. 2004]. A primeira avalia e indica subconjuntos que contêm atributos mais altamente correlacionados com a variável resposta e menos correlacionados entre si. Já a segunda identifica atributos fracamente e fortemente relevantes no dado, considerando relações entre os registros mais e menos próximos na base de dados, e indicando um *ranking* dos atributos. A seleção foi feita utilizando o software WEKA 3.8.6.

Além disso, é necessário realizar a codificação das variáveis nominais no formato binário, mais especificamente *one-hot encoding*. Alguns modelos como redes neurais e *Support Vector Regressors* também requerem a normalização dos dados, e deve ser feita nesta etapa de pré-processamento. A normalização utilizada foi a MinMax, que deixa os valores dos atributos numéricos em um intervalo de 0 e 1. Como

a base de dados não possuía uma quantidade extensa de registros, foi feito também um aumento da base, apenas do dado de treinamento que não tiveram a remoção de *outliers*. Este aumento foi realizado com uma técnica chamada *Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise* (SMOTEGN), que combina uma subamostragem aleatória com duas técnicas de sobre-amostragem: SMOTE para regressão (SMOTER) e introdução de ruído gaussiano [Branco et al. 2017].

3.4. Treinamento dos modelos

Foram selecionados nove algoritmos de regressão para serem utilizados: Regressão Linear, *Multilayer Perceptron* (MLP) [Haykin 1994], *Random Forest* (RF) [Breiman 2001], Árvore de Decisão [Quinlan 1986], *AdaBoost* (ADA) [Schapire 2013], *Gradient Boosting* (GB) [Friedman 2001], *Support Vector Regression* (SVR) [Vapnik 2000], *XGBoost* (XGB) [Chen and Guestrin 2016] e *Catboost* (CAT) [Prokhorenkova et al. 2017]. Além disso, foram utilizadas três versões da base de dados para serem avaliadas:

- *Dataset 1*: Sem remoção de *outliers*;
- *Dataset 2*: Com a remoção de *outliers*;
- *Dataset 3*: Sem a remoção de *outliers* e com a base de treinamento aumentada através do SMOTEGN.

Para determinar os conjuntos de hiperparâmetros que melhoram o ajuste de cada modelo, foi realizada uma busca de hiperparâmetros através do *GridSearch* com um *k-fold Cross-Validation* com $k = 10$. A métrica para medir o desempenho dos modelos avaliados é a média do erro quadrático negativo (negative MSE), e por ser uma métrica de erro, quanto mais próximo de zero for o resultado, melhor o modelo. Depois que os valores dos hiperparâmetros foram escolhidos, treinou-se o modelo usando o conjunto de treinamento completo. Para o *Stacking Regressor*, foram escolhidos os três melhores modelos de cada *dataset*, a partir do erro médio de validação.

3.5. Avaliação dos modelos

Para avaliar os modelos desenvolvidos, foram selecionadas como métricas de avaliação: o Erro Absoluto Médio (MAE), Erro Quadrático Médio (MSE), o Erro Percentual Médio Absoluto (MAPE) e o R2 score. Os melhores modelos foram definidos a partir de cada uma destas métricas.

4. Resultados e Discussão

4.1. Análise exploratória

Inicialmente foi realizada uma análise da variável resposta, que seria a duração dos carregamentos, em dias. A Figura 2 mostra um *boxplot* das durações existentes na base de dados. É possível visualizar que a mediana se encontra em aproximadamente 3 dias de duração, e os *outliers* são os registros com duração acima de 9 dias, aproximadamente.

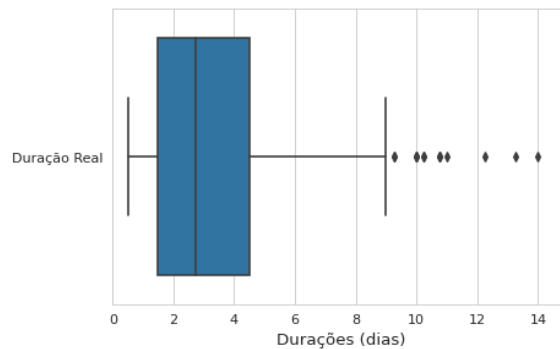


Figura 2: Boxplot da duração real dos carregamentos

Além disso, foi realizada uma análise quantitativa dos atributos numéricos da base de dados. Com isso, pode-se observar que os atributos de quantidade de UEHs, comprimento das UEHs, quantidade de flutuadores e quantidade de bobinas possuem mais de 75% de seus valores iguais a zero, indicando que são poucos os casos de carregamentos que recebem esses materiais para realizar os serviços das embarcações PLSV. Dessa forma, é provável que estes atributos não tenham bom desempenho no modelo preditivo.

Como os tramos e as UEHs são linhas submarinas para conectar o poço à plataforma produtora, com a diferença de que o primeiro é flexível e o segundo é rígido, foi considerado que esses atributos poderiam ser combinados, criando dois atributos que armazenam a informação de quantidade de total linhas e outro de comprimento total de linhas. Portanto, estes atributos foram incluídos na base e os antigos foram removidos. A Figura 3 mostra a distribuição das variáveis numéricas em comparação com duração dos carregamentos, em dias, normalizada.

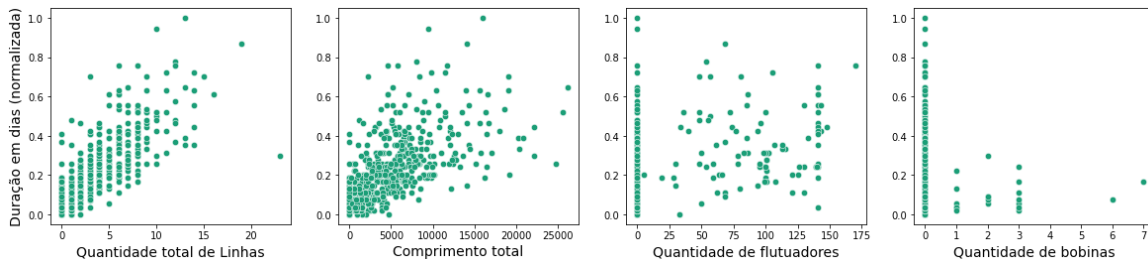
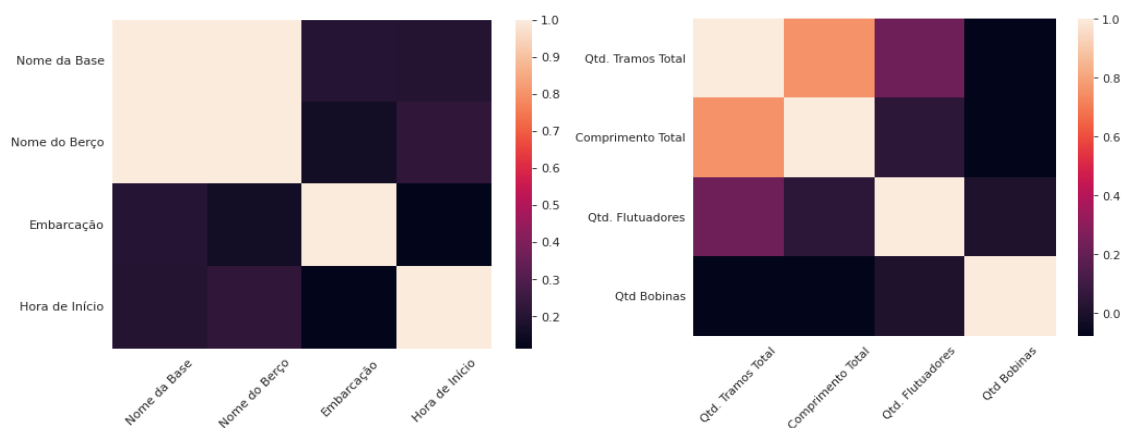


Figura 3: Gráfico da distribuição das variáveis numéricas com a duração normalizada

A partir desta figura, é possível perceber que a quantidade de linhas totais e o comprimento total possuem uma relação aproximadamente linear com a duração, ou seja, quanto maior esses atributos, maior a duração do carregamento. Já para os atributos de quantidade de flutuadores e bobinas, esse padrão não é tão perceptível.

Uma análise de correlação entre os atributos também foi realizada, utilizando a correlação de Pearson para avaliar a correlação entre atributos numéricos, e estatística Cramer's V para os atributos categóricos. O resultado está apresentado na Figura 4.



(a) Correlação entre as variáveis categóricas (b) Correlação entre as variáveis numéricas

Figura 4: Análise de Correlação

Pela análise apresentada, os atributos categóricos que possuem alta correlação entre si são o nome da base e nome do berço, o que faz sentido dado que os berços são locais de atracação dentro das bases. Já para os atributos numéricos, a correlação é alta entre a quantidade de linhas total e o comprimento total, o que também é coerente, visto que quanto mais linhas temos, maior também é o comprimento total.

4.2. Seleção das variáveis

A técnica *Correlation Based Feature Selection* indicou que a melhor combinação de atributos da base foi com os atributos de Quantidade de linhas total, Quantidade de flutuadores e Nome da base, com um mérito de 0.774 deste subconjunto. Já a técnica *Relief-F* retornou um ranking dos atributos da base, apresentado na Tabela 1. Pelo ranking, pode-se observar que os melhores atributos encontrados estão de acordo com a primeira técnica, com exceção do comprimento total, que não foi incluído no melhor subconjunto da *Correlation Based Feature Selection* por ser altamente correlacionado com o atributo Quantidade total de linhas. Com isso, os três atributos destacados na Tabela 1 foram selecionados como as variáveis explicativas do modelo.

Tabela 1: Resultado da seleção de atributos com o *Relief-F*

Ranking	Atributo
0.06551	Quantidade total de linhas
0.05490	Comprimento total
0.04287	Quantidade de flutuadores
0.00123	Nome da base
-0.00052	Nome do berço
-0.00440	Quantidade de bobinas
-0.01145	Hora de início
-0.07916	Embarcação

4.3. Treinamento dos modelos

No processo de treinamento dos modelos foi utilizada a base de treinamento, que corresponde a 80% dos valores da base de dados. Foram treinados e comparados nove algoritmos de regressão, e para cada um foi avaliado conjunto de hiperparâmetros

correspondente. Os intervalos e os melhores hiperparâmetros para os modelos, em cada *dataset* estão apresentados na Tabela 2.

A partir do resultado das métricas de erro da validação, foram selecionados os três melhores modelos de cada *dataset* para compor o Stacking Regressor. Para o *dataset 1*, os modelo escolhidos foram o *Catboost*, *Multilayer Perceptron* e *Random Forest*, e para os *datasets 2 e 3* foram o *Catboost*, *Random Forest* e *Gradient Boosting*. O *Stacking Regressor* foi criado combinando estes modelos, usando uma Regressão Linear como estimador final, para tentar utilizar a diversidade dos diferentes modelos para criar um estimador melhor.

Tabela 2: Intervalo e melhores hiperparâmetros para cada modelo e *dataset*

Algoritmo	Intervalo de Hiperparâmetros	Melhores Hiperparâmetros			
		Dataset 1	Dataset 2	Dataset 3	
LinearRegression	Whether to calculate the intercept for this model: [True, False]	True	True	True	
	Number of jobs to use for the computation: [None, -1]	None	None	None	
		Métrica:	-1.73	-1.40	-4.93
MLPRegressor	Hidden layer sizes: [(5),(10),(15),(20)]	5	5	20	
	Activation: ['logistic', 'tanh', 'relu']	tanh	logistic	tanh	
	Solver: ['sgd', 'lbfgs', 'adam']	adam	lbfgs	lbfgs	
	Batch size: [32, 'auto']	auto	32	32	
	Learning rate: ['constant', 'invscaling', 'adaptive']	constant	constant	constant	
	Initial learning rate: [0.001, 0.01, 0.05, 0.1]	0.01	0.001	0.001	
Maximum number of iterations: [200, 500, 1000, 1500]	500	500	500		
Early stopping: [True, False]	False	True	True		
		Métrica:	-1.64	-1.25	-2.49
RandomForestRegressor	Number of trees in the forest: [50, 100, 200]	200	50	50	
	Function to measure the quality of a split: ['squared_error', 'absolute_error', 'poisson']	absolute_error	squared_error	squared_error	
	Maximum depth of the tree: [3, 4, 5, 6, None]	6	5	None	
	Minimum number of samples to split an internal node: [2, 4, 8, 12, 16]	2	4	2	
	Minimum number of samples to be at a leaf node: [1, 4, 6, 8]	1	1	4	
	Number of features to consider when looking for the best split: ['auto', 'sqrt', None]	sqrt	sqrt	sqrt	
Bootstrap: [True, False]	True	True	False		
		Métrica:	-1.71	-1.24	-2.24
DecisionTreeRegressor	Function to measure the quality of a split: ['squared_error', 'friedman_mse', 'absolute_error', 'poisson']	squared_error	squared_error	squared_error	
	Maximum depth of the tree: [3, 4, 5, 6, None]	None	6	None	
	Minimum number of samples to split an internal node: [2, 4, 8, 12, 16]	16	8	8	
	Minimum number of samples to be at a leaf node: [1, 4, 6, 8]	4	1	1	
	Number of features to consider when looking for the best split: ['auto', 'sqrt', None]	sqrt	sqrt	sqrt	
		Métrica:	-1.87	-1.30	-2.60
AdaBoostRegressor	Maximum number of estimators at which boosting is terminated: [50, 100, 150]	50	50	50	
	Loss function: ['linear', 'square', 'exponential']	exponential	linear	linear	
	Learning rate: [1.0, 0.5, 2.0]	1.0	1.0	2.0	
		Métrica:	-1.98	-1.46	-3.33
GradientBoostingRegressor	Number of boosting stages to perform: [100, 150, 200]	100	100	100	
	Learning rate: [0.1, 0.5]	0.1	0.1	0.1	
	Minimum number of samples to split an internal node: [2, 4, 8, 12]	2	2	2	
	Minimum number of samples to be at a leaf node: [1, 4, 6, 8]	4	6	8	
	Maximum depth of the individual regression estimators: [3, 4, 5, None]	3	4	None	
		Métrica:	-1.82	-1.27	-2.44
SVR	Kernel: ['linear', 'poly', 'rbf', 'sigmoid']	linear	rbf	rbf	
	Gamma: ['scale', 'auto']	scale	scale	scale	
		Métrica:	-1.80	-1.43	-3.94
XGBRegressor	Number of gradient boosted trees: [50, 100, 150, 200]	50	100	100	
	Maximum tree depth for base learners: [4, 5, 6]	4	4	5	
	Learning rate: [0.05, 0.1, 0.2]	0.1	0.05	0.05	
		Métrica:	-1.84	-1.29	-2.49
CatBoostRegressor	Learning rate: [0.01, 0.05, 0.1, 0.2]	0.1	0.1	0.1	
	Depth of the tree: [4, 6, 8, 10]	4	4	8	
	Maximum number of trees that can be built: [50, 100, 200]	50	100	100	
		Métrica:	-1.64	-1.20	-2.16

4.4. Avaliação dos modelos

Com os melhores hiperparâmetros selecionados (vide Tabela 2) e os modelos treinados, estes foram avaliados na base de teste, que corresponde a 20% da base de dados, inicialmente separados da base de treinamento (*holdout*). Analisando o resultado individual de cada modelo em cada *dataset* na Tabela 3, pode-se perceber que os modelos obtiveram os melhores resultados foram o *Gradient Boosting*, *XBGoost*, *Support Vector Regression*, Regressão Linear e o *Stacking Regressor*. Além disso, é

possível observar que os resultados foram próximos para os *datasets* 1 e 2, enquanto os modelos para o *dataset* 3 não obtiveram um desempenho tão bom.

Destacando os resultados dos *datasets* 1 e 2, que foram melhores que os do *dataset* 3, pode-se ver que no *dataset* 1 o *XGBoost* apresentou melhor resultado para a métrica de MAE, o *Gradient Boosting* para as métricas de MSE e R². Já no *dataset* 2, o *Catboost* apresentou melhor resultado para a métrica de MAE, a Regressão Linear para as métricas de MSE e R². Em ambos os *datasets*, o *Support Vector Regression* apresentou o menor MAPE.

Para o *dataset* 3, o *Stacking Regressor* foi o modelo que apresentou o melhor resultado em todas as métricas avaliadas, enquanto nos outros *datasets*, o *Stacking* não obteve resultados tão bons quanto os obtidos pelos modelos isoladamente, como a *Multilayer Perceptron* ou o *Gradient Boosting*.

Tabela 3: Avaliação dos modelos na base de teste (*holdout*) de cada *dataset*

Modelo	Dataset 1				Dataset 2				Dataset 3			
	MAE	MSE	R ²	MAPE	MAE	MSE	R ²	MAPE	MAE	MSE	R ²	MAPE
LR	1.039	2.323	0.651	0.370	0.915	1.620	0.592	0.342	1.351	3.321	0.500	0.528
MLP	1.083	2.534	0.619	0.356	0.950	1.736	0.562	0.337	1.275	3.942	0.407	0.392
RF	1.044	2.485	0.626	0.332	0.928	1.694	0.573	0.351	1.196	3.812	0.427	0.408
DT	1.067	2.542	0.618	0.338	1.009	1.908	0.519	0.361	1.268	4.592	0.309	0.378
ADA	1.134	2.343	0.647	0.486	0.935	1.676	0.577	0.365	1.371	3.792	0.430	0.528
GB	1.023	2.259	0.660	0.336	0.923	1.725	0.565	0.337	1.226	4.550	0.316	0.398
SVR	1.059	2.612	0.607	0.327	0.965	2.049	0.484	0.325	1.233	3.655	0.450	0.404
XGB	1.008	2.328	0.650	0.334	0.909	1.631	0.589	0.334	1.224	4.164	0.374	0.394
CAT	1.057	2.468	0.629	0.367	0.904	1.638	0.587	0.335	1.214	3.871	0.418	0.396
StackReg	1.044	2.436	0.633	0.350	0.909	1.644	0.586	0.340	1.175	2.843	0.572	0.362

Utilizando os modelos que apresentaram melhor resultado em cada uma das métricas avaliadas, feita uma análise da previsão, em cada, *dataset* com a duração real dos carregamentos, apresentada na Figura 5. Para o *dataset* 1, o modelo selecionado foi o *Gradient Boosting*, devido à melhor performance em duas das quatro métricas (MSE e R²), e para o *dataset* 2, a Regressão Linear, pelo mesmo motivo. Para o *dataset* 3, o melhor modelo foi o *Stacking Regressor*. É possível observar na Figura 5, que para o *dataset* 1, o *Gradient Boosting* obteve resultados concentrados perto da diagonal principal. Para o *dataset* 2, pode-se observar que a Regressão Linear apresenta resultados bem próximos da linha diagonal, com alguns casos subestimando a duração dos carregamentos. Já no *dataset* 3, existem mais casos em que o modelo superestima a duração.

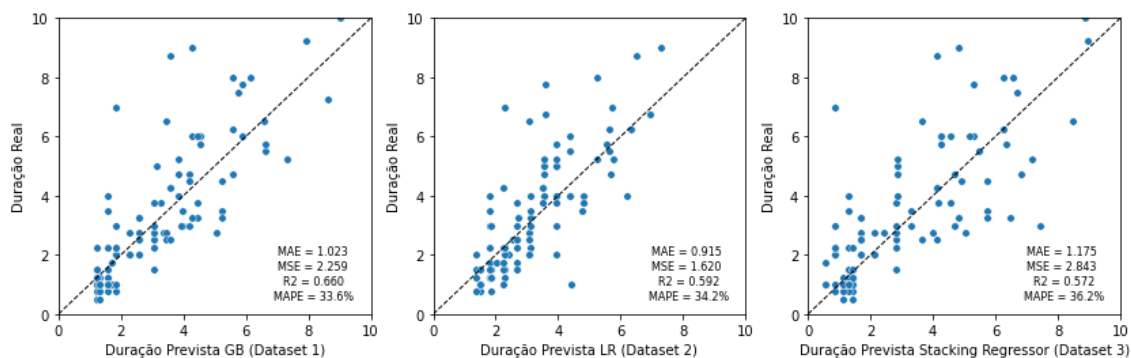


Figura 5: Comparação da previsão da duração dos melhores modelos na base de teste (*holdout*) para cada *dataset*

4.5. Teste com os dados mais recentes da empresa

Utilizando os modelos que apresentaram melhor desempenho, foi realizado um novo teste com os dados dos carregamentos mais recentes da empresa, totalizando 58 carregamentos. Isto foi feito para comparar o desempenho dos modelos com o planejamento feito atualmente pela empresa.

Analisando o planejamento feito pela empresa, junto o resultado dos modelos e com a duração real dos carregamentos, na Figura 6, pode-se concluir que o planejamento atual realizado é bem diferente da duração real observada, sendo possível observar a tendência de superestimar as durações, o que resultou em um erro percentual médio absoluto desta previsão, atinge o valor de 84%. Isso provavelmente é feito para que possíveis atrasos no carregamento não impactem tanto no cronograma de atividades as embarcações. Em comparação, observa-se que os erros de predição dos modelos derivados dos *datasets* 1 (GB) e 2 (LR) e 3 (*Stacking Regressor*) foram 55.2%, 55.0% e 55.9%. Em termos da relação entre a duração prevista e observada, os modelos dos *datasets* 1 e 2 apresentaram resultados bem próximos da linha diagonal ($R^2 = 0.67$ e $R^2=0.71$, respectivamente) do que do *dataset* 3 (*Stacking Regressor*). Além disso, o modelo LR também mostra uma tendência a superestimar as durações, com poucos casos acima da linha diagonal.

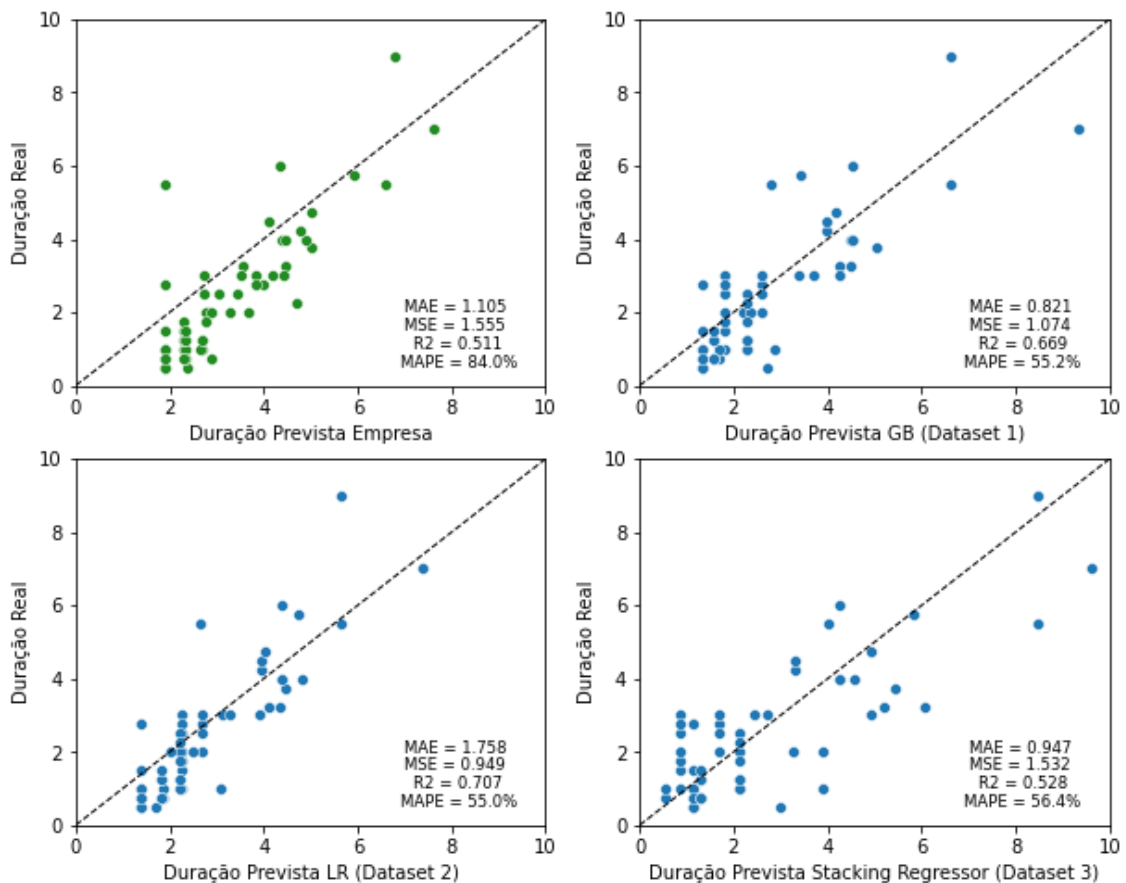


Figura 6: Comparação da previsão da duração dos modelos e da empresa com a duração real nos dados mais recentes de carregamentos

5. Conclusão

As embarcações PLSV são recursos críticos na exploração e produção de petróleo e gás *offshore*, que realizam operações de interligação de poços submarinos. Para isso é necessário que os navios sejam carregados com os materiais referentes à suas operações. Estes carregamentos podem ser realizados em diferentes berços, e a sua duração, apesar de ser planejada, está sujeita a fatores como condições climáticas, atrasos das embarcações ou aspectos humanos da operação. Atualmente a programação destes carregamentos é feita de maneira manual, e por isso pode não ser o planejamento mais adequado.

Com o intuito de auxiliar no planejamento operacional dos carregamentos, este trabalho propôs a elaboração de um modelo preditivo para realizar a previsão da duração dos carregamentos. Para isso, foram feitas comparações entre o desempenho da *Multilayer Perceptron* com outros modelos como Regressão Linear, *Decision Trees*, *Random Forest*, *Adaboost*, *XGBoost*, *Catboost*, *Support Vector Machines*, entre outros.

Analisando os modelos desenvolvidos, pode-se concluir que os melhores *datasets* para serem utilizados pelo modelo seriam o *dataset* 1 e 2. O *dataset* 3, que foi utilizado com um aumento da base de treinamento a partir do SMOTEGN não apresentou resultados tão bons. A previsão dos modelos preditivos desenvolvidos apresentou resultados melhores do que o planejamento da duração dos carregamentos feito atualmente pela empresa, que possui um erro percentual médio absoluto de 84%.

Dentre os *datasets* 1 e 2, os modelos que apresentaram melhor resultado foram o *Gradient Boosting* e a Regressão Linear, apresentando melhores desempenhos em duas das quatro métricas avaliadas (MSE e R^2). Por fim, analisando a comparação entre a previsão da empresa e dos modelos na validação externa, pode-se concluir que os modelos preditivos desenvolvidos apresentam uma predição mais assertiva para a duração do que a previsão realizada atualmente pela empresa, sendo possível auxiliar melhor no planejamento dos carregamentos de embarcações PLSV.

Como trabalhos futuros, sugere-se testar outros modelos, como modelos baseados em redes recorrentes LSTM e testar outros intervalos de hiperparâmetros e métricas de avaliação.

7. References

- Bnamericas (2021). Petrobras to hire brazilian pipelay vessels. Acessado em 20 de junho de 2022.
- Branco, P., Torgo, L., and Ribeiro, R. P. (2017). Smogn: a pre-processing approach for imbalanced regression.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.
- Dantas, L. (2020). *Predicting the acquisition of resistant pathogens in ICUs using machine learning techniques*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro.

- Ferreira, D. (2013). As principais operações das embarcações plsv. Master's thesis.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92.
- Gkerekos, C., Lazakis, I., and Theotokatos, G. (2019). Machine learning models for predicting ship main engine fuel oil consumption: A comparative study. *Ocean Engineering*.
- Hall, M. A. (1999). Correlation-based feature selection for machine learning. Technical report.
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- Jeon, M., Noh, Y., Shin, Y., Lim, O.-K., Lee, I., and Cho, D. S. (2018). Prediction of ship fuel consumption by using an artificial neural network. *Journal of Mechanical Science and Technology*, 32:5785–5796.
- Kim, Y.-R., Jung, M., and Park, J.-B. (2021). Development of a fuel consumption prediction model based on machine learning using ship in-service data. *Journal of Marine Science and Engineering*, 9:137.
- Kononenko, I., Simec, E., and Robnik-Sikonja, M. (2004). Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7:39–55.
- Mekkaoui, S. E., Benabbou, L., and Berrado, A. (2020). Predicting ships estimated time of arrival based on ais data. *Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*.
- Peng, Y., Liu, H., Li, X., Huang, J., and Wang, W. (2020). Machine learning method for energy consumption prediction of ships in port considering green ports. *Journal of Cleaner Production*, 264:121564.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2017). Catboost: unbiased boosting with categorical features.
- Quinlan, J. R. (1986). Induction of decision trees. *MACH. LEARN*, 1:81–106.
- Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference*, pages 37–52. Springer.
- Stepec, D., Martincic, T., Klein, F., Vladusic, D., and Costa, J. P. (2020). Machine learning based system for vessel turnaround time prediction. *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 258–263.
- Tariq, Z., Aljawad, M. S., Hasan, A., Murtaza, M., Mohammed, E. S., El-Husseiny, A., Alarifi, S. A., Mahmoud, M., and Abdulraheem, A. (2021). A systematic review of data science and machine learning applications to the oil and gas industry. *Journal of Petroleum Exploration and Production Technology*, 11:4339 – 4374.
- Vapnik, V. N. (2000). *Methods of Pattern Recognition*, pages 123–180. Springer New York, New York, NY.