

Article

Application of Machine Learning Techniques to the Maritime Industry

Miguel Ángel Gómez Ruiz ^{*}, Iván Martín de Almeida  and Rodrigo Pérez Fernández 

Escuela Técnica Superior de Ingenieros Navales, Universidad Politécnica de Madrid, 28040 Madrid, Spain; ivan.martin.dealmeida@alumnos.upm.es (I.M.d.A.); rodrigo.perez.fernandez@upm.es (R.P.F.)

^{*} Correspondence: miguel.gruiz@alumnos.upm.es

Abstract: The maritime industry is displaying notable interest in the adoption of cutting-edge technologies within the scope of Industry 4.0, aiming to digitalize both companies and processes. At the core of data science lies machine learning, which serves as the focal point of this article. This study seeks to accomplish two main objectives: firstly, an exploration of various machine learning algorithms, and subsequently, the application of these techniques to analyze predictions within the propulsion system of a 9500 TEU container ship. The outcomes of the study reveal that utilizing distinct machine learning algorithms for predicting braking performance yields a lower mean square error (MSE) when compared to the discrepancy introduced by the J. Mau formula, as evident in the container ship database. The selection of propulsion engines was based on predictions for a 9500 TEU container ship. Similarly, promising outcomes were achieved in predicting propeller diameter in comparison to conventional methods. Thus, these predictions can also effectively guide the appropriate choice of propeller diameter.

Keywords: machine learning; Industry 4.0; ship design; container ship; maritime industry



Citation: Gómez Ruiz, M.Á.; de Almeida, I.M.; Pérez Fernández, R. Application of Machine Learning Techniques to the Maritime Industry. *J. Mar. Sci. Eng.* **2023**, *11*, 1820. <https://doi.org/10.3390/jmse11091820>

Academic Editor: Mihalis Golias

Received: 25 July 2023

Revised: 6 September 2023

Accepted: 12 September 2023

Published: 18 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The broader industry is currently undergoing transformative shifts that align with the Fourth Industrial Revolution, widely recognized as Industry 4.0. This significant industrial progression is primarily propelled by the enhancement of resource management within production processes through data utilization. As a result, companies are actively embracing digital transformation to unlock heightened value.

Within the naval industry, the concept of Industry 4.0 has evolved into more distinct terms like Shipyard 4.0 [1] and Port 4.0 [2]. It is essential to emphasize that Industry 4.0 encompasses more than just the production phase of products (in this context, referring to ships and other maritime items); it encompasses tracking a product throughout its lifecycle and even into material recycling post-use. This ideology has given rise to remote inspections as well. Given the inevitability of digital transformation for numerous enterprises, not solely shipyards, the term Industry 4.0 is employed as a comprehensive descriptor within this paper.

The distinctive aspect of Industry 4.0, in contrast to prior revolutions, is its multi-pronged approach, depicted in Figure 1. These diverse facets of progress do not advance uniformly; while certain areas have experienced rapid evolution, others have undergone more gradual shifts.

The interconnected nature of these facets adds to their intrigue as an advancement in one realm can trigger improvements in others. Nevertheless, it is crucial to recognize that these facets are distinct since they pertain to varying modes of data manipulation and treatment.

The marine sector represents a strategic domain of considerable complexity, as a ship's impact extends beyond its mere construction to subsequent operational phases.

This implies that various factors come into play, including the vessel's intended operating location, prevailing national policies, global diplomatic relations, climate considerations, ongoing regulations, and more.

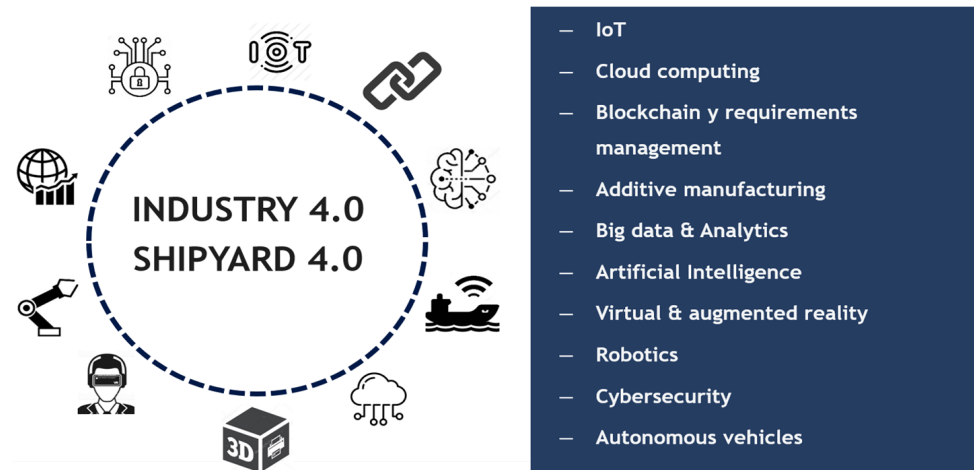


Figure 1. Industry 4.0/Shipyard 4.0 action fronts.

A ship's lifecycle spans approximately 20–25 years. Given their extended operational span, ships often undergo maritime transformations, such as extensions in length or beam, alterations to primary and auxiliary machinery systems, and similar modifications. These transformations can lead to deviations from the cost projections made during the initial design phase of a ship. Industry 4.0 is poised to facilitate the development of more effective strategies through data analysis, ultimately resulting in reduced expenses.

The wealth of ship-related data, ranging from data emanating from onboard sensors to information gathered through worker-conducted surveys, previously posed a challenge due to the multitude of variables influencing it (for example, variables that measure the movement of the ship, structural resistance, and weather conditions). However, Industry 4.0 has now made it feasible to manage and harness this data effectively.

“Among the ten most important strategic technological trends that the consulting firm Gartner has pointed out for the coming years, artificial intelligence, data analytics and digital twins stand out.” [3] (cited in [4] p. 45).

The job displacement attributed to Industry 4.0 is closely linked to the workforce aboard autonomous ships, often referred to as smart ships. The introduction of unmanned engine rooms and checkpoints will undoubtedly yield a substantial impact on employment. However, it is vital not to overlook the intricate nature of the Industry 4.0 landscape and smart ships. Consequently, skilled personnel will be indispensable for managing the artificial intelligence algorithms that bestow autonomy upon these ships. Additionally, a workforce will be necessary to process the ever-expanding data volume, considering its exponential growth over time.

In the maritime realm, advancements like virtual, augmented, and mixed reality have emerged, not solely for simulation purposes but also to diagnose systems or mechanisms without requiring physical disassembly. This approach leads to time and cost savings. Likewise, digital twins have found their place in this domain, serving to study ships or shipyards under various operational scenarios through simulations. The unique advantage of digital twins, as exact replicas of their digitized counterparts, lies in their ability to yield predictions closely aligned with real-world behavior. As we embrace the journey toward decarbonization, digital twins will occupy a pivotal role.

In summary, the digitalization of the maritime industry, coupled with the application of novel machine-learning methodologies, is ushering in uncharted avenues of work that are poised to revolutionize the marine sector.

2. Materials and Methods: Machine Learning and Its Algorithms

Machine learning encompasses a collection of artificial intelligence algorithms employed in data analysis, enabling machines to learn and execute tasks autonomously. This capability is pivotal for systems to effectively analyze and glean insights from extensive datasets. The ultimate aim is to enhance productivity, curtail expenses, and engender favorable outcomes for the industry.

Data analysis can be broadly categorized into two distinct groups, as illustrated in Figure 2, based on their intended objectives:

- **Supervised analyses:** These are employed for predictive purposes and necessitate historical data for problem modeling. In this scenario, one variable is designated as the target variable, while the remaining variables function as predictor variables.
- **Unsupervised analyses:** These are utilized for data structuring and do not rely on historical data for problem modeling. They are reliant on data extracted at the current moment, obviating the need for past behavior records (this is particularly applicable to certain classification algorithms, for instance). All variables within the dataset serve as predictor variables.

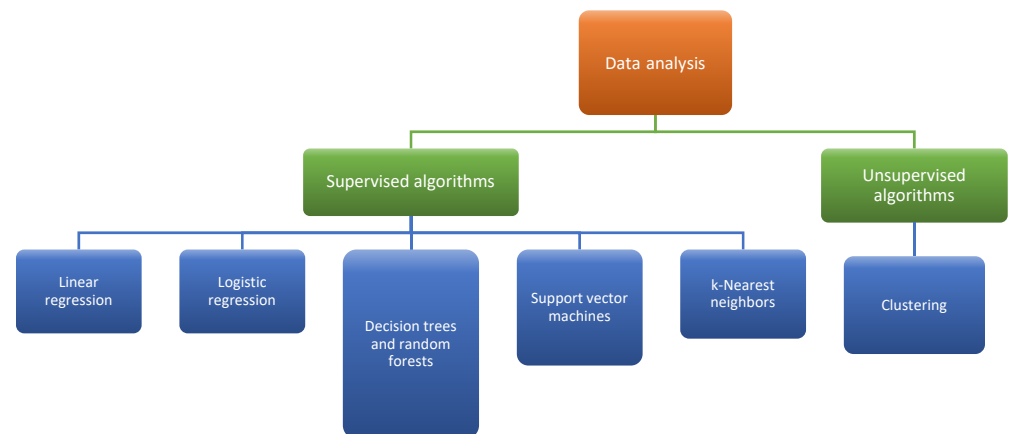


Figure 2. Classification of machine learning algorithms.

The ensuing steps in the data analysis undertaken in this paper, as well as those recommended to be followed, are elucidated in Figure 3.

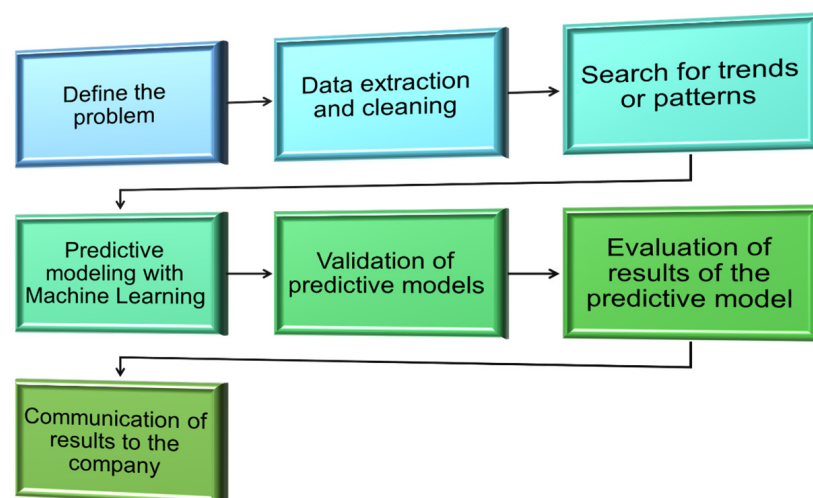


Figure 3. Scheme of the process of data analysis.

2.1. Machine Learning Algorithms

Continuing, we will delve into succinct explanations of several machine learning algorithms to provide a foundational grasp of their functions.

Machine learning methods are extensively used in the maritime industry, for example:

1. Prediction of ship motion and trajectory [5].
2. Evaluation of navigation risk [6].
3. Monitoring ship fuel consumption and optimizing ship design [7].
4. Development and application of green and intelligent inland vessels [8].
5. Estimation of the berthing state of autonomous maritime surface ships [9].
6. Quantifying Arctic oil spill event risk [10].

2.1.1. Linear/Nonlinear Regression

Widely adopted by the scientific community, this algorithm proves instrumental in data analysis. Input variables can span the numeric or categorical spectrum, with the latter entailing binary values: 1 if a prediction pertains to that category, and 0 otherwise.

In this context, an output variable (target variable: Y) is selected, while the remaining variables act as predictors (predictor variables: X). The linear regression equation takes shape as:

$$Y \approx \beta_0 + \beta X \quad (1)$$

where

- Y : Represents the value attributed to the prediction within the regression process.
- X : Reflects the value assumed by the predictor variable (numeric or categorical).
- β_0 : Denotes the coefficient termed the intercept, signifying the y-intercept within the regression equation.
- β : Stands for the coefficient of the input variable X .

Equation (1) entails only two unknowns: the coefficients β_0 and β . The algorithm seeks these coefficients' values that minimize the sum of squared errors, as manifested in Equation (2).

$$\min \sum_{i=1}^n e_i^2 = \min \sum_{i=1}^n (y_i - (\beta_0 + \beta x_i))^2 \quad (2)$$

where

- e_i : Represents the error corresponding to the i -th data point concerning the regression equation.
- y_i : Stands for the value of the output variable within the dataset (i.e., the dataset's output value).
- n : Signifies the database's size, indicative of the number of elements in the dataset.

The same foundational concept extends to scenarios involving multiple variables; the key divergence is the involvement of more than one predictor variable. A parallel application pertains to nonlinear regression, where it is important to note that the resultant regression expression is not a first-degree algebraic form.

2.1.2. Clustering

Clustering, an unsupervised algorithm, operates without the need for historical data. A current database suffices for application. In this scenario, all variables are employed as inputs.

This algorithm segments data through the creation of groups containing similar elements. Both similarities and differences among the data are taken into account during this process. These groups are referred to as clusters.

The foundation for forming these groups lies in the distances between points representing individual data. The distance metric serves as a critical hyperparameter in

clustering. While there are multiple ways to compute distances, for simplicity, this analysis will exclusively employ Euclidean distances, as portrayed in the general Equation (3).

$$D_{Euclidean}(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2} \\ = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3)$$

where

- x_i, x_j : Represent two points for which the distance is calculated.
- Subscripts i or j denote the respective points.
- Subscript k pertains to each variable.

This concept extends into three-dimensional space (each variable occupies a distinct axis: X, Y, or Z) as depicted in Equation (4).

$$D_{Euclidean}(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (4)$$

It is crucial to acknowledge the varying ranges of different variables, which necessitates distance normalization. This is accomplished through Equation (5).

$$Unnormalized\ data \rightarrow z_{i,j} = \frac{x_{i,j} - mn(x_{*,j})}{mx(x_{*,j}) - mn(x_{*,j})} \rightarrow Normalized\ data \quad (5)$$

where

- $z_{i,j}$: The element in the normalized database.
- $x_{i,j}$: The element in the unnormalized database.
- $min(x_{*,j})$: The minimum element in column j of the unnormalized database.
- $max(x_{*,j})$: The maximum element in column j of the unnormalized database.

In this study, agglomerative hierarchical clustering will be employed. This algorithm initially creates n clusters (one per data point) and iteratively merges them until a predetermined number of clusters is achieved.

Hierarchical clustering is often represented visually using a dendrogram, similar to the illustration shown in Figure 4. The positioning of a threshold holds significant importance, as it determines whether data points are organized into a larger or smaller number of clusters. Raising the threshold on the dendrogram results in fewer clusters while lowering it leads to a greater number of clusters being formed.

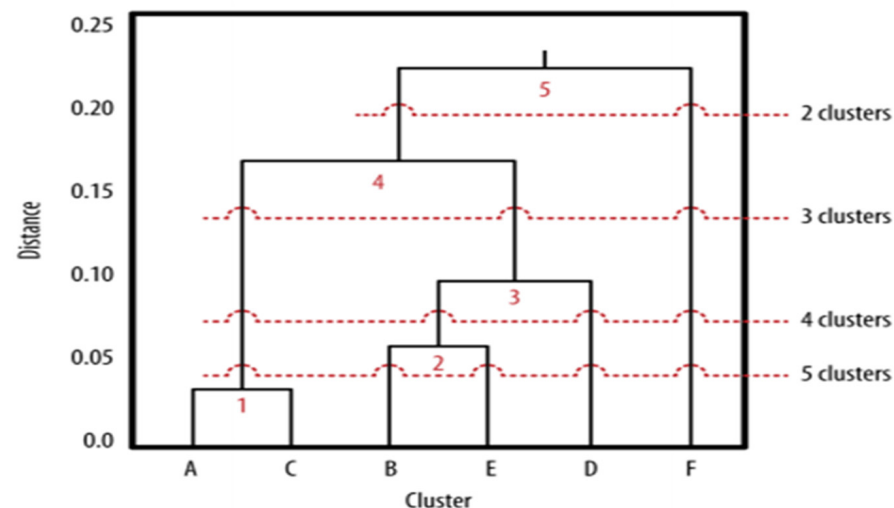


Figure 4. Dendrogram in a hierarchical clustering. Own elaboration from [4].

2.1.3. Decision Trees and Random Forests

As members of the supervised algorithm category, decision trees demand historical data for their functioning. Their core role involves data classification, a process resembling clustering. Nonetheless, a significant divergence emerges in their application intent. While clustering aims to create structural organization within data, decision trees are specifically designed for making predictions.

For a visual depiction of decision trees, please refer to Figure 5:

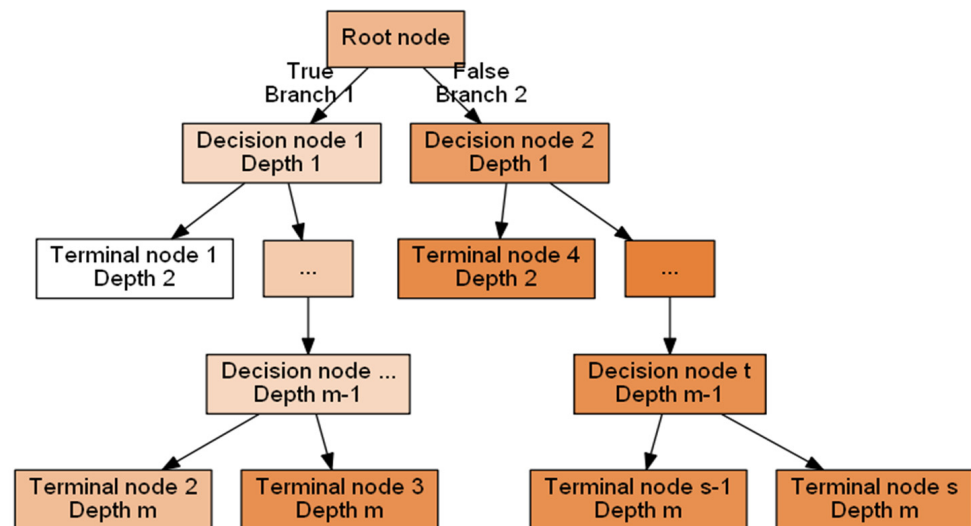


Figure 5. Decision tree structure.

As illustrated in Figure 5, the nomenclature of this algorithm stems from its visual development, reminiscent of a tree with its branching structure.

The tree's inception lies within a root node. At each node, decisions are made based on variable values. If a condition is met, the data branch left; failing that, the condition leads to a rightward branch. Notably, terminal nodes—also known as leaf nodes—abstain from decision-making. Instead, they furnish the final prediction. Figure 5 underscores this concept by displaying the absence of further branches beyond these terminal nodes.

For classification tasks, it is advisable to attain maximum homogeneity within the data. Higher homogeneity reduces the information required to represent the decision tree. Homogeneity assessment employs two metrics: information entropy and information gain/loss.

The entropy formulation for a target variable is encapsulated in Equation (6):

$$H(X) = \sum_{i=1}^n -p_i \log_2(p_i) \quad (6)$$

where

- p_i : The probability of occurrence (it is the proportion occurrence of the class that is predicted since it is a statistic that is obtained by maximum likelihood) of the class i .
- n : Number of categories for each target variable.

Every node in the tree possesses an associated information entropy value. An analogy to thermodynamic entropy reveals that the null value represents an optimal outcome, signifying a defect-free structure. The algorithm strives to minimize entropy, with zero being the ideal value. The process of entropy reduction is termed information gain, with greater reduction yielding higher information gain. Smaller entropy and greater information gain lead to more homogenous information for each target variable.

2.1.4. Regression Trees

Regression trees bear a resemblance to decision trees. Employed when the decision variable is continuous, regression trees address data with non-linear or non-logistic behavior. These trees create models to capture complex nonlinear relationships. Notably, if output variable values are missing, they can be averaged over terminal nodes.

- In each node, samples are present: $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$
- The prediction is calculated as the average: $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$

2.1.5. Random Forests

In a manner that is intuitive and can be likened to an analogy, a random forest is comprised of a fusion of multiple decision or regression trees. However, the noteworthy aspect here is that the hyperparameters (such as the minimum number of samples per node, minimum and maximum tree depth, etc.) are not rigidly set. Instead, these hyperparameters exhibit random variations for each tree constituting the forest.

Within the forest, a diversity of trees is present, some with greater depth and others with shallower depth. Similarly, there are trees with more terminal nodes, and those with fewer. Each tree stands as a distinct model. By incorporating an array of models, the reduction of variance is optimized.

Several parameters hold importance in this context:

- Node size. Typically set at a default value of 1 in Python[®] 3.11.5 (Beaverton, OR, USA).
- Number of trees. A starting point of around 500 trees is recommended, although careful consideration is advised when increasing this parameter, as larger values lead to longer algorithm execution times.
- Number of predictor variables sampled. A default range of 2 to 5 is often employed.

2.1.6. Support Vector Machines (SVM)

Support Vector Machines (SVM) stand out as a potent machine learning algorithm, especially valuable for classification tasks, such as clustering, when dealing with an abundance of input variables within a high-dimensional space. Notably, SVM is a supervised algorithm, necessitating historical data for its proper application.

The fundamental operation of SVM involves employing kernel functions to achieve a linear transformation into a higher-dimensional vector space. This transformation simplifies the classification process significantly. Essentially, when dealing with a dataset existing within a vector space whose dimensions match the number of predictor variables, the algorithm elevates it to a more intricate p -dimensional vector space, boasting greater dimensionality. Following this transformation, the algorithm endeavors to identify a hyperplane with $p - 1$ dimensions. This hyperplane serves as the means to segregate the data into distinct groups.

Initially, the algorithm constrains errors within defined margins, often referred to as a “corridor” in machine learning parlance. The margins are ultimately shaped by the support vectors—essential data points pivotal to the model’s efficacy. The algorithm’s primary mission is to locate a dividing hyperplane equipped with the widest feasible corridor. Data situated on one side of this corridor are classified into a designated group, while the opposite side houses a distinct group. Any data extending beyond this corridor signify a misclassification.

In the pursuit of robust predictions, broader corridors hold an advantage over narrower ones, as the latter can often result in overfitting, a phenomenon illustrated in Figure 6.

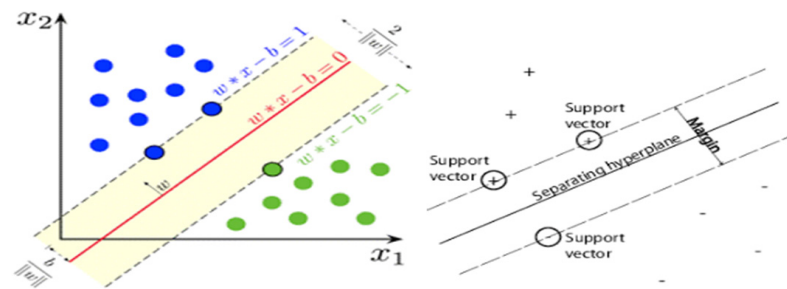


Figure 6. On the left, classification using SVM [11]. On the right, elements of support vector machines [12].

C is a hyperparameter of the model:

- Choosing a high C hyperparameter can cause overfitting problems. The corridor narrows, so there will be a greater restriction in the error bands.
- Choosing a low C hyperparameter can cause underfitting problems. The corridor widens, so there will be fewer restrictions on the error bands.

2.1.6.1. k Nearest Neighbors

The k nearest neighbors algorithm operates as a supervised algorithm, necessitating historical data for its functionality. Its primary objective involves classifying new data within established classifications, whether through logistic regression, clustering, or similar techniques.

When incorporating new observations, the algorithm identifies the k nearest neighbors from previously classified data. Subsequently, the new data point is classified within the cluster boasting the highest neighbor count. Here, the parameter k , initially provided by the engineer, dictates the number of neighbors to consider.

Referencing Figure 7, the algorithm's purpose becomes evident: classifying a fresh observation (designated as x) within a pre-existing clustering context. To achieve this, the user sets $k = 3$, signifying the selection of the three nearest neighbors (these points lie within the circular region). Given the greater representation of neighbors from the red cluster (2) compared to the blue cluster (1), the algorithm classifies the new point within the red cluster.

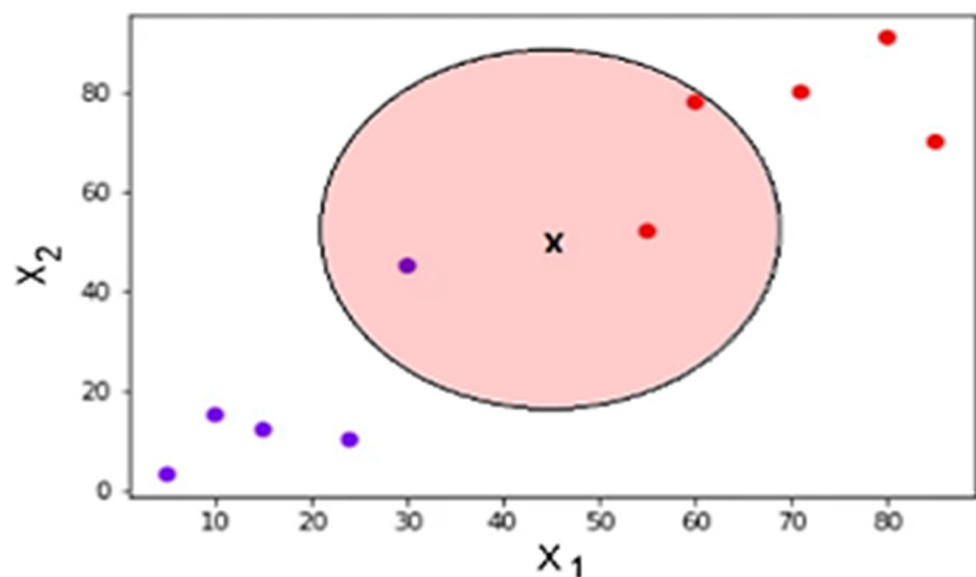


Figure 7. Clustering a new point with the k -nearest neighbors' algorithm. Own elaboration from [13].

2.2. Choice of Algorithm

Choosing the most suitable algorithm can sometimes prove to be a complex decision, and even deploying multiple algorithms remains a viable option.

Figure 8 shows a schematic illustration aiding the algorithm selection process.

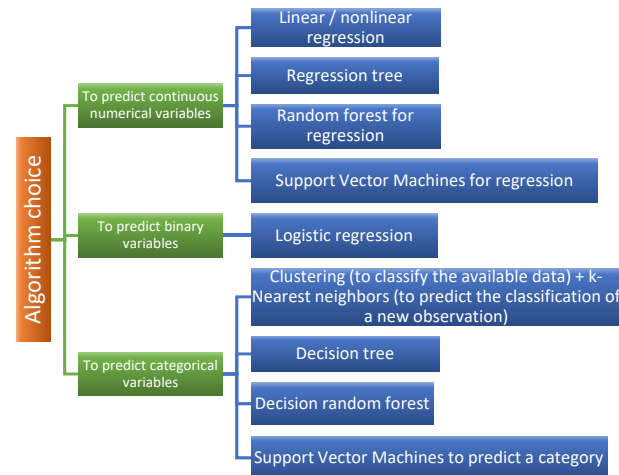


Figure 8. Scheme to choose the algorithm to use.

In Figure 8, a diagram is provided to assist in the selection of algorithms from those covered in this chapter, based on the specific output variable.

In essence, the optimal model would be the one yielding superior predictive performance, essentially minimizing errors. However, a common practice is to amalgamate multiple algorithms. This approach allows us to take into account predictions from all the models developed, thereby enhancing the overall understanding and predictive power.

2.3. Validation and Evaluation of the Models

The validation process, in its essence, is straightforward. It involves partitioning the database into two distinct sets: a training set (comprising the larger portion, approximately 80% of the data) and a test set (consisting of the smaller portion, roughly 20% of the data).

The training set serves as the foundation for the previously mentioned algorithms; it is used to construct the predictive model.

Conversely, the test set is employed to assess the performance of the constructed model's predictions. This is where we calculate the error incurred by the predictive model.

Upon confirming the reliability of the predictive model, it is crucial to delve into the outcomes yielded by the exploratory data analysis. Grasping the context of these results is vital for evaluating their coherence. Should any inconsistencies arise, a revisit to the preceding steps might be necessary to rectify such disparities.

3. Results Applied on a Container Ship

We will now apply the techniques detailed above to analyze container ships. Given the focus on container ships, the available database comprises fixed and controllable pitch propellers, along with maneuvering propellers.

The propellers utilized in this context are fabricated from materials like *Mn* and *Ni–Al* bronze, brass, and stainless steel. Notably, the container ships within the compiled database, sourced from the Significant Ships database [14–17], are constructed using bronze and *Ni–Al* alloys for their propellers.

In the conventional setup, a single line of shafts is employed, with fixed-pitch propellers in cases where no reduction gear is utilized.

For our specific use case, we have selected the following ship data:

- Ship Type: cellular container ship.
- TEUs: 9500 (post-Panamax family).

- Service speed (at 90% of MCR): 22.5 knots.
- Deadweight: 123,500 tons.
- Number of propellers: 1.
- Propeller type: fixed pitch.
- Length between perpendiculars: 303 m.
- Scantling draught: 14.5 m.

4. Brake Power Prediction

4.1. Estimation of Propulsive Power before Analysis

Before delving into the realm of machine learning, we will first carry out an estimation of the propulsive power using the J. Mau formula [18], from which Equation (7) is derived. The Holtrop and Mennen method is not being employed at this stage, as it necessitates preliminary ship sizing and data obtained from onboard sensors—prerequisites that align with the application of machine learning. This initial estimation will serve as a benchmark against which we can evaluate any enhancements achieved through the implementation of predictive models using machine learning.

$$BHP(bhp) \approx 0.0114 \cdot V_s^3 \cdot DWT^{0.55} \quad (7)$$

where

- DWT : Deadweight in tons.
- V_s : Speed in knots, under average service conditions.
- BHP : Power delivered by the drive motor to the driveline, measured in imperial units as brake power.

Now, we will substitute the ship's specifications, as detailed in Section 3, into Equation (7).

$$\begin{aligned} BHP(bhp) &\approx 0.0114 \cdot (22.5)^3 \cdot (123,500)^{0.55} = 82,010.46(bhp) \\ 82,010.46(bhp) \cdot 0.7355(kW/bhp) &= 60,318.69(kW) = 60.32(MW) \end{aligned}$$

Prior to embarking on the development of machine learning models, our initial step will involve defining the problem, following the structure outlined in Figure 3 within the context of data analysis.

4.2. Definition of the Problem

In order to establish the problem's framework, it is imperative to initially define the objective variable, followed by the determination of the input variables. In this study, the focal point is the brake power (BHP), as depicted in Equation (8), which functions as a derivative of:

$$BHP = BHP(N, V_s, L_{BP}, DWT, TEUs) \quad (8)$$

where

- N (rpm): Engine revolution per minute (which will be equal to the propeller revolutions per minute, except for very few cases that will be considered outliers since they are vessels that have a reduction gear).
- V_s (kn): Ship service speed.
- L_{BP} (m): Length between perpendiculars.
- DWT (t): Deadweight.
- $TEUs$: Number of containers (Twenty Equivalent Units).

4.3. Preparation of the Database

A database has been compiled from the documents of several *Significant Ships* [14–17] from various years (see Table 1). The database has been collected in a CSV file and from this file, a reading was made in Python® to be able to implement the machine learning algorithms with the Statsmodels and Scikit-learn libraries.

Note that in the database there are both numerical and categorical variables.

Table 1. Sample of the first 10 rows and columns of the database. Own elaboration from [14–17].

	Name	Year	TEUs	Length over All (m)	Length between Perpendiculars (m)	Breadth Moulded (m)	Depth Moulded to Main Deck (m)	Depth Moulded to Upper Deck (m)	Depth Moulded to Other Decks (m)	Width of Double Skin Side (m)
0	MSC Altamira	2012	8900	299.18	286.00	48.2	24.80	24.8	20.140	48.20
1	CAP Ines	2010	4600	264.32	251.00	32.2	21.50	21.5	17.358	2.06
2	Frisia Bonn	2010	2000	189.50	178.00	16.5	NaN	NaN	NaN	2.19
3	Hanjin Korea	2010	10,000	349.70	334.00	45.2	NaN	27.2	NaN	2.33
4	Maule	2010	6600	305.60	293.16	40.0	24.20	NaN	NaN	NaN
5	MSC Berly	2010	13,000	365.80	349.80	48.4	22.94	29.9	NaN	2.49
6	APL Southampton	2012	10,640	347.00	331.00	45.2	22.59	29.7	NaN	2.17
7	BBC Amethyst	2012	985	153.80	145.50	23.2	12.80	NaN	NaN	2.60
8	CMA Marco Polo	2012	16,000	396.00	378.40	53.6	29.90	NaN	NaN	2.59
9	Ever Lambent	2012	8500	335.00	317.00	45.8	NaN	25.0	NaN	14.20

4.4. Pattern Visualization

To observe data trends, visual representations such as graphs can be employed. Additionally, the correlation between variables can be assessed through correlation matrices, as illustrated in Equation (9).

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\text{std}(X)\text{std}(Y)} \quad (9)$$

The correlation coefficient operates within the domain of $[-1, 1]$ and is dimensionless.

- When the coefficient approaches **1**, there is a stronger direct correlation (as one variable increases, the other also increases proportionally).
- When it approaches **−1**, there is a strong inverse correlation (as one variable increases, the other decreases proportionally).
- As the coefficient approaches **0**, the variables exhibit less interdependence.

Given this consideration, it is intriguing to note that the correlations between the input variables and the output variable exhibit significant levels of direct or inverse proportionality.

In the context of this study, correlation values falling within the ranges of **−1** to **−0.5** and **0.5** to **1** are considered high.

In Section 5, the correlation matrix will be depicted, allowing for an examination of the correlations between the brake power and the other variables, as well as between the propeller diameter and the remaining variables.

Visualized in the form of a heatmap within Figure 9, positive correlations are indicated by warm colors, while cool colors signify negative correlations.

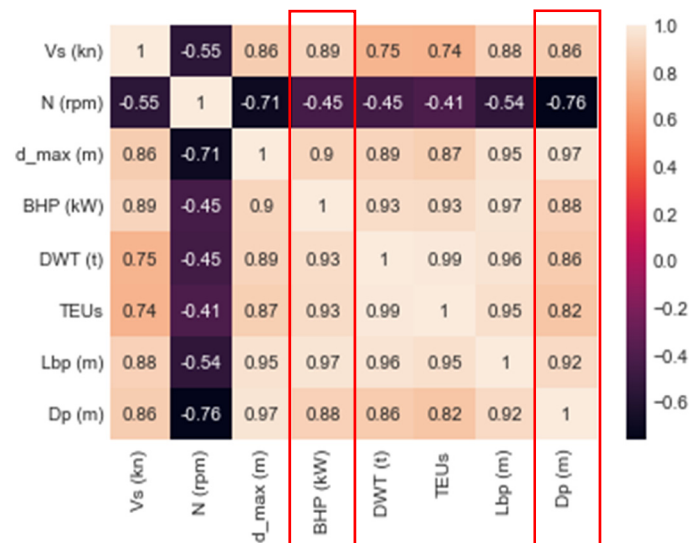


Figure 9. Correlation matrix represented in a heat map.

For deriving brake power, there are notable correlations, excluding engine revolutions per minute. This is understandable as brake power is also influenced by engine displacement.

4.5. Elaboration of the Model with Machine Learning

Since the target variable to be predicted (brake power) is a continuous numerical value, following the framework depicted in Figure 8, the chosen algorithms include:

- Linear/nonlinear regression.
- Regression trees.
- Random forests for regression.

4.5.1. Linear/Nonlinear Regression

To commence the analysis, regression has been selected due to its widespread recognition in the scientific community.

A summarized overview of the outcomes from Model 17, along with the regression coefficient, can be found in Table 2.

$$\text{Comparative : } \begin{cases} J.Mau : BHP(\text{bhp}) \approx 0.0114 \cdot Vs^3 \cdot DWT^{0.55} \\ \text{Model 17 : } BHP(\text{kW}) \approx 0.0069 \cdot Vs^3 \cdot DWT^{0.55} \end{cases} \quad (10)$$

Both equations utilize the parameters: V_s (kn) and DWT (tons).

In this scenario, the regression within the parameter space can be graphically depicted, as seen in Figure 10. This illustration is applicable because it involves two predictor variables (V_s and DWT) and one target variable (BHP).

4.5.2. Regression Tree

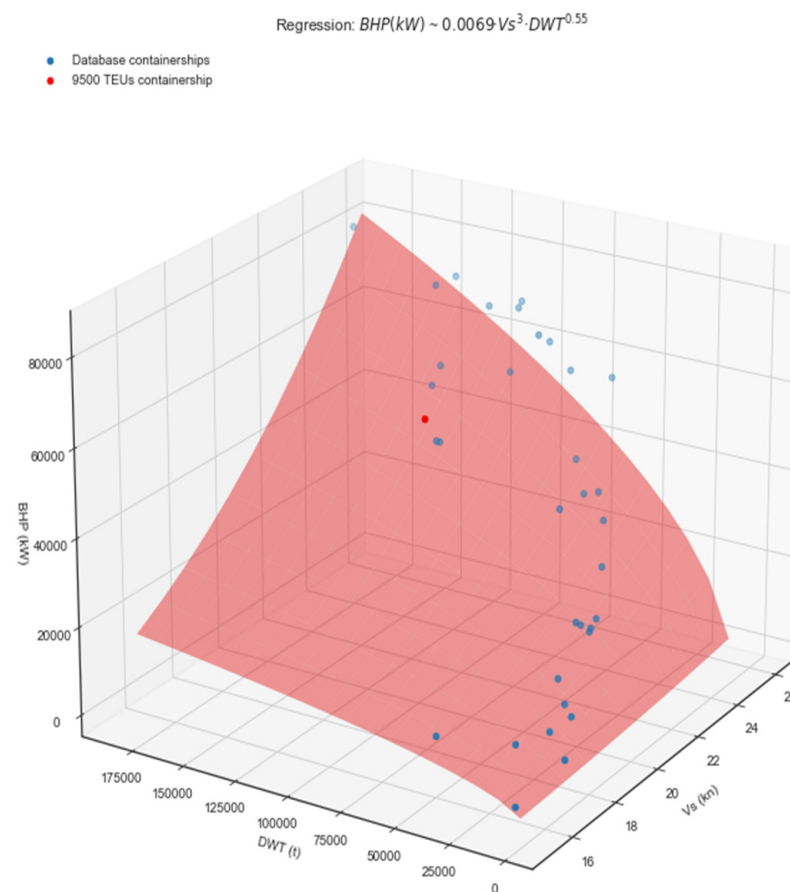
The power development has previously been analyzed, revealing its non-linear nature. A regression tree can serve as another valuable algorithm for predicting the brake power that aligns well with the ship's specifications. Furthermore, its advantage over regression lies in not requiring a predetermined function to outline the variable relationships.

Upon applying this model, the ensuing result has been achieved:

As shown in Figure 11, the tree performs a classification based on the input variables and based on this classification, returns the power prediction for said classification.

Table 2. Model 17 analysis summary—prediction of BHP (kW)—performed in Python®.

OLS Regression Results						
Dep. Variable:	BHP_KW	R-squared(uncentered):	0.994			
Model:	OLS	Adj.R-squared(uncentered):	0.994			
Method:	Least Squares	F-statistic:	5604			
Date:	Tue, 22 February 2022	Prob (F-statistic)	2.02×10^{-38}			
Time:	14:19:09	Log-Likelihood:	−325.33			
No. Observations:	34	AIC:	652.7			
Df Residuals:	33	BIC:	654.2			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	p > [t]	[0.025	0.975]
np.power(Vs, 3):np.power(DWT, 0.55)	0.0069	9.24×10^{-5}	74.857	0.000	0.007	0.007

**Figure 10.** Spatial representation of the regression of brake power.

4.5.3. Random Forest for Regression

The random forest is a derivation of the previous model. What this model generates are multiple random trees and performs the average of said predictions.

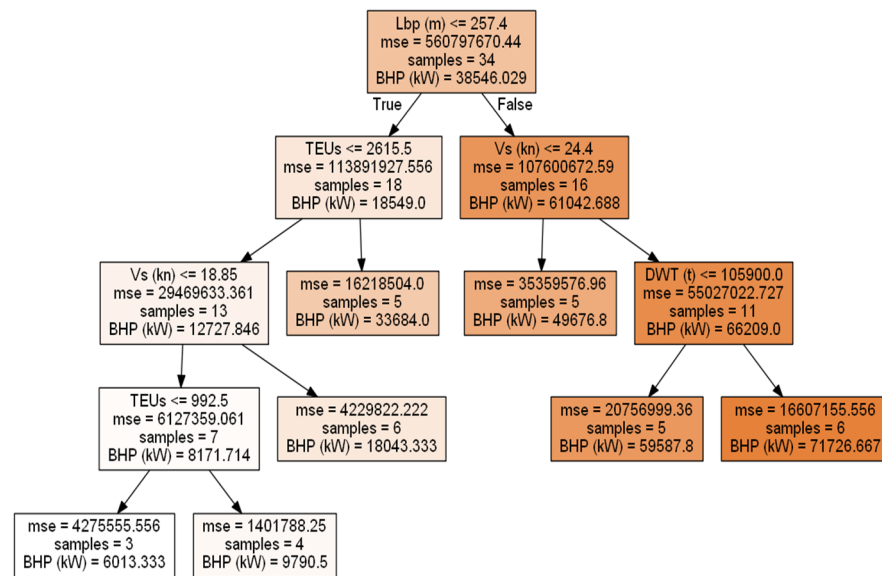


Figure 11. Regression tree to estimate brake power BHP (kW)—generated in Python®.

A technique that is used very often in the validation of models is the technique known as K Fold Cross Validation. This technique consists of splitting the dataset into training and test sets in k iterations. MSE is obtained in the following way, with Equations (11)–(13).

$$SSD_j = \sum_{i=1}^n (y_{i,j} - y(x_{i,j}))^2; \text{ para } j = 1, \dots, k \quad (11)$$

$$MSE_j = \frac{SSD_j}{n} \quad (12)$$

$$MSE = \frac{\sum_{j=1}^k MSE_j}{k} \quad (13)$$

where

- $y_{i,j}$: The value of the output variable, in this case, *BHP* (kW), that we have in the test data set of the j -th iteration.
- $y(x_{i,j})$: The value of the prediction that is estimated from the output variable, in this case, *BHP* (kW), which is found in the test data set of the j -th iteration.
- SSD_j : The sum of the squares of the differences at the j -th iteration.
- n : The size of the test data set, which in all iterations for this case is going to be 9 vessels.
- MSE : The average of the mean square error.
- MSE_j : The mean square error that is committed for the j -th iteration. That is, the error made for the partition into the training and test set at the j -th iteration.
- i : i -th position of the data in the test set.
- j : j -th position of the cross-validation iteration.
- k : Total number of iterations performed when using the cross-validation technique.

As can be seen in Figure 12, lower errors are achieved with machine learning compared to the estimates provided by [19], pp. 595,602–603.

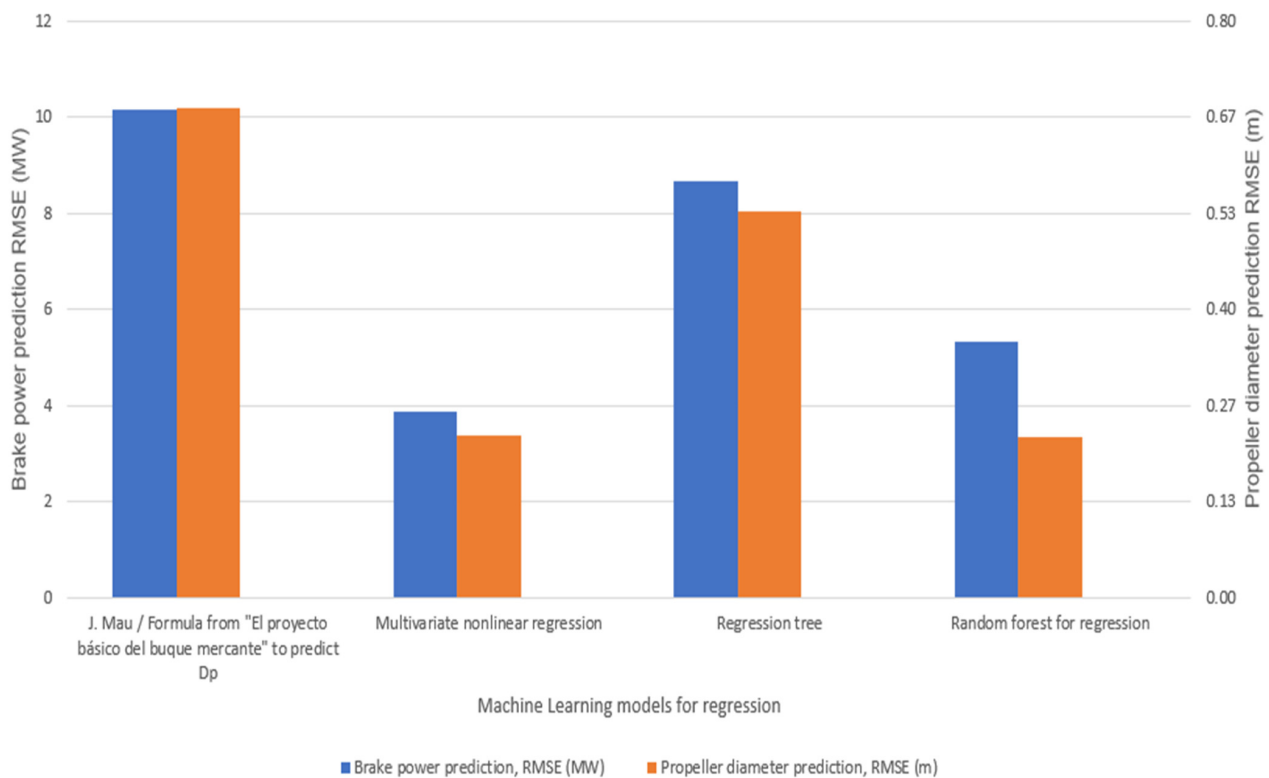


Figure 12. Errors committed with each model in the predictions.

4.6. Brake Power Estimation after Analysis

Recalling the results of the estimation prior to the application of machine learning, the brake power had been estimated with the J. Mau formula.

$$BHP \text{ (MW)} = 60.32 \text{ (MW)} \pm 10.15 \text{ (MW)}$$

Figure 13 shows the brake power of the 9500 TEUs container ship according to machine learning models.

```

1 predictors
['Vs (kn)', 'TEUs', 'N (rpm)', 'DWT (t)', 'Lbp (m)']

1 #Brake power is predicted using the specification data
2 Vs = 22.5 #(kn)
3 TEUs = 9500
4 N = 78 #(rpm)
5 DWT = 123500 #(t)
6 Lbp = 303 #(m)
7
8 print('Brake power - Nonlinear regression, BHP =',[0.0069*(Vs**3)*(DWT**0.55)],'(kW)')
9 print('Brake power - Regression tree, BHP =',regtree.predict([[Vs, TEUs, N , DWT, Lbp]]),'(kW)')
10 print('Brake power - Random forest for regression, BHP =',regforest.predict([[Vs, TEUs, N , DWT, Lbp]]),'(kW)')

Brake power - Nonlinear regression, BHP = [49637.908224960236] (kW)
Brake power - Regression tree, BHP = [49676.8] (kW)
Brake power - Random forest for regression, BHP = [51153.755] (kW)

```

Figure 13. Brake power of the 9500 TEUs container ship according to machine learning models.

According to Model 17 of the regression algorithm, using Equation (10):

$$\begin{aligned}\text{Model 17 : } BHP \quad (\text{kW}) &= 49,637.908 (\text{kW}) \pm 3879.304 (\text{kW}) \\ &= 49.64 (\text{MW}) \pm 3.88 (\text{MW})\end{aligned}$$

According to the regression tree:

$$BHP (\text{kW}) = 49,676.8 (\text{kW}) \pm 8653.039 (\text{kW}) = 49.68 (\text{MW}) \pm 8.65 (\text{MW})$$

According to the random forest for regression:

$$BHP (\text{kW}) = 51,153.755 (\text{kW}) \pm 5342.73 (\text{kW}) = 51.15 (\text{MW}) \pm 5.34 (\text{MW})$$

The algorithm with the least error committed is the nonlinear regression of Model 17. Therefore:

$$BHP = 49.64 (\text{MW}) \pm 3.88 (\text{MW})$$

5. Prediction in the Propeller Diameter

Since the steps to follow are the same as in the previous section, the details of the procedure will not be entered again and only the results obtained will be discussed.

5.1. Estimation of Propeller Diameter before Analysis

For an estimation (prior to the analyses) of the propeller diameter, we used the formula provided by [19], pp. 602–603.

To use this formula, it is enough to know the brake power and the propeller revolutions per minute.

$$Dp (\text{m}) \approx 15.75 \cdot \frac{BHP^{0.2}}{N^{0.6}} \quad (14)$$

where

- Dp (m): The propeller diameter in meters.
- BHP (bhp): The brake horsepower in imperial units.
- N (rpm): The propeller revolutions. For the ship that is being studied, it will not have a reduction gear since the selected engine, in Section 4.6, is a low-speed engine: 84 (rpm); therefore, $N_{engine} = N_{propeller}$.

Therefore, it is possible to calculate the propeller diameter in meters Dp (m) with the following data:

$$\begin{aligned}N &= 84 (\text{rpm}) \\ BHP &= 49,920 (\text{kW}) / 0.7355 (\text{kW/bhp}) = 67,872.2 (\text{bhp}) \\ Dp (\text{m}) &\approx 15.75 \cdot \frac{BHP^{0.2}}{N^{0.6}} = 15.75 \cdot \frac{(67,872.2)^{0.2}}{84^{0.6}} = 10.2 (\text{m})\end{aligned}$$

5.2. Definition of the Problem

The target variable is the propeller diameter, Dp .

For the study to be conducted, the propeller diameter, Dp , will be considered as a function of:

$$Dp = Dp(BHP, N, V_S, DWT, TEUs, L_{BP}, d_{max}) \quad (15)$$

where

- BHP (kW): Brake power.
- N (rpm): Propeller revolutions.
- V_S (kn): Ship service speed.
- DWT (tons): Deadweight.
- $TEUs$: Number of containers $TEUs$ (twenty equivalent units).
- L_{BP} (m): Length between perpendiculars.

- d_{max} (m): Scantling draught

5.3. Elaboration of the Model with Machine Learning

The machine learning algorithms to be used are the same as for Section 4.5 of this document.

5.3.1. Linear/Nonlinear Regression

On this occasion, only a regression that coincides dimensionally with the formula provided by [19], pp. 602–603, for which Table 3 is detailed, is used.

Table 3. Summary of analysis of the applied model—prediction of D_p (m)—performed in Python®.

Applied Model						
Dep. Variable:	Dp	R-squared(uncentered):	0.999			
Model:	OLS	Adj.R-squared(uncentered):	0.999			
Method:	Least Squares	F-statistic:	3.789×10^4			
Date:	Tue, 2 February 2021	Prob (F-statistic)	4.91×10^{-44}			
Time:	14:13:53	Log-Likelihood:	3.9340			
No. Observations:	28	AIC:	−5.868			
Df Residuals:	27	BIC:	−4.536			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	p > [t]	[0.025	0.975]
np.power(Vs, 3):np.power(DWT, 0.55)	15.4705	0.079	194.665	0.000	15.307	15.634

The result of the model is:

$$D_p \approx 15.4705 \cdot \frac{BHP^{0.2}}{N^{0.6}} \quad (16)$$

5.3.2. Regression Tree

The propeller diameter has already been studied and it is known that it is not a linear expression.

A regression tree can be another useful algorithm to predict the propeller diameter that would best fit the ship's specifications. In addition, it has the advantage over regression that it is not necessary to predefine with a function how the variables are related.

With the same input and output variables, a regression tree was constructed, see the result in Figure 14.

It is possible to enlarge the tree a little more by varying the hyperparameters, but care must be taken not to overfit the model.

5.3.3. Random Forest for Regression

Likewise, with the same input and output variables, a random forest for regression (composed of 1000 trees) was constructed. Remember that the rest of the hyperparameters vary randomly.

5.4. Goodness of Fit of the Models

Once again, the error is quantified following the k fold cross-validation technique that was explained in Section 4.6.

See in Figure 12 the errors committed in the prediction of the propeller diameter, D_p .

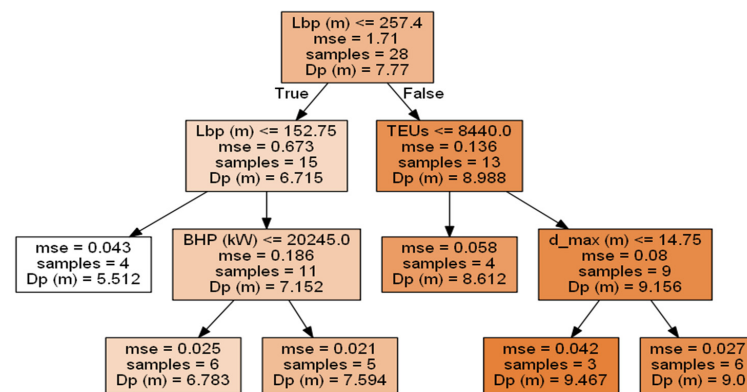


Figure 14. Regression tree to estimate the propeller diameter D_p (m)—generated in Python®.

5.5. Estimation of Propeller Diameter after Analysis

According to Equation (14), provided by [19], pp. 602–603 (applied in Section 5.2), one has:

$$D_p = 10.2 \text{ (m)} \pm 0.68 \text{ (m)}$$

According to nonlinear regression using Equation (16):

$$D_p = 15.4705 \cdot \frac{BHP^{0.2}}{N^{0.6}} = 15.4705 \cdot \frac{49,920^{0.2}}{84^{0.6}}$$

$$D_p = 9.43 \text{ (m)} \pm 0.2247 \text{ (m)}$$

According to the regression tree:

$$D_p = 9.4667 \text{ (m)} \pm 0.5349 \text{ (m)}$$

According to the random forest for regression:

$$D_p = 9.4529 \text{ (m)} \pm 0.2236 \text{ (m)}$$

The best predictive algorithms are random forest for regression and nonlinear regression, as it is stated in Figure 15. Both can be considered good. Therefore, one option is to choose the average value of the diameter predictions for both algorithms.

```

1 predictors
['Vs (kn)', 'TEUs', 'N (rpm)', 'DWT (t)', 'Lbp (m)', 'BHP (kW)', 'd_max (m)']

1 #Propeller diameter is predicted using the specification data
2 Vs = 22.5 #(kn)
3 TEUs = 9500
4 N = 84 #(rpm)
5 DWT = 123500 #(t)
6 Lbp = 303 #(m)
7 BHP = 49920 #(kW)
8 d_max = 14.5 #(m)
9
10 print('Propeller diameter - Nonlinear regression, Dp = ', [15.4705*(BHP**0.2)*(N**-0.6)], '(m)')
11 print('Propeller diameter - Regression tree, Dp = ', regtree.predict([[Vs, TEUs, N, DWT, Lbp, BHP, d_max]]), '(m)')
12 print('Propeller diameter - Random forest for regression, Dp = ', regforest.predict([[Vs, TEUs, N, DWT, Lbp, BHP, d_max]]),
13       '(m)')

Propeller diameter - Nonlinear regression, Dp = [9.43173725408558] (m)
Propeller diameter - Regression tree, Dp = [9.46666667] (m)
Propeller diameter - Random forest for regression, Dp = [9.45295] (m)
  
```

Figure 15. Propeller diameter of the 9500 TEUs container ship according to machine learning models.

6. Classification of Materials Used in the Propulsive System

When devising a system, a crucial task involves selecting the appropriate materials and considering a balance between the demanded mechanical behavior and material cost. Opting for a material that best aligns with performance needs is pivotal. In this regard, machine learning techniques can be employed to achieve classifications via clustering. This entails organizing a database comprised of materials for maritime usage, wherein materials with similarities are grouped while those with greater disparities are separated. The parameter guiding this distinction is the distance among data points. This approach offers a substantial advantage as it permits consideration of which material closely matches the imposed requirements. This algorithm could be directly integrated with an optimization problem.

For the purpose of this study, the problem will be streamlined by focusing solely on the main propulsion system.

Before delving further into the application of these algorithms, it is prudent to have a preliminary understanding of the materials used in propulsion systems.

We will start with the shaft. The shaft is primarily affected by corrosion, fatigue, and fretting corrosion. Coatings are employed due to the highly corrosive environment, although they remain vulnerable at the junctures between the metal components and coatings. Enhanced frictional fatigue of the tail shaft can be achieved by using a cold-rolled shaft surface; while this may not entirely prevent crack formation, it does impede their propagation [20], p. 938.

Weldability is another vital aspect, as cracks stemming from corrosion fatigue can be rectified through grinding followed by the application of filler material through welding, thereby extending the shaft's lifespan.

Commonly used materials for merchant ship shaft lines include high-strength steels which boast high carbon content, although their mechanical properties can diminish due to the corrosive environment. High-strength steel is employed for merchant ships with substantial power, such as the container ship project covered in this document.

Additional materials used encompass nickel–copper alloys (MONEL), precipitation-hardened stainless steels, manganese bronzes (α – β brasses), and more.

Now, turning to the propeller. When selecting propeller materials, the primary aim revolves around meeting the following properties:

- Favorable castability traits;
- Cavitation resistance;
- Corrosion–erosion resistance;
- Strength and ductility;
- Weldability;
- Cost.

Copper-based alloys have gained widespread use for marine propellers. These alloys offer sound mechanical properties and enhanced resistance to biofouling due to copper's biocidal properties. Over the years, the trend has shifted toward employing high-strength nickel–aluminum bronzes and manganese–nickel–aluminum bronzes [20], p. 938.

In certain specialized applications, materials like 12% Cr stainless steels, austenitic stainless steels, or titanium alloys may be used (though titanium comes with high mechanical performance but a corresponding high cost) [20], p. 938.

6.1. Preparation of the Database

A database extracted from GRANTA EduPack 2020®, Figure 16, is elaborated from the Level 3 database, which is a very extensive and detailed database of materials. In this database, the materials used in propulsion systems have been filtered from the entry: propeller.

The collected variables encompass the mechanical properties of materials, price, corrosion resistance, durability in diverse environments, CO₂ footprint for production and recycling, weldability, and castability properties, and more.

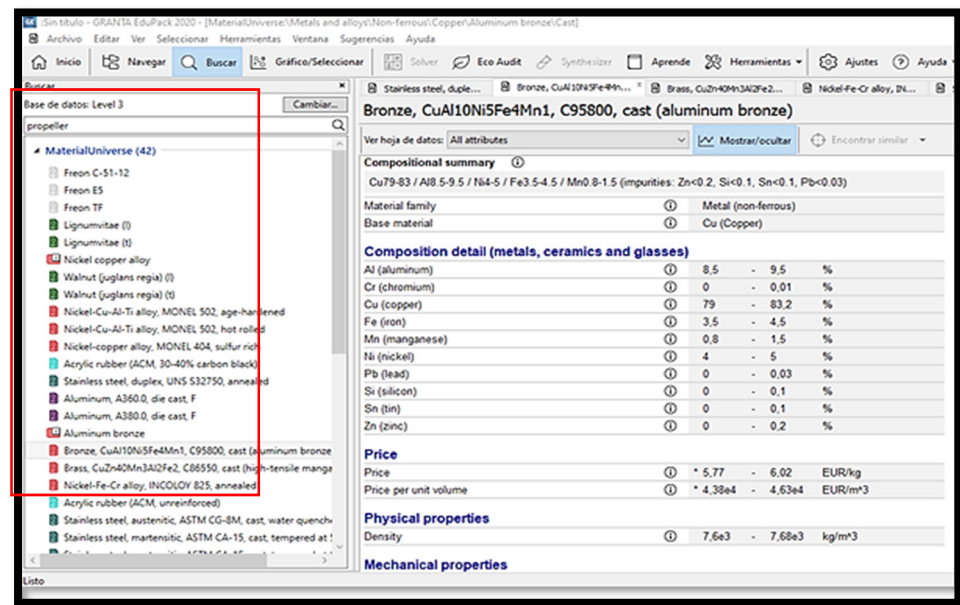


Figure 16. GRANTA EduPack 2020® database—Level 3 [21].

6.2. Elaboration of the Model with Machine Learning

The material selection process is divided into two distinct stages, each serving a unique purpose.

In the first stage, machine learning algorithms come into play. This begins with clustering, which sorts the database data. Subsequently, these clustered data points are utilized as inputs for various supervised algorithms, each designed to predict a categorical variable. Refer to Figure 8 for guidance on choosing the appropriate algorithms based on the output variable types. The categorical output variable in question corresponds to the cluster formed through the initial clustering process. Following the class prediction, the algorithms are aggregated, combining their predictions via voting, as elucidated later in Section 6.2. This initial stage aims to filter the materials that most closely align with desired specifications. Here, “assembly” refers to utilizing the output from one algorithm as input for another.

Moving to the second stage, an optimization process (see Figure 17) is undertaken to identify the material boasting the highest tensile strength from the subset of materials filtered during the first stage. This selection must adhere to a set of constraints, ultimately resulting in the identification of a single material.

The first stage commences with the application of an unsupervised algorithm to structure the data. Employing an agglomerative hierarchical clustering technique and examining the resultant dendrogram, one can observe the grouping that the algorithm determines most suitable. This takes into account the similarities and differences among data attributes, essentially analyzing material properties to facilitate data grouping and segmentation.

Figure 18 illustrates the clustering outcome, showcasing the grouping of materials based on their similarity levels as indicated by the Euclidean distances among data points along the ordinate axis.

The algorithm’s classification proves coherent, effectively grouping materials within the same family to which they belong. In total, five clusters were identified.

Having grouped the data, we proceed to apply the supervised algorithms which serve to predict.

For the supervised algorithms, only six input variables (X) will be used (of which three are numerical variables and the other three are categorical variables). The output is a categorical variable Y .

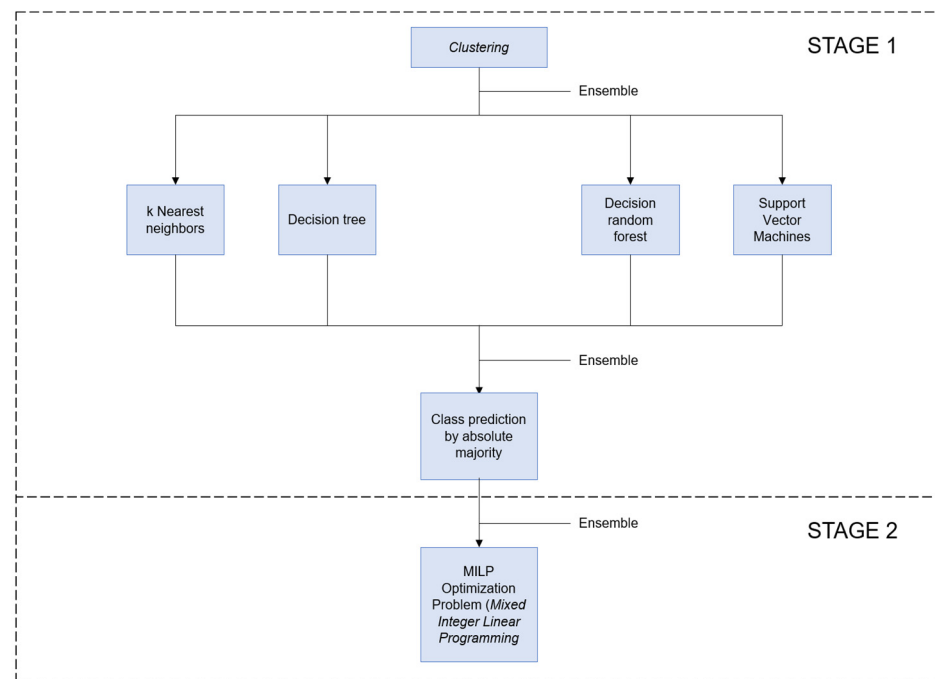


Figure 17. Conceptual map of the material selection program with the implementation of machine learning.

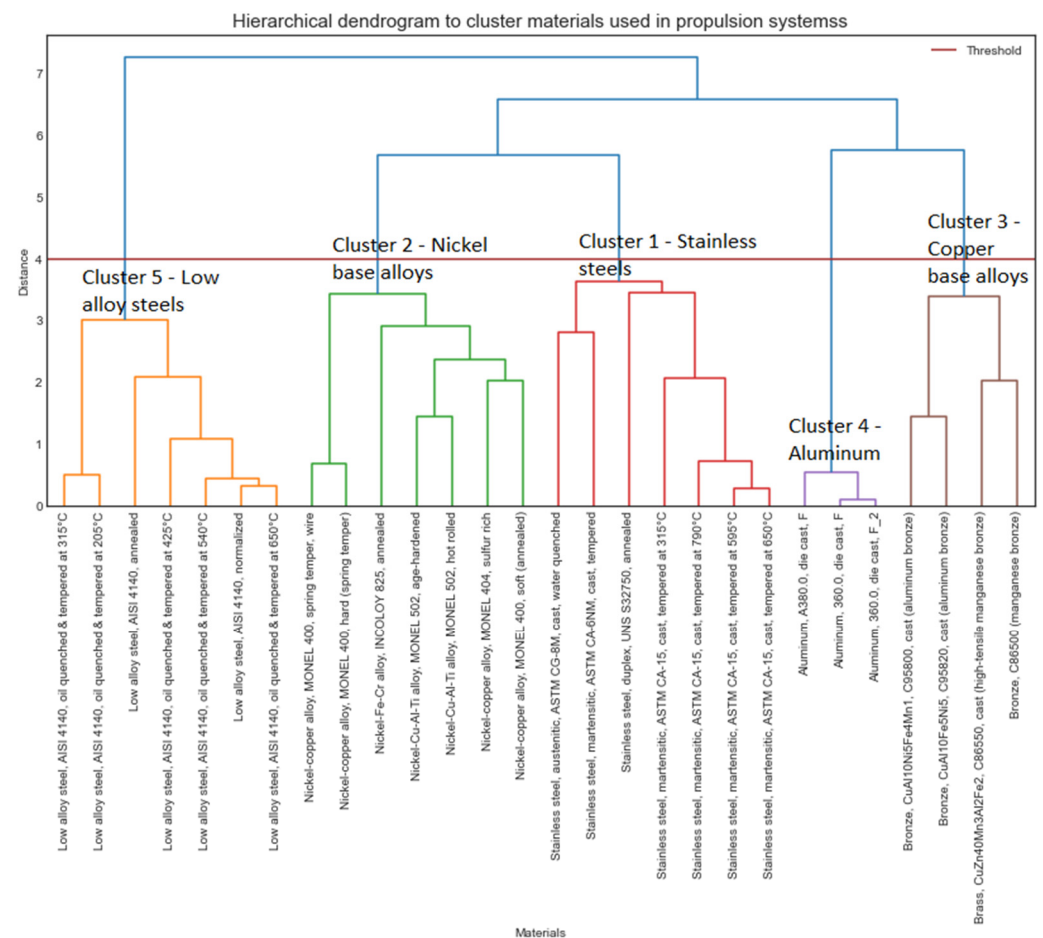


Figure 18. Agglomerative hierarchical dendrogram to group the materials used in propellers and/or shaft lines.

- The input variables (X) for the material are the price (EUR/kg), density (kg/m^3), tensile strength (MPa), properties for casting, properties for welding, and resistance properties against stress corrosion cracking.
- The output variable (Y) is the cluster to which the material would belong based on the desired specifications.

To confirm the models, a cross-validation is performed (with k equal to ten iterations). The procedure is the same as that calculated in previous sections, it is randomly split into a training set and a test set for each iteration. With the training set, the model is elaborated, and with the test set, the efficiency of the model is measured; the difference is that now what is going to be measured is the accuracy in each iteration for the test set.

$$Accuracy_j = \frac{N \text{ of correct predictions in } j - \text{th test set}}{N \text{ of total predictions in } j - \text{th test set}} \quad (17)$$

For $j = 1, \dots, k$

$$Accuracy = \frac{\sum_{j=1}^k Accuracy_j}{k} \quad (18)$$

where

- j : The position of the iteration.
- k : The number of iterations.

This technique will be used with every one of the independent models obtained from the supervised algorithms and the accuracy of assembling the models will also be quantified, which basically consists of an absolute majority vote.

This vote is a soft vote. This means that in the hypothetical case of a tie vote, the probability of occurrence of the class predicted by each model will be considered. The one with the highest probability will be the class selected in the final prediction of *STAGE I* of the program.

Figure 19 shows a slight improvement when assembling the models since a higher accuracy value has been obtained.

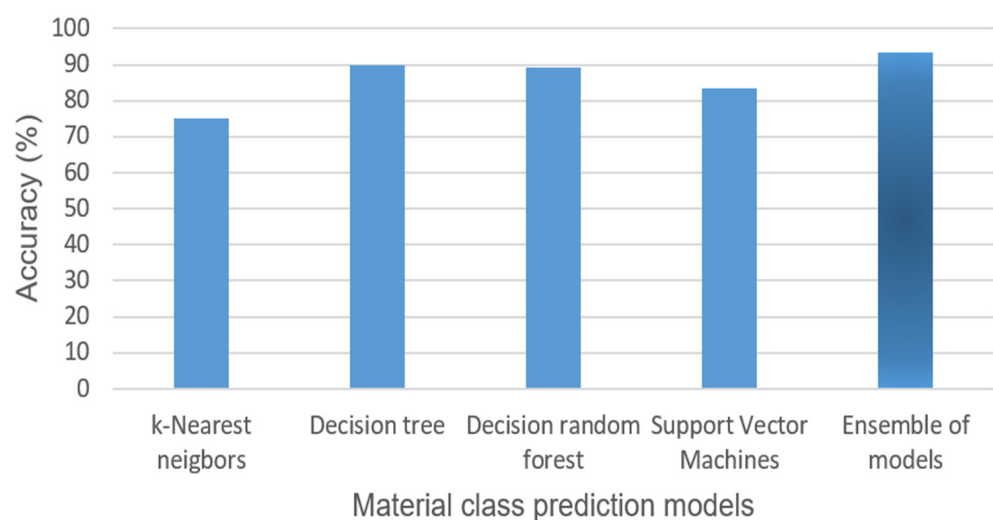


Figure 19. Accuracies obtained with the independent models and with the ensemble of models.

6.3. Assembly with MILP Optimization Problem

After completing the first stage, which aims to filter the family of materials that most closely resembles the desired specifications, the second stage in the selection of the material begins, which consists of applying a linear optimization problem.

Finally, to obtain a material that provides the best performance within the family of materials, the results obtained are assembled with an optimization problem.

The goal of the optimization problem is to find the highest tensile strength.

Based on the results obtained, only the materials that belong to the family of copper-based alloys have been filtered.

The *Mixed Integer Linear Programming* (MILP) optimization problem is defined.

- Decision variables (unknown to the problem): $Material_0, Material_1, \dots, Material_i, \dots, Material_{n-1}$

For $i = 0, 1, \dots, n - 1$

$Material_i$ is a binary variable (it is 0 if the material is not chosen and 1 if it is the chosen material).

Where

- i : The subscript that identifies the material.
- n : The total number of materials inside the cluster obtained from machine learning.
- Parameters (known to the problem):

Price _{i} , Density _{i} , Tensile strength _{i} ,

CO₂ footprint in the production _{i} ,

CO₂ footprint in the recycling _{i}

Where

- i : The subscript that identifies the material. Each parameter is defined by the subscript i since the parameters are known within the dataset.
- Objective function: $\max \sum_{i=0}^{n-1} Material_i \cdot \text{Tensile strength}_i$.
- Restrictions: Subject to

$$\left\{ \begin{array}{l} C_1 \leq \sum_{i=0}^{n-1} Material_i \cdot \text{Price}_i \leq C_2 \\ \sum_{i=0}^{n-1} Material_i \cdot \text{Density}_i \leq C_3 \\ C_4 \leq \sum_{i=0}^{n-1} Material_i \cdot \text{Tensile strength}_i \leq C_5 \\ \sum_{i=0}^{n-1} Material_i \cdot \text{CO}_2 \text{ footprint in the production}_i \leq C_6 \\ \sum_{i=0}^{n-1} Material_i \cdot \text{CO}_2 \text{ footprint in the recycling}_i \leq C_7 \end{array} \right.$$

where

- C_1 : Minimum price allowed (EUR/kg);
- C_2 : Maximum price allowed (EUR /kg);
- C_3 : Maximum density allowed (EUR /kg);
- C_4 : Minimum tensile strength allowed (MPa);
- C_5 : Maximum tensile strength allowed (MPa);
- C_6 : Maximum CO₂ footprint in the production allowed (kg/kg);
- C_7 : Maximum CO₂ footprint in the recycling allowed (kg/kg).

The problem may or may not have a solution. The more restrictive the problem, the less chance there is of finding a solution.

The problem has been solved with the *Optlang* library [22]. In ANNEX VI, it has been further solved with the *PuLP* library [23]. The same results were obtained for both libraries.

6.4. Interactive Program

Obviously, implementing all these algorithms requires an investment in time when extracting the data, cleaning it, and applying the models. However, it is always sought to automate the code so that it is as comfortable as possible when it comes back to future projects.

In the Jupyter® V7.0.3 notebook that is attached to this work, all the code has been grouped in a single cell and an interactive program has been developed. To use the program, it is only necessary to run the three code cells, and the screen shown in Figures 20 and 21 will appear.

STAGE 1 - Prediction of the family of materials that is sought (Solved using Machine Learning algorithms)

Select the specifications that you are looking for the material which you want to use in the propeller or in the shaft of the ship.

The program will search for the group of materials that most closely matches the desired specifications.

Price (€/kg)

Density (kg/m³)

Tensile strength (MPa)

Metal casting

Weldability

Stress corrosion cracking

Prediction with Clustering + k-Nearest Neighbors: ['Cluster 1 - Stainless steels']

Prediction with Clustering + decision tree: ['Cluster 3 - Copper base alloys']

Prediction with Clustering + decision random forest: ['Cluster 3 - Copper base alloys']

Prediction with Clustering + support vector machines: ['Cluster 3 - Copper base alloys']

Final prediction, ensemble model: Cluster 3 - Copper base alloys

	Material	Average price (€/kg)	Average density (kg/m ³)	Average tensile strength (MPa)	Metal casting	Weldability	Stress corrosion cracking
3	Bronze, CuAl10Ni5Fe4Mn1, C95800, cast (aluminum bronze)	5.895	7640	655.0	Acceptable	Good	Not susceptible
4	Brass, CuZn40Mn3Al2Fe2, C86550, cast (high-tensile manganese bronze)	5.005	7900	535.0	Excellent	Poor	Highly susceptible
6	Bronze, C86500 (manganese bronze)	4.965	8330	471.5	Excellent	Poor	Highly susceptible
9	Bronze, CuAl10Fe5Ni5, C95820, cast (aluminum bronze)	5.955	7600	670.0	Acceptable	Good	Not susceptible

Figure 20. The first stage of the interactive program to predict the cluster of the material.

It is easy to use. Only input variables and the restrictions to be imposed are introduced.

During the initial stage, the desired specifications are designated by manipulating the control bars and selecting options from choice boxes. Following this, the first stage yields a cluster prediction along with a corresponding table listing the materials affiliated with that cluster. Once the initial stage concludes, the program advances to the second stage.

In the second stage, the cluster derived from the ultimate prediction of the first stage is chosen, alongside the specifications for the MILP optimization problem.

STAGE 2 - Optimal material (Solved by MILP problem)

Select the cluster obtained in STAGE 1.

Next, select the constraints, if any, for the material you are looking for.

The program will return the material with the highest tensile strength within the imposed restrictions.

To see the material chosen, see the column of the DataFrame named <<Decision>>. It is observed that the row that takes value 1 will be the chosen material. If all materials take value 0 then none of them solves the problem for the restrictions imposed

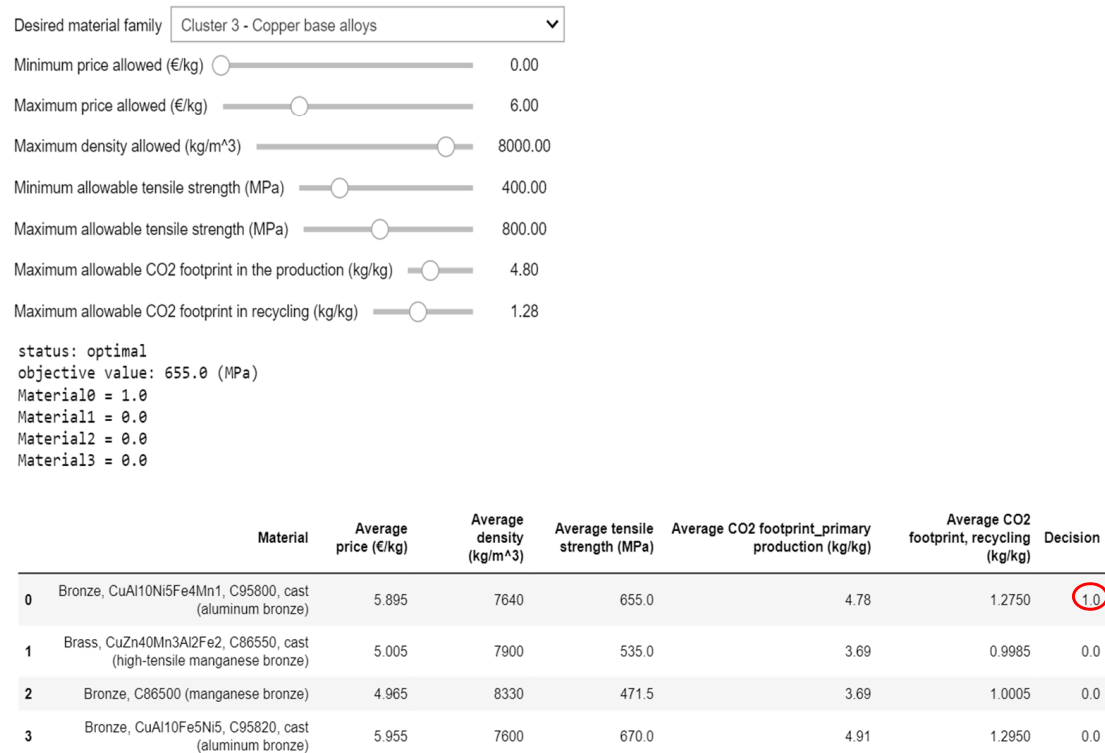


Figure 21. The second stage of the interactive program to predict the material with the highest tensile strength.

7. Discussion: Deep Learning and Future Lines of Study

The concept of deep learning is often regarded as an advancement over machine learning. In this scenario, the level of interaction between users and machines is minimized. In the realm of deep learning, the focus revolves around crafting the architecture of a neural network that meticulously processes the data.

Within the domain of deep learning, users play a pivotal role in defining hyperparameters. These parameters encompass crucial aspects such as the count of neuron layers, where a minimum of two layers—a designated input layer and a corresponding output layer—is mandatory.

Notably, intermediate hidden layers frequently intercede between the input and output layers, significantly enhancing the precision of prediction and classification outcomes.

Moreover, the hyperparameters encompass determining the number of neurons per layer.

This entails having a corresponding number of neurons in the input layer as there are input variables. Similarly, the number of neurons in the output layer aligns with the target variable's count, underlining the comprehensive nature of the parameter configuration in deep learning.

Neurons within a neural network are interconnected, with each connection being assigned a specific weight.

As illustrated in Figure 22, a neuron fundamentally operates by aggregating the weighted summation of its inputs and associated weights. Subsequently, an activation function, functioning as a filter, is applied. This function serves to activate neurons that

demonstrate favorable outcomes. Figure 23 showcases some commonly employed activation functions.

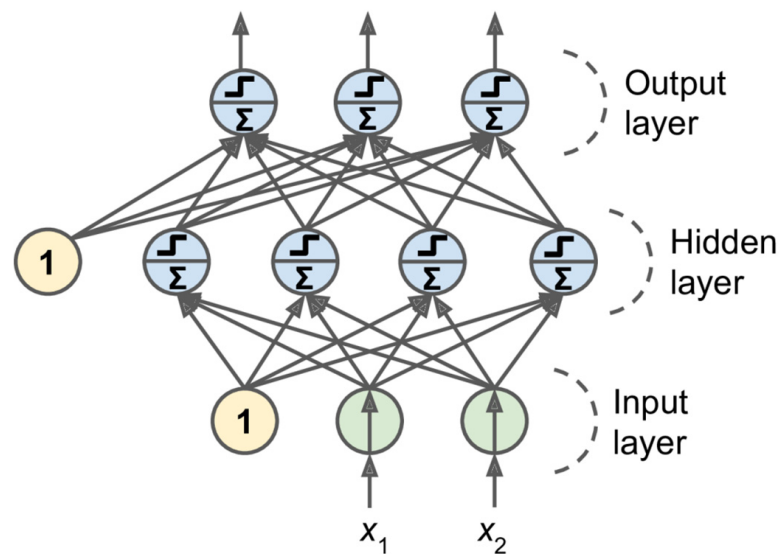


Figure 22. Multilayer perceptron with two input neurons, two bias neurons, four neurons in the hidden layer, and three output neurons [24], p. 299.

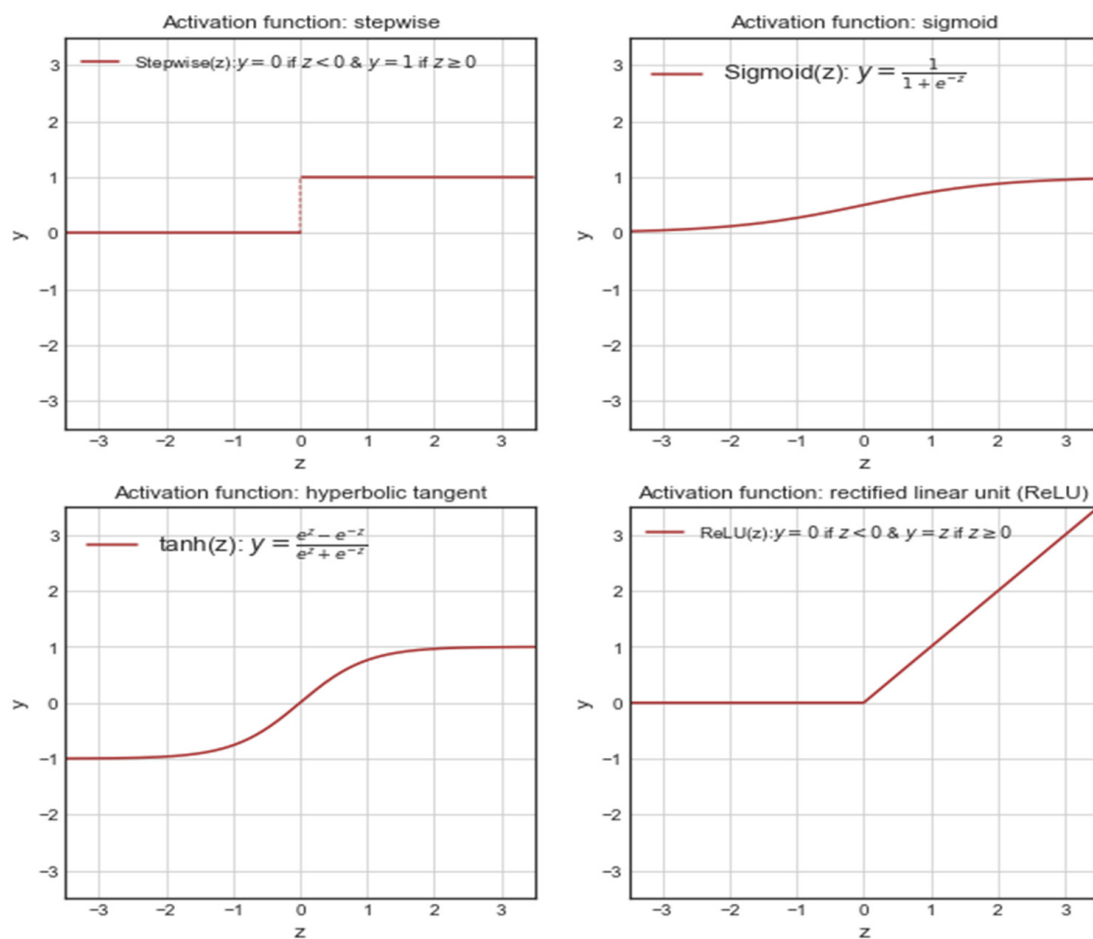


Figure 23. Activation functions of artificial neurons.

Activated neurons generate outputs, contributing to the iterative training of the neural network. During the initial iteration, the network's predictions may be inaccurate, as learning has not yet transpired. Here, the pivotal role of backpropagation emerges. Neurons predict layer by layer, culminating in the output layer. After prediction, the algorithm assesses errors by traversing the network backward. Correct predictions reinforce the weights linked to neuronal connections, intensifying the network's intelligence. Consequently, an increased number of iterations enhances weight reinforcement, resulting in heightened neural network competence.

For neural network construction in Python3®, TensorFlow and Keras libraries are commonly employed. These libraries, the most widely used, facilitate the development of neural networks. A distinctive advantage of deep learning over machine learning lies in its adeptness at handling unstructured data more efficiently. While this study utilizes structured, tabulated data, deep learning is particularly adept at handling unstructured data, such as images, videos, audio, and text.

The forthcoming research trajectory envisioned for this study involves predicting ship propeller cavitation wear rates using images obtained during periodic inspections. This entails employing intricate neural network architectures and gathering a database of propeller images accompanied by wear rate study outcomes. If the images are in color, a conversion to black and white will be undertaken, as color intensity values per pixel are more interpretable with a neural network. Refer to Figure 24 for an example of the images constituting the database.

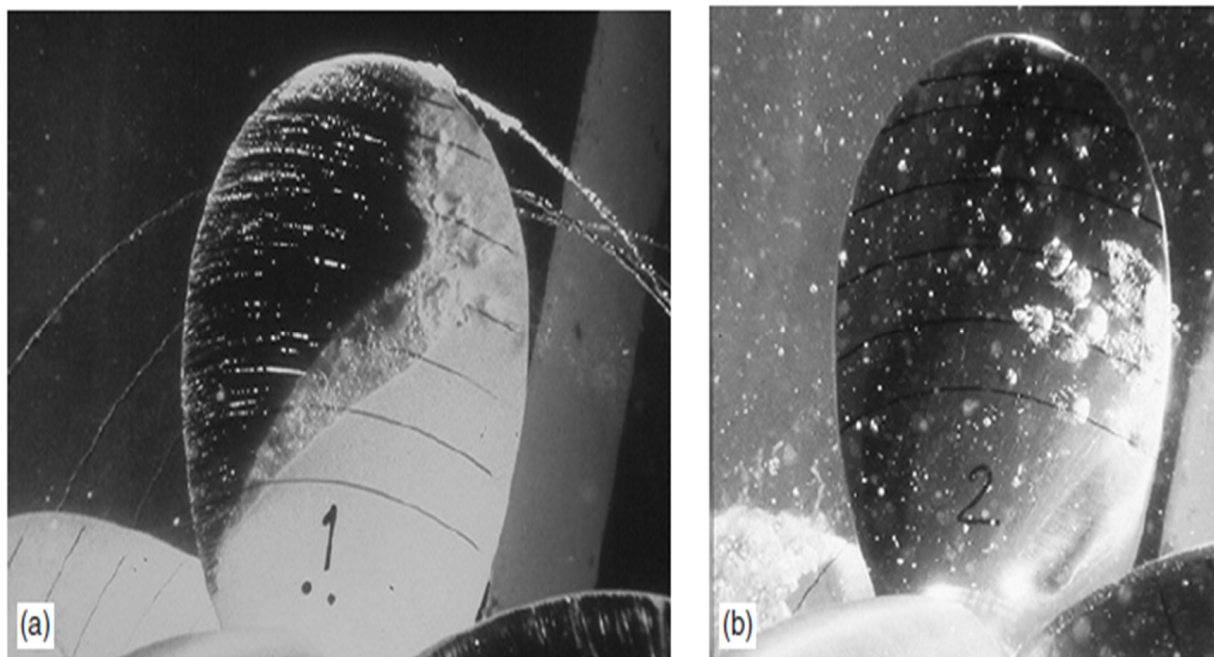


Figure 24. (a,b) Photos taken of propellers subjected to cavitation studies [25].

Subsequent phases involve formulating neural network architectures based on the acquired knowledge and training them using the image database as input variables, with the cavitation wear rate as the output variable. The neural network's potential lies in utilizing propeller images to detect wear due to air bubble implosion from cavitation and assessing the resultant cavity shape. Linking each image to a parameter quantifying corrosion-induced wear—such as material loss—enables the neural network to predict based on insights gleaned from the image database. Following training, the neural network can swiftly assess the ship propeller's condition during routine inspections by predicting future images captured during such evaluations.

8. Findings and Conclusions

The investigation yielded noteworthy findings regarding the performance of each machine-learning algorithm in predicting braking power and propeller diameter. The root mean square error (RMSE) values for these predictions were consistently lower than those derived from J. Mau's braking power calculation formula and the propeller diameter formula [19], p. 602. These improved predictions effectively address the initial query about the quality of the machine learning predictions posed at the study's outset.

Furthermore, it is worth noting that there exists a significant correlation among predictor variables. This observation is not surprising given the interconnectedness of variables related to cargo capacity and ship size. This high correlation substantiates the validity of the predictive model.

The efficacy of clustering was demonstrated and particularly evident in its application to material classification. The algorithm was adept at structuring data and grouping elements solely based on the information they offered. This significance amplifies when dealing with data whose underlying behavior is less apparent.

Ensemble models, constructed through the amalgamation of algorithms, displayed an accuracy of 93.33%. This accuracy surpasses that of individual algorithms, highlighting the enhanced robustness attained through algorithm assembly.

The obtained results are gratifying and signify a marked improvement in prediction accuracy. The maritime industry stands to benefit from the diverse applications of these algorithms. While this study concentrated on specific machine learning algorithms, there remain unexplored avenues, and the realm of deep learning warrants further exploration, especially concerning neural networks' capacity to predict propeller cavitation wear rates using images.

In line with this investigation's outcomes, it is evident that the developed models possess the ability to scrutinize vast volumes of intricate data, thereby producing more precise outcomes.

Author Contributions: Conceptualization, M.Á.G.R. and I.M.d.A.; methodology, M.Á.G.R. and R.P.F.; software, M.Á.G.R. and I.M.d.A.; validation, M.Á.G.R. and R.P.F.; investigación, M.Á.G.R. and I.M.d.A.; resources, M.Á.G.R. and R.P.F.; data curation, M.Á.G.R. and I.M.d.A.; writing, review and editing, M.Á.G.R., I.M.d.A. and R.P.F.; supervisión, M.Á.G.R. and R.P.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting results are available in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Venesa, S.; Hadjina, M.; Fafandjel, N. Toward Shipbuilding 4.0-an Industry 4.0 Changing the Face of the Shipbuilding Industry. 2018. Available online: https://www.researchgate.net/publication/325715397_Toward_shipbuilding_40-an_industry_40_changing_the_face_of_the_shipbuilding_industry (accessed on 18 February 2023).
2. Puertos del Estado Gobierno de España. Available online: <https://www.puertos.es/es-es/Paginas/ports40.aspx> (accessed on 18 February 2023).
3. Gartner. Gartner. 2018. Available online: <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019/> (accessed on 16 February 2021).
4. Rodal Montero, E. *Industria 4.0 Conceptos, Tecnologías Habilitadoras y Reto*; Pirámide: Madrid, Spain, 2020.
5. Zhang, M.; Kujala, P.; Musharraf, M.; Zhang, J.; Hirdaris, S. Machine Learning Method for the Prediction of Ship Motion Trajectories in Real Operational Conditions. *Ocean. Eng.* **2023**, *283*, 114905. [CrossRef]
6. Chou, C.C.; Wang, C.N.; Hsu, H.P. A novel quantitative and qualitative model for forecasting the navigational risks of Maritime Autonomous Surface Ships. *Ocean Eng.* **2022**, *15*, 248. [CrossRef]
7. Khan, S.; Goucher-Lambert, K.; Kostas, K.; Kaklis, P. ShipHullGAN: A Generic Parametric Modeller for Ship Hull Design Using Deep Convolutional Generative Model. *Comput. Methods Appl. Mech. Eng.* **2023**, *411*, 116051. [CrossRef]

8. Liu, J.; Yan, X.; Liu, C.; Fan, A.; Ma, F. Developments and Applications of Green and Intelligent Inland Vessels in China. *J. Mar. Sci. Eng.* **2023**, *11*, 318. [CrossRef]
9. Hu, B.; Liu, X.; Jing, Q.; Lyu, H.; Yin, Y. Estimation of berthing state of maritime autonomous surface ships based on 3D LiDAR. *Ocean Eng.* **2022**, *251*, 111131. [CrossRef]
10. Chen, X.; Liu, S.; Liu, R.W.; Wu, H.; Han, B.; Zhao, J. Quantifying Arctic oil spilling event risk by integrating an analytic network process and a fuzzy comprehensive evaluation model. *Ocean Coast. Manag.* **2022**, *228*, 106326. [CrossRef]
11. Colaboradores de Wikipedia. Wikipedia, La Enciclopedia Libre. Support-Vector Machine. 2020. Available online: https://en.wikipedia.org/wiki/Support-vector_machine (accessed on 28 November 2020).
12. The MathWorks, Inc. MathWorks. Máquinas Vectoriales de Soporte Para la Clasificación Binaria. Available online: <https://es.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html> (accessed on 28 November 2020).
13. Robinson, S. Stack Abuse. K-Nearest Neighbors Algorithm in Python and Scikit-Learn. Available online: <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/> (accessed on 28 November 2020).
14. The Royal Institution of Naval Architects. *Significant Ships of 1998*; The Royal Institution of Naval Architects: London, UK, 1998.
15. The Royal Institution of Naval Architects. *Significant Ships of 2007*; The Royal Institution of Naval Architects: London, UK, 2007.
16. The Royal Institution of Naval Architects. *Significant Ships of 2010*; The Royal Institution of Naval Architects: London, UK, 2010.
17. The Royal Institution of Naval Architects. *Significant Ships of 2012*; The Royal Institution of Naval Architects: London, UK, 2012.
18. Mau, J. La Rapport de Deplacement au Port en Lourd des Cargos Pour Merchandises Générales. Bulletin Technique du Bureau Veritas. April 1969.
19. Alvariano, R.; Azpíroz, J.J.; Meizoso, M. *El Proyecto Básico del Buque Mercante*; Colegio Naval de Ingenieros Navales y Oceánicos de España: Madrid, Spain, 2007.
20. The Society of Naval Architects and Marine Engineers. *Marine Engineering*; Sname: New York, NY, USA, 1992.
21. Granta Design. CES EduPack Software. Material Level 3. 2020. Available online: <https://www.ansys.com/products/materials/granta-edupack> (accessed on 14 February 2021).
22. Kristian, J.; Cardoso, J.G.; Sonnenschein, N. Optlang: An algebraic modeling language for mathematical optimization. *J. Open Source Softw.* **2017**, *2*, 139.
23. Mitchell, S.; O'Sullivan, M.; Dunning, I. PuLP 2.4. 2011. Available online: http://www.optimization-online.org/DB_FILE/2011/09/3178.pdf (accessed on 12 April 2021).
24. Géron, A. *Aprende Machine Learning Con Scikit-Learn, Keras Y Tensorflow. Conceptos, Herramientas y Técnicas Para Conseguir Sistemas Inteligentes*; Anaya Multimedia: Madrid, Spain, 2020.
25. Molland, M. The Maritime Engineering Reference Book. In *A Guide to Ship Design, Construction and Operation*; Butterworth-Heinemann: Oxford, UK, 2008.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.