Universidade Federal de Juiz de Fora
Curso de Graduação em Engenharia de Produção

Victor Hugo Soares Pereira

A Machine Learning approach to predict Length of Stay of vehicles in an inbound logistics operation

Juiz de Fora
2022

Victor Hugo Soares Pereira

A Machine Learning approach to predict Length of Stay of vehicles in an inbound logistics operation

Trabalho de Conclusão de Curso apresentado a Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito parcial para a obtenção do título de Engenheiro de Produção.

Orientador: Eduardo Pestana de Aguiar

Coorientador: Kaike Sa Teles Rocha Alves

Juiz de Fora

2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Soares Pereira, Victor Hugo.
A Machine Learning approach to predict Length of Stay of vehicles in an inbound logistics operation / Victor Hugo Soares Pereira. – 2022.
31 f. : il.

Orientador: Eduardo Pestana de Aguiar
Coorientador: Kaike Sa Teles Rocha Alves
Trabalho de Conclusão de Curso (Graduação) – Universidade Federal de Juiz de Fora, . Curso de Graduação em Engenharia de Produção, 2022.

1. Machine Learning. 2. Steel Industry. 3. Logistics. 4. Regression models I. Pestana de Aguiar, Eduardo, orient. II. Sa Teles Rocha Alves, Kaike, coorient.) III. Título.

Victor Hugo Soares Pereira

A Machine Learning approach to predict Length of Stay of vehicles in an inbound logistics operation

Trabalho de Conclusão de Curso apresentado a Faculdade de Engenharia da Universidade Federal de Juiz de Fora, como requisito parcial para a obtenção do título de Engenheiro de Produção.

Aprovado em 22 de Fevereiro de 2022

BANCA EXAMINADORA

D. Sc, Eduardo Pestana de Aguiar - Orientador
Universidade Federal de Juiz de Fora

M. Sc, Kaike Sa Teles Rocha Alves - Coorientador
Universidade Federal de Juiz de Fora

D. Sc, Antonio Angelo Missiaggia Picorone
Universidade Federal de Juiz de Fora

D. Sc, Cristina Márcia Barros de Castro
Universidade Federal de Juiz de Fora

RESUMO

Uma usina siderúrgica de grande porte recebe cerca de milhões de toneladas de sucata por ano a partir do transporte rodoviário. O Tempo de Permanência de Veículos é uma das métricas mais importantes que representa o desempenho do processo de recebimento de matéria-prima. Previsões precisas desse indicador possibilitam que gerentes façam decisões baseadas em dados sobre aspectos operacionais, táticos e estratégicos da operação. Esse trabalho propõe uma abordagem de Aprendizado de Máquina para a previsão do Tempo de Permanência de Veículos da operação de recebimento de sucata de uma grande usina siderúgica brasileira. Cinco modelos de Aprendizado de Máquina foram treinados e testados com nove meses de dados do processo. Os resultados foram comparados com o método de previsão atual e estatisticamente validados com um teste ANOVA. A abordagem de Aprendizado de Máquina proposta nesse trabalho alcançou uma melhor precisão, reduzindo em 64% o RMSE, e tem o potencial de permitir decisões mais eficazes baseadas em dados para a empresa.

Palavras-chave: Aprendizado de Máquina. Indústria Siderúrgica. Logística. Modelos de regressão.

ABSTRACT

A large steel plant receives up to millions of tons of scrap metal through road transportation each year in its inbound logistics process. The Length of Stay of vehicles is one of the most important metric that represent the performance of the unloading operation of raw materials. The provision of accurate prediction for this metric enables managers to make data-driven decisions in operational, tactical and strategic level. This study proposes a Machine Learning approach for the prediction of Length of Stay of vehicles loaded with scrap metal in the inbound operation of a large Brazilian steel plant. Five Machine Learning models were trained and tested with nine months of data from the process. The results are compared with the current method of prediction and statically validated through ANOVA test. The Machine Learning approach proposed in this study achieved better accuracy, reducing in 64% the RMSE, and has the potential to enable more reliable data-driven decisions for the company.

Key-words: Machine Learning. Steel Industry. Logistics. Regression models.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANOVA	Analysis of Variance
KNN	k-Nearest Neighbors
KPI	Key Process Indicator
LOS	Length of Stay
MAE	Mean Absolute Error
ML	Machine Learning
NDEI	Non-Dimensional Index Error
RMSE	Root Mean Squared Error

CONTENTS

1 INTRODUCTION

1.1 Background of the study

Steel production is an operation that consumes a large number of raw materials, in particular, scrap metal (YUZOV; SEDYKH, 2003). The operation to transfer this material from scrapyards to a steel plant is a complex system that moves up to millions of tons of scrap metal each year by road transportation.

Length of stay (LOS) of vehicles is one of the most critical performance indicators in inbound logistic operations. The provision of accurate prediction of the unloading time of cargo, or the average unloading time of the day, enables managers to make better-informed decisions about planning at the operational level (e.g., manpower, machinery allocation, extra hours, and process capacity), or tactical and strategic level. (HYNDMAN; ATHANASOPOULOS, 2018). According to (ZHAO; XIE, 2002), the accuracy of the predictions can also benefit the performance of the whole operation by minimizing the negative impact of uncertainty in a context where information is carried throughout the entire supply chain.

1.2 Justification

The company in which this study was conducted, a large steel plant in Brazil, makes a rough prediction of the LOS metric. The method that is currently used is the division of the net weight (ton) of each cargo by a flow rate factor (ton/minute) - which is set by the managers of the steel plant and is not reviewed frequently - plus a fixed amount of time to contemplate other activities of the inbound process. The currently used method presents high values and high variability of the error metrics.

The prediction of the LOS metric for the inbound operation is a challenging issue since there is a great variety of vehicles and materials combinations to be considered. Also, external factors such as machinery breakdown, process interruptions, shift change and first-time inside the plant vehicle drivers often create fluctuations that are very hard to predict.

According to (KELLEHER; NAMEE; D'ARCY, 2020), Machine Learning is a field that is used to make predictive models by extracting patterns from datasets. In order to predict a continuous target, regression algorithms model the relationship between the target and one or more predictors using supervised learning.

Further justification is stated as follows:

- A accurate prediction of the LOS metric enables more reliable data-driven decisions about the inbound process of scrap metal;

- The company has a large database with enough information to characterize the LOS values and enable a machine learning approach to the problem.
- The proposed Machine Learning approach achieved better accuracy than the method that it's currently used at the steel plant;

1.3 Scope

- An exploratory analysis of the LOS metric with the company's data;
- An analysis of the current method of prediction;
- A proposal of a Machine Learning approach for improving the prediction of the steel plant's logistic KPI;
- A comparison of the performance of different regression machine learning models to predict a continuous variable;
- The validation of a framework of training, testing, and statistical validation of the results of machine learning models.

1.4 Statement of the objectives

This work aims to propose Machine Learning models as an efficient approach for the prediction of the LOS metric. The key factor is to consider not only the net weight of the cargo, but also other characteristics of the material, the vehicle, and Key Process Indicators (KPIs) of the process to predict more accurately the Length of Stay. In order to assemble all this information into a feasible solution, this study proposes the application of Machine Learning Regression Models, trained with the company's historical data from the process to predict the LOS metric of the inbound operation of scrap metal.

1.5 Methodology

The methodology that will be conducted throughout the study is presented below:

- Data collection from the inbound process of scrap metal.
- A descriptive analysis of the LOS metric and the current method of prediction;
- Selection of features, data cleaning and preparation of the datasets with random train/test split;
- Selection of the Machine Learning Models to be tested;

- Evaluation of the models based on error metrics;
- Statistical validation of the results with One-way ANOVA.

1.6 Work organization

This work is organized as follows:

- Section 2 the LOS metric and its importance to the company is presented. Then, the current method of prediction is discussed. And finally, the dataset is presented.
- Section 3 briefly recalls the Machine Learning models implemented to predict the LOS metric.
- Section 4 presents the tools used in this study and discusses the errors of the Machine Learning predictions, comparing the presented models. Additionally, a statistical test validates the performance of the proposal.
- Section 5 presents the main conclusions of this work.

2 PROBLEM FORMULATION

In this chapter, the LOS metric and its importance to the company is presented. Then, the current method of prediction is discussed. And finally, the dataset is presented.

2.1 The LOS metric

The Length of Stay metric is an industrial KPI that is constantly monitored by the steel plant and by higher administration. It represents the total time spent by a vehicle, loaded with raw material, to unload its cargo at the steel plant. Managers of the company keep a close eye on this KPI - hourly, daily e monthly mean values - and whenever it presents an abnormal behavior, a committee takes place to investigate possible causes and address solutions. The importance of this indicator relies on fact that it directly impacts productivity, safety, and also laws that protect truck drivers against long hours of unloading cargo.

The data of the inbound process of scrap metal was obtained from the company's database and the histogram of the LOS metric is presented in Figure 1. Based on the histogram, it's possible to observe a wide range of values for LOS, from less than an hour up to 19 hours.

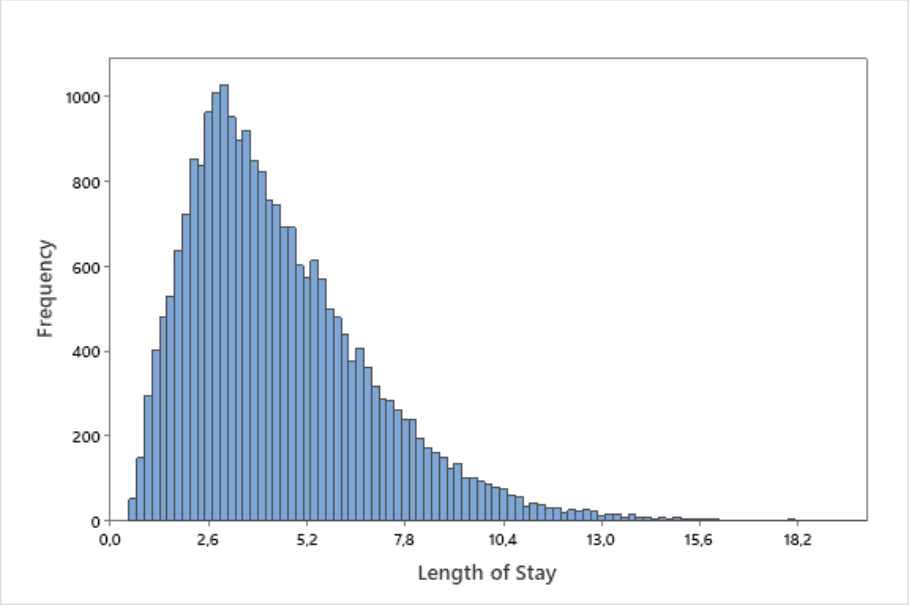

Figure 1 – Histogram of LOS values

Source: The author

In total, 23,974 observations were recorded and divided into 9 datasets, one for each month. Each observation represents a truck whose cargo was unloaded at the steel plant. Table 1 presents mean values, standard deviation (SD), the 25th, 50th, and 75th percentiles of the LOS metric. LOS is recorded in hours.

Table 1 – Descriptive statistics of the LOS metric

Dataset	No. of observations	Mean	SD	25th	50th	75th
DS_1	3,009	4.93	6.32	2.92	4.30	6.05
DS_2	3,026	4.74	3.04	2.86	4.18	5.94
DS_3	3,577	4.91	2.51	3.09	4.45	6.23
DS_4	2,889	4.94	3.68	3.01	4.32	6.36
DS_5	2,492	4.56	5.49	2.75	3.98	5.72
DS_6	2,884	4.48	6.59	2.40	3.69	5.76
DS_7	1,814	4.67	2.91	2.54	3.89	6.24
DS_8	2,196	3.85	2.52	2.15	3.27	5.03
DS_9	2,087	3.81	2.64	2.45	3.36	4.74
Entire Dataset	23,974	4.60	4.37	2.71	3.99	5.84

Source: The author

The descriptive statistics of the LOS metric in Table 1 show high values of standard deviation when compared to the mean values for each dataset. The main causes for variation in the total time a vehicle spends at the plant are the great variety of vehicles and materials combinations to be considered, machinery breakdown, process interruptions and overload, shift change, queues, and first-time inside the plant vehicle drivers.

2.2 Current method of prediction

Equation 2.1 represents the current method of prediction of the LOS metric. The method consists of the division of the net weight (ton) of each cargo by a flow rate factor (ton/minute) - which is set by the managers of the steel plant and is not reviewed frequently - plus a fixed amount of time to absorb other activities of the process, such as clearance, weighing and motion inside the plant.

$$LOS = \frac{1}{60}\left(\frac{\text{Net weight of the cargo [ton]}}{\text{Flow rate factor [ton/min]}} + \text{Fixed amount of time [min]}\right) \tag{2.1}$$

Table 2 summarizes three error measures of the current method of LOS prediction: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Non-Dimensional Index Error(NDEI). Equations 2.2, 2.4, 2.3 describe the error metrics . Also, the values of Table 2 are plotted in Figure 2. The RMSE of the method presents high variability among the datasets, with values ranging from 2.55 hours up to 6.63 hours. The MAE presents more stability than the RMSE, but it's notably smaller. This discrepancy in the magnitude of the error metrics shows that the current predictions have large residues, due to the quadratic nature of the RMSE Equation (2.2). (CHAI; DRAXLER, 2014).

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(y^k - \hat{y}^k)^2}, \tag{2.2}$$

$$NDEI = \frac{RMSE}{std([y^1, ..., y^T])}, \tag{2.3}$$

$$MAE = \frac{1}{T}\sum_{k=1}^{T}|y^k - \hat{y}^k|, \tag{2.4}$$

where $\hat{y}^k$ is the $k-$th forecasted value, y^k the $k-$th actual value and T is the sample size.

Table 2 – Error metrics for the current method of LOS prediction

Dataset	RMSE	NDEI	MAE
DS_1	6.37	1.01	2.05
DS_2	3.12	1.05	1.95
DS_3	2.55	1.03	1.88
DS_4	3.74	1.02	2.00
DS_5	5.53	1.00	1.94
DS_6	6.63	1.00	2.25
DS_7	2.95	1.02	2.24
DS_8	2.65	1.05	1.97
DS_9	2.70	1.02	1.65
Mean	4.03	1.02	1.99

Source: The author

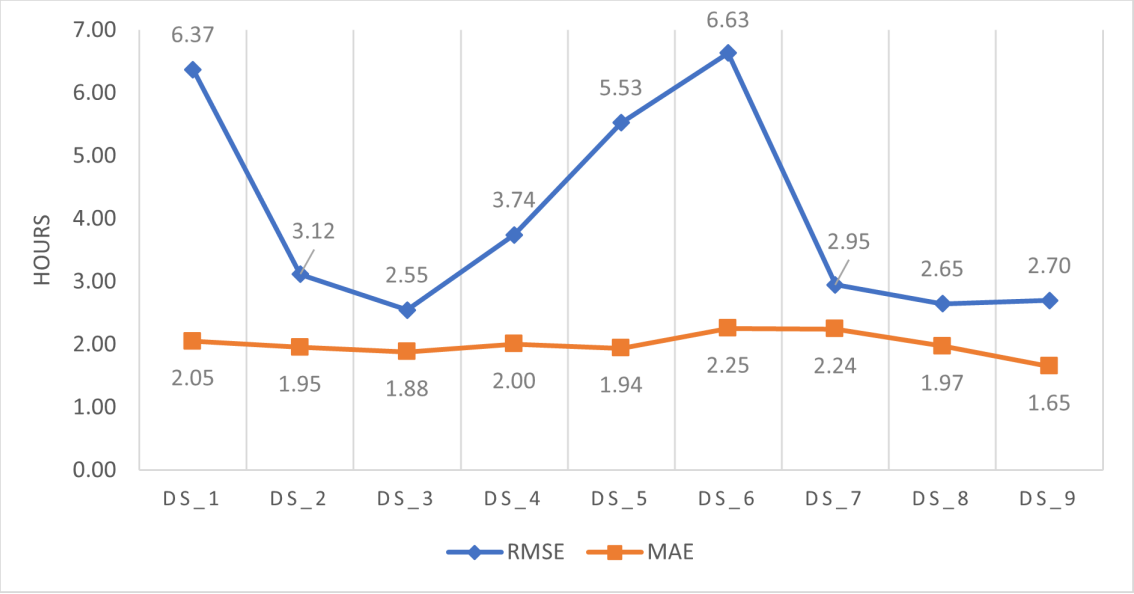

Figure 2 – Current method: RMSE and MAE for each dataset

Source: The author

2.3 The dataset

Table 3 presents a random sample of three vehicles in order to describe the dataset used in this study. The first column shows 12 attributes that characterize the cargo, the vehicle, and the current state of the inbound process. The following columns are values for each of these attributes of the vehicles. Each observation collected from the company's database contains the twelve attributes and the LOS time.

Table 3 – Example of the dataset

Attributes	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**
day_of_the_week	2	6	5
No._half_of_the_month	1	1	1
group_of_vehicle	1	0	0
type_of_vehicle	75	127	127
bin_sweeping	1	1	1
sweeping_time	2.58	0.88	0.76
unload_location	27	14	14
id_material	32	30	30
multi_material	1	1	1
net_weight	48,220	72,960	72,920
amount_of_vehicles_day	119	95	125
amount_of_vehicles_inside	40	40	40
LOS	6.06	5.68	5.37

Source: The author

The day_of_the_week and No._half_of_the_month are time-related variables that use integers from one to seven [1, 2, ..., 7] to represent the day of the week [Sunday, Monday, ..., Saturday] and integers one or two [1,2] for the first or second half of the month, respectively. The group_of_vehicle and the type_of_vehicle describe the physical characteristics of the vehicle. The first is related to whether the vehicle needs external help to unload the cargo, with an industrial claw for example, and then receives the value of 1, or it can unload itself by lifting the front and dumping the material, receiving the value of 0. The latter describes what kind of vehicle was carrying the cargo and what is the maximum gross weight it can carry. The values of this attribute are based on a standard list that the company uses for transportation modes. The binary variable bin_sweeping indicates if the vehicle needed sweeping after unloading the cargo. This information is useful since the sweeping_time (variable, in hours) is completely contained in the LOS metric and, due to queues in the process, it may take more than an hour to be completed. The unload_location is a variable that identifies the location where the cargo was unloaded in the scrapyard. The id_material variable is related to the type of material that was carried by the vehicle. The binary variable multi_material indicates

whether more than one type of material was unloaded by the same vehicle or not. This information is critical since vehicles with more than one type of material need to unload each material at a time and be weighted by the end of each unloading process. By doing this, the vehicle faces multiple queues and presents higher values of LOS. The net_weight is the weight of the material unloaded, in kilograms. The amount_of_vehicles_day and amount_of_vehicles_inside are directly related to the operation of the plant. The first indicates how many vehicles were received that day and the latter indicates how many vehicles were inside the plant at the moment that particular vehicle entered it. These two variables are also related to queues in the process.

A Machine Learning approach is justifiable by the comparison of the number of attributes that can be obtained from the company's database to characterize each LOS metric, presented in Table 3, to the current method of prediction (Equation 2.1), that only uses the net weight of the cargo.

3 PROPOSED MODELS

This section presents the proposed models for the prediction of the LOS metric using the data described in Table 3. In order to predict a continuous variable, the chosen models solve regression problems.

3.1 Linear Ridge Regression

Linear Regression models predict the targets using a linear combination of the features. Proposed by (HOERL; KENNARD, 1970), Linear Ridge Regression addresses the Ordinary Least Squares method with a penalty on the size of the coefficients as a possible solution to the imprecision of the method when the variables are highly correlated. This penalty is based on a α parameter to minimize a penalized residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Ridge regression does not perform feature selection. It shrinks coefficients towards zero, including all of the features in the final model.

3.2 k-Nearest Neighbors Regressor

The k-Nearest Neighbors Regressor (KNN) is an intuitive and efficient algorithm that has been used extensively for regression problems (HU et al., 2014), (KOHLI; GODWIN; UROLAGIN, 2021), (BAN et al., 2013). The application of KNN is based on the assumption that for data generated by a given process, there may be observations of repeated patterns of behavior (BAN et al., 2013). Proposed by (YAKOWITZ, 1987), KNN algorithm predicts a new value based on past feature similarity. For each prediction, the model identifies the k most similar past patterns and combines their values.

The KNN algorithm holds a collection of training instances. The i-th training instance is a vector of n-features as in $\{f_1^i, f_2^i, ..., f_n^i\}$ and a associated target value $\{t_1^i, t_2^i, ..., t_m^i\}$ with size m. For a new prediction whose features are known $\{q_1^i, q_2^i, ..., q_n^i\}$, the k most similar training instances are combined to predict the target value, based on a similarity or distance metric.

3.3 Gradient Boosting Regressor

A boosting process is a method for improving the accuracy of learning algorithms by fitting an initial model to the data and then building a second model focused on accurately predicting the cases in which the first had a bad performance (SCHAPIRE, 1999). Proposed by (FRIEDMAN, 2001), the Gradient Boosting Regressor uses a differentiable loss function (e.g. squared error) to guide an additive method of creating weak learners in a greedy way, following a gradient descent procedure and, thus, minimizing loss.

In the regression problem, the algorithm's objective is to find a function $F(x)$ that minimizes the loss function $L(y, F(x))$, given a training set $\{(x_1, y_1), ..., (x_T, y_T)\}$ with size T, input variables (i.e. predictors) x_t and the corresponding output value y_t. The additive method to find the optimal solution $\hat{F}(x)$ weights the week learners $h(x_t)$ gradually throughout the descent procedure. The initialization of the algorithm is made with a constant function $F_0(x)$ as follows:

$$F_0(x) = \arg\min_{\gamma} \sum_{t=1}^{T} L(y_t, \gamma) \tag{3.1}$$

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{3.2}$$

where $F_m(x)$ is the integration of the values of the basic regression trees, $h_m(x)$ is m-th regression tree and γ_m is the weighting coefficient of the m-th tree.

The algorithm is optimized by the negative gradient, as follows:

$$z_m(x_t) = -\frac{\partial L(y_t, F_{m-1}(x_t))}{\partial F_{m-1}(x_t)} \tag{3.3}$$

The next regression tree $h_m(x)$ is build based on the values of $z_m(x_t)$ and x. The γ_m coefficients are determined as follows:

$$\gamma_m = \arg\min_{\gamma} \sum_{t=1}^{T} L(y_t, F_{m-1}(x_t) - \gamma_m h_m(x_t)) \tag{3.4}$$

According to (ZHAN et al., 2020), the performance of the Gradient Boosting Regressor can be affected by three parameters: maximum number of trees, learning rate, and max-depth of the tree. The best combination of the parameters enables the optimal result of the model. The first refers to the total number of trees (i.e. weak learners) that are integrated into Gradient Boosting Regressor. The second parameter sets the contribution of each weak learner to the final results, with values between 0 and 1. The third parameter expresses the complexity of the tree. Gradient Boosting Regressor is a strong learner formed by the combination of weak learners. Therefore, the max-depth of each tree must be controlled in order to limit the complexity of the whole system.

In order to summarize the algorithm, Table 4 presents the Gradient Boosting Regressor's pseudo-code, based on (ZHAN et al., 2020).

Table 4 – Gradient Boosting Regressor's Pseudo-code

Input: Training set $\{(x_1, y_1), ..., (x_T, y_T)\}$, differentiable loss function $L(y, F(x))$, Maximum number of trees (**M**).
Initial value: $F_0(x) = \arg\min_\gamma \sum_{t=1}^{T} L(y_t, \gamma)$

For t=1 to M:
  For t=1 to T:
    Calculate $z_m(x_t) = -\dfrac{\partial L(y_t\, F_{m-1}(x_t))}{\partial F_{m-1}(x_t)}$
  End For
  Fit regression tree $h_m(x)$ to predict the negative gradient z_m using input variables x.
  Compute the gradient descent step size (learning rate) given by:
  $\gamma_m = \arg\min_\gamma \sum_{t=1}^{T} L(y_t\, F_{m-1}(x_t) - \gamma_m h_m(x_t))$.
  Update Model $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$

End For
Output model $F_m(x)$.

3.4 Decision Tree Regressor

Among the first statistical algorithms to be implemented in electronic form, Decision Tree is a widely used algorithm for regression problems (SCAVUZZO et al., 2018), (SAGHAFI; ARABLOO, 2017), (CHOUDHURY et al., 2020). The main characteristic of this model is the recursive subsetting of the data according to the values of the predictors in order to progressively narrow the possible values into decision nodes until the model is able to reach a prediction (leaf nodes). (VILLE, 2013).

Figure 3 is a visual representation of a decision tree. For each level of the tree, it's possible to observe the reduction of the mean squared error and the progressive subsetting of the data until only one data point is left at the leaf node.

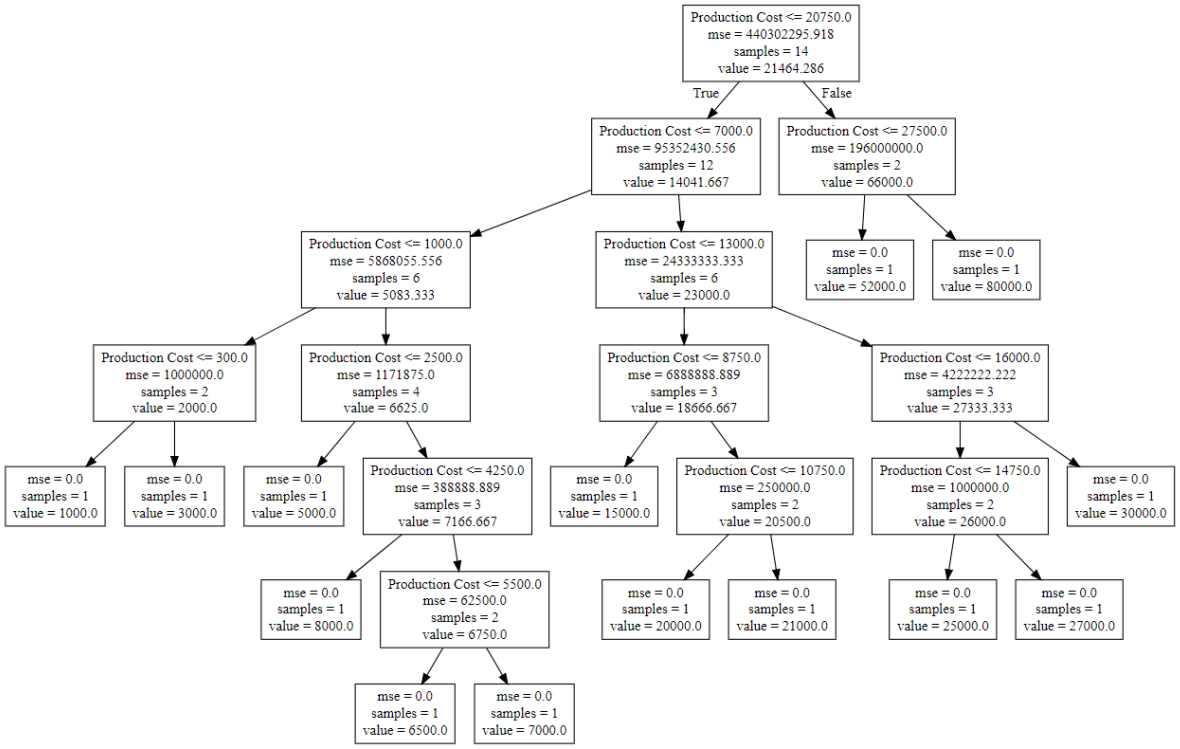

Figure 3 – Decision tree representation

Source: (GEEKSFORGEEKS. . . ,)

Splitting and pruning are two concepts that explain how decision trees work and can be optimized, respectively. According to (KORSTANJE, 2021), the splitting starts at the root node, with the whole dataset. For a set of attributes that characterize the target variable, the first decision node splits the dataset by choosing, among all possible options, the split that results in the lowest error. Then, the original dataset is split into two groups. The procedure is repeated until no further splitting is possible, i.e. there's only one data point left. Pruning is a procedure to reduce complexity and avoid overfitting. Two examples of pruning are cutting branches that are least needed after a decision tree grew completely and adding a complexity parameter that prevents the trees from becoming too detailed.

3.5 ePL-KRLS-DISCO

The ePL-KRLS-DISCO model is a fuzzy rule-based system (ALVES; AGUIAR, 2021). Fuzzy logic is an approach based on degrees of truth instead of the usual binary boolean logic. Fuzzy rule-based systems express their knowledge base with a collection of fuzzy if-then rules (PEDRYCZ, 1993). The algorithm creates rules (Equation 3.5) by clustering similar inputs. For each new observation from the database, the algorithm computes the output using the most suitable rule and updates its rules' quality by improving the quality of the clusters.

$$Ri: \quad \textbf{IF} \quad \underbrace{x \quad \text{is} \quad Ai}_{\text{Antecedent}} \quad \textbf{THEN} \quad \underbrace{yi = f_i(x, \theta_i)}_{\text{Consequent}} \tag{3.5}$$

where Ri is the i-th fuzzy rule, $i = \{$ 1,2,...,$R\}$, R is the number of fuzzy rules, $x = [x_1, \ldots, x_m]^T \in IR^m$ is the input, m is the number of attributes in the input vector, Ai is the fuzzy set of the i-th fuzzy rule, and yi is the output of the i-th rule calculated as a function of the input and the consequent parameters.

4 EXPERIMENTAL RESULTS

The five Machine Learning models mentioned in Section 3 were trained and tested using the free version of Google Colaboratory platform, which is a serverless Jupyter notebook environment for interactive development (BISONG, 2019). The Python Notebook file and the datasets used for this study can be found on: <https://bit.ly/33qoJZe>

Since most of the transports' characteristics were categorical variables (Table 3), it was necessary to perform some preprocessing in the database to have only integers and floats as inputs to the models. For this, the technique Label-Encoding (HANCOCK; KHOSHGOFTAAR, 2020) was implemented, converting the categorical variables into an associated integer number. To preserve the company's sensitive information, the original data is not shown in this study. Due to the quality of the data extracted, with a 0% missing rate, no additional work was needed to replace missing values.

Then, each of the 9 datasets was separated into random training and test subsets on the ratio of 85:15 using the function train_test_split from the scikit-learn Python machine learning library (PEDREGOSA et al., 2011). A parameter of this function, named random_state, is a pseudo-random number generator and controls the shuffling applied to the data before applying the split (PEDREGOSA et al., 2011). The algorithms were put inside a loop structure, altering the random_state parameter from 1 to 50. The results of each iteration were recorded in order to compare the average outcome of each model and its standard deviation. Table 5 presents the parameters used with the ML Models. A heuristic analysis was performed to set the values for the models' parameters.

Table 5 – Machine Learning models' parameters

Model	**Parameters**
Linear Model Ridge	$\alpha = 0.1$
KNN Regressor	n_neighbors = 2
Gradient Boosting Regressor	n_estimators = 500, max_depth = 4, min_samples_split = 5, learning_rate = 0.15, loss = squared_error
Decision Tree Regressor	max_depth=2
ePL-KRLS-DISCO	$\alpha = 0.001$, $\beta = 0.06$, $\lambda = 0.0000001$, $\sigma = 0.3$, e_utility = 0.05

Source: The author

4.1 Evaluation method

The evaluation of the models was measured with three error measures - Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Non-Dimensional Index Error (NDEI). Equations 2.2, 2.4, 2.3 describe the error metrics. Also, another relevant metric to compare the performance of the models is computational complexity. The model execution time is usually a good way to represent it since faster computational speed increases the possibility of the algorithm deployment. (CORMEN et al., 2009).

4.2 Models' results

Table 6 summarizes the results of the models for each dataset, evaluated with the metrics presented above.

Table 6 – Results of the predicitons

Dataset	ML Model	RMSE	NDEI	MAE	Time (s)
DS_1	Linear Ridge Regression	1.49 ± 0.14	0.98 ± 0.07	1.00 ± 0.06	0.02 ± 0.01
	KNN Regressor	3.01 ± 0.13	1.98 ± 0.08	1.41 ± 0.02	0.02 ± 0.00
	Gradient Boosting Regressor	1.36 ± 0.12	0.91 ± 0.07	0.92 ± 0.04	1.78 ± 0.33
	Decision Tree Regressor	1.54 ± 0.11	1.03 ± 0.06	1.01 ± 0.04	0.03 ± 0.02
	ePL-KRLS-DISCO	1.31 ± 0.11	0.87 ± 0.07	0.97 ± 0.07	651.99 ± 750.32
DS_2	Linear Ridge Regression	1.59 ± 0.11	1.05 ± 0.06	1.03 ± 0.04	0.01 ± 0.01
	KNN Regressor	2.88 ± 0.18	1.90 ± 0.10	1.35 ± 0.04	0.02 ± 0.00
	Gradient Boosting Regressor	1.52 ± 0.11	1.01 ± 0.06	0.98 ± 0.04	1.85 ± 0.48
	Decision Tree Regressor	1.64 ± 0.10	1.08 ± 0.06	1.06 ± 0.04	0.01 ± 0.00
	ePL-KRLS-DISCO	1.49 ± 0.18	0.98 ± 0.12	1.07 ± 0.05	558.88 ± 682.01
DS_3	Linear Ridge Regression	1.46 ± 0.10	0.98 ± 0.05	0.97 ± 0.04	0.01 ± 0.00
	KNN Regressor	2.87 ± 0.17	1.92 ± 0.11	1.36 ± 0.04	0.01 ± 0.00
	Gradient Boosting Regressor	1.45 ± 0.09	0.97 ± 0.06	0.96 ± 0.03	2.22 ± 0.67
	Decision Tree Regressor	1.49 ± 0.09	1.00 ± 0.05	1.00 ± 0.04	0.02 ± 0.00
	ePL-KRLS-DISCO	1.40 ± 0.15	0.94 ± 0.10	1.04 ± 0.10	600.13 ± 771.50
DS_4	Linear Ridge Regression	1.69 ± 0.10	1.13 ± 0.05	1.06 ± 0.04	0.02 ± 0.01
	KNN Regressor	3.10 ± 0.15	2.06 ± 0.09	1.42 ± 0.04	0.02 ± 0.01
	Gradient Boosting Regressor	1.75 ± 0.12	1.16 ± 0.07	1.05 ± 0.04	1.96 ± 0.68
	Decision Tree Regressor	1.71 ± 0.10	1.14 ± 0.05	1.07 ± 0.04	0.01 ± 0.00
	ePL-KRLS-DISCO	1.55 ± 0.19	1.03 ± 0.12	1.11 ± 0.07	443.58 ± 568.89
DS_5	Linear Ridge Regression	1.55 ± 0.10	1.01 ± 0.05	1.00 ± 0.04	0.02 ± 0.01
	KNN Regressor	2.91 ± 0.21	1.90 ± 0.12	1.36 ± 0.05	0.02 ± 0.01
	Gradient Boosting Regressor	1.47 ± 0.11	0.96 ± 0.07	0.95 ± 0.04	1.56 ± 0.46
	Decision Tree Regressor	1.57 ± 0.11	1.02 ± 0.06	1.00 ± 0.04	0.01 ± 0.00
	ePL-KRLS-DISCO	1.44 ± 0.16	0.94 ± 0.10	1.03 ± 0.05	375.96 ± 474.09
DS_6	Linear Ridge Regression	1.71 ± 0.11	1.04 ± 0.05	1.06 ± 0.03	0.01 ± 0.00
	KNN Regressor	2.53 ± 0.17	1.54 ± 0.09	1.23 ± 0.04	0.01 ± 0.00
	Gradient Boosting Regressor	1.24 ± 0.11	0.75 ± 0.06	0.80 ± 0.03	1.71 ± 0.45

Continues on next page

Table 6 – *Continuation*

Dataset	ML Model	RMSE	NDEI	MAE	Time (s)
	Decision Tree Regressor	1.63 ± 0.08	0.99 ± 0.04	1.04 ± 0.03	0.01 ± 0.00
	ePL-KRLS-DISCO	1.48 ± 0.16	0.90 ± 0.10	1.06 ± 0.05	420.83 ± 544.15
DS_7	Linear Ridge Regression	1.82 ± 0.15	1.13 ± 0.08	1.09 ± 0.05	0.04 ± 0.02
	KNN Regressor	3.49 ± 0.31	2.17 ± 0.17	1.49 ± 0.07	0.03 ± 0.01
	Gradient Boosting Regressor	1.76 ± 0.18	1.10 ± 0.10	1.03 ± 0.06	1.16 ± 0.35
	Decision Tree Regressor	1.82 ± 0.14	1.13 ± 0.07	1.10 ± 0.05	0.01 ± 0.00
	ePL-KRLS-DISCO	1.60 ± 0.25	1.00 ± 0.15	1.11 ± 0.06	171.51 ± 221.97
DS_8	Linear Ridge Regression	1.55 ± 0.11	0.99 ± 0.06	0.99 ± 0.04	0.01 ± 0.00
	KNN Regressor	2.93 ± 0.25	1.86 ± 0.14	1.34 ± 0.06	0.01 ± 0.00
	Gradient Boosting Regressor	1.54 ± 0.10	0.98 ± 0.07	0.96 ± 0.04	1.61 ± 0.52
	Decision Tree Regressor	1.48 ± 0.10	0.94 ± 0.06	0.97 ± 0.04	0.01 ± 0.00
	ePL-KRLS-DISCO	1.38 ± 0.13	0.87 ± 0.08	1.00 ± 0.05	266.15 ± 344.02
DS_9	Linear Ridge Regression	1.27 ± 0.11	0.89 ± 0.07	0.90 ± 0.04	0.02 ± 0.01
	KNN Regressor	2.42 ± 0.21	1.70 ± 0.12	1.24 ± 0.06	0.02 ± 0.01
	Gradient Boosting Regressor	1.34 ± 0.14	0.94 ± 0.09	0.90 ± 0.05	1.45 ± 0.41
	Decision Tree Regressor	1.34 ± 0.12	0.94 ± 0.07	0.91 ± 0.04	0.01 ± 0.00
	ePL-KRLS-DISCO	1.27 ± 0.11	0.89 ± 0.07	0.94 ± 0.05	287.04 ± 366.55

End of table

Source: The author

In order to statistically validate the results, a One-Way ANOVA test was performed. According to (SCHEFFE, 1999), the procedure uses the variances of the groups to determine whether the means are different or not. The comparison of variance between group means against the variance within groups works as a way to determine whether the groups are all part of a larger population or distinct populations with different characteristics. The null hypothesis states that all populations' means are equal while the alternative hypothesis states that at least one is different. (MONTGOMERY; RUNGER, 2003). The categorical factor used in the test is the ML models, while the continuous response variable is the RMSE of each model, for each dataset.

Table 7 presents the results of the tests, considering a significance level (α) of 0.05. If the $p - value$ is lower than α, there's not enough information to conclude the null hypothesis is true and the statement that all the models have equal accuracy is rejected.

According to Table 7, for all datasets, at least one of the models has a different accuracy. Analyzing the means comparison outcome from One-way ANOVA, it's possible to identify to which models each model has different accuracy. Table 8 presents the ranking of the models in terms of better accuracy for each dataset. Also, it presents for each model, the models to which it does not overlap the confidence interval for its RMSE value.

Table 7 – ANOVA results

Dataset	***p-value***	**Observation**
DS_1	< 0.001	H_0 rejected
DS_2	< 0.001	H_0 rejected
DS_3	< 0.001	H_0 rejected
DS_4	< 0.001	H_0 rejected
DS_5	< 0.001	H_0 rejected
DS_6	< 0.001	H_0 rejected
DS_7	< 0.001	H_0 rejected
DS_8	< 0.001	H_0 rejected
DS_9	< 0.001	H_0 rejected

Source: The author

Table 8 – Means comparison

Dataset	**# Rank**	**ML Model**	**Differs from**
DS_1	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Gradient Boosting Regressor	1, 3, 4, 5
	3	Linear Model Ridge	1, 2, 4, 5
	4	Decision Tree Regressor	1, 2, 3 ,5
	5	KNN Regressor	1, 2, 3, 4
DS_2	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Gradient Boosting Regressor	1, 3, 4, 5
	3	Linear Model Ridge	1, 2, 5
	4	Decision Tree Regressor	1, 2, 5
	5	KNN Regressor	1, 2, 3, 4
DS_3	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Gradient Boosting Regressor	1, 5
	3	Linear Model Ridge	1, 5
	4	Decision Tree Regressor	1, 5
	5	KNN Regressor	1, 2, 3, 4
DS_4	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Linear Model Ridge	1, 5
	3	Decision Tree Regressor	1, 5
	4	Gradient Boosting Regressor	1, 5
	5	KNN Regressor	1, 2, 3, 4
DS_5	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Gradient Boosting Regressor	1, 3, 4, 5
	3	Linear Model Ridge	1, 2, 5
	4	Decision Tree Regressor	1, 2, 5

Continues on next page

Table 8 – *Continuation*

Dataset	# Rank	ML Model	Differs from
	5	KNN Regressor	1, 2, 3, 4
DS_6	1	Gradient Boosting Regressor	2, 3, 4, 5
	2	ePL-KRLS-DISCO	1, 3, 4, 5
	3	Decision Tree Regressor	1, 2, 4 ,5
	4	Linear Model Ridge	1, 2, 3, 5
	5	KNN Regressor	1, 2, 3, 4
DS_7	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Gradient Boosting Regressor	1, 5
	3	Decision Tree Regressor	1, 5
	4	Linear Model Ridge	1, 5
	5	KNN Regressor	1, 2, 3, 4
DS_8	1	ePL-KRLS-DISCO	2, 3, 4, 5
	2	Decision Tree Regressor	1, 3, 4, 5
	3	Gradient Boosting Regressor	1, 2, 5
	4	Linear Model Ridge	1, 2, 5
	5	KNN Regressor	1, 2, 3, 4
DS_9	1	ePL-KRLS-DISCO	3, 4, 5
	2	Linear Model Ridge	3, 4, 5
	3	Decision Tree Regressor	1, 2, 5
	4	Gradient Boosting Regressor	1, 2, 5
	5	KNN Regressor	1, 2, 3, 4

End of table

Source: The author

The results of the statistical test presented in Tables 7 and 8, show that ePL-KRLS-DISCO demonstrated the best accuracy for datasets 1, 2, 3, 4, 5, 7, 8, and 9, compared to the other presented ML Models, with a 95% confidence level, achieving the lowest values of RMSE and not overlapping the confidence interval of its mean RMSE with any other model. Considering all datasets, the model's mean value of RMSE presents a reduction of 64% when compared to the current method of prediction. Regarding dataset 6, Gradient Boosting Regressor demonstrated the best result, with an RMSE of 1.24 ± 0.11, followed by ePL-KRLS-DISCO, with an RMSE of 1.48 ± 0.16. The KNN Regressor performed the worst results of accuracy in all datasets. Linear Model Ridge and Decision Tree Regressor performed average results when compared with the other models. Also, according to the

One-way ANOVA Test, these models' accuracy results are statistically equal for datasets 2, 3, 4, 5, and 7.

Regarding the computational cost, estimated by the algorithm's execution time, ePL-KRLS-DISCO presented values starting at 171.51 ± 221.97 seconds up to 651.99 ± 750.32 seconds. Gradient Boosting Regressor performed the second-worst results with values ranging from 1.16 ± 0.35 seconds up to 2.22 ± 0.67 seconds. All other models presented execution time values lower than 0.04 seconds with almost zero standard deviation.

4.3 Discussions

Figures 4 and 5 present the performance of the Machine Learning approach (Table 6) in comparison with the performance of the current method of prediction (Table 2). Despite the KNeigbors Regressor, the Machine Learning models tested show that it is possible to reduce the variance and achieve significantly lower values of RMSE than the method that is currently used, which presents values ranging from 2.55 hours up to 6.37 hours. RMSE values that are significantly higher than MAE are a good indicator of large residues in the prediction due to the quadratic nature of the Equation (2.2) (CHAI; DRAXLER, 2014). The Machine Learning approach also presented lower values of MAE in all datasets tested, with errors inferior to 1 hour. The results of this approach reveal the potential to enable more reliable data-driven decisions regarding the inbound process of scrap metal.

Figure 4 – Plot of the RMSE values of the models and current method of prediction

Source: The author

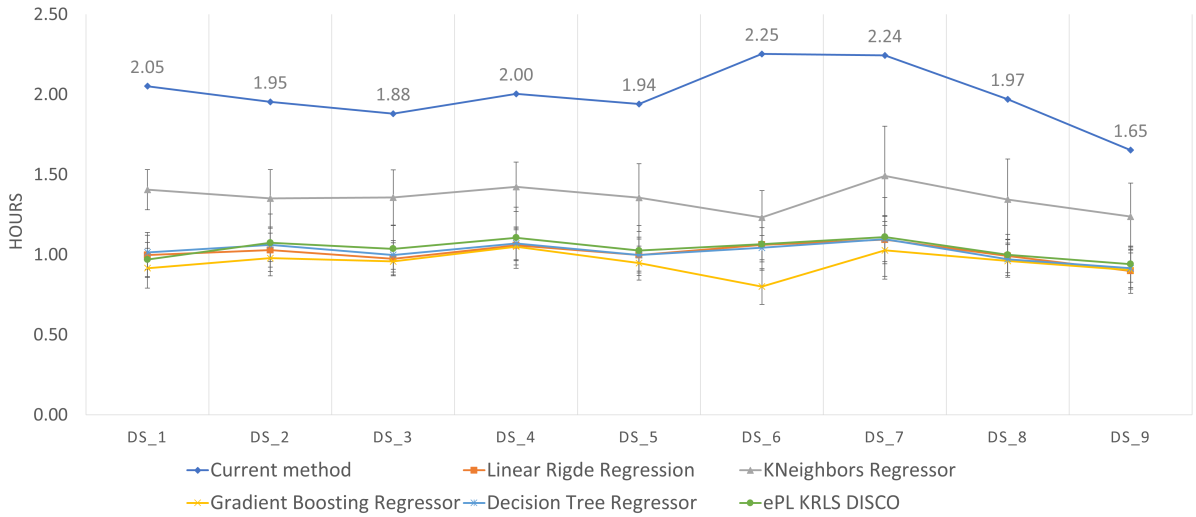

Figure 5 – Plot of the MAE values of the models and current method of prediction

Source: The author

5 CONCLUSIONS

This study presented the results of a Machine Learning approach to the prediction of vehicles' Length of Stay for the inbound operation of scrap metal of a Brazilian steel plant. The results' simulations show a feasible solution to improve the accuracy of the LOS prediction issue.

The current method of prediction does not have a good performance to contribute to decision-making in the day-to-day operations of the steel plant, nor to tactical or strategical level. Furthermore, this method only considers one attribute to make a prediction. Therefore, the Machine Learning approach is justifiable by the importance of the KPI to the company and by the amount of data available that is not used for prediction.

The results of this study presented five Machine Learning models that are more accurate than the current method that is used. This represents an opportunity for the company to consider using Machine Learning to predict important indicators and enable more robust data-driven decisions.

Future work includes the evaluation of other related data sources to improve accuracy (e.g. data from the scrap metal purchase plan, stock level, weather conditions). Also, this study is an initial step for a decision model to be implemented, based on the predicted LOS metric. In the State-Of-The-Art of the 4.0 Industry, a decision model would be able to optimize the process in real-time. In the context of the inbound process of scrap metal, entrance anticipation, selection of unloading location and priority pass of vehicles are routine decisions that could be optimized and automated.

REFERENCES

ALVES, K. S. T. R.; AGUIAR, E. P. de. A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers. *Applied Soft Computing*, Elsevier, v. 112, p. 107764, 2021.

BAN, T. et al. Referential knn regression for financial time series forecasting. In: SPRINGER. *International Conference on Neural Information Processing*. [S.l.], 2013. p. 601–608.

BISONG, E. Google colaboratory. In: _____. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019. p. 59–64. ISBN 978-1-4842-4470-8. Disponível em: <https://doi.org/10.1007/978-1-4842-4470-8_7>.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific Model Development Discussions*, v. 7, n. 1, p. 1525–1534, 2014.

CHOUDHURY, S. et al. Predicting crack through a well generalized and optimal tree-based regressor. *International Journal of Structural Integrity*, Emerald Group Holdings Ltd., v. 11, p. 783–807, 9 2020. ISSN 17579872.

CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT Press, 2009.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, JSTOR, p. 1189–1232, 2001.

GEEKSFORGEEKS Python | Decision Tree Regression using sklearn. Disponível em: <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/>.

HANCOCK, J. T.; KHOSHGOFTAAR, T. M. Survey on categorical data for neural networks. *Journal of Big Data*, SpringerOpen, v. 7, n. 1, p. 1–41, 2020.

HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, [Taylor Francis, Ltd., American Statistical Association, American Society for Quality], v. 12, n. 1, p. 55–67, 1970. ISSN 00401706. Disponível em: <http://www.jstor.org/stable/1267351>.

HU, C. et al. Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. *Applied Energy*, Elsevier, v. 129, p. 49–55, 2014.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2018.

KELLEHER, J. D.; NAMEE, B. M.; D'ARCY, A. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. [S.l.]: MIT Press, 2020.

KOHLI, S.; GODWIN, G. T.; UROLAGIN, S. Sales prediction using linear and knn regression. In: *Advances in Machine Learning and Computational Intelligence*. [S.l.]: Springer, 2021. p. 321–329.

KORSTANJE, J. *Advanced Forecasting with Python*. [S.l.]: Apress, 2021.

MONTGOMERY, D. C.; RUNGER, G. C. Estatística aplicada e probabilidade para engenheiros, 2ª. *Ed. Rio de Janeiro: Editora LTC*, 2003.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PEDRYCZ, W. *Fuzzy control and fuzzy systems*. [S.l.]: Research Studies Press Ltd., 1993.

SAGHAFI, H.; ARABLOO, M. Modeling of co2 solubility in mea, dea, tea, and mdea aqueous solutions using adaboost-decision tree and artificial neural network. *International Journal of Greenhouse Gas Control*, Elsevier Ltd, v. 58, p. 256–265, 3 2017. ISSN 17505836.

SCAVUZZO, J. M. et al. Modeling dengue vector population using remotely sensed data and machine learning. *Acta Tropica*, Elsevier B.V., v. 185, p. 167–175, 9 2018. ISSN 18736254.

SCHAPIRE, R. E. A brief introduction to boosting. In: CITESEER. *Ijcai*. [S.l.], 1999. v. 99, p. 1401–1406.

SCHEFFE, H. *The analysis of variance*. [S.l.]: John Wiley & Sons, 1999. v. 72.

VILLE, B. D. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, Wiley Online Library, v. 5, n. 6, p. 448–455, 2013.

YAKOWITZ, S. Nearest-neighbour methods for time series analysis. *Journal of Time Series Analysis*, Wiley Online Library, v. 8, n. 2, p. 235–247, 1987.

YUZOV, O. V.; SEDYKH, A. M. *Metallurgist*, v. 47, n. 5/6, p. 201–205, 2003.

ZHAN, X. et al. Multi-step-ahead traffic speed forecasting using multi-output gradient boosting regression tree. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, Taylor and Francis Inc., v. 24, p. 125–141, 3 2020. ISSN 15472442.

ZHAO, X.; XIE, J. Forecasting errors and the value of information sharing in a supply chain. *International Journal of Production Research*, Taylor Francis, v. 40, n. 2, p. 311–335, 2002. Disponível em: <https://doi.org/10.1080/00207540110079121>.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA

Termo de Declaração de Autenticidade de Autoria

Declaro, sob as penas da lei e para os devidos fins, junto à Universidade Federal de Juiz de Fora, que meu Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Produção é original, de minha única e exclusiva autoria. E não se trata de cópia integral ou parcial de textos e trabalhos de autoria de outrem, seja em formato de papel, eletrônico, digital, áudio-visual ou qualquer outro meio.
Declaro ainda ter total conhecimento e compreensão do que é considerado plágio, não apenas a cópia integral do trabalho, mas também de parte dele, inclusive de artigos e/ou parágrafos, sem citação do autor ou de sua fonte.
Declaro, por fim, ter total conhecimento e compreensão das punições decorrentes da prática de plágio, através das sanções civis previstas na lei do direito autoral[1] e criminais previstas no Código Penal[2], além das cominações administrativas e acadêmicas que poderão resultar em reprovação no Trabalho de Conclusão de Curso.

Juiz de Fora, 23 de Fevereiro de 2022.

Victor Hugo Soares Pereira	201749011
NOME LEGÍVEL DO ALUNO (A)	Matrícula
Victor HS Pereira	177.774.817-89
ASSINATURA	CPF

[1] LEI N° 9.610, DE 19 DE FEVEREIRO DE 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.
[2] Art. 184. Violar direitos de autor e os que lhe são conexos: Pena – detenção, de 3 (três) meses a 1 (um) ano, ou multa.