Multiple Linear Regression & Random Forest Approach for Productivity Estimations of Bulldozers

Sara Moradi Jafarbeglo, Sinem Karaomeroglu, Tugberk Erdogmus, Ugur Eylik

Applied AI for Digital Production Management, Deggendorf Institute of Technology

Abstract

This paper investigates the application of machine learning models to estimate the productivity of bulldozers using synthetic datasets generated through Uniform Distribution, Monte Carlo Simulation, and Latin Hypercube Sampling. In the first phase, we used Multiple Linear Regression and Random Forest algorithms. In the second phase, we expanded our analysis by incorporating Ridge Regression, Lasso Regression, and Elastic Net, resulting in 15 different model-dataset combinations. Our results indicate that Random Forest and Elastic Net consistently outperform other models across all dataset types, providing valuable insights for construction industry applications.

Keywords: *Linear Regression, Random Forest, Ridge Regression, Lasso Regression, Elastic Net, Uniform Distribution, Monte Carlo Simulation, Latin Hypercube Sampling, Productivity Estimation, Bulldozers*

1.Introduction

Productiveness estimation in development is valuable for effective task planning and administration. Correct predictions permit higher useful resource allocation, scheduling, and rate management, directly impacting the overall success of construction initiatives. Normal ways for estimating bulldozer productivity predominantly depend on empirical formulation and knowledgeable judgment. These traditional approaches, at the same time broadly used, mainly suffer from massive inaccuracies and inefficiencies because of their inherent subjectivity and reliance on historic data that may not accurately mirror present venture conditions.

In recent years, computer studying has emerged as a robust software across more than a few

domains, including development. Its purposes span apparatus performance monitoring, predictive preservation, and optimization of operational workflows. For example, reports have validated the abilities of computing device learning in predicting gear failure and optimizing protection schedules, mainly to diminished downtime and operational fees. Nonetheless, despite these promising developments, there remains an incredible gap in the software of desktop learning especially for estimating bulldozer productivity. The important project lies in the scarcity of actual-world datasets vital for coaching powerful and accurate machine finding out units. The first step in timing and estimating the assignment's working fees is the significant work/hours of quite a few machines the success of which may also be viable with the aid of two approaches. One is to make use of the experience and proficient recommendation and the reverse use of the producers' manuals. A series of modified coefficients for making right corrections headquartered on the special operating stipulations and environmental settings inside which the venture is implemented are supplied. These comprise the category of soil to work on, potential of operators, website manager, and other explanations which affect the effectiveness of the machineries [1], [2]. Accordingly, present study is constrained, and there is a want for revolutionary methods to information new release and mannequin coaching to bridge this gap.

Building on the insights from the literature, this study aims to address the existing gap by investigating how machine learning models can be leveraged to improve the accuracy of productivity estimation for bulldozers. Specifically, it will explore the efficacy of utilizing synthetic datasets generated through Uniform Distribution, Monte Carlo Simulation, and Latin Hypercube Sampling. By employing these techniques, the research seeks to overcome the limitations posed by the lack of real-world data. Thus, the central research question of this study is: How can machine learning models, utilizing synthetic datasets generated through Uniform Distribution, Monte Carlo Simulation, and Latin Hypercube Sampling, improve the accuracy of productivity estimation for bulldozers?

2. Methods & Methodology

Simulating Bulldozer Productivity with Uncertainty

Based on the multiple factors which are given from Table 1 [3], the factors that have impact on production and the data collected for a sample bulldozer are detailed in the table. The techniques for generating datasets include Uniform Distribution, Monte Carlo Simulation, and Latin Hypercube Sampling.

Table 1. Factors influencing production and data collected for a sample bulldozer [3]

NO.	FACTOR	STATUS	SAMPLE
1	Total service life time (hours)	0 – 150,000	100,000
2	Service and maintenance condition	Good/Average/Rather poor/Poor	Good
3	Type of blade	Straight tilt dozer/U-tilt dozer/Semi U-tilt dozer/Angle dozer	U-tilt
4	Maximum blade capacity (m^3)	4.8/6.8/8.8/11.8	8.8
5	Blade sharpness	Good/Average/Rather poor/Poor	Average
6	Ripper used?	Yes/No	Yes
7	Time between gear shifting (seconds)	Less than 5/Between 5~10/More than 10	Less than 5
8	Operator's skill	Good/Average/Rather poor/Poor	Good
9	Overall operator's condition during the operation	Good/Average/Rather poor/Poor	Good
10	Site management quality	Good/Average/Rather poor/Poor	Average
11	Number of consecutive operational days	Between 0 ~100	7
12	Predominant soil type	Sand/Sandy clay/Clay/Gravel/Broken rocks	Broken rocks
13	Big pieces of rock exist on the site?	No/Rarely/Commonly	Commonly
14	Equipment maneuvering space	Easy/Average/Rather difficult/Difficult	Easy
15	Ground grade (%)	-25~25	-10%
16	Dozing distance (m)	0~150	20
17	Operation time	Morning/Afternoon/Night	Morning
18	Average temperature during operation (°C)	-15~45	20
		Actual Measured Productivity (Lm^3/Hour)	**150**

2.1 Generating the Datasets

The table will serve as the base for the model. It includes different factors that will affect bulldozer productivity, such as service and maintenance condition, blade type, and material characteristics. For each factor, the table will define its range of possible values or list the discrete categories it can fall under (e.g., excellent, good, fair for service condition).

Uniform Distribution

Uniform distribution comes into play when we are working with factors represented by categories. The uniform distribution is a continuous probability distribution and is concerned with events that are equally likely to happen [4]. By implementing an equal probability to each category that are in the table, we can improve random sampling to select a category for each data point in the modeling. This action gives an insurance of a balanced representation of the different possibilities for each factor.

Monte Carlo simulation

Monte Carlo simulation enters the scene as the overarching framework. Monte Carlo Simulation is a type of computational algorithm that uses repeated random sampling to obtain

the likelihood of a range of results occurring [5]. It uses random sampling to model the complex system of bulldozer productivity. We will draw random values from the probability distributions that are assigned to each factor in our table (including the uniform distributions for categorical factors). This approach will create a set of data points which each of them represents a unique scenario with its own combination of factor values. Basically, the model generates a multitude of possible situations that a bulldozer might face during operation.

Latin Hypercube Sampling

While random sampling is powerful, Latin Hypercube Sampling can make more improvement. Latin hypercube sampling (LHS) is a method for generating samples of random variations from a given probability distribution function [6]. It enhances the simulation by ensuring each variable (factor) in our table is represented throughout the generated data points. This stratified approach guarantees that the simulation does not skew towards specific categories or values within a factor's range. This is particularly useful for simulations that include numerous variables, like ours.

2.2 Feature Engineering and Preprocessing

Feature engineering is a process that we select, manipulate and transform raw data into features, and we can use in supervised machine learning. In general, it consists of five processes: feature creation, transformations, feature extraction, exploratory data analysis and benchmarking [7].

The table indicates several factors that can affect bulldozer productivity. These factors can be mostly categorized into two types:

1. Numerical Factors: These factors have numerical values, and they can be directly used by machine learning models. For example, "total service lifetime (hours)" or "ground grade (%)" are numerical factors.

2. Categorical Factors: These factors represent distinct categories, and before being used in machine learning models they need to be encoded. Examples in our table include "service condition" (good/fair/poor) or "blade type" (straight tilt/U-tilt/etc.).

Variance Inflation Factor

Preprocessing is a strong way to prevent a model from being overfitted. In machine learning models, two problems can arise during training and testing, overfitting and underfitting [8]. Underfitting means that when the generated model does not perform well in both training and

testing, overfitting explains that the generated model only performs well in training data and fails with testing data [8]. The variance inflation factor (VIF) is used for determining multicollinearity between variables and the result [9]. The factor is preferred in the study to understand collinearity. If the factor is 1, it means variables are not correlated, and if it is greater than 5 that causes the multicollinearity problem [9]. This threshold value can be set from 5 to 10, and in the study, the VIF multicollinearity threshold value is taken as 10 for VIF comparison and variable elimination.

2.3 Machine Learning Methods

There are many models that can be used, but these have been found to be more suitable:

1. Multiple Linear Regression: This provides a basic understanding of the factors' effects and finds linear relationships.

2. Random Forest: This method will investigate the possibility of non-linear relationships and make it easier for feature importance to obtain insights into the most important factors.

3. Regularization Methods: In situations when overfitting is a concern or we have many factors, applying ridge regression, lasso regression, or elastic net regression. Regularization methods are preferred: Ridge regression, lasso regression, or elastic net regression.

Combination of these methods can get a more general picture of the factors that can affect bulldozer productivity. Also, we can find the solution that gives the most accurate and understandable results for our specific data.

Our prediction methodology is:

- Multiple Linear Regression
- Random Forest
- Ridge Regression
- Lasso Regression
- Elastic Net Regression

Regression models are used to describe relationships between variables by fitting a line to the observed data. Regression estimates how a dependent variable changes or is related to the independent variables change [10].

Multiple linear regression is used when we want to predict the relationship between two or more independent variables and one dependent variable. in other words, it can be used when we want to know:

1. How strong the relationship is between independent variables and dependent variables [10].

2. The value of the dependent variable at a certain value of the independent variables [10].

The formula for a multiple linear regression is [10]:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_P x_{ip} + \varepsilon_i = X_i^T \beta + \varepsilon_i, \qquad i = 1, \ldots, n, \qquad (1)$$

Random forest is a popular supervised machine learning method for classification and regression. This method consists of using several Decision Trees and combining the trees predictions into an overall prediction [11]. In general, Random Forest is a suitable method for analyzing this data and predicting productivity. One reason is its capability for Complex Relationships. Random forest is great in working with complex and non-linear variables [11]. This is an important advantage because the factors listed in the table might not have perfectly linear relationships with bulldozer productivity.

One of the most aspect of Random Forest is finding the features importance [11]. This tool can find the factors affecting bulldozer productivity in our dataset. In other words, it can guide our decision-making to improve productivity by working on the most important factors.

Table 2. Random Forest's Feature Importance Output

Feature	Importance
Dozing distance (m) Ripper used?_1	0.137197
Average temperature during operation (°C) Rippe...	0.082818
Total service life time (hours) Ripper used?_1	0.063476
Number of consecutive operational days Ripper u...	0.053587
Ground grade (%) Ripper used?_1	0.044007
Dozing distance (m) Operation time 3	0.037214
Dozing distance (m) Average temperature during	0.029165
Dozing distance (m) Operation time_ 2	0.028696
Dozing distance (m) Type of blade_4	0.025764
Dozing distance (m) Type of blade_3	0.024649
Type of blade_4 Ripper used?_1	0.022516
Ripper used?1 Operation time 3	0.020467
Dozing distance (m) Maximum blade capacity (m3)...	0.020332
Dozing distance (m) Type of blade_2	0.018802
Dozing distance (m) Maximum blade capacity (m3)...	0.018063
Total service life time (hours) Dozing distance...	0.017663
Dozing distance (m)^2	0.017225
Number of consecutive operational days Dozing d...	0.016895
Dozing distance (m) Maximum blade capacity (ma)...	0.015735
Ripper used?_1 Operation time_2	0.015505
—	—
Mean feature importance:	**0.0104**

Dozing distance (m) can be considered as the most important feature because of its highest value (0.1372) in the "importance" column. This tells us that dozing distance has a strong effect on bulldozer productivity in this dataset.

Regularization

In this paper, it is focused on methods for handling overfitting. To investigate the relationship between variables (predictors) and output (predicted value), Least Squares calculates coefficients of variables by minimizing Residual Sum of Squares (RSS) [12]. RSS shows whether a linear regression model suits initial data [13]. During this investigation, overfitting may happen because although the model's accuracy is high and bias is low with training data, variance is high with testing/unseen data [13]. To handle overfitting, ways for reducing model complexity were searched. Defining a penalty for variable coefficients is introduced in the concept of regularization. Adding a penalty that shrinks the coefficients is called Coefficient Shrinkage [13]. The functionality of penalty is adding some bias to the model so it can perform well by resulting in low variance in the testing data and called as "Bias-Variance Trade Off" [13]. For penalty determination, Cross Validation (CV) is used, and its purpose is dividing the dataset into two subsets, which are training and testing [14]. In the study, K-fold cross validation is utilized with GridSearch from Scikit library in Python. K-fold cross-validation treats each k subsets as training data iterative processes and finds the best combination for one fold for training and k-1 fold for testing data [14]. Without regularization, the model created by the Least Squares technique results with high variance so that output becomes sensitive to minor changes in input data [13]. With regularization, predictions become less sensitive to the training data.

Ridge Regression

Ridge Regression, in other words, L2 regularization technique contributes to reduce model complexity and to prevent multicollinearity of the model [15], [13]. Ridge regression shrinks coefficients by introducing a penalty term into the RSS function [13]. Lambda, regularization parameter, is the penalty term that denotes the amount of shrinkage and can be varied from 0 to plus infinity [13], [16]. Lambda is determined by using Cross Validation with the types K-fold cross validation and leave-one-out cross validation [14]. Ridge estimator calculates new regression coefficients that reduce RSS. Predictor's effects are minimized and overfitting is reduced. Ridge regression does not shrink every coefficient by the same value; high-value coefficients are penalized greater than low-value coefficients [13].

Lasso Regression

Similar to L2 regularization, Lasso (Least Absolute Shrinkage Selector Operator) in other words L1 regularization contributes to reduce model complexity and to prevent multicollinearity of the model by shrinking coefficients with penalty [17]. While in Ridge Regression coefficients are shrunk towards zero, in Lasso Regression coefficients may be shrunk exactly to zero and get eliminated from the model [17]. Less important features in a dataset are eliminated by penalty and that also is called Feature Selection [17]. It is used when the number of features are more, because it automatically does feature selection [17].

Elastic Net Regression

Elastic net is a regression method for coefficient shrinkage with the combination of both L1 and L2 regularization [18]. Elastic net's aim is to cover shortcomings of each regularization technique [18]. Firstly, coefficients are found with the ridge regression. and after that lasso shrinkage method is applied [18]. In figure 1, it can be depicted that Elastic net prevents all correlated group's variables from being eliminated except one, and also it prevents all correlated variables stayed in the model with penalized coefficients together [19].

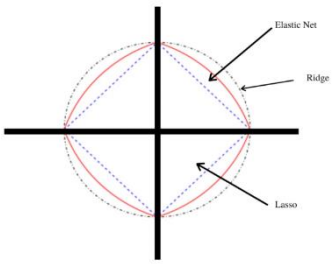

Figure 1. Regularization Techniques [19]

2.4 Model Training and Validation

Data Split

Data splitting is important for model generalization with reducing dependency to initial dataset. After preprocessing, polynomial feature transformation and eliminating multicollinear variables with VIF>10, data is splitted. As it is mentioned in the regularization section, splitting the data set for training and testing, some methods are preferred and K-fold cross validation technique is one of them. Treating one subset of the data is for training, and the remaining is used in testing process [20]. In the project, the data is splitted with 80% for training and 20% for testing.

Evaluation Metrics

In the study, two evaluation metrics are utilized when comparing results of the combination of different data generation methods and machine learning methods with respect to the model performance. These metrics are R-squared (R2) and Mean Squared Error (MSE). R2 means that the coefficient of determination is calculated using the formula displayed in equation (2) [16]. If R2 is calculated as 1, it means that there is strong correlation [21]. MSE, the formula is shown in equation (3), is used for comparison of line and best fit [21]. When MSE is close to zero, it means prediction is good [21], [16].

$$R^2 = 1 - \frac{SSR}{TSS} \tag{2}$$

SSR: Sum of Squares of residuals

TSS: Total Sum of squares

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - P_i)^2 \tag{3}$$

MSE: Mean Squared Error

n: Number of predictions

Y_i: Observed Values

P_i: Predicted Values

3. Results

Phase 1 Results

The Linear Regression model applied to the dataset generated using Uniform Distribution achieved an R^2 of 0.8291 and an MSE of 11.2443.

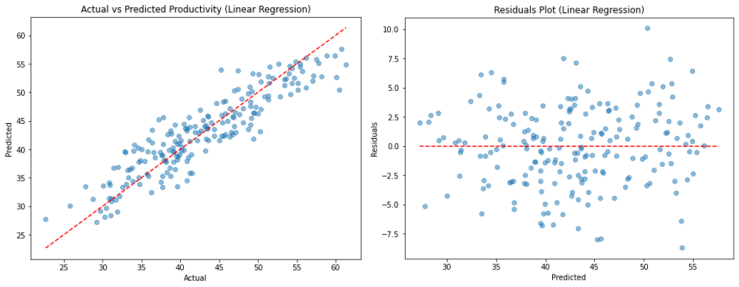

Figure 2. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Uniform Distribution Data

The Random Forest model applied to the dataset generated using Uniform Distribution

achieved an R^2 of 0.7840 and an MSE of 14.2111.

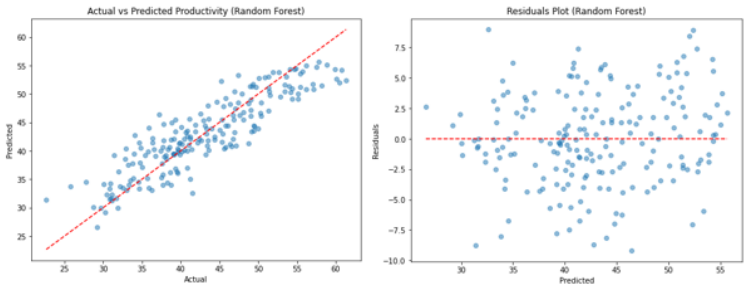

Figure 3. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Uniform Distribution Data

The Linear Regression model applied to the dataset generated using Monte Carlo Simulation achieved an R^2 of 0.9160 and an MSE of 4.1119.

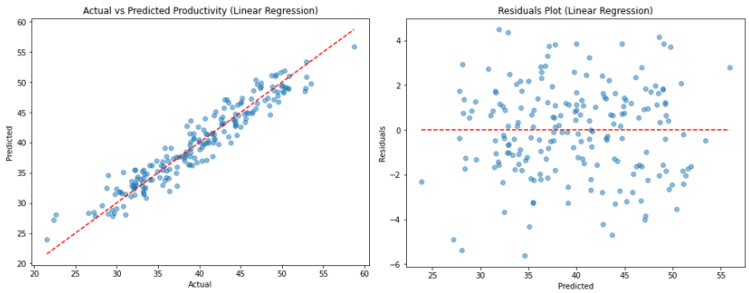

Figure 4. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Monte Carlo Simulation Data

The Random Forest model applied to the dataset generated using Monte Carlo Simulation achieved an R^2 of 0.8055 and an MSE of 9.5275.

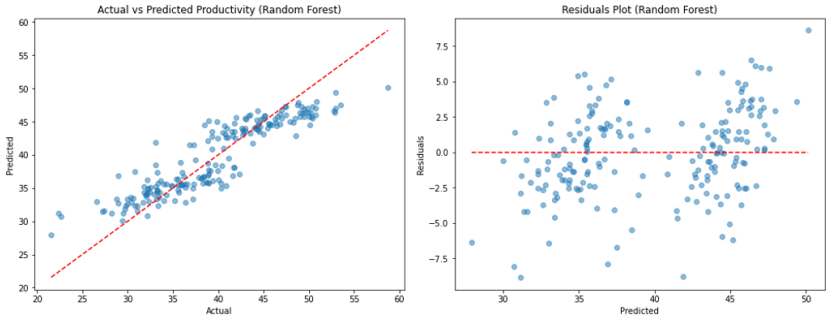

Figure 5. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Monte Carlo Simulation Data

The Linear Regression model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.9234 and an MSE of 5.3756.

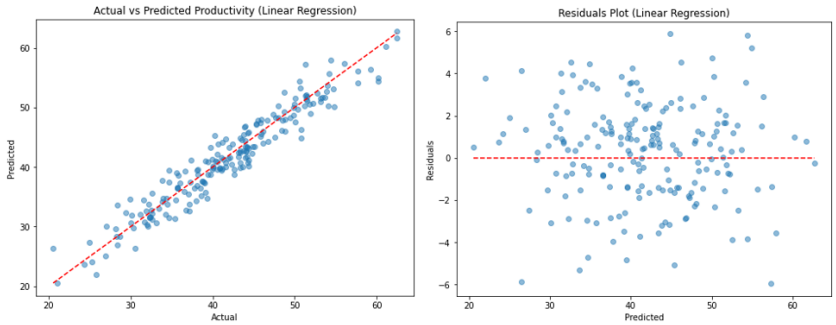

Figure 6. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Latin Hypercube Sampling Data

The Random Forest model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.8195 and an MSE of 11.2527.

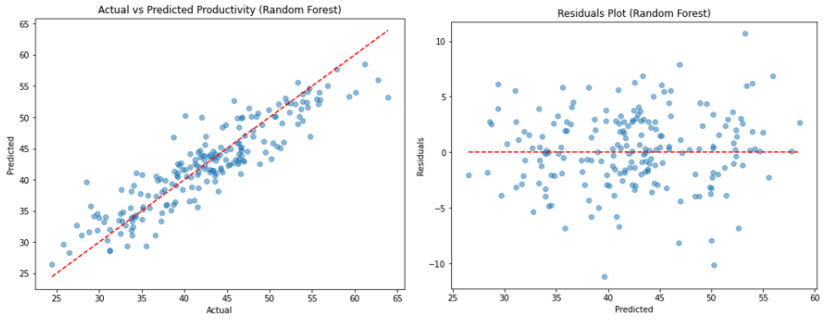

Figure 7. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Latin Hypercube Sampling Data

Table 3. Phase 1 Results

Model	R^2	MSE
Uniform Distribution with Linear Regression	0.9160	4.1119
Uniform Distribution with Random Forest	0.8055	9.5275
Monte Carlo Simulation with Linear Regression	0.9160	4.1119
Monte Carlo Simulation with Random Forest	0.8055	9.5275
Latin Hypercube Sampling with Linear Regression	0.9234	5.3756
Latin Hypercube Sampling with Random Forest	0.8195	11.2527

Phase 2 Results

The final Linear Regression model applied to the dataset generated using Uniform Distribution achieved an R^2 of 0.6910 and an MSE of 20.4564.

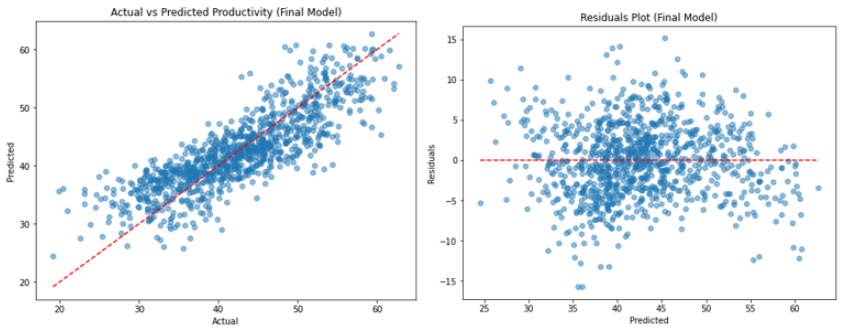

Figure 8. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Uniform Distribution Data

The final Random Forest model applied to the dataset generated using Uniform Distribution achieved an R^2 of 0.7486 and an MSE of 16.6451.

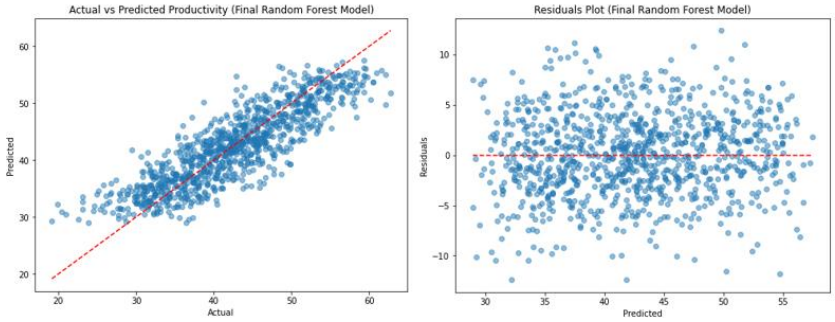

Figure 9. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Uniform Distribution Data

The final Ridge Regression model applied to the dataset generated using Uniform Distribution achieved an R^2 of 0.6912 and an MSE of 20.4442.

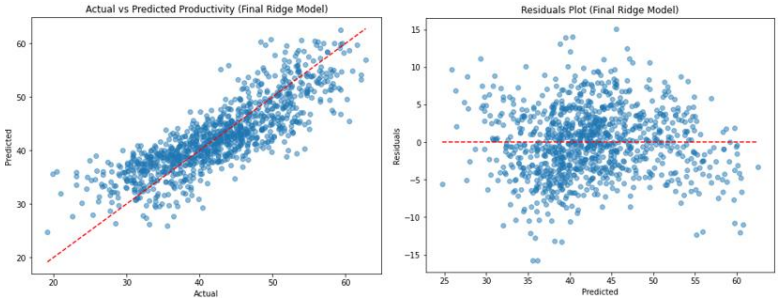

Figure 10. Actual vs Predicted Productivity and Residuals Plot for Ridge Regression Model on Uniform Distribution Data

The final Lasso Regression model applied to the dataset generated using Uniform Distribution

achieved an R^2 of 0.6914 and an MSE of 20.4325.

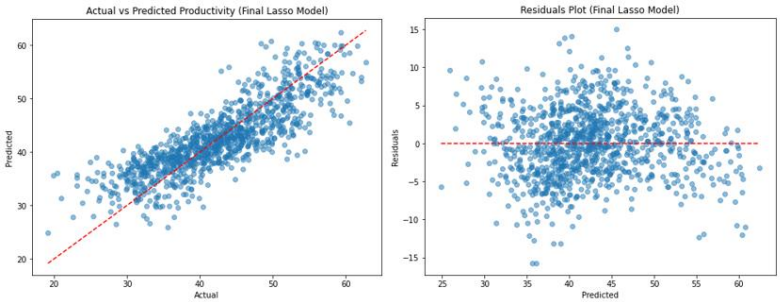

Figure 11. Actual vs Predicted Productivity and Residuals Plot for Lasso Regression Model on Uniform Distribution Data

The final Elastic Net Regression model applied to the dataset generated using Uniform Distribution achieved an R^2 of 0.6914 and an MSE of 20.4339.

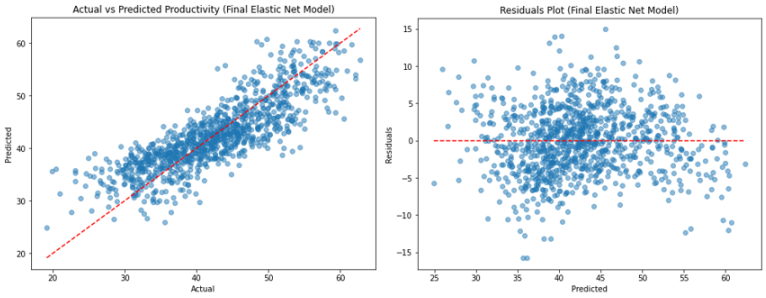

Figure 12. Actual vs Predicted Productivity and Residuals Plot for Elastic Net Regularization Model on Uniform Distribution Data

The final Linear Regression model applied to the dataset generated using Monte Carlo Simulation achieved an R^2 of 0.5900 and an MSE of 17.4087.

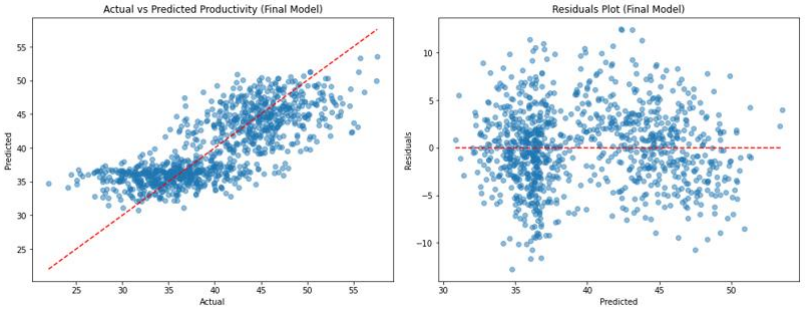

Figure 13. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Monte Carlo Simulation Data

The final Random Forest model applied to the dataset generated using Monte Carlo Simulation

achieved an R^2 of 0.7486 and an MSE of 16.6451.

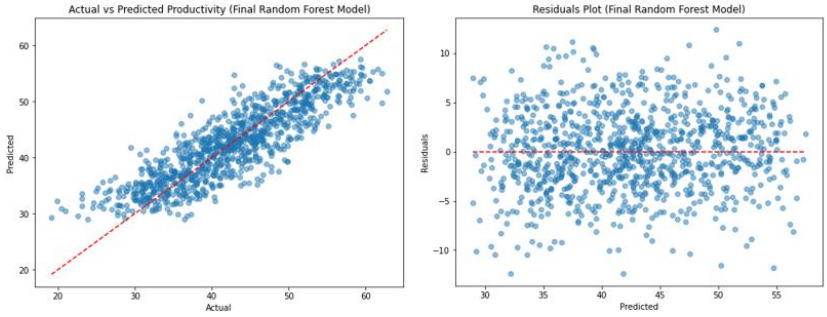

Figure 14. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Monte Carlo Simulation Data

The final Ridge Regression model applied to the dataset generated using Monte Carlo Simulation achieved an R^2 of 0.5900 and an MSE of 17.4116.

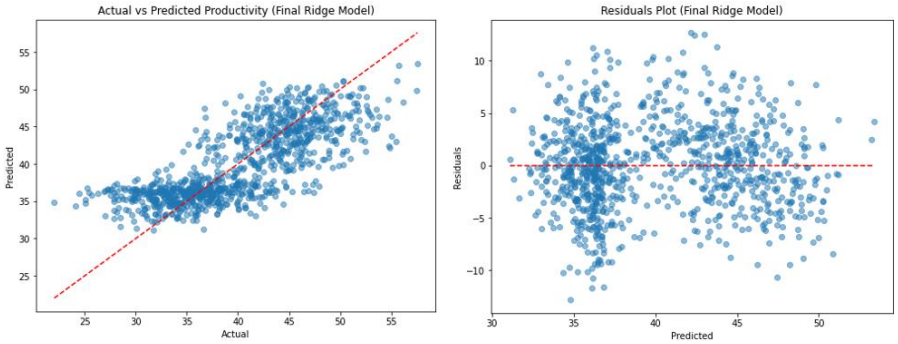

Figure 15. Actual vs Predicted Productivity and Residuals Plot for Ridge Regression Model on Monte Carlo Simulation Data

The final Lasso Regression model applied to the dataset generated using Monte Carlo Simulation achieved an R^2 of 0.5891 and an MSE of 17.4482.

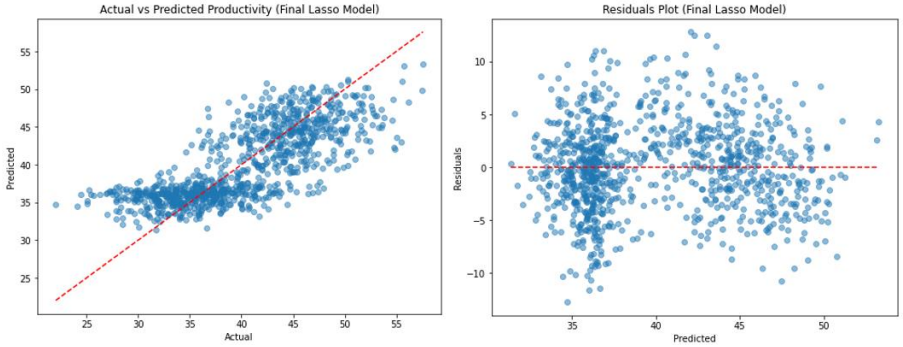

Figure 16. Actual vs Predicted Productivity and Residuals Plot for Lasso Regression Model on Monte Carlo Simulation Data

The final Elastic Net Regression model applied to the dataset generated using Monte Carlo

Simulation achieved an R^2 of 0.5891 and an MSE of 17.4487.

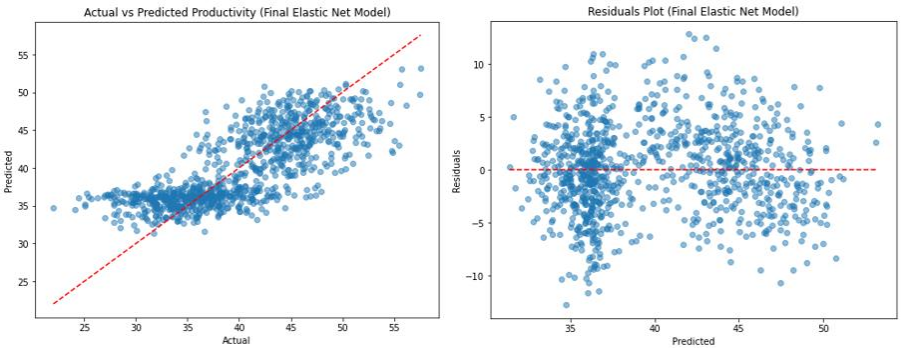

Figure 17. Actual vs Predicted Productivity and Residuals Plot for Elastic Net Regularization Model on Monte Carlo Simulation Data

The final Linear Regression model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.7058 and an MSE of 18.8080.

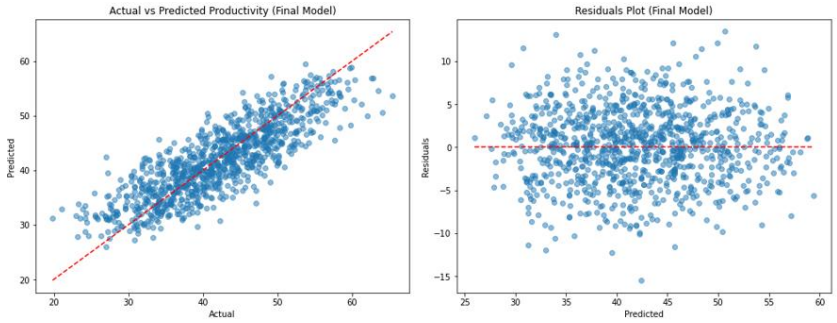

Figure 18. Actual vs Predicted Productivity and Residuals Plot for Linear Regression Model on Latin Hypercube Sampling Data

The final Random Forest model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.7239 and an MSE of 16.5157.

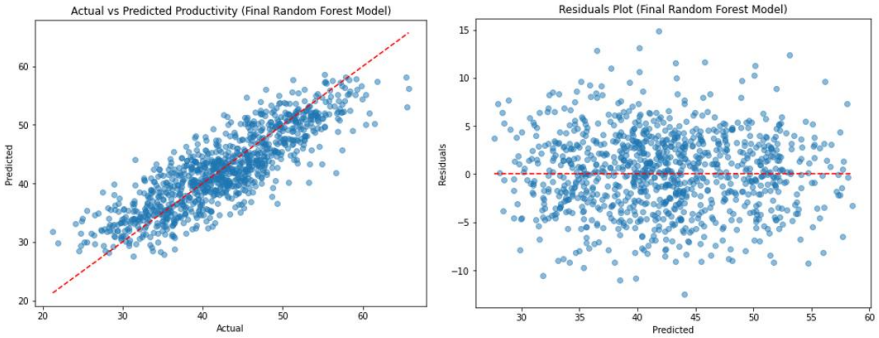

Figure 19. Actual vs Predicted Productivity and Residuals Plot for Random Forest Model on Latin Hypercube Sampling Data

The final Ridge Regression model applied to the dataset generated using Latin Hypercube

Sampling achieved an R^2 of 0.6991 and an MSE of 18.7784.

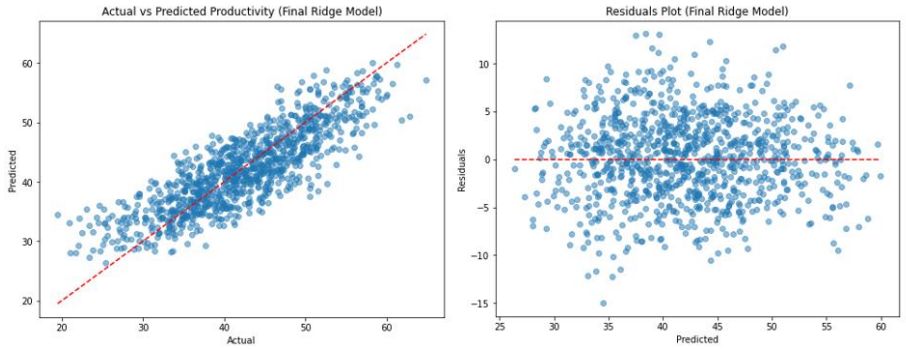

Figure 20. Actual vs Predicted Productivity and Residuals Plot for Ridge Regression Model on Latin Hypercube Sampling Data

The final Lasso Regression model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.7168 and an MSE of 18.9227.

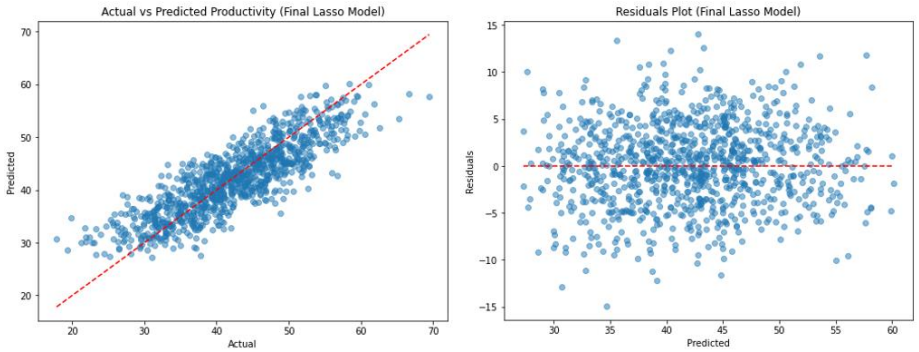

Figure 21. Actual vs Predicted Productivity and Residuals Plot for Lasso Regression Model on Latin Hypercube Sampling Data

The final Elastic Net Regression model applied to the dataset generated using Latin Hypercube Sampling achieved an R^2 of 0.6976 and an MSE of 18.8724.

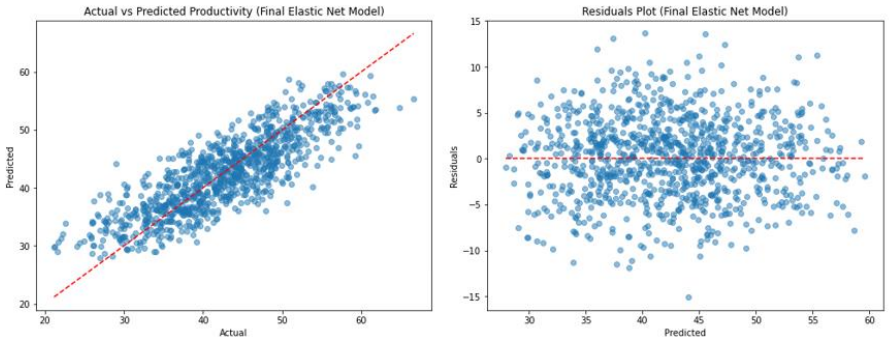

Figure 22. Actual vs Predicted Productivity and Residuals Plot for Lasso Regression Model on Latin Hypercube Sampling Data

Table 4. Phase 2 Results

Model	R^2	MSE
Uniform Distribution with Linear Regression	0.6910	20.4564
Uniform Distribution with Random Forest	0.7486	16.6451
Uniform Distribution with Ridge Regression	0.6912	20.4442
Uniform Distribution with Lasso Regression	0.6914	20.4325
Uniform Distribution with Elastic Net Regression	0.6914	20.4339
Monte Carlo Simulation with Linear Regression	0.5900	17.4087
Monte Carlo Simulation with Random Forest	0.7486	16.6451
Monte Carlo Simulation with Ridge Regression	0.5900	17.4116
Monte Carlo Simulation with Lasso Regression	0.5891	17.4482
Monte Carlo Simulation with Elastic Net Regression	0.5891	17.4487
Latin Hypercube Sampling with Linear Regression	0.7058	18.8080
Latin Hypercube Sampling with Random Forest	0.7239	16.5157
Latin Hypercube Sampling with Ridge Regression	0.6991	18.7784
Latin Hypercube Sampling with Lasso Regression	0.7168	18.9227
Latin Hypercube Sampling with Elastic Net Regression	0.6976	18.8724

4. Discussions

This study's main goal was to investigate how well different machine learning techniques work for determining bulldozer productivity. In particular, the efficiency of several machine learning models and data generation methods in forecasting productivity was evaluated.

To generate datasets for assessing the effectiveness of various machine learning models, such as Elastic Net, ridge regression, lasso regression, linear regression, random forest models, and three data generation techniques were used: uniform distribution, Monte Carlo simulation, and Latin hypercube sampling.

One thousand data points were used in the first stage. Across all data generating methods, it was shown that linear regression models consistently outperformed other models. Particularly, when combined with linear regression, the uniform distribution and Monte Carlo simulation produced R^2 values of 0.9160 and MSE values of 4.1119. Latin hypercube sampling with linear regression produced the best results, with an R^2 value of 0.9234 and an MSE of 5.3756. In comparison, the random forest model exhibited lower performance, with R^2 values of 0.8055 (MSE = 9.5275) for uniform distribution and Monte Carlo simulation, and an R^2 of 0.8195 (MSE = 11.2527) for Latin hypercube sampling.

The dataset size was expanded to 5000 in the second phase, as additional feature engineering and preprocessing methods were applied. These included polynomial features, encoding types, and hyperparameter tuning with cross-validation (CV) and Variance Inflation Factor (VIF) analysis. The goal of these improvements was to raise the accuracy of the predictions and model performance.

The phase's results provided more specific performance differences between various models and data generation techniques. Using Monte Carlo simulation and uniform distribution, for example, the random forest model performed robustly with both ($R^2 = 0.7486$, MSE = 16.6451). However, the R^2 values and MSE values of linear models, such as ridge, lasso, and Elastic Net regression, were comparatively lower R^2 values, ranging from 0.6910 to 0.7168 and MSE values between 17.4087 to 20.4564.

Latin hypercube sampling kept proving to be successful; it performed especially well with lasso regression (R2 = 0.7168, MSE = 18.9227) and random forest (R2 = 0.7239, MSE = 16.5157).

The clustering effect, as seen in the relevant graphs, was an interesting finding in the Monte Carlo simulation using the larger data set. The following was observed in the outputs received; Monte Carlo simulation did not display very healthy distributions depending on the increases in data. This may have happened because Monte Carlo Simulation tried to create a real-world dataset. This distribution can have an impact on how broadly applicable the model is and underscores the need to choose the right data collection techniques.

Strengths and Limitations

The thorough comparison of various data generation methods and machine learning models, which is reinforced using feature engineering and preprocessing approaches, is one of this study's strongest points. The model performed more accurately and precisely thanks to these techniques, which included encoding types, VIF analysis, polynomial features, and hyperparameter tuning using CV. There are restrictions to consider, though. Even using effective sampling techniques, the synthetic nature of the data could not fully represent the nuances of real-world data. Furthermore, the generalizability of the model can be impacted by the clustering effect that was seen in the Monte Carlo simulation with larger dataset sizes. The results' ability to be applied generally may be impacted by these restrictions. In the future, these results should be verified using real-world datasets, and the effects of other variables that can influence bulldozer productivity should be investigated.

Future Research

The findings point to several directions for future study. Using real-world datasets to test and improve this machine learning models' efficiency is a crucial topic. Future research should also think about including more complicated elements in the data generation process, such

as operator behavior and environmental factors. Examining further sophisticated machine learning methods, such as deep learning, may yield more information on productivity estimation. Ultimately, a more thorough analysis of various feature engineering strategies may result in the creation of prediction models that are even more robust.

Conclusion

In conclusion, a variety of feature engineering and data production strategies were used to assess the efficacy of several machine learning models for estimating bulldozer productivity. The findings show that the combination of robust models such as random forest with data production approaches such as Latin hypercube sampling leads to more accurate predictions. The study offers insightful information on the potential of machine learning in this field, despite some limitations like the artificial character of the data and clustering effects in Monte Carlo simulations. All things considered, these results improve our knowledge of bulldozer productivity estimation and highlight the need for more study using real-world data, more factors, and sophisticated modeling methods.

5. References

[1] G.D. Anon, Caterpillar performance Handbook, Caterpillar, Illinois, USA, 1997.

[2] L.Chao, M.J.Skibniewski, Estimating Construction Productivity: Neural-Network-Based Approach, ASCE Journal of Computing in Civil Engineering, 8 (2) (1994) 234-251.

[3] Rashidi, A., Rashidi Nejad, H., & Behzadan, A. H. (2009). Multiple linear regression approach for productivity estimation of bulldozers. In International Conference on Construction Engineering and Project Management (pp. 1140-1147). Korea Institute of Construction Engineering and Management.

[4] Gnewuch Michael, Kuo Frances, Niederreiter Harald, Woźniakowski Henryk: Uniform Distribution Theory and Applications. Oberwolfach Rep. 10 (2013), 2837-2917. doi: 10.4171/OWR/2013/49

[5] Reiter, D. (2008). The Monte Carlo Method, an Introduction. In: Fehske, H., Schneider, R., Weiße, A. (eds) Computational Many-Particle Physics. Lecture Notes in Physics, vol 739. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74686-7_3

[6] Hung, Y. (2013). Optimal Experiment Design, Latin Hypercube. In: Dubitzky, W., Wolkenhauer, O., Cho, KH., Yokota, H. (eds) Encyclopedia of Systems Biology. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_1233

[7] Verdonck, T., Baesens, B., Óskarsdóttir, M. et al. Special issue on feature engineering editorial. Mach Learn 113, 3917–3928 (2024). https://doi.org/10.1007/s10994-021-06042-2

[8] "Overfitting vs. Underfitting Explained," Built In. Accessed: Jul. 01, 2024. [Online]. Available: https://builtin.com/articles/overfitting-vs-underfitting

[9] M. O. Akinwande, H. G. Dikko, and A. Samson, "Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis," Open Journal of Statistics, vol. 05, no. 07, Art. no. 07, 2015, doi: 10.4236/ojs.2015.57075.

[10] Krzywinski, M., Altman, N. Multiple linear regression. Nat Methods 12, 1103–1104 (2015). https://doi.org/10.1038/nmeth.3665#

[11] Breiman, L. (2001) Random Forests. Machine Learning, 45, 5-32. http://dx.doi.org/10.1023/A:1010933404324

[12] "Least Squares Method: What It Means, How to Use It, With Examples," Investopedia. Accessed: Jul. 01, 2024. [Online]. Available: https://www.investopedia.com/terms/l/least-squares-method.asp

[13] "What Is Ridge Regression? | IBM." Accessed: Jul. 01, 2024. [Online]. Available: https://www.ibm.com/topics/ridge-regression

[14] S. Liu and E. Dobriban, "Ridge Regression: Structure, Cross-Validation, and Sketching." arXiv, Mar. 29, 2020. doi: 10.48550/arXiv.1910.02373.

[15] A. E. Hoerl and R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” Technometrics, vol. 12, no. 1, pp. 55–67, Feb. 1970, doi: 10.1080/00401706.1970.10488634.

[16] A. V. Tatachar, “Comparative assessment of regression models based on model evaluation metrics,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 9, pp. 853–860, 2021.

[17] What Is Lasso Regression? | IBM.” Accessed: Jan. 18, 2024. [Online]. Available: https://www.ibm.com/topics/lasso-regression

[18] W. Liu and Q. Li, “An Efficient Elastic Net with Regression Coefficients Method for Variable Selection of Spectrum Data,” PLOS ONE, vol. 12, no. 2, p. e0171122, ub 2017, doi: 10.1371/journal.pone.0171122.

[19] “Elastic Net,” Corporate Finance Institute. Accessed: Jul. 01, 2024. [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/elastic-net/

[20] S. Jarvis, “Lionfish Bot 2.0 2018-2019 Final Report,” PhD Thesis, WORCESTER POLYTECHNIC INSTITUTE, 2019. Accessed: Jul. 01, 2024. [Online]. Available: https://digital.wpi.edu/downloads/x059c9968

[21] M. Z. Naser and A. Alavi, “Insights into Performance Fitness and Error Metrics for Machine Learning,” Archit. Struct. Constr., vol. 3, no. 4, pp. 499–517, Dec. 2023, doi: 10.1007/s44150-021-00015-8.