CONSTRUCTION MATERIALS OBTAINED BY RECYCLING ASH FROM COAL-FIRED POWER PLANTS ASH PONDS. ESTIMATION OF BASIC MECHANICAL PROPERTIES BY MEANS OF MACHINE LEARNING ALGORITHMS. PART II – MACHINE LEARNING ALGORITHMS BENCHMARKING

Lucica Anghelescu, *University "Constantin Brâncuși" of Tg-Jiu, ROMANIA*
Bogdan Diaconu, *University "Constantin Brâncuși" of Tg-Jiu, ROMANIA*
Mihai Cruceru, *University "Constantin Brâncuși" of Tg-Jiu, ROMANIA*
Tzvetelin K. Gueorguiev, *University "Angel Kanchev" of Ruse, BULGARIA*

ABSTRACT: It is a fact that engineering properties of the building materials are particularly difficult to model analytically. Given the importance of their values in any application, it is critical to have an estimation of every engineering parameter that is required. This two-part paper will present a dataset containing three engineering properties of some new materials obtained through recycling waste from petroleum industry and from coal-based power. The second part of the paper will present the application of several Machine Learning algorithms to the dataset mentioned above. The performance of each model was assessed and discussed. It was found that Bagging (with a Decision Tree based algorithm) and XGBoost algorithm have the best performance.

Key-Words: - Machine Learning algorithms, Clay, Ash, Compressive strength, Density.

1.INTRODUCTION

Machine Learning algorithms have significant advantages over conventional approaches in predicting properties of cement-based materials [1], compressive strength of self-compacting concrete [2], compressive strength of fly ash-based geopolymer [3], etc. As the algorithms of choice, it is commonly agreed that ensemble methods (especially based on Decision Tree algorithm) perform better than standalone algorithms [4]. Hybrid models were also reported, such as a SVM – Genetic Algorithm [5].

The main Machine Learning algorithms deemed suitable for the dataset considered in this study were described in the first part of this paper. In the second part, the algorithms will be benchmarked on the mechanical properties dataset.

2.PRELIMINARY ANALYSIS OF THE DATASET

The dataset consisted of 207 lines with the following structure:

- Features:
  - Clay percentage
  - Ash percentage
  - Waste drilling fluid percentage (WDF)
  - Temperature (T)
- Targets:
  - Compressive strength (CS)
  - Density (D)
  - Pore density (PD)

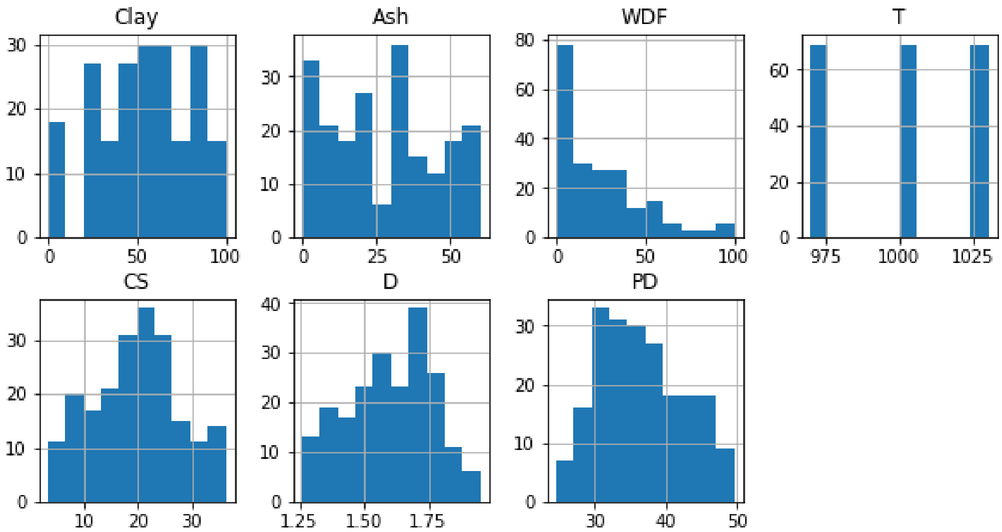

Figure 1. Histograms (ten bins) of the dataset variables.

No missing values or outliers were present in the dataset due to the collecting procedure requirements. The main statistic parameters of the data set are presented in Table 1.

Table 1. Main statistic parameters of each variable in the dataset

	Clay	**Ash**	**WDF**	**T**	**CS**	**D**	**PD**
count	207.000000	207.000000	207.000000	207.000000	207.000000	207.000000	207.000000
mean	51.181159	26.681159	22.137681	1000.000000	19.762126	1.598357	36.640531
std	26.659937	18.081927	23.407101	24.554279	8.150619	0.166635	5.990664
min	0.000000	0.000000	0.000000	970.000000	3.120000	1.260000	24.720000
25%	32.000000	10.000000	0.000000	970.000000	14.220000	1.475000	31.955000
50%	50.000000	25.000000	18.000000	1000.000000	20.050000	1.610000	35.790000
75%	75.000000	40.000000	35.000000	1030.000000	25.315000	1.730000	41.335000

In order to analyse the mutual influences and cross-correlations between various variables in the dataset a cross-correlation matrix has been determined (Figure 2). It can be noticed that the compressive strength has a distribution closed to the normal while density and pore density are somewhat skewed to the left and to the right respectively. Several clear trends can be identified (increasing the clay content results in a higher density and compressive strength and a lower pore density; increasing the ash content results in a higher pore density and in a lower density while the effect on compressive strength is not well defined). The WDF variable deserves a special comment: from Figure 1 it can be noticed that most of the WDF content values were les than 10% (WDF content reduced dramatically the compressive strength, so during the experiments it was found that some specimens did not comply with the minimum requirements to undergo compressive strength, density and pore density measurements). It is therefore difficult to assess the effect of WDF on compressive strength, density and pore density (however a slight trend can be observed, in the sense that increasing the WDF content results in a lower compressive strength and higher pore density). The effect of the temperature cannot be assessed from these plots since only three values were used in the experiments.

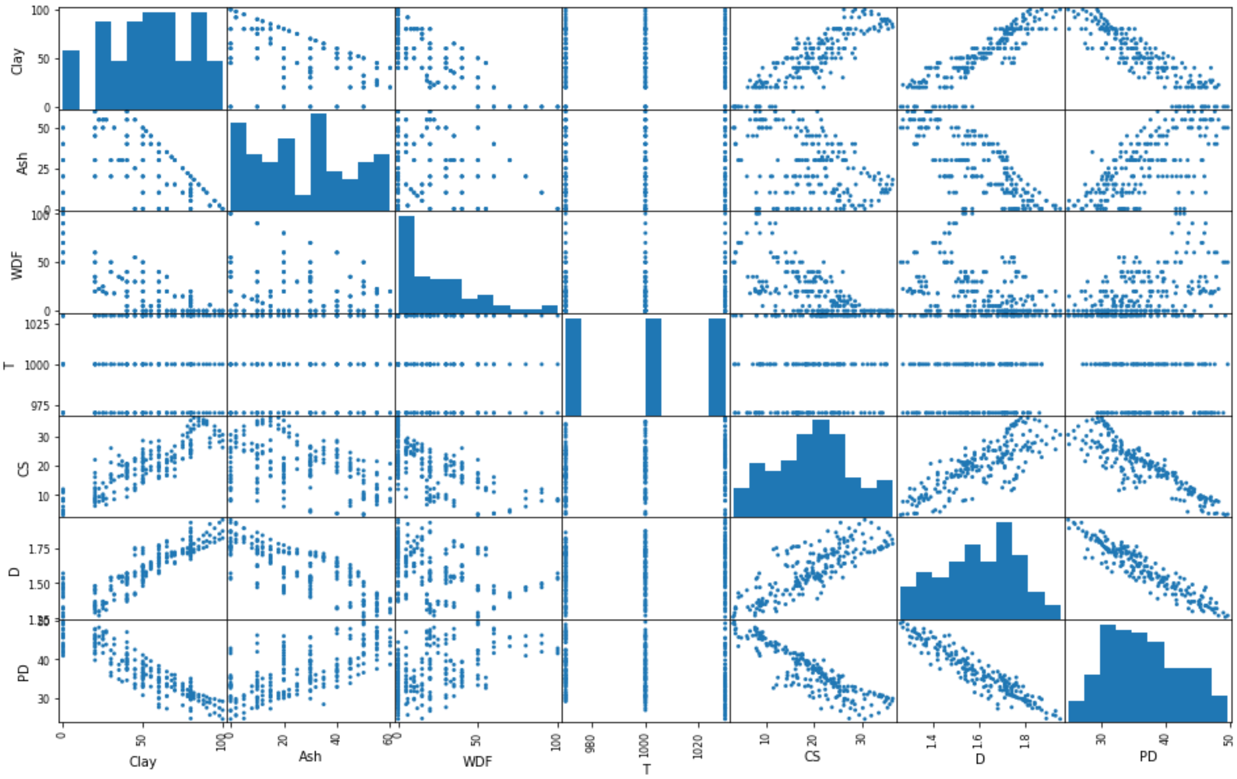

Figure 1. Cross-correlation plots for all variables in the dataset

3.MACHINE LEARNING MODEL EXPERIMENTS

The dataset described in the previous section will be used for training and test with several Machine Learning algorithms implemented in scikit-learn library. The metrics used to compare the results of the algorithms against each other are mean squared error, mean absolute error and mean absolute percentage error. The features values were scaled to [0, 1] as most Machine Learning algorithms work better with scaled values. Since the number of features is already small, it is out of the question to perform feature selection. It is however interesting to establish if the temperature has any effect on the results since it has a very small number of values.

3.1.Support Vector Regressor

The scikit-learn implementation of the Support Vector Regressor requires several important parameters. The first parameter to be discussed is `kernel`, for which the value *linear* has been used in this paper. For the parameter C the default value 1 was used. Several tests were performed to establish the influence of ϵ. It was found that no significant difference occurs when varying ϵ from 0.1 to 1E-4.

The metric values for Support Vector Regressor are presented in Table 2.

Table 2. Support Vector Regressor. Metric values

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape

Compressive Strength	3.229188	17.174875	0.162177	3.529106	20.507019	0.181709
Density	0.043850	0.002582	0.027289	0.048401	0.003106	0.030051
Pore Density	1.280040	2.898557	0.034724	1.955872	5.702066	0.053639

3.2.Decision Tree

Although Decision Tree is a more robust algorithm and it is relatively insensitive to features scale. However, the scaling from previous algorithm was preserved. An important parameter of the Decision Tree algorithm is `max_depth`, which controls the maximum depth of the tree. The metrics values for the `max_depth` parameter value 3 are presented in Table 3.a. However, it was found that the default value for the `max_depth` parameter used by the scikit-learn implementation (*None*) results in significantly better performance, especially for compressive strength. The metrics values for `max_depth = None` are presented in Table 3.b.

Table 3.a. Decision Tree Regressor metrics values for max_depth = 3

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.938462	13.816518	0.184132	2.746735	10.524323	0.179425
Density	0.041697	0.003522	0.026806	0.041697	0.003522	0.026806
Pore Density	1.722098	5.310888	0.049134	1.722098	5.310888	0.049134

Table 3.b. Decision Tree Regressor metrics values for max_depth = None

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.024524	8.692321	0.105466	1.859563	6.469421	0.110321
Density	0.017381	0.000550	0.011189	0.022698	0.000928	0.013847
Pore Density	1.261190	2.758826	0.035792	1.846905	5.731231	0.051811

3.3.Linear Regression

The most basic algorithm, Linear Regression results in the metrics values presented in the Table 4. The results are comparable to those obtained by Support Vector Regressor and considerably worse than those provided by Decision Tree algorithm.

Table 4. Linear Regression metrics values

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.916663	12.271482	0.169075	2.940396	12.726532	0.188403
Density	0.041298	0.002305	0.026351	0.044555	0.002638	0.028270
Pore Density	1.377503	3.445922	0.037910	1.553998	4.827806	0.043454

3.4.Linear Regression with L1 regularization (Lasso)

The parameter that controls the amount of regularization is alpha (higher the value, stronger the regularization). The results for Lasso regression are presented in Table 5.a for the default value of the L1 regularization parameter.

Table 5.a Lasso Regression with the default value of `alpha`=1.0

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	4.093048	26.925335	0.215454	4.093048	26.925335	0.215454
Density	0.138429	0.025205	0.086626	0.138429	0.025205	0.086626
Pore Density	3.770655	20.308864	0.102570	3.770655	20.308864	0.102570

Lasso regression produces the same values of the metrics no matter if temperature is included in the features or not. Reducing the amount of regularization by setting the regularization parameter value 0.1, the metrics will take the values presented in Table 5.b.

Table 5.b Lasso Regression with the value of alpha=0.1

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.849642	12.345342	0.164365	2.949861	13.071669	0.177216
Density	0.138429	0.025205	0.086626	0.138429	0.025205	0.086626
Pore Density	1.435464	3.751327	0.039345	1.622918	5.288863	0.044809

From analyzing the results presented in Tables 5.a and 5.b it can be noticed that by decreasing regularization the effect of temperature starts to be noticed. This means that a strong regularization will remove some of the features, in this case temperature.

3.5.Linear Regression with L2 regularization (Ridge)

Similar to the Lasso regression, Ridge regression controls the amount of regularization through alpha parameter. Unlike Lasso regression, all features are considered no matter what the value of the L2 regularization parameter is but increasing it will reduce the influence of some features. With the default value of the L2 hyperparameter, the metrics values are presented in Table 6.a.

Table 6.a. Ridge Regression with the default value of `alpha`=1.0

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.879031	12.421199	0.165275	2.961354	12.993707	0.180775
Density	0.042258	0.002440	0.027013	0.045111	0.002796	0.028706
Pore Density	1.409173	3.588732	0.038669	1.583756	5.062063	0.043972

In the next test, the amount of L2 regularization will be reduced by setting the L2 regularization hyperparameter to 0.1. The results are presented in Table 6.b. It can be noticed that no significant difference exists between the L2 regularization hyperparameter values 1.0 and 0.1.

Table 6.b. Ridge Regression with the default value of alpha=0.1

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.906201	12.276718	0.167648	2.942531	12.744227	0.187484
Density	0.041372	0.002315	0.026403	0.044596	0.002651	0.028304
Pore Density	1.379930	3.455920	0.037963	1.556642	4.847866	0.043496

3.6.AdaBoost.

The scikit-learn implementation of the AdaBoost algorithm [7] starts by fitting a regressor on the original dataset and then continuing fitting copies of the regressor on the same dataset with weights adjusted according to the error of the current prediction. In such way, subsequent regressors will focus on instances where prediction error is higher.

Table 7. AdaBoost Regression metrics values

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	2.148098	6.930690	0.121905	2.086592	6.552493	0.120948
Density	0.026003	0.000982	0.016385	0.028779	0.001202	0.018035
Pore Density	1.439784	3.196552	0.040819	1.626779	4.672220	0.046310

The default regressor used in the implementation of AdaBoost is DecisionTreeRegressor with the default value of the `max_depth` parameter 3. Experiments with `max_depth=None` showed a small improvement in the values of the metrics.

3.7.Bagging Regressor

Bagging – short for bootstrap aggregating – consists of using the same training algorithm for every predictor but training them on random subsets of the training set. The random sampling of the dataset can be performed with replacement, in which case the method is called bagging or without replacement and the method is called pasting. Bagging Regressor is typically used as a way to reduce the variance of a black-box estimator such as a Decision Tree, by introducing randomization into its construction procedure and then making an ensemble out of it.

The values of the metric obtained by employing a Bagging Regressor are presented in Table 8.a.

Table 8.a. Bagging Regressor metrics values. Out-of-bag evaluation set to **False**

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	1.332946	2.906534	0.083195	1.835834	5.142862	0.113702
Density	0.019450	0.000863	0.012392	0.023978	0.001152	0.014820
Pore Density	1.124883	2.249006	0.032390	1.583572	4.726846	0.044652

Further improvement can be achieved by setting the `oob_score` parameter to True. This forces the predictor to be evaluated on the set of instances that it never sees (details in [6])

Table 8.b. Bagging Regressor metrics values. Out-of-bag evaluation set to **True**

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	1.238918	2.564622	0.078041	1.811776	5.210920	0.111734
Density	0.017988	0.000703	0.011463	0.023203	0.001052	0.014295
Pore Density	1.106171	2.182096	0.031803	1.635455	4.880250	0.046006

The Bagging Regressor with Out-of-bag evaluation set to True produces by far the best results. However, overfitting can be noticed if the metrics for the train set are analyzed, as shown in Table 8.c

Table 8.c. Bagging Regressor metrics for the train set

	Temperature feature included		
Target	Mae	Mse	mape
Compressive Strength	0.664283	0.825856	0.045133
Density	0.009201	0.000145	0.005825
Pore Density	0.562173	0.534737	0.015413

A comparison between predictions and targets for the test set is presented in Figure 3.a.

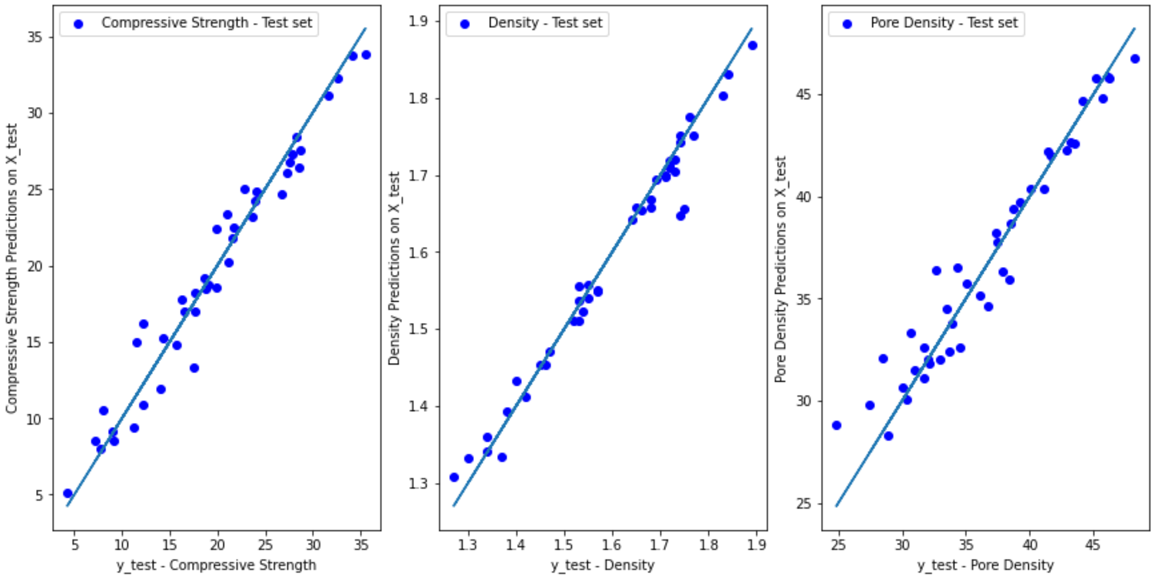

Figure 3.a. Bagging Regressor – comparison between predictions and targets for the test set

In contrast, the results for the train set are presented in Figure 3.b. Comparing Figure 3.a with Figure 3.b it is obvious that the model overfits significantly.

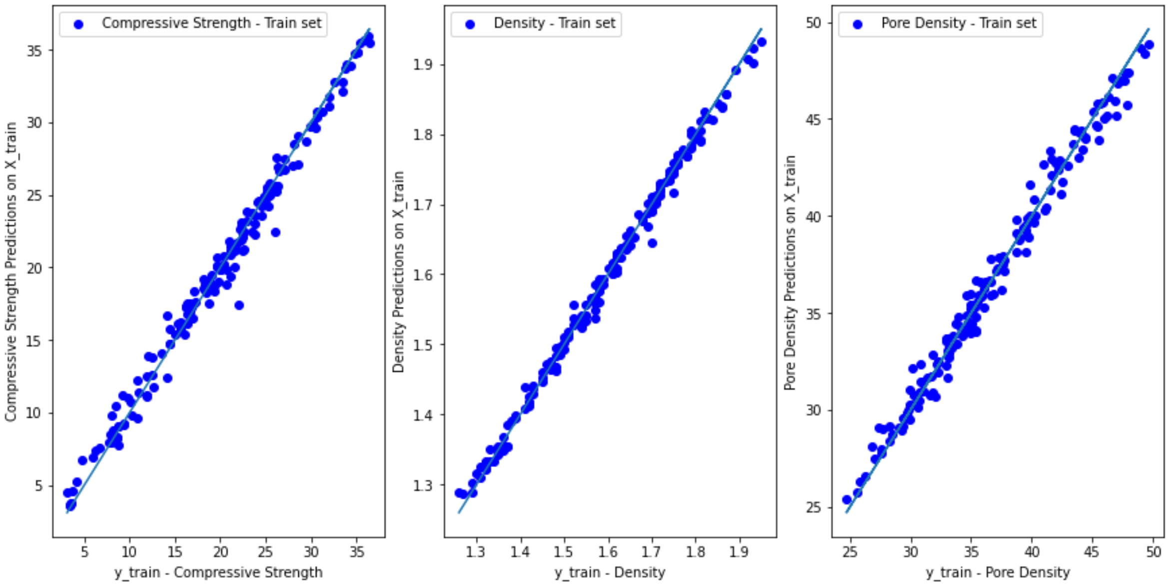

Figure 3.b. Bagging Regressor – comparison between predictions and targets for the train set

3.8.Random Forests Regressor

Table 9. Random Forests Regressor metrics values. Out-of-bag evaluation set to **True**

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	1.601407	4.190459	0.096428	1.827735	5.221197	0.109622
Density	0.021698	0.000875	0.013775	0.024935	0.001105	0.015587
Pore Density	1.296026	3.005651	0.037258	1.572180	4.644227	0.044411

Random Forests family algorithms can measure the relative importance of each feature, as presented in Table 10.

Table 10. Relative importance of the features as determined by the Random Forests Regressor

	Compressive Strength	**Density**	**Pore Density**
Clay	0.825993	0.807865	0.872886
Ash	0.055522	0.170469	0.066074
WDF	0.088666	0.013287	0.014213
Temperature	0.029820	0.008380	0.046826

3.9.Gradient Boosting Regressor

This is another boosting algorithm from the family of Ensemble learning.

Table 11. Gradient Boosting Regressor metrics

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape

Compressive Strength	1.780641	5.696110	0.100393	2.033219	7.401983	0.116788
Density	0.015483	0.000434	0.009975	0.020615	0.000783	0.012539
Pore Density	1.296582	2.923086	0.036860	1.784169	5.671942	0.050087

From Tables 11 and 8 it can be noticed that the boosting algorithm performs almost as good as the bagging algorithm, which demonstrates one more time the efficiency of ensemble methods.

3.10.Voting Regressor

The Voting Regressor takes the arithmetic average between a number of estimators provided as arguments.
In this example, a number of four estimators (Gradient Boosting, Random Forest, Linear Regression and AdaBoosting) will be considered.

Table 12. Voting Regressor

	Temperature feature included			Temperature feature not included		
Target	Mae	Mse	mape	Mae	Mse	mape
Compressive Strength	1.654241	4.280269	0.096342	2.006209	5.458830	0.121971
Density	0.022847	0.000707	0.014479	0.025979	0.001038	0.016098
Pore Density	1.129023	2.229244	0.032356	1.560110	4.381273	0.044062

4.CONCLUSIONS

The common Machine Learning algorithms tested on the dataset produced different results, from worst (Linear Regression) to the best performing algorithm Bagging Regressor, which demonstrates the power of Ensemble Learning methods. However, due to the small volume of the dataset, it is not possible to avoid overfitting, even though the dataset was carefully collected and pre-processed. By using the Random Forest Algorithm, it was possible to quantify the relative influence of the features to the target values. It was found that the Clay content influence dominates (with more than 80%) the other three features for all targets. The temperature influence is the least significant (less than 5%) for all targets. The waste drilling fluid component influences mostly the compressive strength and less density and pore density.
It is expected that a higher volume dataset will produce better results and will reduce overfitting. Another possible way to increase the performance of the models is to add more features to the dataset. Regarding the performance of different algorithms, it can be clearly noticed that Linear Regression performed the worst. L1 and L2 regularization did not improve the standard Linear Regression (this is not surprising as the number of features of this dataset is anyway small). A comparison of the algorithm performance is presented in Figure 4.a (mean absolute error criteria) and Figure 4.b. (mean percentage error criteria).

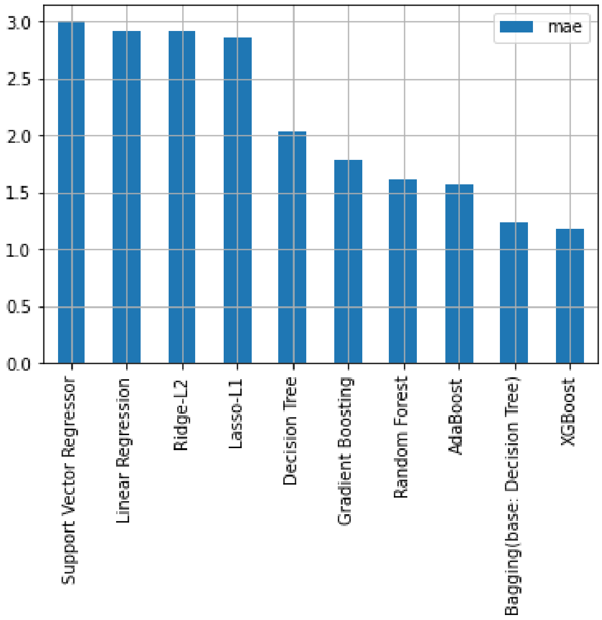

Figure 4.a. Algorithm performance (*mae* criteria)

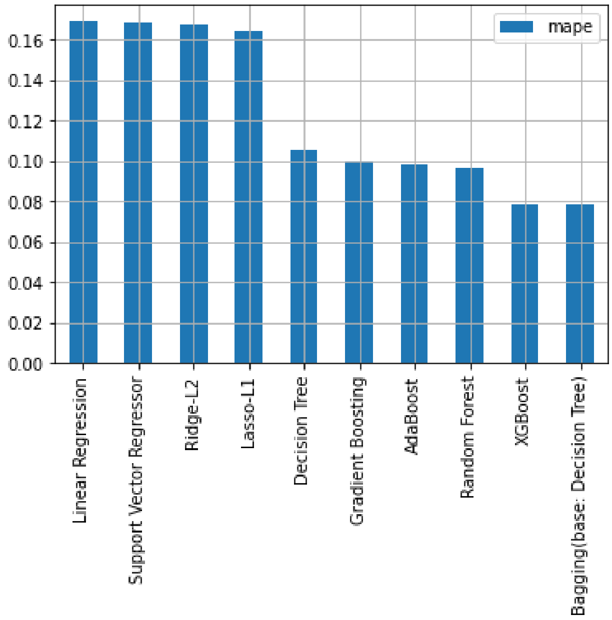

Figure 4b. Algorithm performance (*mape* criteria)

The study presented in this paper attempted to establish if the temperature feature does influence significantly the performance of the model. The dataset contains only three distinct values of the temperature, which, intuitively, is not expected to bring much information. Therefore, the performance metrics were determined in two cases, with temperature included as the fourth feature and without temperature (the feature number was limited in this case to three). The influence of the temperature feature is almost insignificant, which was expected due to the very small number of values. A better approach could have been splitting the dataset into three subsets, one for each value of the temperature. In this case, the advantage or removing a feature is offset by reducing significantly the datasets volume.

REFERENCES

[1] Zhuxin Cheng, Yingchun Yang, Haoyou Zhang, Interpretable ensemble machine-learning models for strength activity index prediction of iron ore tailings, Case Studies in Construction Materials 17 (2022)

[2] Jesús de-Prado-Gil, Covadonga Palencia, Neemias Silva-Monteiro, Rebeca Martínez-García, To predict the compressive strength of self compacting concrete with recycled aggregates utilizing ensemble machine learning models, Case Studies in Construction Materials 16 (2022)

[3] Ayaz Ahmad, Waqas Ahmad, Fahid Aslam, Panuwat Joyklad, Compressive strength prediction of fly ash-based geopolymer concrete via advanced machine learning techniques, Case Studies in Construction Materials 16 (2022)

[4] Qichen Wang, Abasal Hussain, Muhammad Usman Farooqi, Ahmed Farouk Deifalla, Artificial intelligence-based estimation of ultra-high-strength concrete's flexural property, Case Studies in Construction Materials 17 (2022)

[5] Hamdi A. Al-Jamimi, Walid A. Al-Kutti, Saleh Alwahaishi, Khalid Saqer Alotaibi,mPrediction of compressive strength in plain and blended cement concretes using a hybrid artificial intelligence model, Case Studies in Construction Materials 17 (2022)

[6] Aurélien Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly 2019 (2nd Edition)

[7] Yoav Freund, Robert E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences 55, 119_139 (1997)

[8] scikit-learn.org (accessed on October 15, 2022)

[9] Racoceanu Cristinel, REDUCTION OF NITROGEN OXIDE EMISSIONS BY ORGANIZING COMBUSTION, Annals of the" Contantin Brancusi " University of Târgu Jiu, Engineering Series, Nr.3/2021, pag. 51-54, 2537-530X (online), ISSN-L 1842-4856.

[10] Racoceanu Cristinel, REDUCTION OF GREENHOUSE GAS EMISSIONS, Annals of the" Contantin Brancusi " University of Târgu Jiu, Engineering Series, Nr.3/2021, pag.17-20, 2537-530X (online), ISSN-L 1842-4856.