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 I – DISCUSSION ON SUITABILITY OF SEVERAL MACHINE LEARNING ALGORITHMS FOR THE PROBLEM

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: Machine Learning algorithms have demonstrated their ability to model complex processes that are otherwise impossible to model analytically or through standard techniques. However, the amount of data and more importantly, the data quality, are critical factors for obtaining a good performance with such algorithms. This two-part paper will present an experimental dataset and will discuss several commonly used Machine Learning algorithms adequate for such problems. In the first part of the paper an introduction to several algorithms will be given; the inputs of the algorithms and some special constraints and limitations will be introduced.

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

1.INTRODUCTION

Machine Learning algorithms gained popularity during the last decades, mainly due to the unprecedented development of the computing power and secondly, the availability of large amounts of data. Since its first major milestone – Yann Lecunn's deep convolutional network for image recognition – significant progress has been reported. Machine Learning and Deep Learning have since expanded exponentially their application domains. Aurélien Géron [1] coined this phenomenon "The Machine Learning Tsunami".

Standard machine learning supervised algorithms such as Linear Regression (and its more advanced variants Lasso, Ridge, Elastic Net), Support Vector Machine, Decision Tree have been for decades and can be considered a standard in Machine Learning. Their applicability has been generalizes in almost all engineering areas as well as in economics, natural language processing, fraud detection and many other.

The area of construction and building materials made no exception and benefited from the power of Machine Learning algorithms. The most frequent applications of ML algorithms were reported in predicting the compressive strength of various types of concrete (especially concrete incorporating recycled aggregates) [2-5], chloride diffusion [6], optimization of mixture proportions [7], dynamic modulus of asphalt concrete mixtures [8,9], clay (soil) mechanical properties [10,11].

This paper discusses several Machine Learning algorithms considered suitable for modelling properties and static/dynamic behaviour of construction materials and elements. The first part of the paper presents several Machine Learning algorithms and discusses their features, advantages and limitations.

2. THE DATASET DESCRIPTION

The data used in this study was collected in 2018 and consists of basic mechanical properties of a new material, which was investigated in order to assess its suitability as a replacement material for clay. The components of the new material were clay, ash from a coal-fired power plant ash pond and waste drilling fluid. The components were processed (details are provided in [5] and [12]) and mixed in various concentrations. Then test samples were fabricated from the mixture (details of the procedure are presented in [12]) and burnt at three values of the temperature. The mechanical properties of the test samples determined in the studies presented in [5] and [12] were compressive strength, density and open pore density. The dataset structure consisted of four features (clay, ash, waste drilling fluid percentages and thermal processing temperature) and three targets (compressive strength, density and open pore density). The dataset consisted of a number of 200 entries.

0. Machine Learning libraries

Libraries, in the sense of computer science, are collections of non-volatile resources used by computer programs in software development. They consist of configuration data, documentation, help files, templates, code, subroutines, classes, values and type specifications. In Machine Learning, a programming language with a high prevalence is Python, which is a high-level, general-purpose programing language, developed in the late 1980. A well-known Machine Learning library for Python programming language is scikit-learn, featuring capabilities such as classification, regression and clustering. Scikit-learn has been design for compatibility with other scientific libraries such as NumPy (offers support for large, multi-dimensional arrays and matrices as well as a large collection of high-level mathematical functions) and SciPy.

Machine Learning algorithms suitable for the problem.

1. Linear Models.

Let X a set of features consisting of m vectors $X_m = (x_1, \dots, x_n)_m$, for each vector in the set a target (observed) value is denoted y_m Linear models estimate the target value – denoted $\hat{y}$ considering it as a linear combination of the features:

$$\hat{y}_m(X_m) = \beta_{m0} + \beta_{m1}x_{m1} + \cdots \beta_{mn}x_{mn}$$

The vector $(\beta_1, \dots, \beta_n)$ is known as coefficient vector and the value β_0 is known as intercept.

The most elementary linear model is the Linear Regression algorithm, which fits a linear model with the coefficient vector $(\beta_1, \dots, \beta_n)$ and intercept β_0 using as cost function the sum of squares between the targets (observed values) and the estimated values of the targets $\hat{y}$ is minimized:

$$min(\| \hat{y} - y\|_m^2)$$

It can be easily proved that the solution to the above minimization problem is given by:

$$\beta = (X^T X)^{-1} X^T y$$

It is important that feature vector values are independent. In case correlations exists between features, the feature matrix becomes close to singular and the least-squares estimate becomes very sensitive to random errors in the observed value of the target eventually resulting in a large variance.
The complexity of the least squares solution is given by $O(m, n^2)$.

1.1.Ridge regression

Ridge regression is a form of regularization (L2 regularization) which is used in cases where multicollinearity is present in the data. In such cases, large variance results and predicted values deviate significantly from targets. In case of ridge regression, the cost function is similar to the one used for plain linear regression with a penalty term which limits the values of the coefficient vector:

$$min(\| \hat{y} - y\|_m^2 + \lambda\|\beta\|_m^2)$$

Ridge regression reduces the complexity of the model but does not reduce the number of features considered important for the model since it does not result in any coefficient being zero.

1.2. Lasso (Least Absolute Shrinkage and Selection Operator) regression is another form of regularization (L1 regularization). In contrast to Ridge regression, Lasso regression reduces the number of features upon which the solution is dependent. It is used in cases where redundant features need to be removed or in cases where features that do not influence significantly the target exist and need to be removed. Lasso regression results in simple, sparse models (the coefficient vector will have some zero values, that is not all features will be considered to be contributing to the target value). The cost function in the case of Lasso regression is given by:

$$min(\| \hat{y} - y\|_m^2 + \lambda\|\beta\|_m)$$

The L1 regularization parameter λ controls the amount of shrinkage (regularization). If $\lambda = 0$ then no L1 regularization is applied and the cost function is equivalent to the one used for simple linear regression. Increasing λ results in a larger bias while decreasing it will increase the variance.

2. Decision Trees

Decision Trees are Machine Learning algorithms that can solve both regression and classification problems. Decision Tree is a supervised machine learning algorithm using a tree-like model: in the sense of computer science, a tree is a hierarchical data structure with a set of connected nodes. Each node on a level can be connected to any number of nodes (children nodes) on the inferior level but it must be connected to exactly (no more and no less) than a node (parent node) on the superior level. Thus, the uppermost node, called root node, has no parent node.

Decision Trees do not require many assumptions on the training data. If no regularization is applied, the algorithm will adapt to the training data, fitting it very closely and most of the times, overfitting it. Decision Tree algorithms belong to the nonparametric model class. A nonparametric model does not have an imposed number of parameters (predetermined before training the data). This makes the model structure flexible and being able to replicate the data closely. Parametric models have a predetermined number of parameters, which limits the number of degrees of freedom. This reduces the risk of overfitting (with an increased risk of overfitting) and can also increase the bias.

In fact, overfitting is a major issue when performing regression with Decision Trees.

The Scikit-Learn implementation of the Decision Tree applies regularization by restricting the maximum depth of the tree. The hyperparameter `max_depth` (with the default value None, which means no depth limit) regularizes the model reducing the risk of overfitting.

3. XGBoost

XGBoost (eXtreme Gradient Boosting) is an open-source powerful supervised learning algorithm suitable for both regression and classification. It uses sequentially-built small-depth decision trees to deliver accurate predictions. XGBoost supports training on distributed cloud platforms such as AWS, Azure or GCP. While the mathematical background is beyond the scope of this paper, a few remarks on the XGBoost parameters are necessary. Before running XGBoost, three types of parameters must be set: general parameters, booster parameters and task parameters:

General parameters relate to which booster will be used: default value is gbtree. Other values are gblinear (uses linear functions) and dart (tree-bases models).

Booster parameters depend on which booster has been selected in the General parameters section. Several important booster parameters are the following:

`eta`: the step size shrinkage used in update to prevents overfitting.

`gamma`: Minimum loss reduction required to make a further partition on a leaf node of the

tree. The larger the value, the more conservative the algorithm will be.

`max_depth`: Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 indicates no limit on depth. It has to be mentioned that XGBoost aggressively consumes memory when training a deep tree.

`lambda`: L2 regularization term on weights. Increasing this value will make model more conservative.

`alpha`: L1 regularization term on weights. Increasing this value will make model more conservative.

Learning task parameters decide on the learning scenario. For example, regression tasks may use different parameters with ranking tasks.

4. Support Vector Machines

The Support Vector Machine (SVM) is a versatile machine learning algorithm, used traditionally for classification of complex but small or medium-sized datasets (either linear or non-linear classification). Being a flexible algorithm, it can be easily adapted for regression or even for outlier detection.

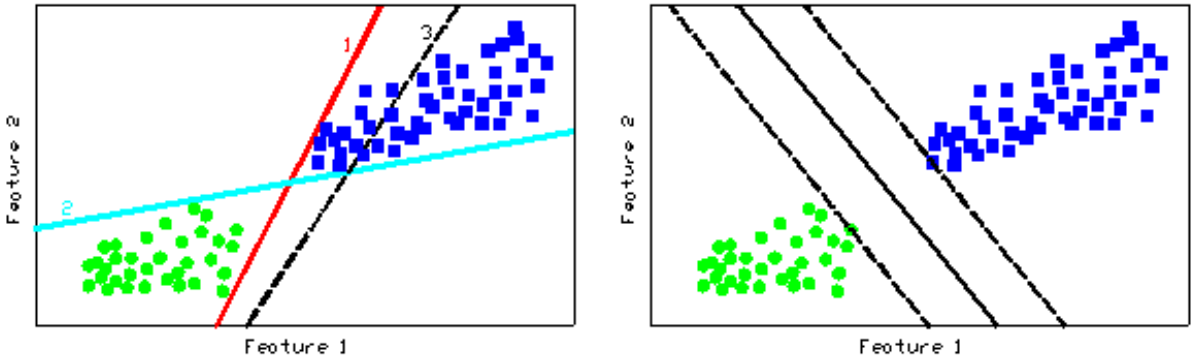

Figure 1. Left: Decision boundaries that separate completely the two classes (1 and 2) or fail to separate them correctly (3); Right: Large margin classification

The principle of the SVM algorithm can be illustrated as shown in Figure 1. Here, a two-class classification problem is presented with a two-element feature vector. In the left plot, the two classes can be clearly separated by either line 1 or line 2. However, the decision boundaries set by either line 1 or 2 are so close to some training set class instances that it is very likely that next instances (not included in the training set) will be misclassified, as shown in Figure 2 (left plot). The solid line on the right-hand plot (Figure 1) is different from lines 1 and 2 (left-hand plot) in that it maximizes the distances from the closest instances of both classes. It is called large margin classification.

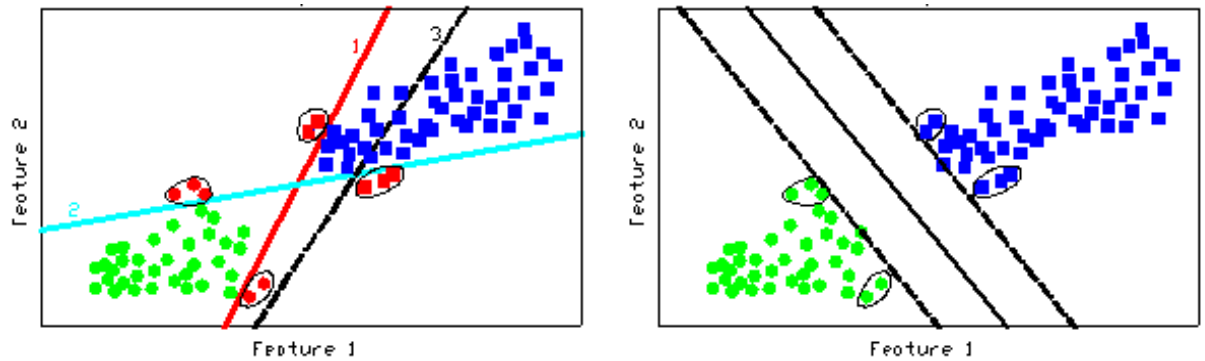

Figure 2. Left: new instances are misclassified by decision boundaries 1 and 2. Right: large margin classification correctly classifies the new instances

Large margin classification avoids misclassification that occurs for new instances in case of decision boundary lines 1 and 2 (Figure 2, left-hand plot).

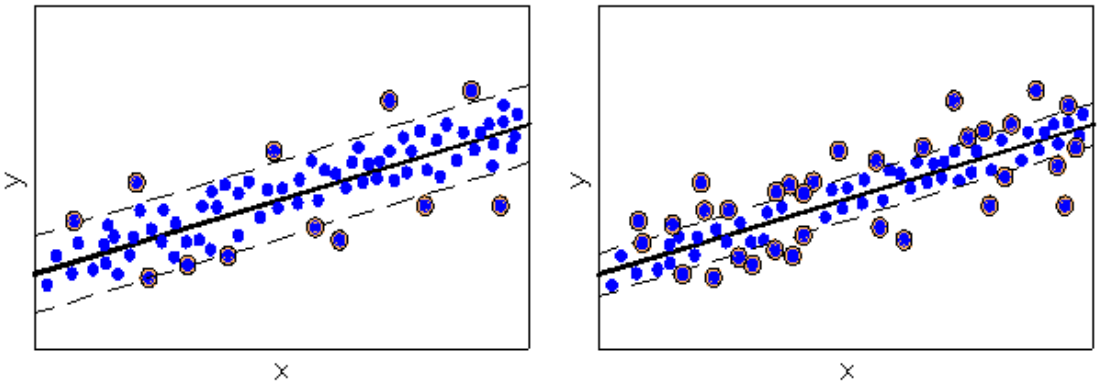

Figure 3. SVM Regression and the effect of hyperparameter ε

SVM Regression. Instead of fitting the larges possible interval between the two classes while avoiding or reducing the misclassification events, SVM Regression fits as many as possible instances inside the interval while limiting margin violations (instances off the interval). The idea of the SVM Regression is illustrated in Figure 3. The width of the interval is controlled by the hyperparameter ε.

5.Ensemble Learning

Ensemble Learning is a very powerful and important concept in Machine Learning. Simply put, an Ensemble Learning algorithm aggregates the predictions from several standalone models. A typical example of Ensemble Learning is the Random Forest algorithm. It consists of training a group of Decision Tree algorithms, each on a different random subset of the training set. In case of a classification problem, the final prediction is the one that accumulates the most votes.

The most widely used Ensemble methods are *bagging, boosting and stacking*. Of interest to this paper is boosting, which will be described in the next section.

Any Ensemble method that combines several weak algorithms into a strong one is a boosting method. The general idea of boosting methods is to train predictors sequentially, each trying to correct its predecessor. The most well-known boosting algorithm is AdaBoost (short for Adaptive Boosting).

AdaBoost algorithm.

The process of creating an AdaBoost regressor can be described as follows: a base regressor (such as Linear Regressor or Decision Tree) is trained and used to make predictions on the training set. Then a set of weights is assigned to the instances in such way that higher the residual error value the higher the weight value. A second classifier takes over and is trained using the weights assigned in the previous step. The weight values are again updated and the process continues. A major drawback of this technique is the fact that it cannot be parallelized since each predictor can only be trained after the previous predictor has been trained and evaluated.

Gradient Boosting.

Gradient Boosting algorithm works in some way similar to AdaBoost adding sequentially predictors to an ensemble, each one correcting its predecessor. The difference from AdaBoost is the fact that instead of tweaking the instance weights at every iteration, Gradient Boosting tries to fit the new predictor to the residual error made by the previous predictor.

3.CONCLUSIONS

Engineering properties of the construction materials are difficult if not impossible to model by means of analytic methods, making it a field where Machine Learning algorithms could demonstrate their power and efficiency. The main issue that has to be thoroughly considered before employing a Machine Learning algorithm is the data quality.

The dataset considered in this paper does not contain outliers or missing values as the authors attempted to obtain a clean dataset before proceeding with the modelling. For the relatively small dataset considered in this study, neural networks and deep learning are not the best choice; it remains to be established in the second part of this paper how well fundamental Machine Learning algorithms can perform.

REFERENCES

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

[2] Jinrui Zhang, Wenjun Niu, Youzhi Yang, Dongshuai Hou, Biqin Dong, Machine learning prediction models for compressive strength of calcined sludge-cement composites, Construction and Building Materials, vol. 346, September 2022

[3] Hong Li et al, A BFRC compressive strength prediction method via kernel extreme learning machine-genetic algorithm, Construction and Building Materials, vol. 344, August 2022

[4] Van Quan Tran, Viet Quoc Dang, Lanh Si Ho, Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach, Construction and Building Materials, vol. 323, March 2022

[5] L. Anghelescu, M. Cruceru, B.M. Diaconu, Building materials obtained by recycling coal ash and waste drilling fluid and characterization of engineering properties by means of Artificial Neural Networks, Construction and Building Materials, vol. 227, December 2019

[6] Van Quan Tran, Machine learning approach for investigating chloride diffusion coefficient of concrete containing supplementary cementitious materials, Construction and Building Materials, vol. 328, April 2022

[7] Junfei Zhang, Yimiao Huang, Yuhang Wang, Guowei Ma, Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms, Construction and Building Materials, vol. 253, August 2020

[8] Yasir Ali, Fizza Hussain, Muhammad Irfan, Abdul Salam Buller, An eXtreme Gradient Boosting model for predicting dynamic modulus of asphalt concrete mixtures, Construction and Building Materials, vol. 295, August 2021

[9] Thanh-Hai Le et al, Artificial Intelligence-Based Model for the Prediction of Dynamic Modulus of Stone Mastic Asphalt, Appl. Sci. 2020 10(15)

[10] Pin Zhang, Zhen-Yu Yin, Yin-Fu Jin, Tommy H.T.Chan, Fu-Ping Gao, Intelligent modelling of clay compressibility using hybrid meta-heuristic and machine learning algorithms, Geoscience Frontiers, vol 12, Issue 3, January 2021

[11] E.U.Eyo, S.J.Abbey, T.T.Lawrence, F.K.Tette, Improved prediction of clay soil expansion using machine learning algorithms and meta-heuristic dichotomous ensemble classifiers, Geoscience Frontiers, vol 13, Issue 1, January 2022

[12] L. Anghelescu, PhD thesis, University of Petrosani, Romania, 2015

[13] Racoceanu., C. Study concerning the application of the nonpolluting ash exhaust technology to the Benson steam generators, Annals of the "Constantin Brâncuși" University of Târgu-Jiu, Engineering Series, 4 (2019), 100-104 [ISSN: 1842-4856].

[14] Racoceanu Cristinel, Study on influence of energy efficiency of a steam boiler benson on environmental pollution, Fiability&Durability, no.2/2016, ISSN 1844-640X.