Project on creating a classifier of lithological types for uranium deposits in Kazakhstan

Yan Kuchin\textsuperscript{1,2}, Kirill Yakunin\textsuperscript{1,2}, Elena Mukhamedyeva\textsuperscript{1} and Ravil Mukhamedyev\textsuperscript{1,2}

\textsuperscript{1} Satbayev University, 22a Satpaev str., 050013, Almaty, The Republic of Kazakhstan
\textsuperscript{2} Institute of Informational and Computational Technologies, 125 Pushkin str., Almaty, The Republic of Kazakhstan

E-mail: ykuchin@mail.ru, yakunin.k@mail.ru, muhamedijeva@gmail.com, ravil.muhamedyev@gmail.com

Abstract. A program that provides the task of classifying electric log data using machine learning methods is described. The program, after the training phase, allows one to automatically determine the composition of the rocks along the wellbore, which is necessary to ensure the technological process of uranium mining. The applied methods of data preprocessing, methods of forming a “floating window” of data and some results are briefly described. When using multilayer perceptron, the obtained average values of precision, recall, f\textsubscript{1}-score with recognition of 7 rock classes are 49\%. At the same time, the developed program allows one to apply a wide range of classification algorithms, including Nearest-Neighbor (k-NN), Logistic regression, Decision Tree, Support Vector Classifier, artificial neural network, boosting, LSTM, etc. For example, when using XGBoost on same data with a change in the size of the floating window and the weight of individual classes (rocks), the indicated accuracy metric was up to 54\%. The results of a comparative analysis of the mentioned methods on an extended data set will be presented in the next article of the authors.

1. Introduction

Over the past seven years, uranium production in Kazakhstan has increased by almost 3.5 times. Uranium mining in Kazakhstan’s deposits is carried out by the method of in-situ leaching, which is one of the least expensive, environmentally friendly methods of production\cite{1}. The production process consists in pumping a weak acid solution into the ore-bearing horizon, which then, enriched in the product, is pumped out through a network of wells. A necessary element of the process is the installation of filters in boreholes. The desired position of filter is that, which ensures the spread of acid through permeable rock (sands, gravel) limited to the top and bottom by impermeable rocks (clays). In addition to a reasonable choice of the filter installation position, knowledge of the lithological structure allows one to calculate the dynamics of leaching, to predict the consumption of reagents and the volume of production.

To obtain data on the lithological structure of the field, electric logging methods are used: induction logging (IL), apparent resistivity logging (AR) and spontaneous polarization logging (SP)\cite{2}. In the process of logging, a probe is lowered into the drilled borehole, which, when lifted, allows one to
obtain the listed data, based on which the expert draws a conclusion about the depth and properties of the rocks.

As a result, the expert determines the lithological type code (usually from 1 to 9 for production wells, a more detailed coding is accepted for exploratory wells), which determines their filtration properties. Having this classification, you can begin to determine the optimal installation location of filters for pumping acid and pumping out the product solution (the position of the filter also depends on the depth of the ore body, and the type of well).

The economic losses from the incorrect interpretation of the logging data in the fields of Kazakhstan can be estimated at from 1 to 4 million dollars per year [3]. Therefore, an expert assistance system is needed that could automatically conduct lithological classification of rocks based on logging data. Note that this problem does not yet have strictly substantiated formal solutions. The present work is devoted to the description of an example of solving this problem using machine learning (ML) methods.

The method of solution consists in the fact that having a dataset of electric logs interpreted by experts, a classifier is trained, which allows one to interpret the data automatically using the log data.

The work consists of the following sections
- The second section provides a brief description of the method used.
- The third section describes the initial data.
- The fourth section presents the results, including a brief description of the developed program.
- In conclusion, we discuss the results obtained and formulate the tasks of further research.

2. Methods

In cases where there are no strict formal methods for solving the problem, the ML methods are usually used. ML is successfully used to solve problems in medicine [4, 5], biology [6], robotics, urban economy [7] and industry [8], environmental modeling [9], when creating a new type of communication system [10], in astronomy [11] etc. In particular, ML is used in solving petrography tasks [12, 13].

ML methods are divided into two broad groups [14–16]:
- Unsupervised Learning (UL) [17].
- Supervised Learning (SL) [18].

UL solve the problem of clustering, when many previously undefined objects are divided into groups by an automatic procedure based on the properties of these objects [19, 20]. SL solve the problem of classification or regression. The classification problem arises when finite groups of objects in some way designated are formed in a potentially infinite set of objects. Usually the formation of these groups is carried out by experts. The classification algorithm should, using this initial classification as a model, assign the following unassigned objects to a particular group based on the properties of these objects.

Since we have data labeled by experts, it is reasonable to use classification algorithms to solve our problem, for example, k-Nearest-Neighbor (k-NN) [21–23], Logistic regression, Decision Tree (DT), Support Vector Classifier (SVM) [24], artificial neural networks (ANN) [25–27], compositions of algorithms, for example, boosting methods [28], deep learning networks, for example, Long short-term memory (LSTM) [29], etc. Among the many ANNs, we have chosen the “classic” feed forward neural networks in the form of a multi-layer perceptron (MLP).

MLP is one of the most popular classifiers especially for the case of multiple classes. To adjust the weights $\Theta$ of a neural network (network training), a cost function is used that resembles the cost function of logistic regression.

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_{ik}(i) \log h_{\Theta}(x(i))_k + (1 - y_{ik}(i)) \log (1 - h_{\Theta}(x(i))_k) + \frac{\lambda}{2m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sum_{j=1}^{n} (\Theta_{ij})^2,$$ (1)
where $L$ is the number of layers of the neural network; $s_l$ – the number of neurons in the layer $l$; $K$ is the number of classes (equal to the number of neurons in the output layer); $\Theta$ – weights matrix.

To minimize the error function of a multilayer ANN (learning), the Back Propagation Error (BPE) algorithm\cite{30} and its modifications aimed at accelerating the learning process Conjugate gradient\cite{31}, BFGS, L-BFGS\cite{32} are used.

3. Initial data
The initial data is presented in the form of two sets of tables. "Synthetic data" Ideal_data is data simulated based on a given lithology. They are needed to assess the upper limit of the classification quality\cite{3}. In turn, Real_data set is genuine log data interpreted by experts. The results obtained using the Real_data set are given below.

In the tables, the columns correspond to the depth, values of AR, SP and rock code, respectively (table 1). Each table corresponds to one of 37 wells. The main rock codes are as follows: 1-gravel; 2-coarse sand; 3-sand medium-grained; 4-fine-grained sand; 5-sandstone; 6-silt; 7-clay.

| Depth | AR   | SP   | Code |
|-------|------|------|------|
| 365.8 | 10.12| 11.60| 3    |
| 365.9 | 9.96 | 10.40| 3    |
| ...   |      |      |      |
| 366.5 | 11.29| 12.00| 3    |
| 366.6 | 11.35| 12.70| 12   |
| ...   |      |      |      |
| 408.3 | 12.71| 8.00 | 4    |
| 408.4 | 13.56| 8.70 | 112  |
| 408.5 | 14.40| 9.90 | 112  |

For example, the first line of table 1 (file 1.txt), means that at a depth of 365.8 meters the apparent resistivity (AR) is 10.12 Ohms per meter, the potential for spontaneous polarization (SP) is 11.6 millivolts. The expert determined that the rock at this depth is a medium-grained sand (code-3).

At a depth of 366.6 meters, the expert identified a rock consisting of a mixture of gravel and coarse sand, where gravel is the main, and at a depth of 408.4 and 408.5 meters gravel with a slight interspersed coarse sand.

Note that in the developed classifier only the first digit of the code is used (for example, 112 will be considered as 1), rocks with code 9 (coalified rocks) are considered as clays (7) due to the fact that there are very few such rocks in the considered data set.

4. Results
To work with the data described above, the GTSTxtReader program, which provides data loading, preprocessing, including the formation of "floating windows" of data, normalization and processing using a feed-forward neural network was developed.

The use of "floating data windows" is a common technique for analyzing data sequences, for example, time series or, as in our case, the dependencies of logging indices on depth. Since the expert, when evaluating the data, takes into account the form of the logging curve, it is natural to submit data to the input by a sliding window with a depth of $n + 1 + n$ points in depth. That is, $n$ measurements above, the current value and $n$ measurements are below the current value. The next window is formed in a similar way, shifting one unit below (Figure 1).
Since the length of the logging tool is 1 meter, which corresponds to 10 measurements in depth, it is natural to set $n = 5$. Supplying data in the form of a floating window allows to take into account the nature of the change in the form or the curve, not just the recorded value at a specific depth. Size of the window can be varied within certain limits, allowing for the consideration of longer or shorter parts of the logging curve. A full description of the program is given in Appendix 1 (https://drive.google.com/open?id=1a715UMosCZfRECY2enzu-j8CrpCMVym). The appendix contains the program code for pre-processing, training and classification of rocks using MLP. The accuracy of the test data is 48.84%. The metrics 'loss', 'acc', 'precision', 'recall' are respectively 1.2184, 0.4884, 0.5131, and 0.38769. A detailed report demonstrates the following results (Table 2.)

5. Conclusions
The obtained results show that the program was not able to determine sandstone (code 5), which is generally predictable, since the number of values of this code is only 17 out of more than 11 thousand values. The indicators of the harmonic measure $f_1$ for only two rock types take acceptable values (codes 1 and 3) of greater than 0.5. At the same time, for impermeable clay (code 7), an error-free determination in the process of interpretation is important, therefore, its recall value should be more than 0.5, however, the program shows only 0.32. Further experiments with resizing the floating data window provided an increase in these indicators for MLP to 52% (accuracy) and 56% (precision for code 7). Similarly, the application of boosting algorithms (XGBoost) [33] yielded 54% and 58%, respectively. Obviously, without changing the methods of data preprocessing, one can apply a wide range of classification algorithms, including Nearest-Neighbor (k-NN), Logistic regression, Decision Tree, Support Vector Classifier, etc. Comparative analysis of qualitative indicators when applying various ML methods is one of tasks for further research. The application of deep learning, for example, LSTM, also seems promising. However, preliminary experiments showed that the available

![Figure 1. Floating data window](image-url)

| confusion matrix | classification report |
|------------------|-----------------------|
| [[1911 925 122 0 113 54] | precision 1.0 recall 0.64 f1-score 0.61 support 3125 |
| [732 2118 808 0 197 74] | 3.0 0.47 0.54 0.51 3929 |
| [164 695 364 0 121 33] | 4.0 0.25 0.26 0.26 1377 |
| [ 3 7 0 0 1 6] | 5.0 0.00 0.00 0.00 17 |
| [ 60 342 125 0 543 110] | 6.0 0.38 0.46 0.42 1180 |
| [ 95 372 34 0 460 461] | 7.0 0.62 0.32 0.43 1422 |
| | micro avg 0.49 0.49 0.49 11050 |
| | macro avg 0.40 0.37 0.37 11050 |
| | weighted avg 0.50 0.49 0.49 11050 |
dataset is not enough for the successful application of deep learning algorithms. Therefore, expanding datasets is another important concern.

Acknowledgments
This work was funded by grant # BR05236447 of the Ministry of Education and Science of the Republic of Kazakhstan.

References
[1] Yashin S A 2008 Podzemnoye skvazhinnoe vyshchelachivaniye urana na mestorozhdeniyakh Kazakhstana Gornyy zhurnal 3 pp 45–49
[2] Mukhamediev R I and Kuchin Y I 2015 Sredstva avtomatizatsii obrabotki dannykh geofizicheskogo issledovaniya skvazhin na mestorozhdeniyakh urana plastovo-infil'tratsionnogo tipa Cloud of science 2 (3)
[3] Kuchin Y I, Mukhamediev R I and Yakunin K O 2019 Kachestvo klassifikatsii dannykh v usloviyakh protivorechivosti ekspertnykh otsenok Cloud of science 6 (1) pp 109–126
[4] Joseph A Cruz and David S Wishart 2006 Applications of Machine Learning in Cancer Prediction and Prognosis Cancer Informatics 2 pp 59–77
[5] Miotto R et al 2017 Deep learning for healthcare: review, opportunities and challenges Briefings in bioinformatics 19 (6) pp 1236–1246
[6] Ballester Pedro J and John BO Mitchell 2010 A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking Bioinformatics 26 9 pp 1169–1175
[7] Mahdavinejad, Mohammad Saeid, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi and Amit P Sheth 2018 Machine learning for Internet of Things data analysis: A survey Digital Communications and Networks 4 (3) pp 161–175
[8] Farrar Charles R and Keith Worden 2012 Structural health monitoring: a machine learning perspective (John Wiley & Sons)
[9] Frederich Recknagel 2001 Application Of machine Learning To Ecological Modelling Ecological Modelling 146 pp 303–310
[10] Clancy Charles, Joe Hecker, Erich Stuntebeck and Tim O Shea 2007 Applications of machine learning to cognitive radio networks IEEE Wireless Communications 14 (4) pp 47–52
[11] Ball Nicholas M and Robert J Brunner 2010 Data mining and machine learning in astronomy Journal of Modern Physics 19 (07) pp 1049–1106
[12] Mukhamediev R, Amirgaliev E, Iskakov S, Kuchin Y and Mukhamedyeva E 2014 Integration of Results of Recognition Algorithms at the Uranium Deposits Journal of ACIII 18 (3) pp 347–352
[13] Amirgaliev Ye N, Iskakov S Kh, Kuchin Y I and Mukhamediyev R I 2013 Machine learning methods in rock recognition problems in uranium deposits Izvestiya NAN RK 3 pp 82–88
[14] Taiwo Oladipupo Ayodele 2010 Types of Machine Learning Algorithms New Advances in Machine Learning pp 19–48
[15] Hamza Awad, Hamza Ibrahim et al 2012 Taxonomy of Machine Learning Algorithms to classify realtime Interactive applications International Journal of Computer Networks and Wireless Communications 2 (1) pp 69–73
[16] Mukhamediev R 2015 Machine learning methods: An overview CMNT 19(6) pp 14–29
[17] Hastie T, Tibshirani R and Friedman J 2009 Unsupervised learning (Springer New York) pp 485–585
[18] Kotsiantis Sotiiris B, Zaharakis I and Pintelas P 2007 Supervised machine learning: A review of classification techniques Emerging artificial Intelligence Applications in Computer Engineering (IOS Press) pp 3–24
[19] Jain A K, Murty M N and Flynn P J 1999 Data clustering: a review ACM computing surveys (CSUR) 31(3) pp 264–323
[20] Wesam Ashour Barbakh, Ying Wu and Colin Fyfe 2009 Review of Clustering Algorithms. Non-Standard Parameter Adaptation for Exploratory Data Analysis Studies in Computational Intelligence 249 pp 7–28

[21] Altman N S 1992 An introduction to kernel and nearest-neighbor nonparametric regression The American Statistician 46 (3) pp 175–185

[22] Dudani Sahibsingh A 1976 The Distance-Weighted $k$-Nearest-Neighbor Rule Systems, Man and Cybernetics SMC-6 (4) pp 325–327

[23] K-nearest neighbor algorithm [Online] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm: 5.07.2012.

[24] Cortes C and Vapnik V 1995 Support-vector networks Machine learning 20 (3) pp 273–297

[25] Guoqiang Peter Zhang 2000 Neural Networks for Classification: A Survey, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS — PART C: APPLICATIONS AND REVIEWS 30 (4)

[26] FJODOR VAN VEEN, SEPTEMBER 14, 2016, THE NEURAL NETWORK ZOO [Online] http://www.asimovinstitute.org/neural-network-zoo/

[27] Schmidhuber Jürgen 2015 Deep learning in neural networks: An overview Neural networks 61 pp 85–117

[28] Friedman Jerome H 2001 Greedy function approximation: a gradient boosting machine Annals of statistics pp 1189–1232

[29] Hochreiter Sepp and Jürgen Schmidhuber 1997 Long short-term memory Neural computation 9 (8) pp 1735–1780

[30] Werbos P 1974 Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (Harvard University)

[31] Martin Fodslette Møller 1993 A scaled conjugate gradient algorithm for fast supervised learning Neural Networks 6 (4) pp 525–533

[32] Dong C Liu and Jorge Nocedal 1989 On the limited memory BFGS method for large scale optimization Mathematical Programming 45 (1-3) pp 503–528

[33] About XGBoost [Online] https://xgboost.ai/about