Supplementary Text

In this study, we provided seven machine learning algorithms: linear regression, principal component analysis (PCA)-based polynomial regression with Elastic Net cost function, artificial neural network regression, decision trees regression, random forest regression, extremely randomized trees (Extra-Trees) regression, and artificial neural network ensemble regression. For all algorithms in this study, predicted $Fe^{3+}/\Sigma Fe$ values below 0 or above 1 were regarded as 0 or 1, respectively, because such out-of-range predicted values may result from uncertainty in the models. Detailed descriptions of the principles of each algorithm are provided here.

1. Linear regression

The most basic ML algorithm, linear regression assumes a linear relationship between the input features and the corresponding output value of the form:

$$\widehat{\mathbf{Y}} = h_{\boldsymbol{\beta}}(\mathbf{X}) = \boldsymbol{\beta} \cdot \mathbf{X} \tag{S1}$$

where $\widehat{\mathbf{Y}}$ is the predicted value. $\boldsymbol{\beta}$ is the model's parameter vector, including the intercept term β_0 and the feature weights β_1 to β_n. $\mathbf{X}$ is the instance's feature vector (X_0 to X_n, with X_0 always equal to 1). $h_{\boldsymbol{\beta}}$ is the hypothesis function, using the model parameters $\boldsymbol{\beta}$, and $\boldsymbol{\beta} \cdot \mathbf{X}$ is the dot product of the vectors $\boldsymbol{\beta}$ and $\mathbf{X}$ (Géron 2017). The formula used in this study is expressed as

$$\widehat{\mathbf{Y}} = \beta_1 * \mathrm{Si} + \beta_2 * \mathrm{Ti} + \beta_3 * \mathrm{Al} + \beta_4 * \mathrm{Cr} + \beta_5 * \mathrm{Fe} + \beta_6 * \mathrm{Mn} + \beta_7 * \mathrm{Mg} + \beta_8 * \mathrm{Ca} + \beta_9 * \mathrm{Na} + \beta_0 \tag{S2}$$

where each input feature is a molar concentration. Training this algorithm consists of setting its weights ($\beta_1 - \beta_9$) and intercept (β_0) to fit the training set, minimizing the loss function (generalized least squares). After obtaining the weights and intercept, we predict $\widehat{\mathbf{Y}}$ from $\mathbf{X}$ using Equation (S2).

2. PCA-based polynomial regression with Elastic Net cost function

Polynomial regression is an extension of linear regression; it adds powers of each original feature as new features, then trains a linear regression on this extended set of features. The power terms make this ML algorithm capable of solving non-linear problems.

Principal component analysis (PCA) is a widely used dimensional reduction technique. It involves representing inter-correlated variables as a new set of independent variables, and extracting the most important information in the original data and summarizing them using fewer variables (Greatorex 2015). Because the *n* features become $\frac{(n+d)!}{d!n!}$ features after a d-degree feature transform, we used PCA to reduce the dimensions of the original features. To avoid overfitting, we applied elastic net regularization, which adds penalty terms to the loss function to reduce the complexity of the model.

3. Neural network regression

The neural network used in this study was an artificial neural network (or multi-layer perceptron), a supervised learning algorithm. It consists of an input layer, one or more hidden layers, and an output layer, and each layer has some number of units (neurons). The features (**X**) of the training set are imported into the input layer. Then, each unit in the hidden layer(s) transforms the values from the previous layer with a weighted linear summation ($w_1x_1 + w_2x_2 + \cdots + w_nx_n$) followed by a non-linear activation function (Pedregosa et al. 2011). The last hidden layer transfers values to the output layer, where the prediction is given.

4. Decision trees regression

Decision trees (Breiman et al. 1984) is a data-driven, model-based, nonparametric estimation machine-learning method for structuring models to make predictions based on data (Loh 2011; Trendowicz and Jeffery 2014). A single decision tree stepwise partitions the data space into a set of subsets and fits a simple regression model to each node to give the predicted value of $\hat{Y}$ (Trendowicz and Jeffery 2014). Decision trees is the base model of the random forest and Extra-Trees regressions. In this study, Na/Mg was added into the features for the decision trees, random forest, and Extra-Trees regression to get better predictions. More detailed information on the decision trees algorithm is available in Breiman et al., 1984.

5. Random forest regression

The random forest algorithm (Breiman 2001) is an ensemble learning algorithm, combining the bagging algorithm (Breiman 1996) with random feature selection (Ho 1998; Petrelli et al. 2020). The bagging (bootstrapping aggregating) algorithm randomly selects a number of subsets with replacement from the entire training set, and each subset is used to train an individual decision tree. The final output from the bagging predictors is the average prediction of the regression.

The random forest algorithm incorporates random feature selection into the bagging algorithm. It also consists of many decision trees, each trained using a subset sampled by bootstrapping. However, when each tree splits at a node, the feature that most satisfies the indicators of feature selection (such as the Gini coefficient and Kullback-Leibler divergence) is selected from the randomly selected feature set instead of the entire feature dataset. The random feature subsets generate much larger differences between component trees compared to bagging predictors, reducing the variance of the model and decreasing the possibility of overfitting.

6. Extremely randomized trees regression

The extremely randomized trees (Extra-Trees) algorithm is similar to the random forest algorithm, as it builds many unpruned regression trees according to the classical top-down procedure (Geurts et al. 2006). The only two differences between the two are that, in Extra-Trees, (i) the entire training dataset, instead of a bootstrapped replica, is input for each component tree, and (ii) the cut-point (feature for splitting) is chosen completely at random when splitting nodes (Geurts et al. 2006).

The Extra-Trees algorithm includes two parameters: the number of attributes randomly selected at each node (K) and the minimum sample size for splitting a node (n_{min}) (Geurts et al. 2006). A number of trees, M, are trained with the full training dataset to generate an ensemble model (Geurts et al. 2006). Thus, K influences the strength of the attribute selection process, n_{min} determines the strength of noise in the averaged output, and M determines the reduction of variance in the ensemble model aggregation (Geurts et al. 2006). The final output is the average of all component tree predictions for regression problems (Geurts et al. 2006).

7. Artificial neural network ensemble regression

In this study, we combined the artificial neural network and bagging algorithms to form artificial neural network ensemble regression, following the principles of these two algorithms as stated above. The hyperparameters of the base artificial neural network models are fixed, as are the initialized connection weights, and each model is trained on a sub-dataset sampled by bootstrapping. The average value of the output of each model is the final output.

References

Breiman, L. (1996) Bagging predictors. Machine Learning, 24, 123–140.

——— (2001) Random forests. Machine learning, 45, 5–32.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984) Classification And Regression Trees, 1st ed. Chapman and Hall/CRC. Chapman and Hall/CRC.

Géron, A. (2017) Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st ed. (T. Nicole, Ed.). O’Reilly Media,Inc.

Geurts, P., Ernst, D., and Wehenkel, L. (2006) Extremely randomized trees. Machine Learning, 63, 3–42.

Greatorex, M. (2015, January 21) Principal Component Analysis. Wiley Encyclopedia of Management.

Ho, T.K. (1998) The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 832–844.

Loh, W.-Y. (2011) Classification and regression trees. Wires Data Mining and Knowledge Discovery, 1, 14–23.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011) Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825–2830.

Petrelli, M., Caricchi, L., and Perugini, D. (2020) Machine Learning Thermo-Barometry: Application to Clinopyroxene-Bearing Magmas. Journal of Geophysical Research: Solid Earth, 125.

Trendowicz, A., and Jeffery, R. (2014) Classification and Regression Trees. In A. Trendowicz and R. Jeffery, Eds., Software Project Effort Estimation pp. 295–304. Springer, Cham.