**ML Algorithms information**

**Note S1. The computational method of the GBR regression algorithm**

The **Gradient Boosted Regression (GBR)** model is an integrated ML algorithm that is generated by the integration of weak regression trees1,2. Given the training samples 𝑫 = {(𝐱𝟏, 𝐲𝟏), (𝐱𝟐, 𝐲𝟐) …, (𝐱𝐧, 𝐲𝐧)}, the number of leaf nodes in every regression tree is ***J***. We divided the input data into J disjoint areas and defined each regression tree as **tm(x)**. The training goal of GBR is to minimize the loss function ***L***, and the parameters of decision tree 𝛉𝐦 are determined through empirical risk minimization:

The process of GBR training is as follows:

(a) Initialize a regression tree function 𝑓0 (𝑥).

(b) Train GBR in the gradient decline direction, and compute the negative gradient value of the loss function as the estimated value of the residual. For the mth iteration, GBR generates a regression tree according to the residual and updates the current function 𝑓m (𝑥).

(c) The final regression model is the weighted sum of several weak regression trees, which is defined as:

**Note S2. The computational method of the RFR regression algorithm**

The **Random Forest Regression (RFR)** model is an ensemble learning method for regression by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees3. Random decision forests correct for decision trees' habit of overfitting to their training set. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x1... xn with responses Y = y1... yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

1. For b = 1... B: Sample, with replacement, n training examples from X, Y; call these Xb, Yb; Train a regression tree fb on Xb, Yb.
2. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x'

**Note S3. The computational method of the SVR regression algorithm**

The **Support Vector Regression (SVR)** model is a supervised learning model with associated learning algorithms that analyze data used for regression analysis4.Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Training the original SVR to:

(a) Minimize.

(b) Subject to

where is a training sample with a target value The inner product plus intercept is the prediction for that sample, and is a free parameter that serves as a threshold: all predictions have to be within a range of the true predictions. Slack variables are usually added into the above to allow for errors and to allow approximation in the case the above problem is infeasible.

**Note S4. The computational method of the KNR regression algorithm**

The ***K*-Neighbour Regression (KNR)** model is a non-parametric method proposed by Thomas Cover used for classification and regression5. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance). Often, the classification accuracy of *k*-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis. This algorithm works as follows:

(a) Compute the Euclidean or Mahalanobis distance from the query example to the labeled examples.

(b) Order the labeled examples by increasing distance.

(c) Find a heuristically optimal number k of nearest neighbors, based on RMSE. This is done using cross-validation.

(d) Calculate an inverse distance weighted average with the k-nearest multivariate neighbors.

**References:**

1. J. H. Friedman, Stochastic gradient boosting. *Computational statistics & data analysis* 2002, **38**, 367-378.

2. J. H. Friedman, Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 2001, 1189-1232.

3. J. Quiñonero-Candela, C. E. Rasmussen, A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* 2005, **6**, 1939-1959.

4. A. Liaw, M. Wiener, Classification and regression by random Forest. *R news*, 2002, **2**, 18-22.

5. A. Seko, T. Maekawa, K. Tsuda, I. Tanaka, Machine learning with systematic density-functional theory calculations: Application to melting temperatures of single-and binary-component solids. *Physical Review B* 2014, **89**, 054303.