Rensselaer

why not change the world?®

More Supervised & Unsupervised Methods

Ahmed Eleish

Data Analytics ITWS-4600/6600 CSCI-4960 MGMT 4600/6600

March 27th 2026

Tetherless World Constellation
Rensselaer Polytechnic Institute

Contents

- Classification
  - XGBoost
- Clustering
  - DBSCAN
- Dimension Reduction
  - Factor Analysis

My box root folder:
https://rpi.box.com/s/7bvcvtj7qrp2c7yllpr6javw713iypen

eXtreme Gradient Boosting (XGBoost)

- Similar to Random Forest, XGBoost is based on decision trees.
- Many trees are built, one at a time, each learning from the errors of previous trees.
- Uses gradients of the loss function to minimize errors.
- Utilizes boosting: combining multiple weak classifiers sequentially to form a strong classifier
- The final prediction is the weighted sum of the predictions from all the trees.

Evolution of Tree Algorithms

https://www.enjoyalgorithms.com/blog/xg-boost-algorithm-in-ml

Algorithm

Onjective Function: $L(\varphi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$

$l(y_i, \hat{y}_i)$ = loss function
$\Omega(f_k)$ = regularization term

1. Initialize predictions (constant values)
2. Calculate initial residuals $Y - \hat{Y}$
3. For m = 1, 2, 3, …, M
    - 3.a. Calculate 1st and 2nd derivatives of loss function for all samples
    - 3.b. Build tree m
        - 3.b.1. Use derivatives to calculate gain to decide best split
        - 3.b.2. Split until stopping criteria (e.g. Max depth reached, min samples in node, etc.)
        - 3.b.3. Calculate leaf weights
    - 3.c. Update predictions: add new tree's predictions scaled by learning rate η
    - 3.d. Apply Regularization
4. Make final prediction

eXtreme Gradient Boosting (XGBoost)

- Pros:
  - Scalability, support for parallel processing and efficient data structures.
  - Handling of missing values.
  - Customizability with multiple hyperparameters.
- Cons:
  - Sensitivity to noise and outliers.
  - Limited interpretability.
  - Tendency to overfit with small datasets.

Iris_XGBoost.R

https://rpi.box.com/s/lzwq9s0p5mrsyp4l53d221nrcxix8mq8

Factor Analysis

- Identifies latent variables (factors) between correlated variables
- Can be used for dimension reduction.
- Commonly used in psychology, marketing, biology and others.
- Used for hypothesis testing and features selection.

Algorithm

1. Standardize data (center + scale by SD)
2. Compute correlation matrix *R*
3. Determine number of factors
4. Extract initial factors
5. Rotate factors
6. Interpret factors

Factor Analysis

- Pros:
  - Reduce data dimension/complexity, remove multicollinearity
- Cons:
  - Unsuitable for smaller datasets
  - Factor interpretation is subjective

mtcars_FA.R

https://rpi.box.com/s/lzwq9s0p5mrsyp4l53d221nrcxix8mq8

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

- A density-based clustering algorithm.
- Identifies clusters by examining the density of points in a given region.
- Two key parameters:
  - epsilon (ε): maximum distance between points to be considered part of the same cluster.
  - minPts: the minimum number of points required to form a dense region.
- Especially effective for datasets containing noise and clusters of varying densities.

Sadeghi, B. (2025). Clustering in geo-data science: Navigating uncertainty to select the most reliable method. *Ore Geology Reviews*, 106591. https://doi.org/10.1016/j.oregeorev.2025.106591

Algorithm

1. Initialize (mark points as unvisited)
2. For each unvisited point *P:*
    - 2.a. Mark *P* as visited
    - 2.b. Find all neighbors *N* of *P* within ε distance
    - 2.c. If |N| < minPts: mark *P* as noise
    - 2.d. Else: create new cluster *C* and add *P* to it
3. Expand cluster *C* from *P,* for each point *Q* in neighorhood N
    - 3.a. Do: 2.a –> 2.d
    - 3.b. If *Q* not in cluster, add to cluster *C*
4. Repeat 2-3 until all points are visited

DBSCAN

- Pros:
  - Find arbitrary, non-spherical clusters.
  - Robust to outliers.
  - Suitable for geo-spatial data.

- Cons:
  - Not suitable for high-dimensional data
  - Assumes clusters of similar density

Iris_DBSCAN.R

https://rpi.box.com/s/lzwq9s0p5mrsyp4l53d221nrcxix8mq8

Thanks!