UNIVERSITY OF TWENTE.

Classification and regression via DECISION TREES AND RANDOM FORESTS

Presented by:

Mahdi Farnaghi

Assistant Prof.

Department of Geo-Information Processing

ITC

FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

Background

- Supervised vs unsupervised learning

Poll#1:

- Are MLPs an example of

  a) supervised or

  b) unsupervised learning?

Background

- Supervised vs unsupervised learning

- Typical tasks of supervised learning
  - Classification (e.g., land cover maps)
  - Regression/prediction (e.g., biomass maps)

- For theses tasks → many methods available in literature
  - Data Driven vs. Process Driven (e.g., Kriging)

- Today’s lecture → Decision trees and Random forests

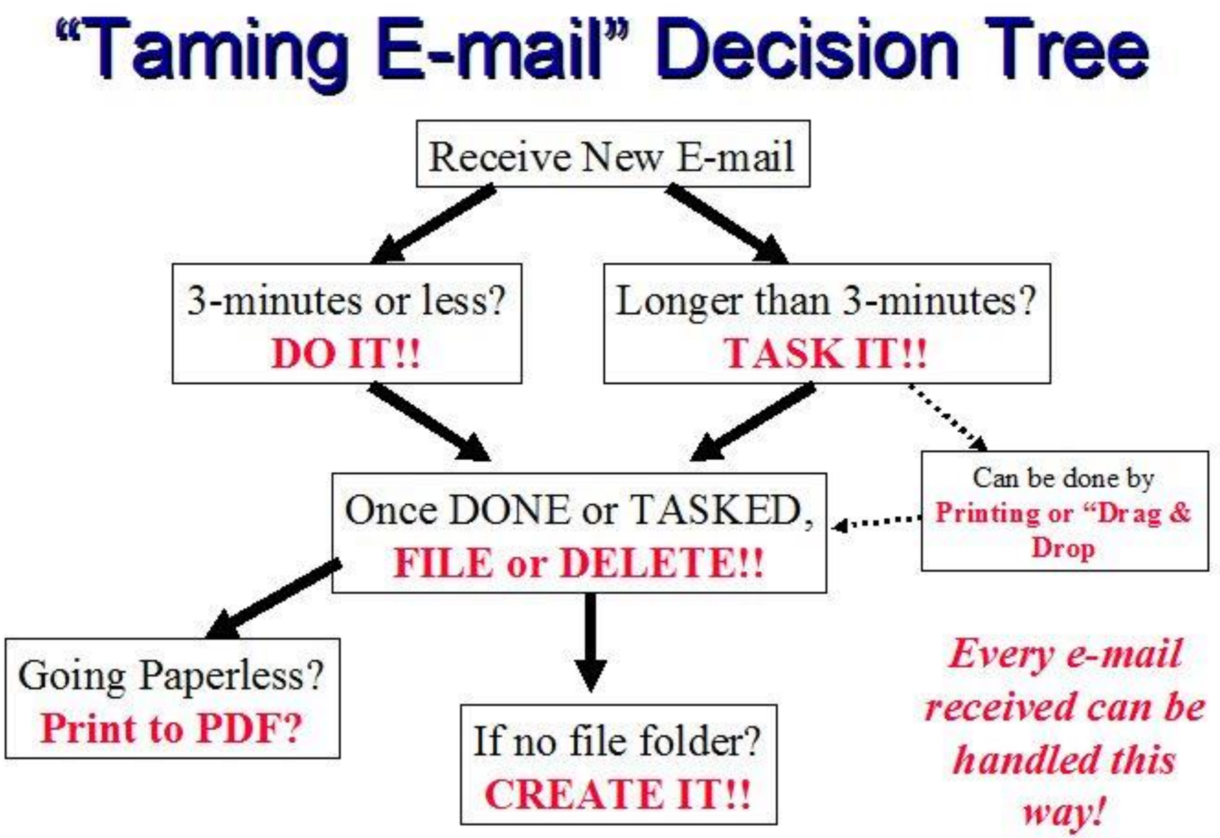
Contents

- Decision trees
- CART: classification and regression trees
- Ensembles: random forests

- Software
- Further reading

Decision Tree - CART

What is a decision tree?



Sorting coins. See here

Decision trees

Decision trees do **recursive partitioning of the data** for classification and/or regression tasks

- Conceptually simple
- Very effective, especially when coupled with randomization techniques

A bit of history

- AID: automatic interactive decision tree (Morgan and Sonquist, 1963)
  - High risk of overfitting → misleading conclusions
  - Lack of analytical rigor

Decision trees: CART

A group of computer scientists found similarities between DT and KNN

- Terminal node trees → dynamical NN classifier (neighborhood)

This group of scientists led by Leo Breiman invented:

- CART: Classification and Regression Trees (Breiman et al., 1984)

- CART is one of the most popular DT methods because
  - It can cope with continuous and categorical data (both as targets and as predictors)
  - It is analytically rigorous

NOTE: CART is also a commercial software package (Salford Systems)

Decision tree algorithms

- ID3
- C4.5
- *CART*
- CHAID (Chi-square automatic interaction detection)
- MARS
- Conditional Inference Trees

Decision trees: CART

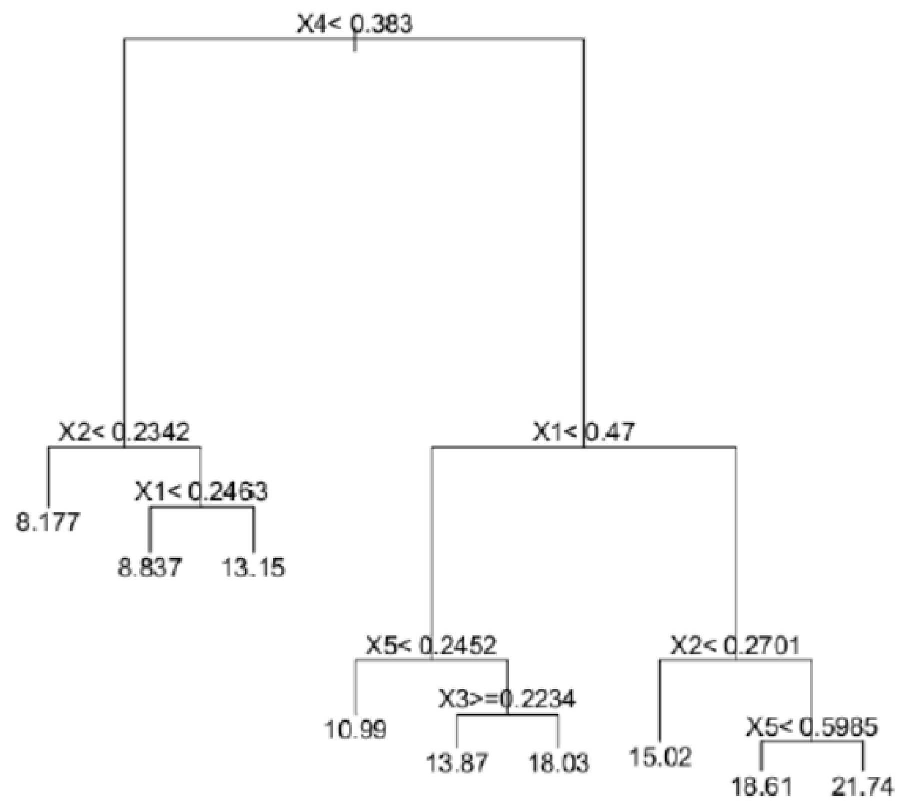
- The main idea behind CART is:
  - To generate a binary tree
  - To minimize the training error in the tree

Decision trees: terminology

- Root node
- Node
- Terminal node

- Branch
- Split

- Attribute or features (X1, X2, …)
- Response or target variables (Y)



UNIVERSITY OF TWENTE.

CART algorithm

- Start by creating the root node (all data)
- Root → 2 children → 4 grandchildren….
- “Grow” the tree until no further splits are possible (lack of data).
- “Prune” back using the cost-complexity method.
  - Splits are pruned sequentially according to their contribution to the performance on the training data.
  - Remove less relevant splits
- Evaluate the set of nested pruned trees by using an independent dataset
  - Or use cross-validation

CART algorithm (II)

- The use of DT require a clear definition of the following 4 elements:

1. A way to select a split at every intermediate node
2. A rule for determining if a node is a terminal one
3. A rule for assigning a value (Y_{est}) to each terminal node

Splitting rules (intermediate nodes)

- An object goes left IF the chosen attribute meets some CONDITION, otherwise it goes right
  - Continuous data: X <= Condition
  - Nominal data: X belongs to set {A,B,C,D}

- The splitter and the split point are chosen by CART
  - Always binary splits
  - An attribute can be used multiple times

Splitting rules

- Use the split (variable and condition) that most decreases a cost function:

For instance:

  - For categorical data, the GINI index
  - For regression, the MSE
- To maximize information gain

- Other cost functions are possible/described in the original CART monograph.

- DT partition the data so that each unit is as homogeneous as possible wrt the response variable (Y)

Splitting algorithm

- Greedy Iterative procedure
  - Starting with a single region -- i.e., all given data
  - At the m-th iteration:

for each region R

    for each attribute x_j in R

        for each possible split s_j of x_j

            record change in <u>score</u> when we partition R into R^l and R^r

Choose (x_j , s_j) giving maximum improvement to fit

Replace R with R^l; add R^r

Rule node is terminal

- CART grows the tree until
  - all the data is the resulting node is homogeneous

    or
  - it contains less elements than a (chosen) threshold

- After a maximum tree has been created, it is pruned back
  - Larger the tree, more likely to overfit training data
  - Pruning finds subtrees that generalize beyond training data
  - Based on trading off tree complexity and goodness of fit to the data (node purity)

Value terminal node

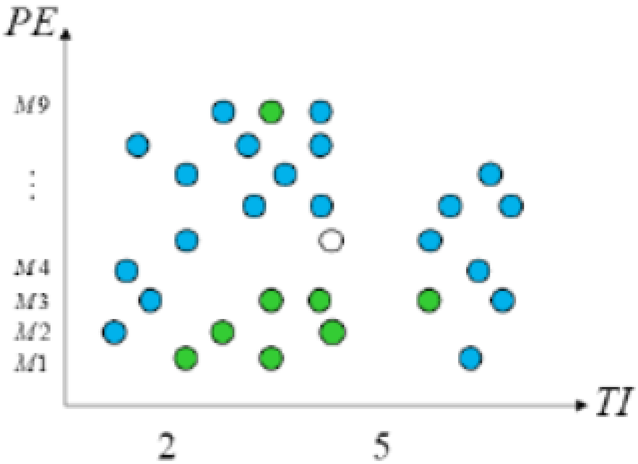
- For categorical data
  - Y_{est} (t)= Mode of the labels of all the elements in the terminal node

- For continuous data
  - $Y_{est}(t) = \frac{1}{n(t)} \sum_{Xi \in t} Y_i$

  So… the mean value of the response variable in the terminal node

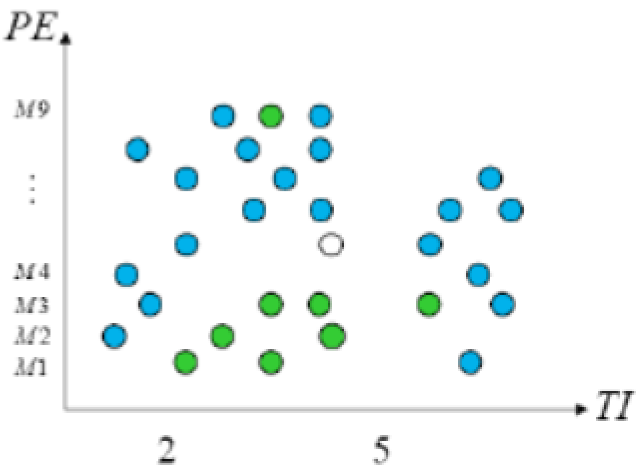
Decision trees: an example

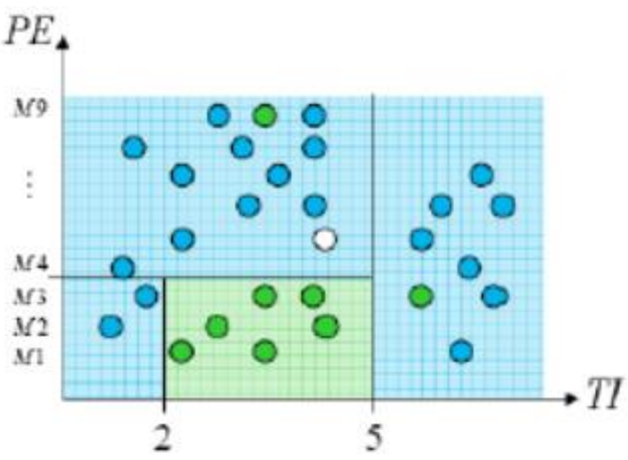
TI	*PE*	*Response*
1.0	*M2*	good
2.0	*M1*	bad
...	...	...
4.5	*M5*	?

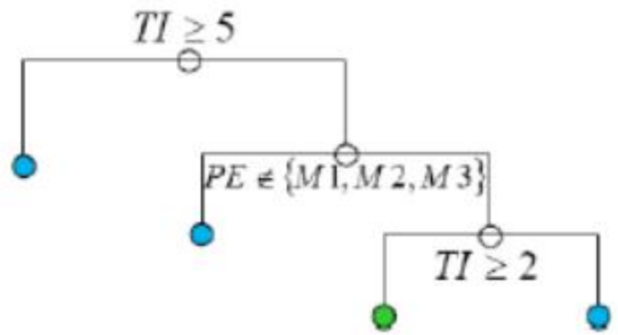


Decision trees: an example

TI	PE	Response
1.0	M2	good
2.0	M1	bad
...	...	...
4.5	M5	?







(Montillo, 2009)

Decision Trees

- Advantages
  - Output is easy to understand
  - Can combine numeric and categorical data
  - Robust (outliers)
  - Fast (after developing the rules)
- Disadvantages
  - Overfitting
  - Limited to the range of the attributes in the training data
  - Unstable (small perturbation input → larger perturbation output)

Random Forest

Random forests

- Leo Breiman continued working on DT and around the year 2000 he found and demonstrated that regression results and classification accuracy can be improved by using
  - ensembles of trees where
  - each tree grown in a “random” fashion.

- This work resulted in “random forests”

- Ensemble = a set of elements.
- Ensemble methods are becoming highly popular → computer power

Random forests (II)

- RF are fast and easy to implement.
- They yield highly accurate predictions
  (even if the input data has a high dimensionality)
- No overfitting
- Provide insight on the importance of each attribute/feature/dimension
- They are easily parallelizable
- Data does not need pre-processing

- They are one of the most popular general-purpose ML methods

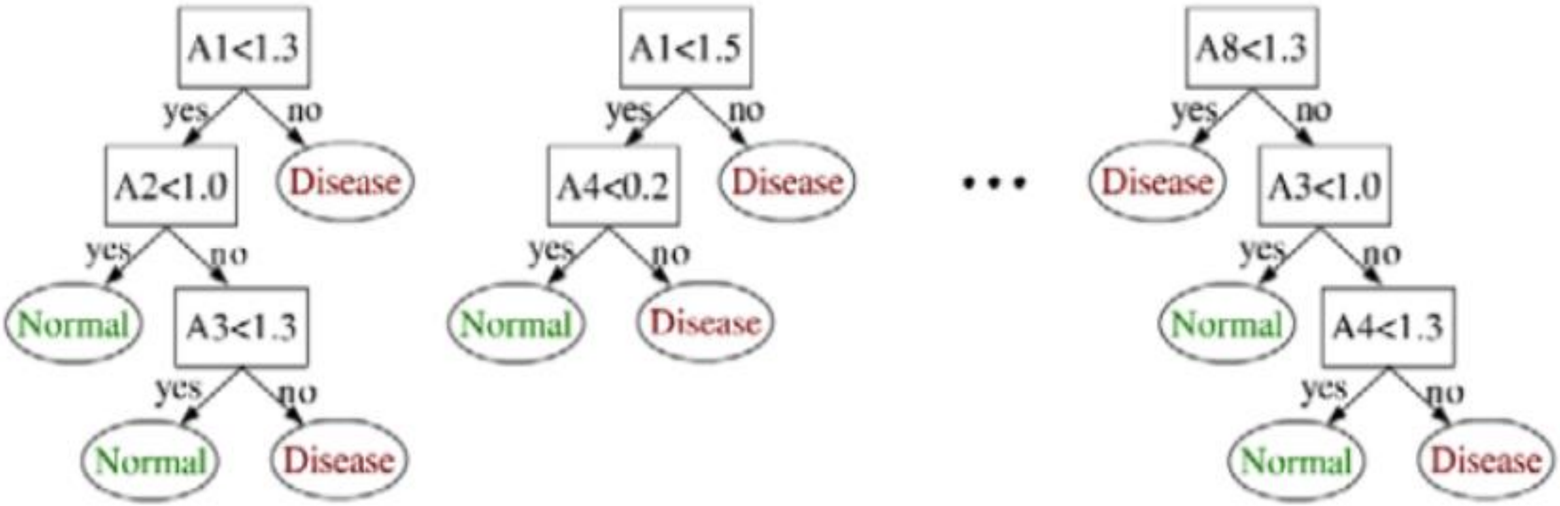
Random forests (III)

- RF produces an ensemble of decision trees during the training
- Each tree is the result of applying the CART method to a random selection of attributes/features at each node.
- And by using a random subset of the original input data (chosen with replacement, -- bootstrapping || Bagging = bootstrapping aggregation)
- Response variables are obtained by voting over the ensemble

Random Forest: algorithm

- Input data: N training cases each with M variables
- n out of N samples are chosen with replacement (bagging).
- Rest of the samples to estimate the error of the tree (out of bag)
- m << M variables are used to determine the decision at a node of the tree
- Each tree is fully grown and <u>not pruned</u>

Random Forest: an example



Random Forest

- Advantages
  - No pruning needed
  - High Accuracy
  - Provides variable importance
  - No overfitting || Not very sensitive to outliers

- Disadvantages
  - Cannot predict (regression) beyond range of input parameters
  - Smoothing extreme values (underestimate high values; overestimate low values)
  - More difficult to visualize/interpret

Spatial and temporal data ?!

- DT (and RF) do not directly use spatio-temporal information
- They only make use of the attributes/values at all the sampled locations and times

- Remember to always examine the spatial variability of the results to check the "validity" of the classification and/or regression.
- Do not forget to make use of maps and other geovisualizations

DT & RF software

- R packages
  - Party
  - Rpart
  - Randomforest
  - …
- Python
  - Scikits learn (sklearn)
  - …

Earthquake Prediction

ELSEVIER

Soil Dynamics and Earthquake Engineering

Volume 144, May 2021, 106663

Spatiotemporally explicit earthquake prediction using deep neural network

Mohsen Yousefzadeh [a], Seyyed Ahmad Hosseini [b], Mahdi Farnaghi [c]

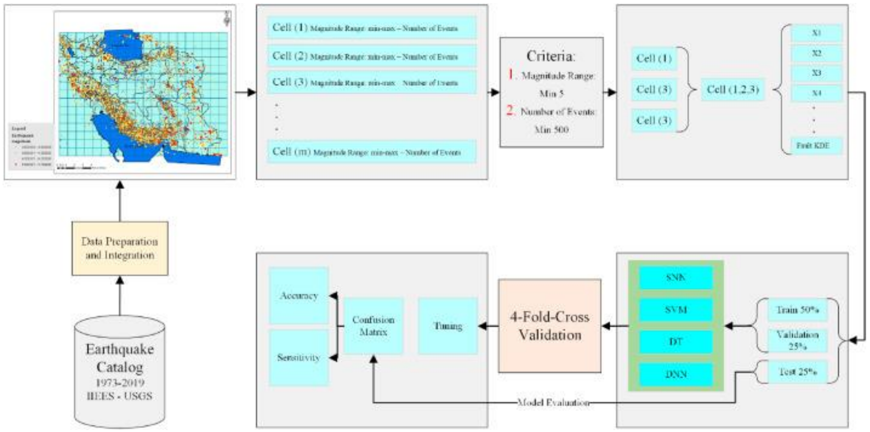
Show more

+ Add to Mendeley   Share   Cite

https://doi.org/10.1016/j.soildyn.2021.106663

- Three machine learning algorithms were compared with a DNN
  - Shallow Neural Network
  - SVM
  - Decision Tree

Earthquake Prediction



- From overall accuracy perspective, DT performance was better than DNN

Table 9. Test data Accuracy.

	SNN	SVM	DT	DNN
Parameter-Set 1	70.4%	78%	82%	78%
Parameter-Set 2	70.0%	78%	80%	78.4%
Parameter-Set 3	61.2%	74.8%	81.2%	79.6%

AIR Pollution Modelling

Environ Monit Assess (2019) 191: 183
https://doi.org/10.1007/s10661-019-7253-2

Check for updates

Proposing and investigating PCAMARS as a novel model for NO_2 interpolation

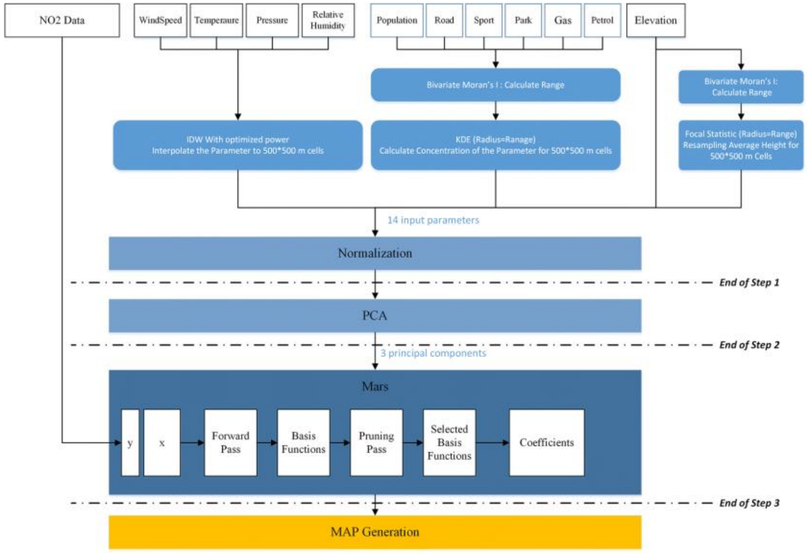
Mohsen Yousefzadeh · Mahdi Farnaghi · Petter Pilesjö · Ali Mansourian

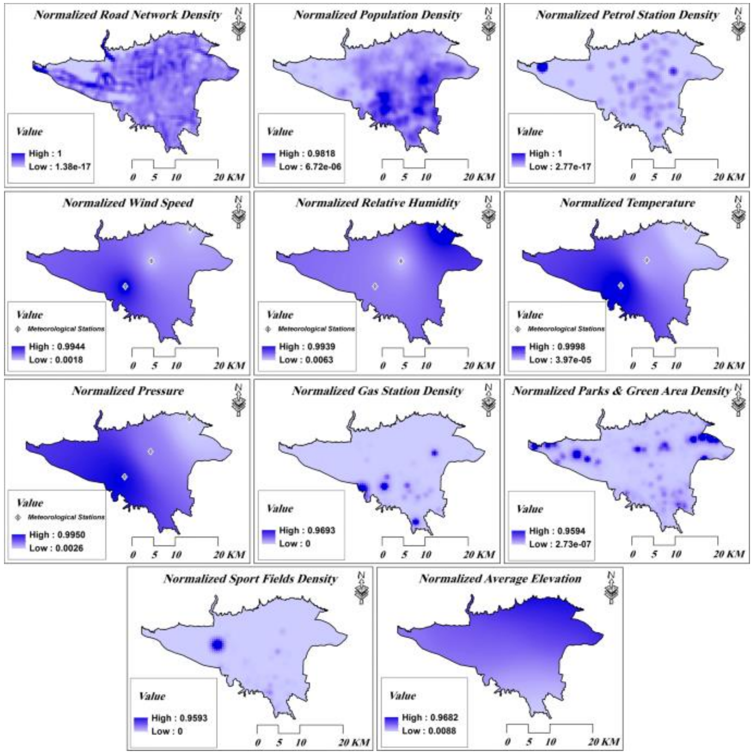
Received: 13 November 2018 / Accepted: 21 January 2019 / Published online: 23 February 2019
© The Author(s) 2019

https://doi.org/10.1007/s10661-019-7253-2

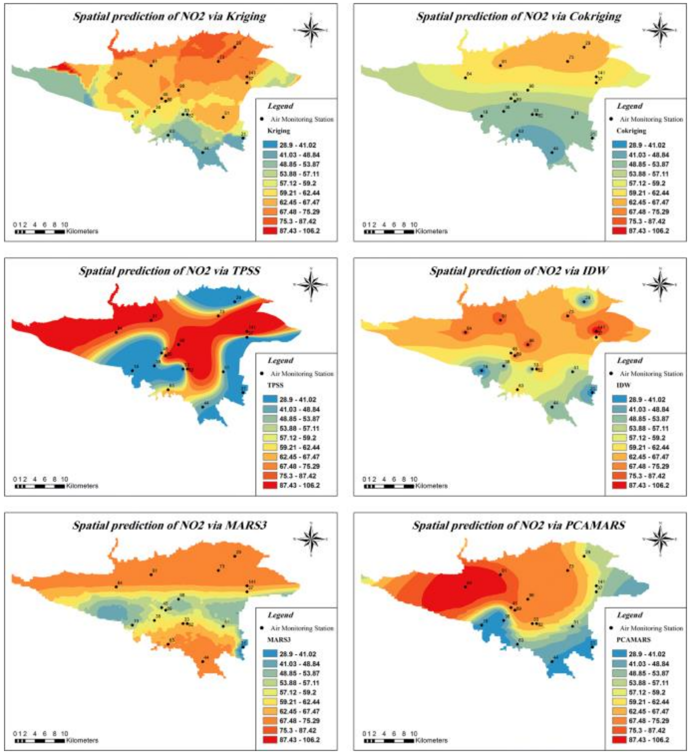
- MARS
  - A particular type of decision tree
  - Coupled with PCA

AIR Pollution Modelling





AIR Pollution Modelling



Method	Average RMSE
IDW	26.61
TPSS	45.83
OK	24.61
CK	22.08
MARS3	19.13
PCAMARS	18.24

Further reading

Books

- Top ten algorithms in data mining.

  Xindong Wu and Vipin Kumar (Eds). 2009
- Machine learning in Action.

  Peter Harrington. 2012.
- Spatial data analysis in ecology and agriculture using R.

  Richard .E. Plant. 2012.
- R and data mining

  Yanchang Zhao. 2012
- Machine Learning With Random Forests And Decision Trees: A Visual Guide For Beginners.

  Scott Hartshorn. 2016

Multiple presentations and material online ☺

Spatio-temporal analytics and modeling

A decisive tree

Questions??