Carnegie Mellon University

MSE

Data Analytics for Materials Science

27-737

A.D. (Tony) Rollett, R.A. LeSar (Iowa State Univ.)

Dept. Materials Sci. Eng., Carnegie Mellon University

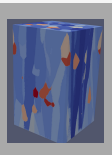
Random Forest

Lecture 6

Revised: 21st Apr., 2021

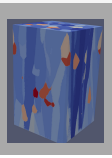
To date, we have discussed:

- linear algebra
- linear regression: prediction
- multiple linear regression: prediction



Useful sources of information (both in Canvas):

- The algorithm for random forests is presented on Page 588 of Hastie et al. Elements of Statistical Learning. linear regression: prediction
- Another useful resource for learning about random forests is: Leo Breiman, Random forests, *Machine learning*, **45**, 5–32 (2001).



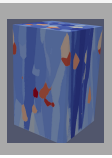
A **decision tree** is a tool for making decisions that uses a tree-like model of decisions and their possible consequences.

A formal decision tree consists of three types of nodes:[1]

- Decision nodes
- Chance nodes
- End nodes

Decision trees are all about information and how to use it in a structured way.

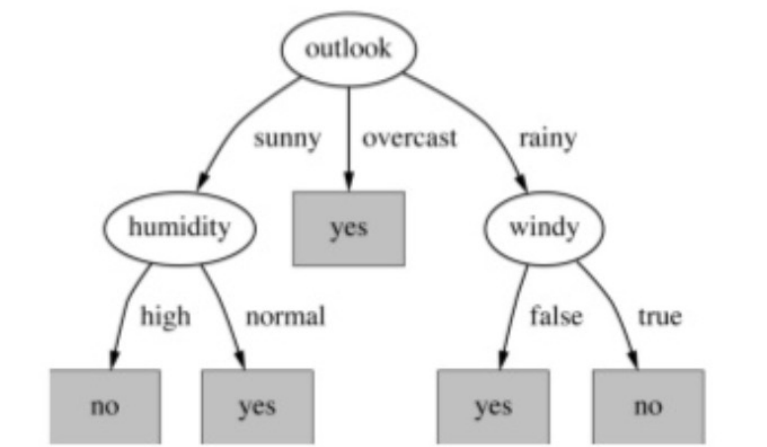
We mention them here because they are the building blocks of the random forest model and useful in their own right.



"What feature will split the observations in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible)?"

Splitting stops when the data cannot be split further.

To play tennis or not to play tennis?

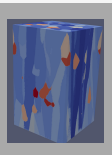


https://www.slideshare.net/marinasantini1/lecture-4-decision-trees-2-entropy-information-gain-gain-ratio-55241087

In a decision tree model, splits are chosen to maximize information gain. For a regression problem, the residual sum of squares (RSS) can be used and for a classification problem, the Gini index or entropy would apply. (See talk on slideshare.)

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

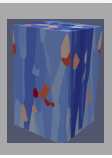
Pruning decision trees is discussed at:
https://en.wikipedia.org/wiki/Decision_tree_pruning



High entropy alloy dataset (we have seen this in the discussion of regular expressions) with composition including 24 elements in five phases.

Can we predict Vicker's hardness based on composition and rule of mixtures (ROM) density?

	Al	Co	Cr	Cu	Fe	Hf	Mo	Nb	B	C	...	Zr	Zn	Y	BCC	FCC	Im	HCP	B2	ROM Density	Vickers Hardness
0	NaN	33.33	NaN	NaN	33.33	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	0	0	0	8.5	125.0
1	NaN	33.33	NaN	NaN	33.33	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	0	0	0	8.5	125.0
3	NaN	30.77	NaN	NaN	30.77	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	0	0	0	7.7	149.0
4	NaN	28.57	NaN	NaN	28.57	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	1	0	0	7.1	287.0
5	NaN	26.67	NaN	NaN	26.67	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	1	0	0	6.6	570.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
349	NaN	17.86	NaN	NaN	17.86	NaN	17.86	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	1	0	0	8.5	520.0
350	NaN	17.24	NaN	NaN	17.24	NaN	17.24	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	1	0	0	8.5	510.0
351	NaN	16.67	NaN	NaN	16.67	NaN	16.67	NaN	NaN	NaN	...	NaN	NaN	NaN	0	1	1	0	0	8.5	382.0
352	NaN	14.29	NaN	NaN	14.29	NaN	14.29	NaN	NaN	NaN	...	14.29	NaN	NaN	0	0	0	0	0	7.3	790.0
353	NaN	NaN	NaN	16.67	16.67	NaN	NaN	NaN	NaN	NaN	...	16.67	NaN	NaN	0	0	0	0	0	6.8	590.0

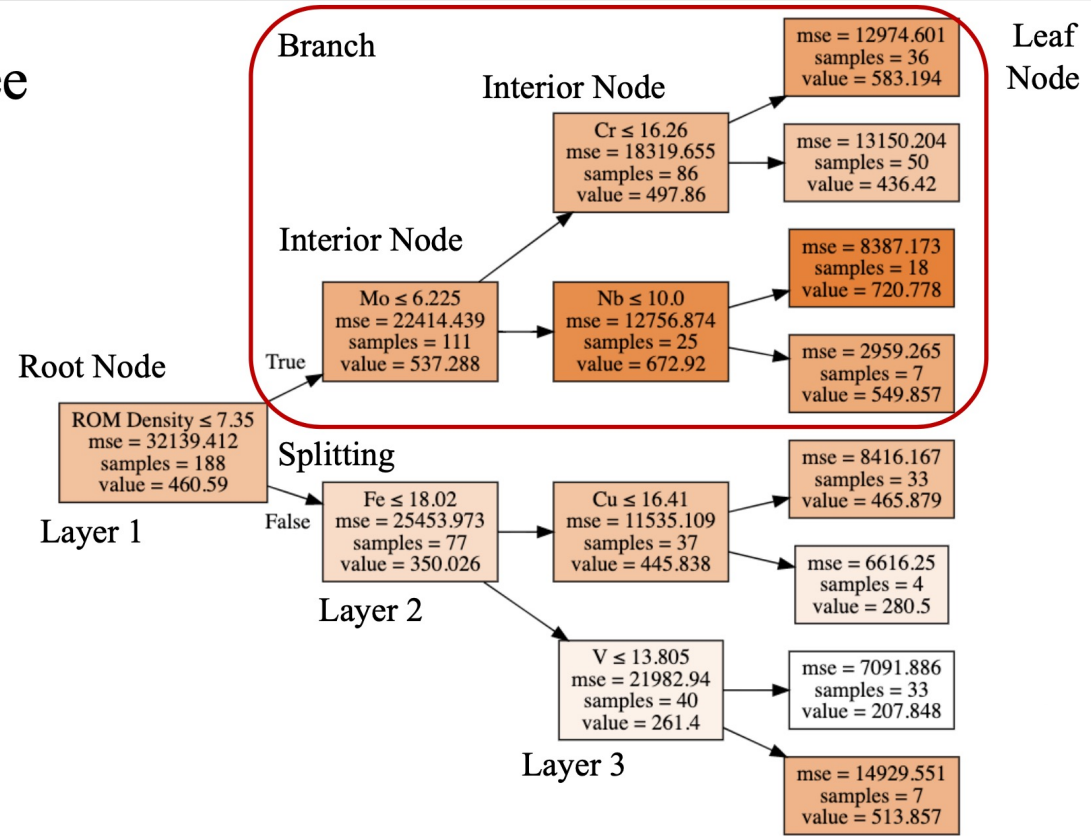


Greedy Approach is based on the concept of Heuristic Problem Solving by making an optimal local choice at each node. By making these local optimal choices, we reach the approximate optimal solution globally."
The algorithm can be summarized as :

1. At each stage (node), pick out the best feature as the test condition.
2. Now split the node into the possible outcomes (internal nodes).
3. Repeat the above steps until all the test conditions have been exhausted into leaf nodes.

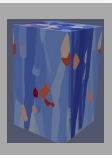
Decision Tree

1. Top-down greedy approach: best split is made at that step
2. Splitting: regions that leads to the greatest possible reduction in RSS
3. Repeat the process



see: https://www.slideshare.net/marinasantini1/lecture-4-decision-trees-2-entropy-information-gain-gain-ratio-55241087

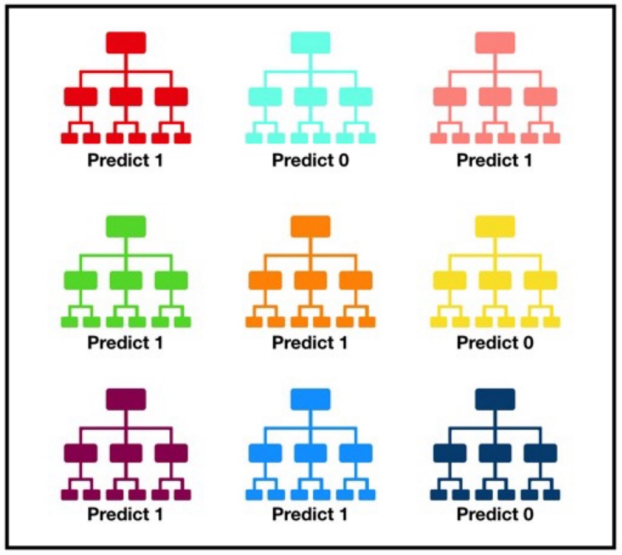
Courtesy of Tony Rollett.

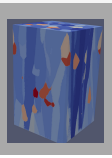


"Random forests are **bagged decision tree** models that split on a **subset of features** on each split." https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706

"Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (see figure)."

https://towardsdatascience.com/understanding-random-forest-58381e0602d2





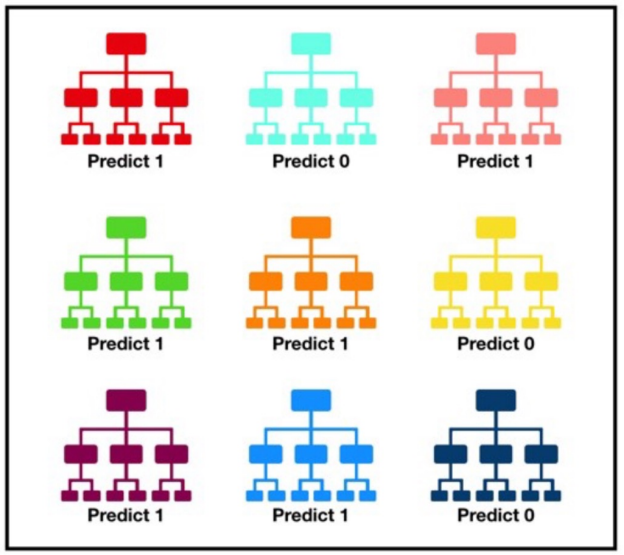
Random Forest model: basic idea

The basic concept behind random forest is based on the *wisdom of* ***crowds***.

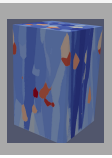
Random forest takes a large number of *uncorrelated* trees (models) that operate as a committee, which will outperform *any* of the individual models.

A key feature is that the models must have low correlation between them.

The low correlation between trees protects each of them from their individual errors.



https://towardsdatascience.com/understanding-random-forest-58381e0602d2



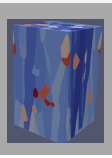
Random Forest model: uncorrelated trees

Decision trees are very sensitive to the data they are trained on — small changes in a training set can result in tree structures with large differences in structure.

Random forest allows individual trees to randomly sample the dataset with replacement.

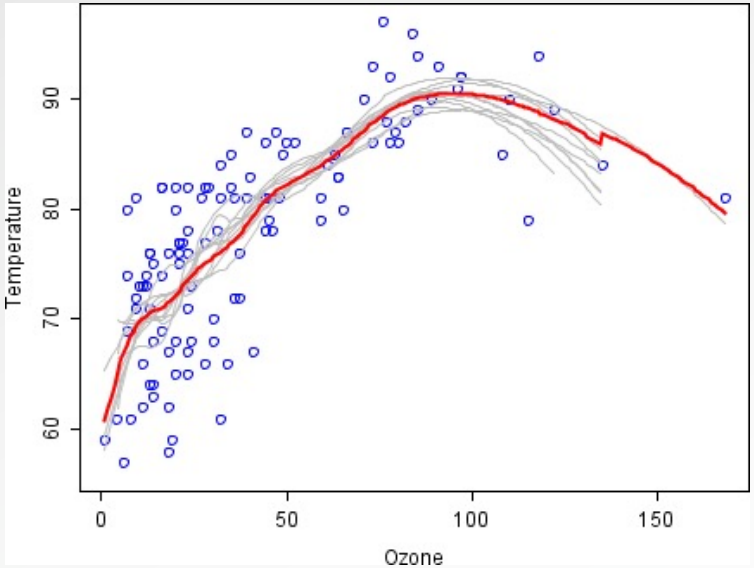
For example, suppose we have a training dataset with N=6 points: $\{1,2,3,4,5,6\}$. Random sampling the data set *with replacement* might lead to something like $\{1,2,2,5,5,6\}$, in which N=6.

Note that bagging can also be used by taking subsets of the data, as we see on the next slide.

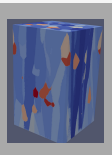


"Instead of building a single smoother from the complete data set, 100 bootstrap samples of the data were drawn. Each sample is different from the original data set, yet resembles it in distribution and variability. For each bootstrap sample, a LOESS smoother was fit. Predictions from these 100 smoothers were then made across the range of the data. The first 10 predicted smooth fits appear as grey lines in the figure below. The lines are clearly very *wiggly* and they overfit the data - a result of the bandwidth being too small."

https://en.wikipedia.org/wiki/Bootstrap_aggregating



By taking the average of the 100 smoothers, we arrive at one bagged predictor (red line). Clearly, the mean is more stable and there is less overfit.



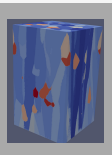
Bootstrap aggregating (bagging)

Reducing variance

- a natural way to reduce the variance and hence increase the prediction accuracy of a statistical learning method is to take many training sets from the population, build a separate prediction model using each training set, and average the resulting predictions

Best practice:

- each bagged tree makes use of around 2/3 of the observations
- remaining 1/3 of the observations are referred to as the out-of-bag (OOB) observations
- each individual tree has high variance, but low bias, averaging these trees reduces the variance
- reduces overfitting; reduce bias; break the bias-variance trade-off
- See later comments for use of OOB data for testing accuracy and feature importance

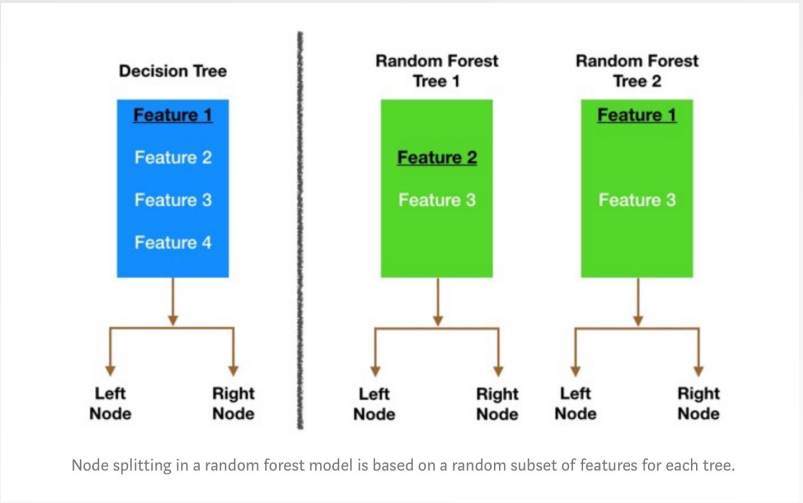


Bagging: advantages

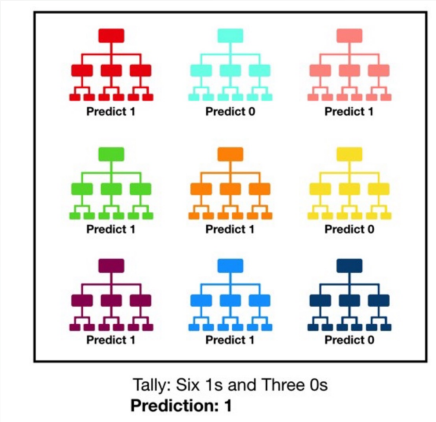
"Random forests are **bagged decision tree** models that split on a **subset of features** on each split."

In addition to bagging, each tree in a random forest bases its split on a random *subset of features*.

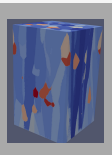
In the example, while a decision tree would include all 4 features, each tree in a random forest would base their split on a subset of features.



Node splitting in a random forest model is based on a random subset of features for each tree.



Tally: Six 1s and Three 0s
Prediction: 1



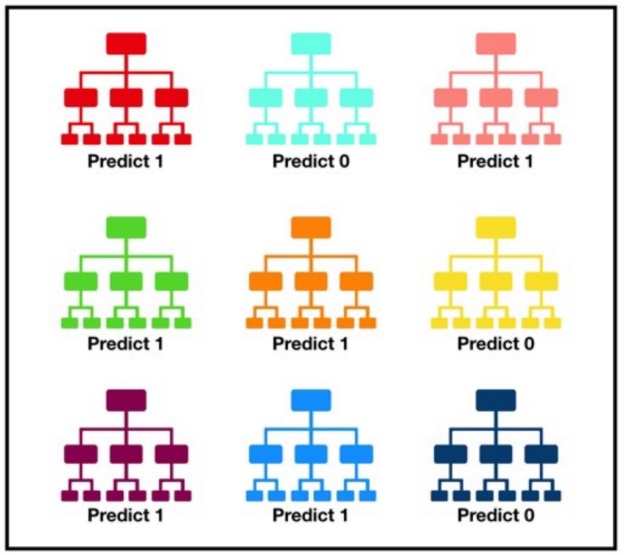
Random Forest model: basic idea

The basic concept behind random forest is based on the *wisdom of* ***crowds***.

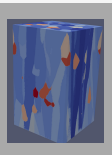
Random forest takes a large number of *uncorrelated* trees (models) that operate as a committee, which will outperform *any* of the individual models.

A key feature is that the models must have low correlation between them.

The low correlation between trees protects each of them from their individual errors.



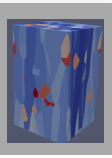
https://towardsdatascience.com/understanding-random-forest-58381e0602d2



Random Forest model: uncorrelated trees

"The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree."

https://towardsdatascience.com/understanding-random-forest-58381e0602d2



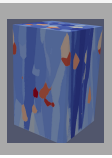
Random Forest model: summary

Decision trees

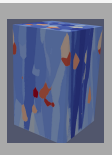
- trees give insight into decision rules
- rather fast computationally
- prediction of trees tend to have high variance

Random Forest

- “Black Box” — rather hard to gain insight into the decision rules
- rather slow computationally
- has smaller prediction variance and thus usually a better performance



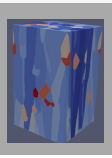
- No statistical assumptions
- Works with any kind of data – continuous / categorical – intrinsically multiclass
- Can express any function – regression / classification
- Works well with small to medium data, unlike neural network which requires large data
- Can handle thousands of input variables without variable selection
  - provides feature importance
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing



Random Forest: attributes

1. How much each feature decreases the variance in a tree
   - For a forest, the variance decrease from each feature can be averaged and the features are ranked according to this measure
   - Biased towards preferring variables with more categories
     (Bias in random forest variable importance measures: Illustrations, sources and a solution — on Canvas)
   - When dataset has two (or more) correlated features, then one shows up high while other as low (applies to other methods too)
     - The effect of this phenomenon is somewhat reduced by random selection of features at each node creation

2. Random shuffling of the variables
   - permute the values of each feature and measure how much the permutation decreases the accuracy of the model
   - The OOB data is passed along each tree to determine the "test error" (since the OOB were not used to train). See section 15.3.1 in Hastie *et al.*
   - For each variable, the values are permuted in the OOB to evaluate the sensitivity to that variable (from the increase in the test error).



Random Forest model: interpretation

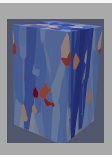
R: `randomForest` package (available on CRAN)

Matlab: `TreeBagger` selects a random subset of predictors to use at each decision split as in the random forest algorithm. (see documentation)

Mathematica: use `Predict[]` with `Method–> “RandomForest”`

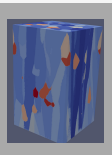
There are also implementations in Python, …

Pick your favorite program and search for random forest in the documentation.



Random Forest model: availability

QUESTIONS?

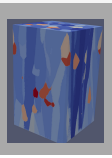


"Despite the recent fast progress in materials informatics and data science, data-driven molecular design of organic photovoltaic (OPV) materials remains challenging. We report a screening of conjugated molecules for polymer−fullerene OPV applications by supervised learning methods (artificial neural network (ANN) and random forest (RF)).

We report a screening of conjugated molecules for polymer−fullerene OPV applications by supervised learning methods (artificial neural network (ANN) and random forest (RF)). Approximately 1000 experimental parameters including power conversion efficiency (PCE), molecular weight, and electronic properties are manually collected from the literature and subjected to machine learning with digitized chemical structures. Contrary to the low correlation coefficient in ANN, RF yields an acceptable accuracy, which is twice that of random classification."

Results based on 1200 points from 500 papers.

Computer-Aided Screening of Conjugated Polymers for Organic Solar Cell: Classification by Random Forest, S. Nagasawa et al, J. Phys. Chem Lett. **9**, 2639 (2018)



Random Forest model: examples from materials research

Artificial Neural Nets (ANN) led to a relation with r=0.37, which is not acceptable.

They represented PCE in 4 groups (e) and used the RF in (d).

Based in part on the RF results, they demonstrated an alternative approach to the design of polymers for OPVs.

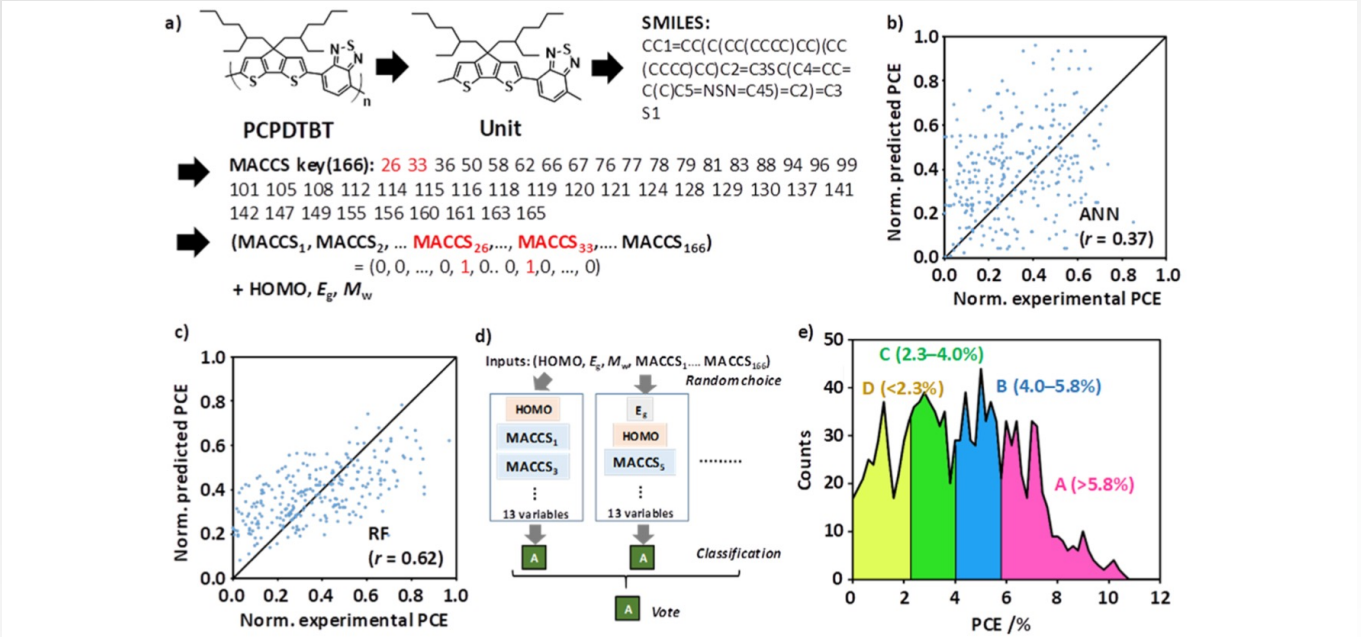
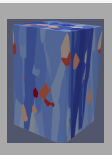


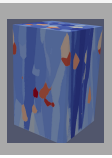
Figure 2. (a) Scheme of converting a chemical structure to digitized data. (b) Results of ANN and (c) results of RF, where the horizontal and vertical axes represent the normalized experimental PCE and predicted PCE, respectively, and *r* is the correlation coefficient. The diagonal black line indicates the perfect positive correlation ($r = 1$). (d) Scheme of a classification using RF. (e) Histogram of the collected experimental PCE data (~1200) and classification of $n = 4$. For example, label "A" corresponds to the highest PCE group.



Random Forest model: examples from materials research

1. How much each feature decreases the variance in a tree
   - For a forest, the variance decrease from each feature can be averaged and the features are ranked according to this measure
   - Biased towards preferring variables with more categories
     (Bias in random forest variable importance measures: Illustrations, sources and a solution — on Canvas)
   - When dataset has two (or more) correlated features, then one shows up high while other as low (applies to other methods too)
     - The effect of this phenomenon is somewhat reduced by random selection of features at each node creation

2. Random shuffling of the variable
   - permute the values of each feature and measure how much the permutation decreases the accuracy of the model



Lecture 17: RF models part II