Lecture 24: Random Forests

CS109A Introduction to Data Science

Pavlos Protopapas, Kevin Rader and Chris Tanner

Outline

- Random Forest (RF)
- Tuning the hyperparameters of a RF
- Feature interpretation in a RF

1. Decision Trees

Review:

To learn a decision tree model, we take a greedy approach:

1. Start with an empty decision tree (undivided feature space)
2. Choose the ‘optimal’ predictor on which to split and choose the ‘optimal’ threshold value for splitting by applying a **splitting criterion,** purity of the regions for classification and MSE for regression.
3. Recurse on each new node until **stopping condition** is met
4. For classification, we label each region in the model with the label of the class to which the plurality of the points within the region belong
5. For regression, we predict with the average of the output values of the training points contained in the region.

2. Bagging

Review:

(Bootstrap) we generate multiple samples of training data, via bootstrapping. We train a large decision tree on each sample of data.

(Aggregate) for a given input, we output the averaged outputs of all the models for that input.

Bagging enjoys the benefits of:

1. High expressiveness, by using larger trees each model is able to approximate complex functions and decision boundaries.
2. Low variance, by averaging the prediction of all the models thus reducing the variance in the final prediction.

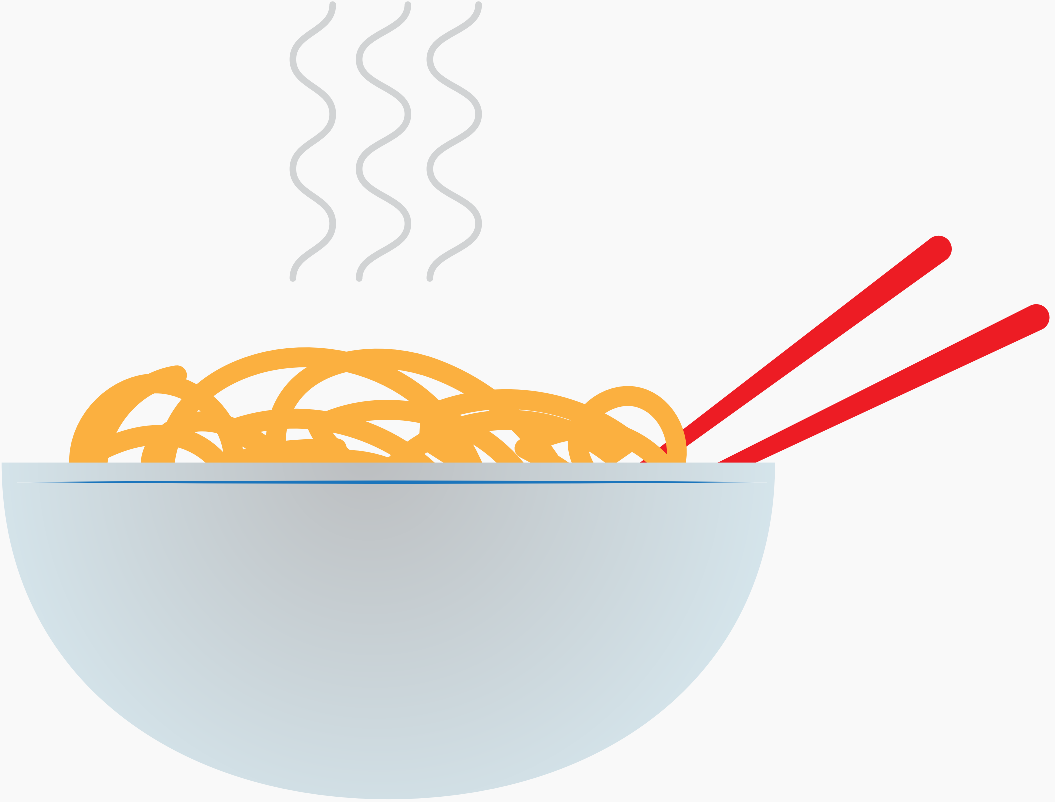
3. Improving on Bagging

Review:

In practice, the ensembles of trees in Bagging tend to be **highly correlated**. Suppose we have an extremely strong predictor, x_j , in the training set amongst moderate predictors. Then the greedy learning algorithm ensures that most of the models in the ensemble will choose to split on x_j in early iterations.

That is, each tree in the ensemble is identically distributed, with the expected output of the averaged model the same as the expected output of any one of the trees.

Ingredients of a Random Forest



Random Forests

A Random Forest is a modified form of bagging that creates ensembles of independent decision trees.

To decorrelate the trees, we:

1. train each tree on a separate bootstrap sample of the full training set (same as in bagging).

2. for each tree, at each split, we ***randomly*** select a set of J predictors from the full set of predictors.

3. From amongst the J predictors, we select the optimal predictor and the optimal corresponding threshold for the split.

Random Forests

(Leo Breiman, 2001)

Random Forest = Decision Tree × number of trees + Bagging + Random selection of features to split each node

Decision Tree: SEEN THAT

Bagging: SEEN THAT

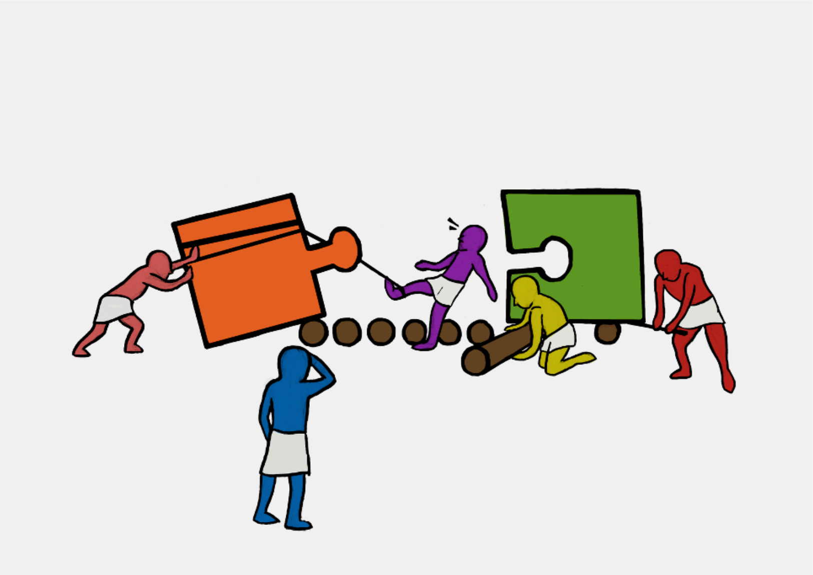
Random selection of features to split each node: NEW !!

WAITING FOR MY
ALGORITHM TO
CONVERGE...

Exercise 1

- Person sharing their screen is the one located **closest to NYC.**

- Be respectful of each other.



- Exercise 1 is about comparing Bagging to RF. You are asked to run a Bagging classification model and a RF classification model and compare the trees.

Tuning Random Forests

Random forest models have multiple hyperparameters to tune:

1. The sampling scheme: number of predictors to randomly select at each split : `max_features {“auto”, “sqrt”, “log2”}, int or float, default=”auto”.`

2. The total number of trees in the forest: `n_estimators`, `int, default=100`

3. The complexity of each tree: stop when a leaf has <= min_samples_leaf samples or when we reach a certain max_depth.

4. In theory, each tree in the random forest is full, but in practice this can be computationally expensive (and added redundancies in the model), thus, imposing a minimum node size is not unusual.

Tuning Random Forests

When the number of predictors is large, but the number of relevant predictors is small, you need to set `max_features` to a larger number.

Question: Why?

If chosen features is small, in each split, the chances of selected a relevant predictor will be low and hence most trees in the ensemble will be weak models.

Tuning Random Forests

There are standard (default) values for each of random forest hyper-parameters recommended by long time practitioners, but generally these parameters should be tuned through the use of out-of-bag **(OOB)** (making them data and problem dependent).

e.g. number of predictors to randomly select at each split:

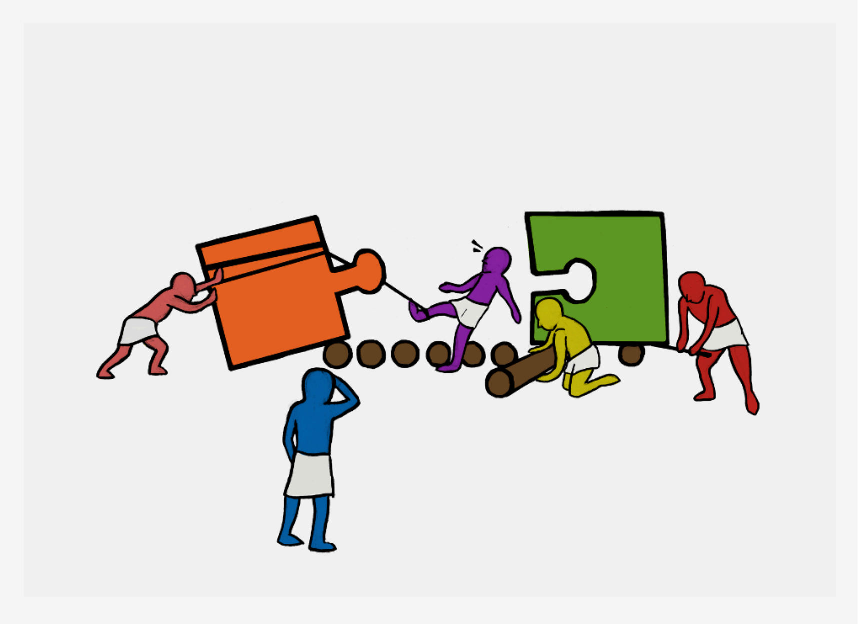
- $\sqrt{N_j}$ for classification
- $\frac{N}{3}$ for regression

Using OOB errors, training and cross validation can be done in a single sequence - we cease training once the OOB error stabilizes.

Exercise 2

- Person sharing their screen is the one located **closest to NYC.**

- Be respectful of each other.

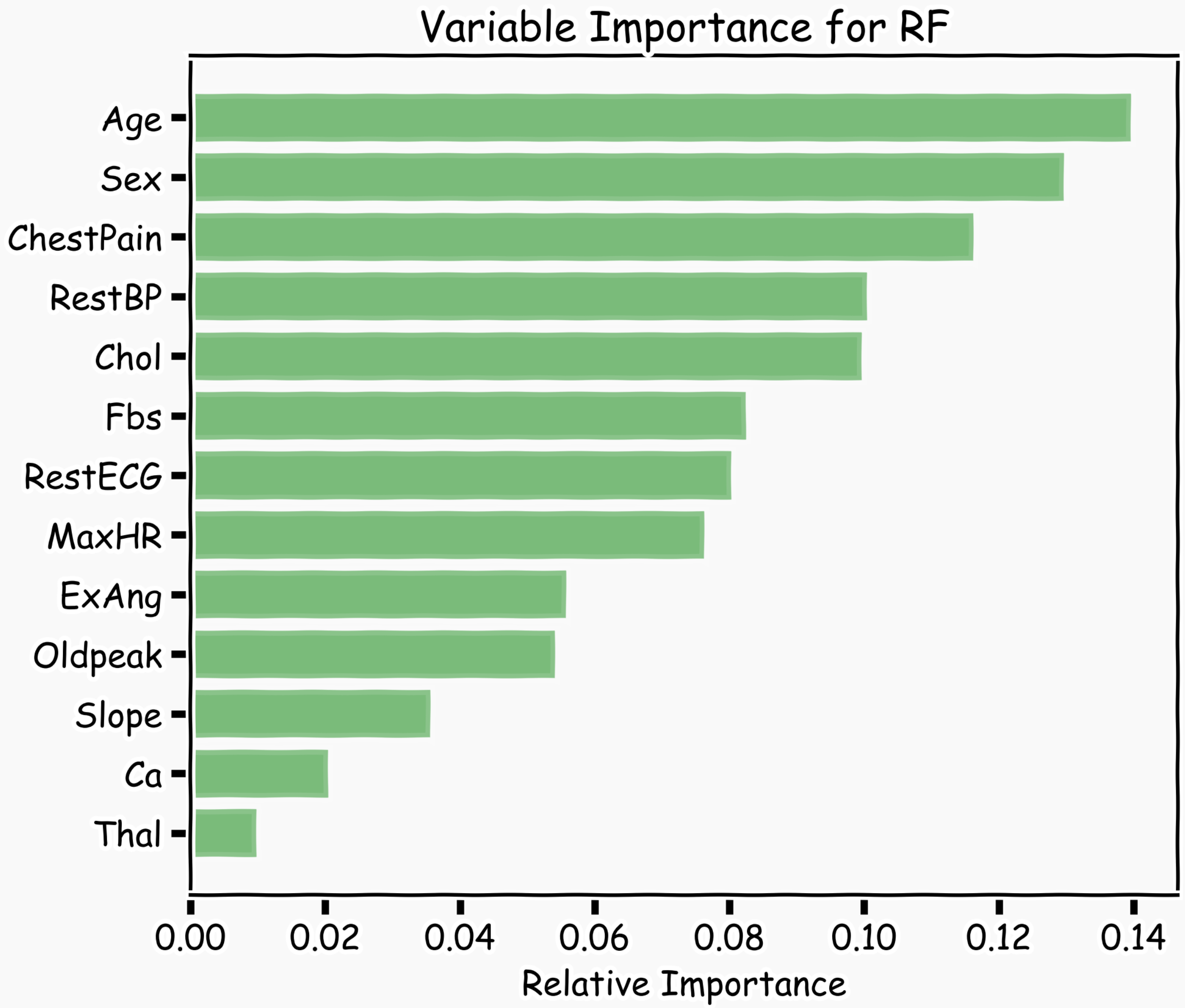


- Exercise 2 is about tuning the hyperparameters in a RF.

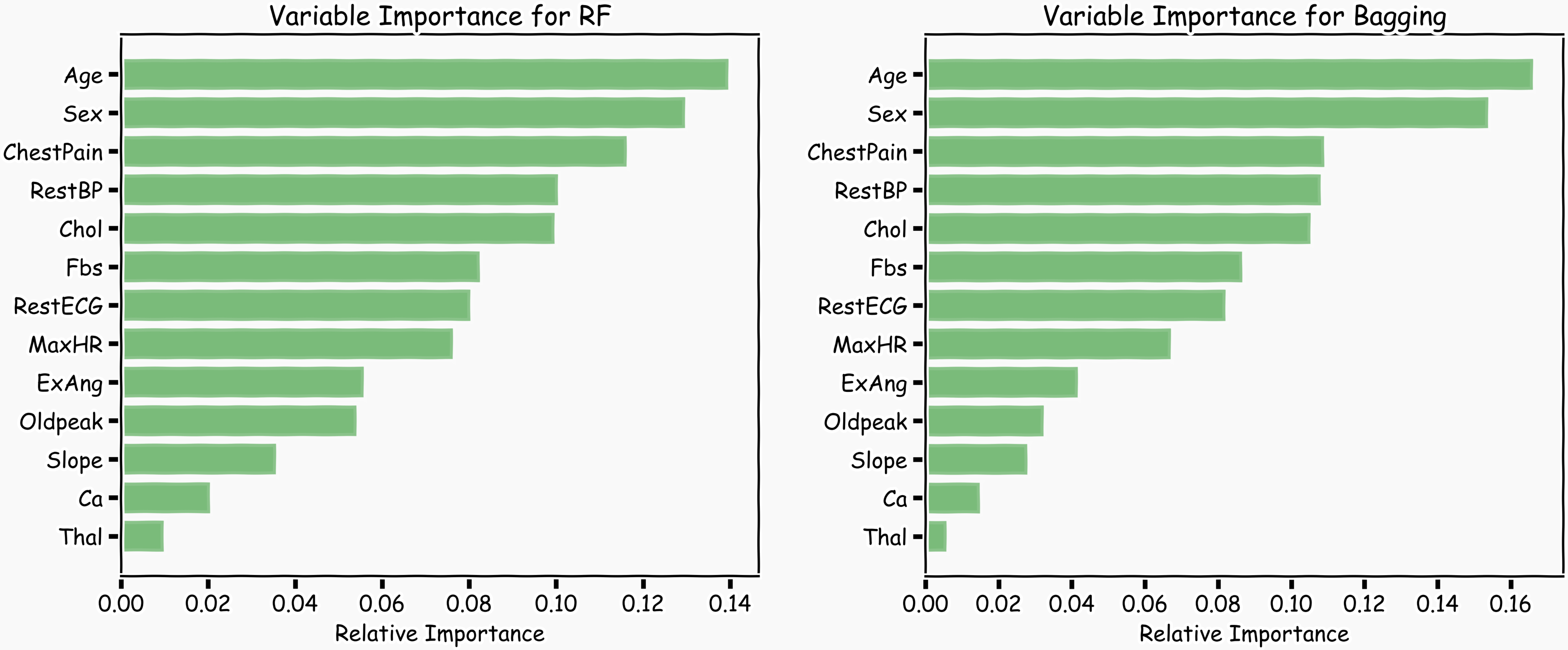
Variable Importance for RF

Explaining predictions from tree models is always desired; the patterns uncovered by a model are, in some applications, more important than the model’s prediction performance.

A drawback of RF, Bagging, and other ***ensemble methods,*** is that the averaged model is no longer easily interpretable - i.e. one can no longer trace the *logic* of an output through a series of decisions based on predictor values!



Variable Importance for RF



100 trees, max_depth=10

Variable Importance for RF

1. Mean Decrease in Impurity (MDI)

- Same as Bagging.
- Record the prediction accuracy on the *oob* samples for each tree.
- Calculate the total amount that the RSS (for regression) or Gini index (for classification) is decreased due to splits over a given predictor, averaged over all trees.
- The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable j in the random forest.
- The default in Scikit-learn `feature_importances_`

Variable Importance for RF

2. Permutation Importance

- Record the prediction accuracy on the *oob* samples for each tree.
- Randomly permute the data for column j in the *oob* samples the record the accuracy again.
- The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable j in the random forest.

3. One step further (SHAP values, LIME)

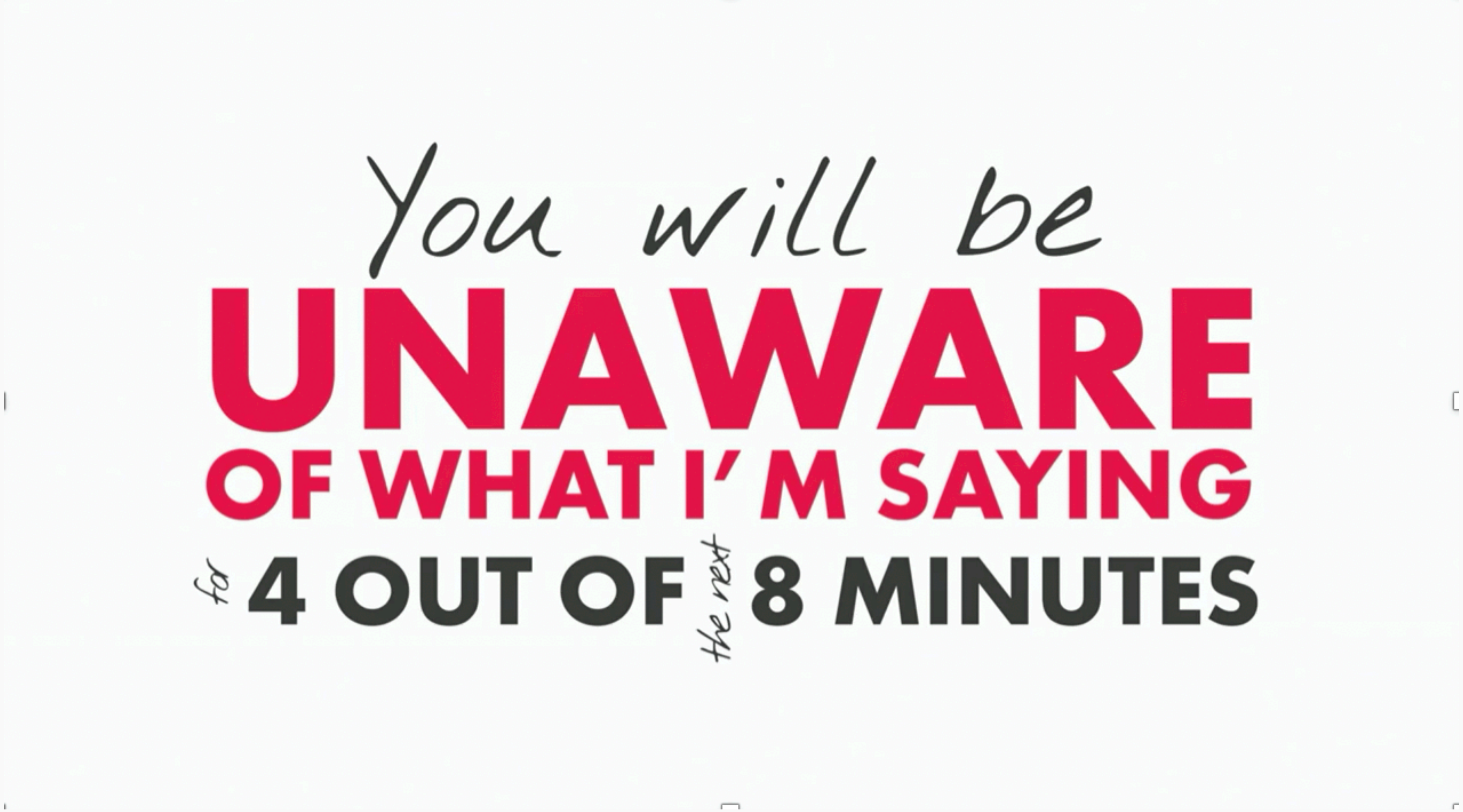
- We will see these methods in later lectures.

Final Thoughts on Random Forests

Increasing the number of trees in the ensemble generally does not increase the risk of overfitting.

Again, by decomposing the generalization error in terms of bias and variance, we see that increasing the number of trees produces a model that is at least as robust as a single tree.

However, if the number of trees is too large, then the trees in the ensemble may become more correlated, increase the variance.



You will be

UNAWARE

OF WHAT I’ M SAYING

for **4 OUT OF** the next **8 MINUTES**

Zeenat Potia

Next Lecture

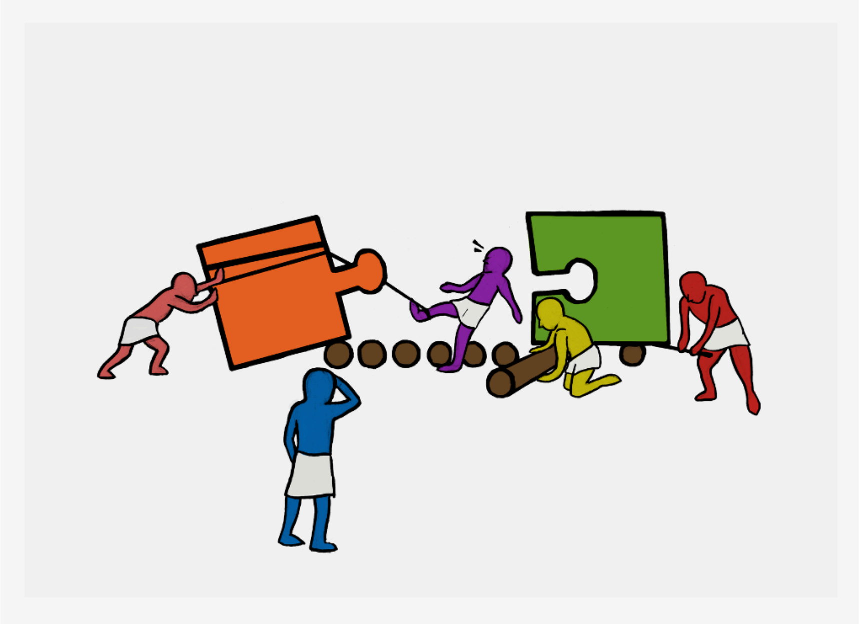
- Imbalanced dataset
  - Weighted samples
- Categorical data
- Missing data

AND BOOSTING!

Exercise 3

- Person sharing their screen is the one located **closest to NYC.**

- Be respectful of each other.



- Exercise 3 is about calculating feature importance in a single tree and a RF, using two methods.