We will use Permutation Feature importance to extract features importance for the random forest algorithm. The permutation importance is a method for which a feature is important if shuffling its values increases the error, and unimportant if shuffling its values leaves the error unchanged.

We set the train data as X, target vector y, and a trained model F. We can use entropy or Gini to calculate the model loss.

1 Calculate the initial error of the given trained model tree F, with error(init) = Gini(F(X), y)

2 For each feature i belongs to {1,2,…,n}:

a. Permute feature i to the same position in every single entry in data X and get the new input Xi.

b. Calculate the error using the given trained tree F, with error(temp) = Gini(F(X),y)

c. Get the difference of the original error and permutated error: error(i) = abs(error(temp) - error(init))

3 Sort features by descending error(i), since a higher difference means higher importance of the feature.

Pros:

1. It does not require retaining the model and we will use one initial trained model throughout

2. easy to implement and understand, as the increase in error means an increase in importance.

Cons:

1. Since the algorithm depends on shuffling the feature, there may be cases in which we get results that vary too much between different permutations.

2. With correlated features, using permutate feature importance could end up with unrealistic data.