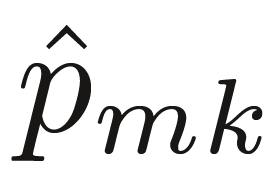
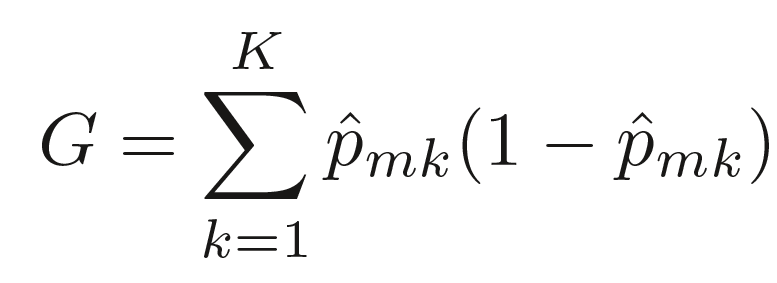
**Classification Trees**

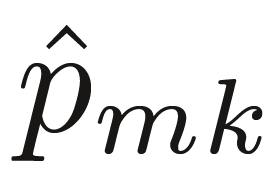
**Classification Trees:**

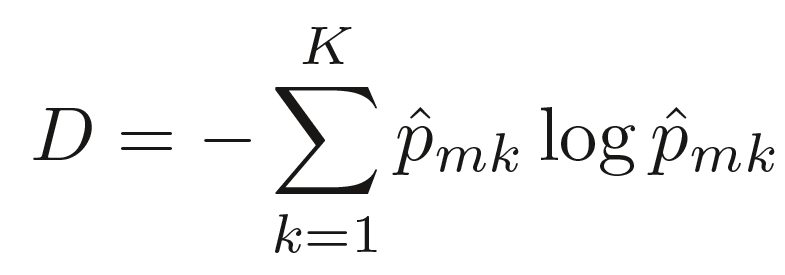
* Unlike regression trees, classification trees are used to predict a qualitative response
* Uses training observation to predict the most common class and the proportion of those classes for different areas of the search space
* Since it is not a regression, RSS cannot be used for making decisions about splitting the observations

**Classification Based Splitting Methods**

* Gini Index
  + Measure of total variance across the K classes
  +  refers to the proportion of training observations that fall within a class



* + Takes on a small value when  is very high or low (close to 0 or 1)
* Entropy
  + Comes from physics where entropy measures the amount of disorder in the system
  + The more even the distribution of training observations into the two classes, the higher the entropy
  + Sometimes referred to as information. Think about it as the information needed to decide about whether to split or not.
  + Small values indicate that we need less extra information in order to classify our observations



**Application in R**

* Packages Needed: ‘rpart’, ‘rattle’, ‘pROC’
* To build a maximal classification tree

**regression.tree = rpart( ~ . , data= " ", method="class",**

**parms = list(split = "information"), control = rpart.control (minsplit=2,**

**minbucket=1, cp=0))**

* To find the cp level to which to prune to (always remember the one standard deviation rule!)
  + Use the simplest model (higher CP) that falls within the xstd of the minimum xerror

**min.xerror <- which.min(max\_tree$cptable[,’xerror’])**

**max\_tree$cptable**

**best\_cp <- max\_tree$cptable[the number row of lowest Xerror, ‘CP’]**

**prune\_tree <- prune(max\_tree, cp = best\_cp)**

* To evaluate the test set

**prune.preds <- predict(prune\_tree, newdata = test\_data, type = 'class')**

**table(test\_data$ThingYouArePredicting, prune.preds,dnn=c("Actual", "Predicted"))**

**round(100\*table(test\_data$ThingYouArePredicting, prune.preds,dnn=c("% Actual", "% Predicted"))/length(prune.preds))**

* Generate ROC Curve

**prune.probs <- predict(prune\_tree, newdata = test\_data)**

**prune.roc <- roc(as.numeric(test\_data$ThingYouArePredicting), prune.probs[,2])**

**plot(prune.roc, col = 'blue', lty = 1)**