Classification

Classification

- Predicting a Qualitative response for an observation can be referred to as **classifying** that observation, since it involves assigning the observation to category or class
- Often the methods used for classification is first predict the **probability** of each of the categories of a qualitative variable, as the basis for making the classification
- There are many classification techniques (classifiers) that one may use to predict a qualitative response
- Logistic Regression is the most widely used classifiers. Other common classifiers are K-nearest neighbors and Decision Trees

Why not Linear Regression for Qualitative Response

- Suppose we are trying to predict the medical condition of a patient in hospital emergency room on the basis of his symptoms. There are three possible diagnoses: stroke, drug overdose and epileptic seizure
- These can be considered to be encoded as quantitative response as:

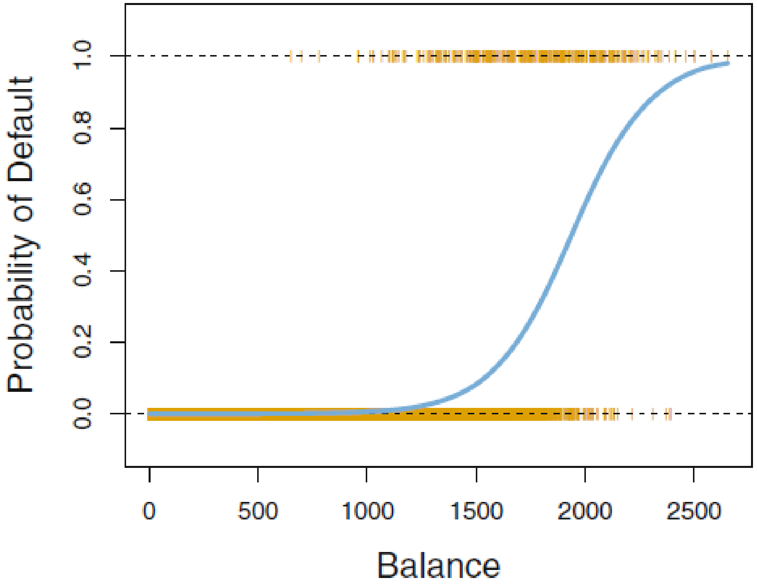
$$Y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$

- Using this coding, least square can be used to fit linear regression
- This coding is forcing drug overdose to be between the other two and insisting successive differences are same which may not be the case
- An interchange of coding is also possible. Each of these set of codings would produce fundamentally different linear models that would ultimately lead to different sets of prediction on test observations
- Situation is better for binary response with 0/1 coding and fitting the linear model and using Y>0.5 to predict 1 and 0 otherwise. But the estimates may go outside [0,1] interval, making them hard to interpret

Logistic Regression

- Unfortunately, in general there is no natural way to convert a qualitative response variable with more than two levels into a quantitative response that is ready for linear regression
- In logistic regression, rather than modelling the response Y directly, it models the probability that Y belongs to a particular category
- Value of probability will range between 0 and 1 and we use Logistic Function for modelling relationship between p(X) and X where p(X) = Pr(Y=1|X)

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$



- Logistic Function will always produce an S-shaped cure
- p(Balance)=Pr(Default=1|Balance) for low Balances will be close to but never below **0** and for high Balances close to but never above **1**

Logistic Regression - Odds

- Another way of writing the Logistic function is

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

- The quantity p(X)/[1-p(X)] is called the odds, and can take any value between 0 and ∞
- Values of the odds close to 0 and ∞ indicate very low and very high probability of default
- For 1 in 5 people will default means p(X) = 0.2 and odds = 0.2/(1-0.2)=1/4
- For 9 in 10 people will default means p(X) = 0.9 and odds = 0.9/(1-0.9) = 9/1
- Odds are traditionally used instead of probabilities in horse racing
- By taking logarithm of both sides we arrive as:

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

- The left hand side is called log-odds or logit. Logistic regression model has a logit which is linear in X

Logistic Regression Coefficients

- In Logistic Regression model, increasing X by 1 unit changes the log odds by β_1, or multiply the odds by $e^{\beta 1}$.
- The amount that p(X) will change due to 1 unit change in X will depend on current value of X
- If β_1 is positive then increasing X will be associated with increasing p(X). If β_1 is negative then increasing X will be associated with decreasing p(X).
- Maximum Likelihood method is used to fit the model. We try to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that plugging these estimates into the model for p(X) yields a number close to 0 for all individuals who defaulted and close to 1 who did not
- Z-statistic associated with $\hat{\beta}_1$ is $\hat{\beta}_1/\mathrm{SE}(\hat{\beta}_1)$ and a large absolute value and low p-value indicates that $\hat{\beta}_1$ is significant and there is an association between X & Y
- Once coefficient has been estimated, $\hat{p}(X)$ can be estimated for a given X
- The Logistic regression can be extended to qualitative predictors

Multiple Logistic Regression

- Multiple Logistic Regression is used for predicting binary response using multiple predictors $X_1, X_2, \ldots, X_p$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots. + \beta_p X_p$$

- The above equation can be re-written as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

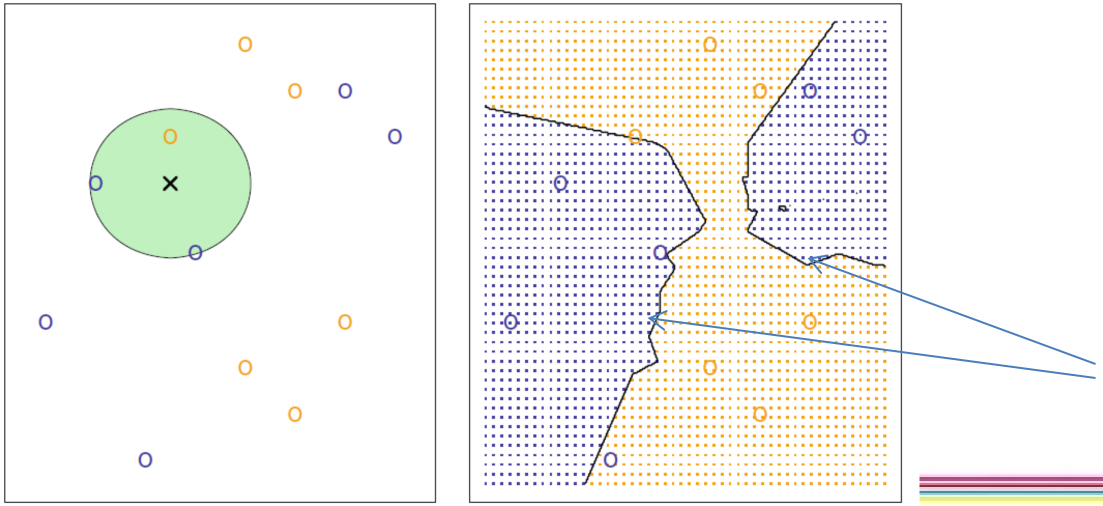
- We use maximum likelihood method to estimate $\beta_{0,}\ \beta_{1,}\ \beta_{2,\ldots,}\ \beta_p$

- Once coefficient has been estimated, $\hat{p}(X)$ can be estimated for given $X_1,..X_p$

K-nearest neighbors (KNN)

- Test error rate is minimized by a classifier that assigns each observation to the most likely class, given its predictor value
- K-nearest neighbors is one of the simplest and best known non-parametric method which follow this route
- KNN starts with a positive integer K and a test observation x_0
- Then it identifies the K points in the training data closest to x_0
- Then it estimates the conditional probability for class j as the fraction of points whose response value equals j
- Finally it classifies the test observation to the class with highest probability

K = 3

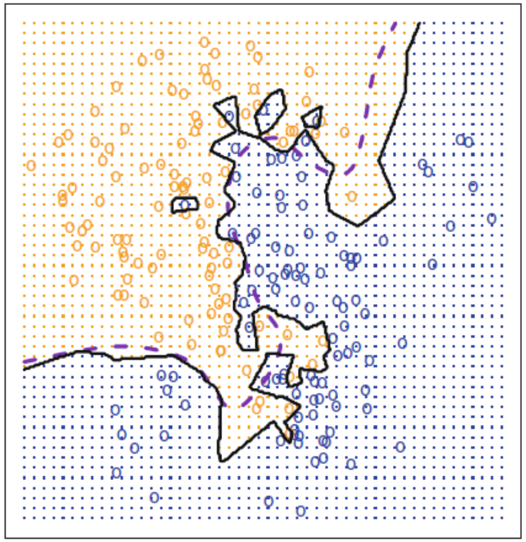


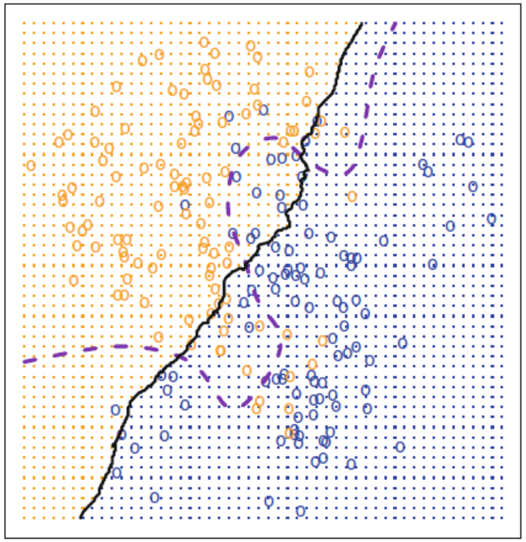
KNN Decision Boundaries

Level of Flexibility for KNN

- The choice of K has drastic effect on the KNN classifier obtained
- For K=1, the decision boundary is overly flexible with low bias but high variance
- As K grows, the method becomes less flexible and K=100 produces a decision boundary that is close to linear with high bias and low variance
- The training error consistently declines as the flexibility increases but test error exhibits a characteristic U-shaped curve
- Choosing correct level of flexibility (K) is important to have low test error rate

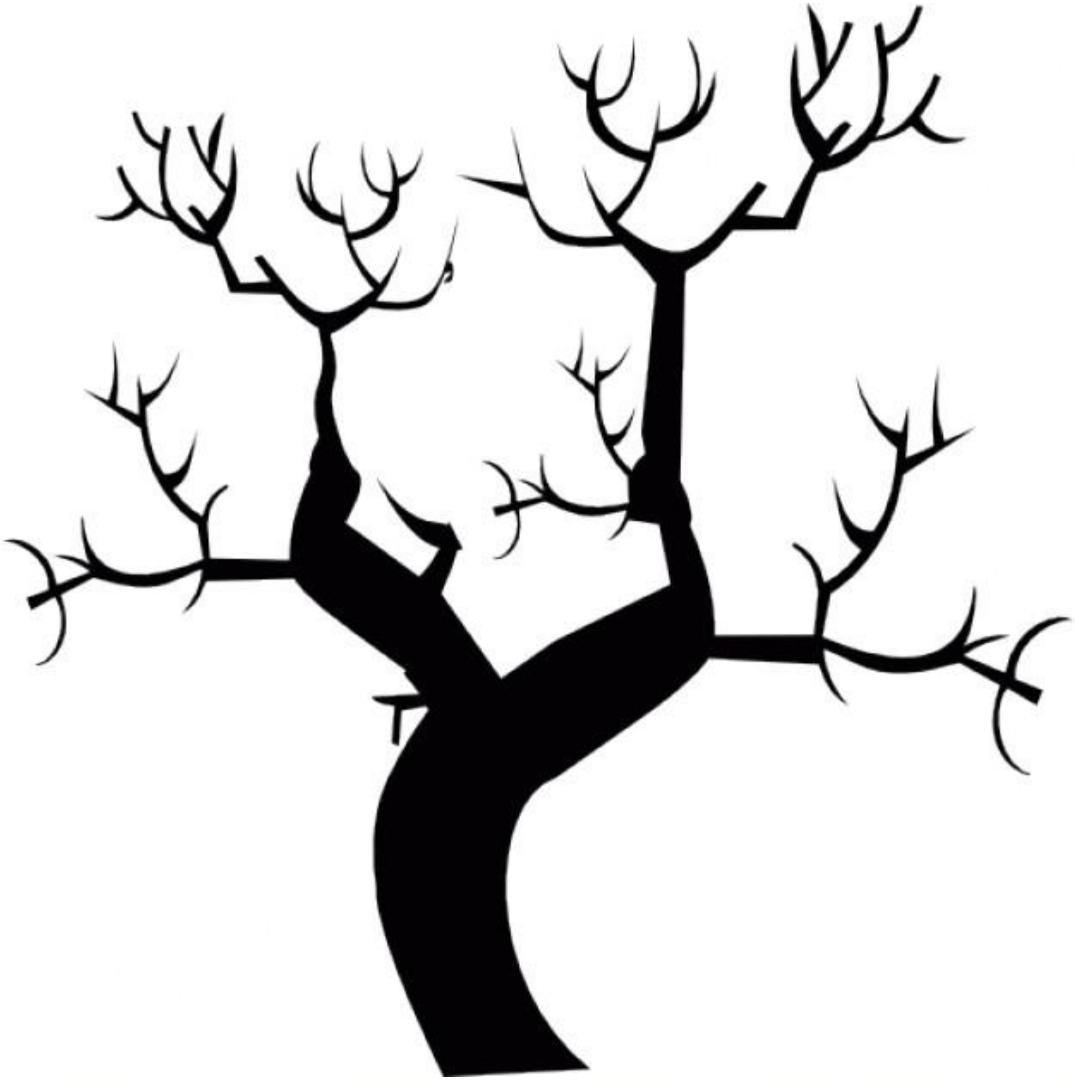
K = 1



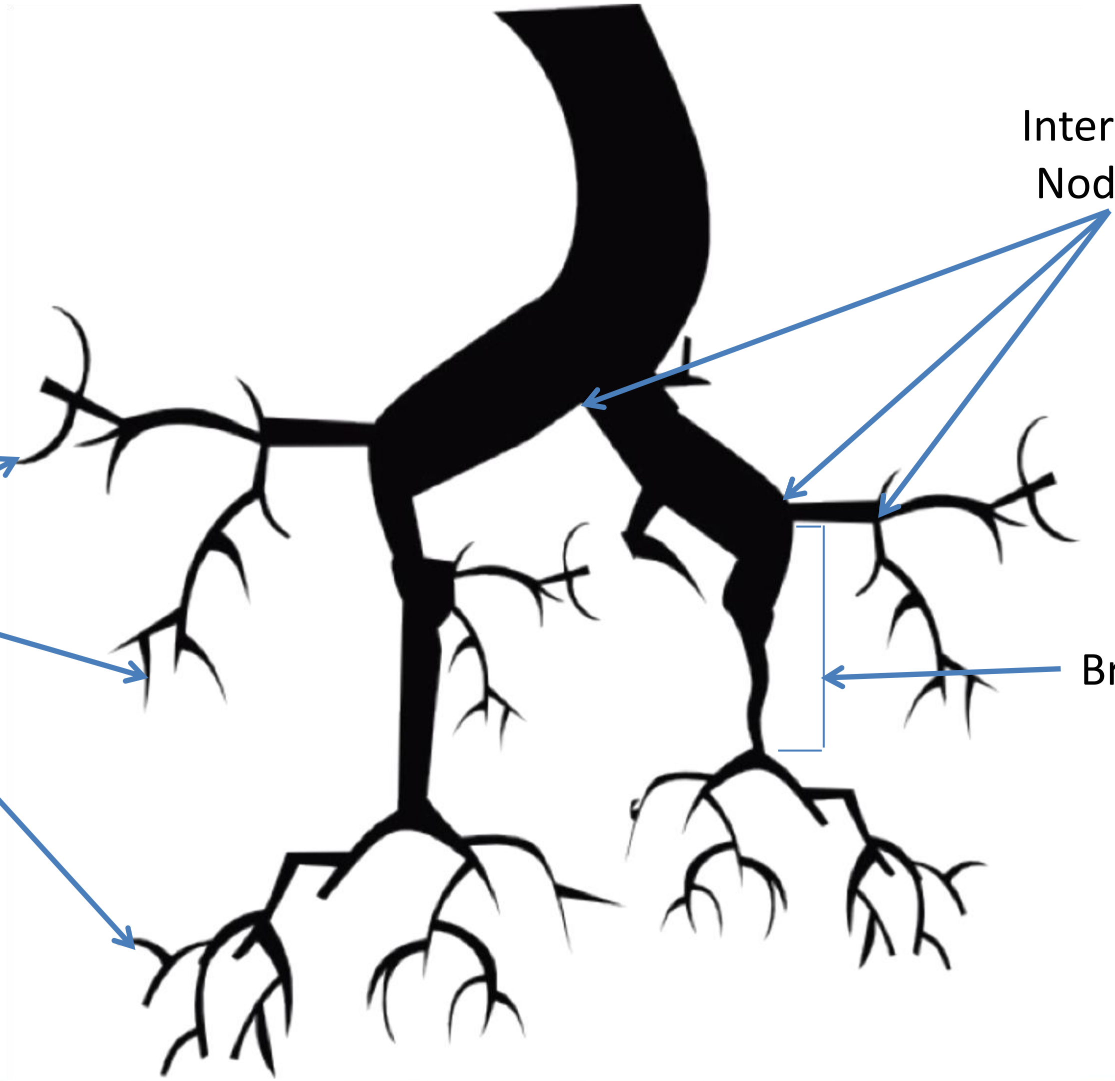


K = 100

Tree



To make a prediction we grow inverted Tree



Points where tree gets split

Internal Nodes

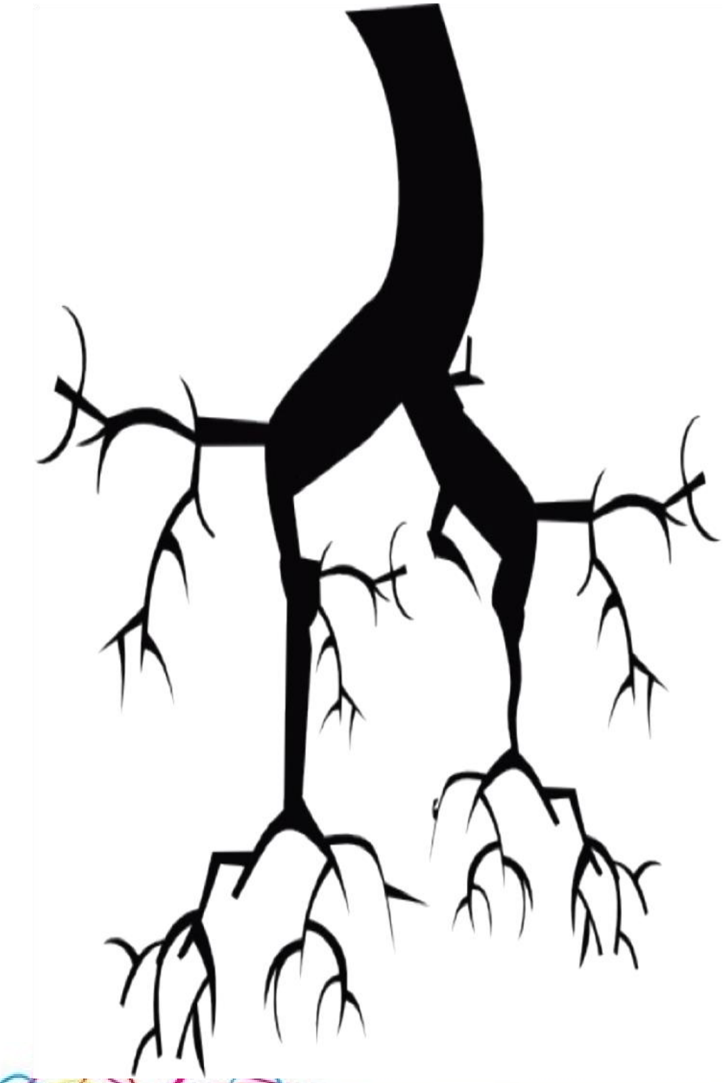
Also known as Terminal Nodes

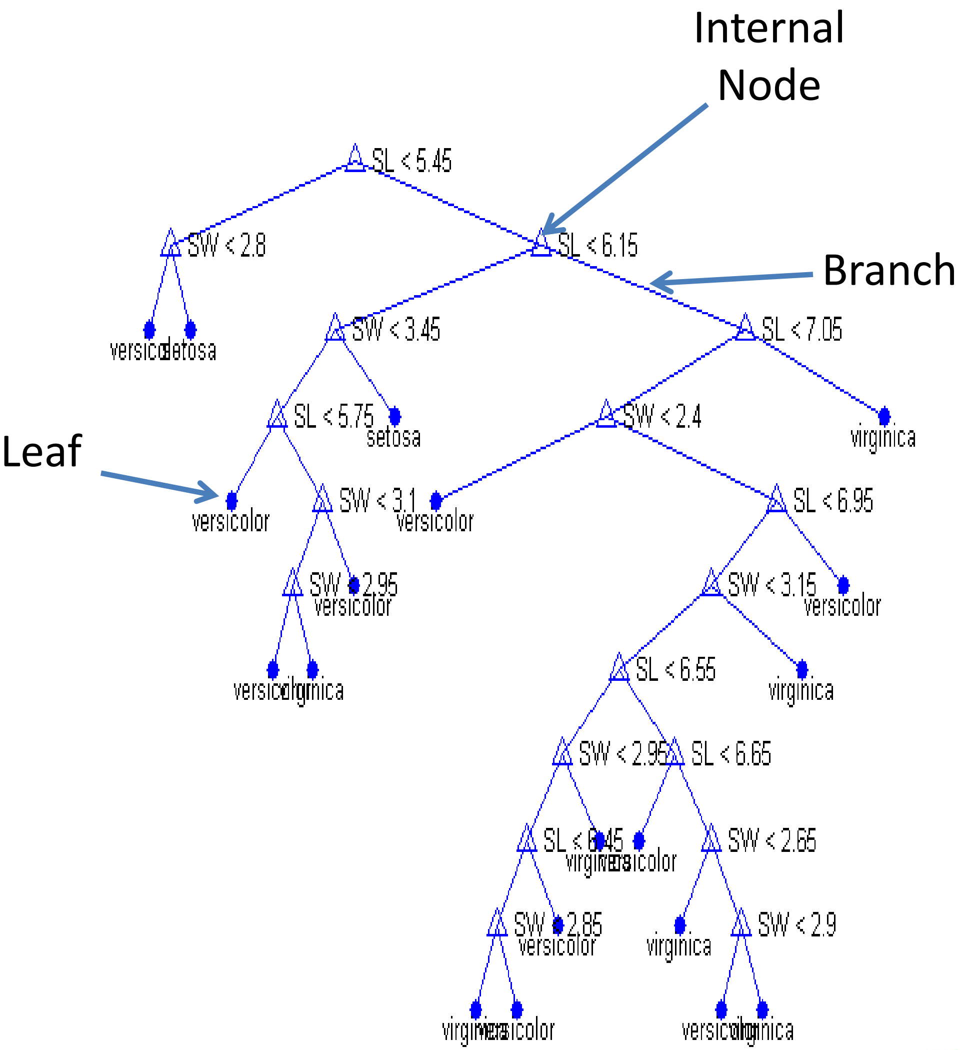
Leaves

Branch

Connect two nodes

Decision Tree





Internal Node

Branch

Leaf

Predictor Space and Decision Tree

Predictor Space: The space consisting of set of possible values for all the Predictors

Tree based method partition the Predictor space into a number of simple regions.

In order to make prediction for a given observation, we first find the region to which it belongs. Then we typically use:

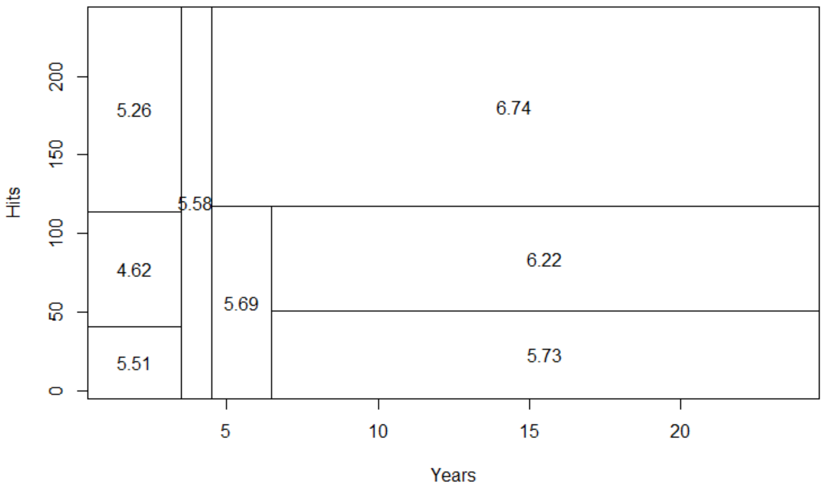
a) Mean or

b) Mode

of the training observations in the region.

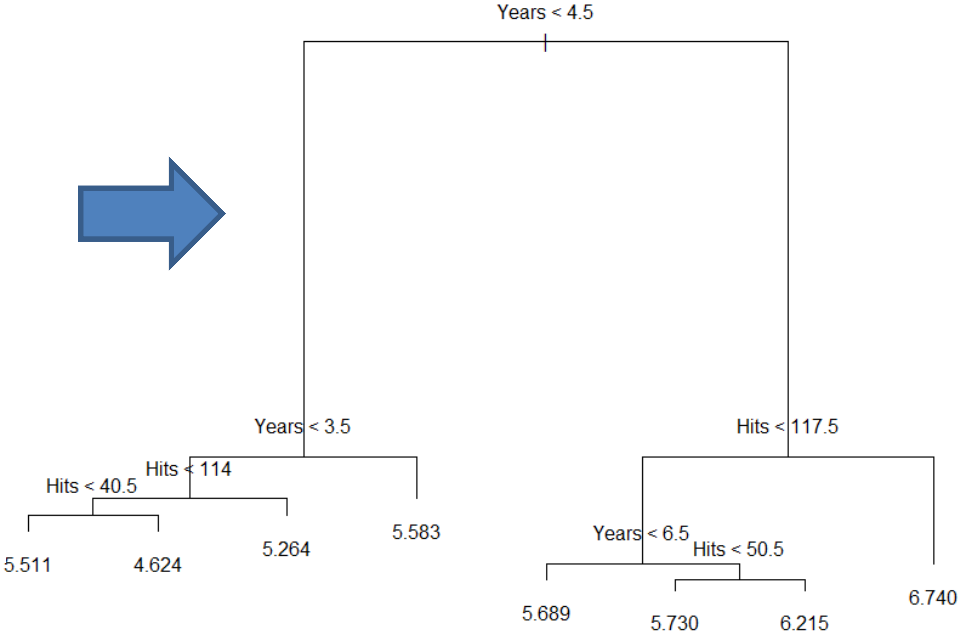
Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as Decision Tree methods.

Partitioned Predictor Space and Decision Tree



Partitioned Predictor Space

(Output of recursive binary splitting)



Decision Tree

Each partitioned region corresponds to one leaf in decision tree. The predictor on which first split is made is the most important factor followed by second and so on.

Decision Trees

Decision Trees can be applied to both:

a) Regression Problems – For predicting Quantitative response

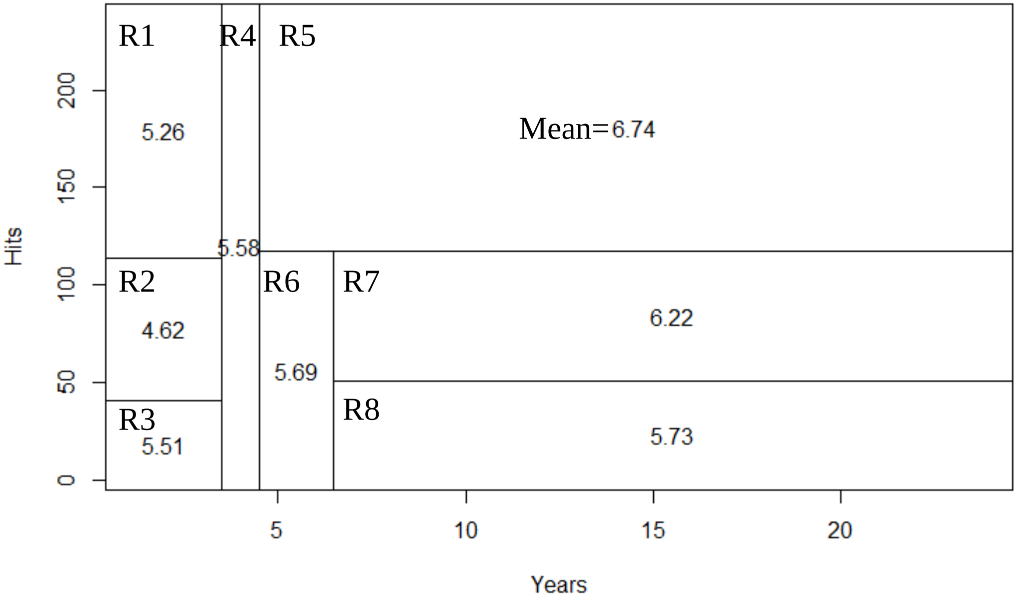
b) Classification Problems – For predicting Qualitative response

Regression Trees

Building Regression Trees:

a) Divide the predictor space into J distinct and non-overlapping regions R1, R2, R3, ….., Rj

b) For every observation that falls into the region Rj, we make the same prediction which is the mean of the response values for the training observations in Rj



Constructing Regions

How do we construct the regions R1, R2,…Rj?

We divide the predictor space into J high dimensional rectangles or boxes so that Residual Sum of Square (RSS) is minimized.

It is computationally infeasible to consider every possible partition of the feature space into J boxes

So we take a top down and greedy approach that is known as recursive binary splitting.

Constructing Regions (Contd.)

The approach is top down because it begins at the top of the tree. At this point all observations belong to a single region.

Then we successively splits the predictor space, each split is indicated via two new branches further down the tree.

It is greedy because at each step of the tree building process, the best split is made at this particular step instead of looking ahead.

Classification Trees

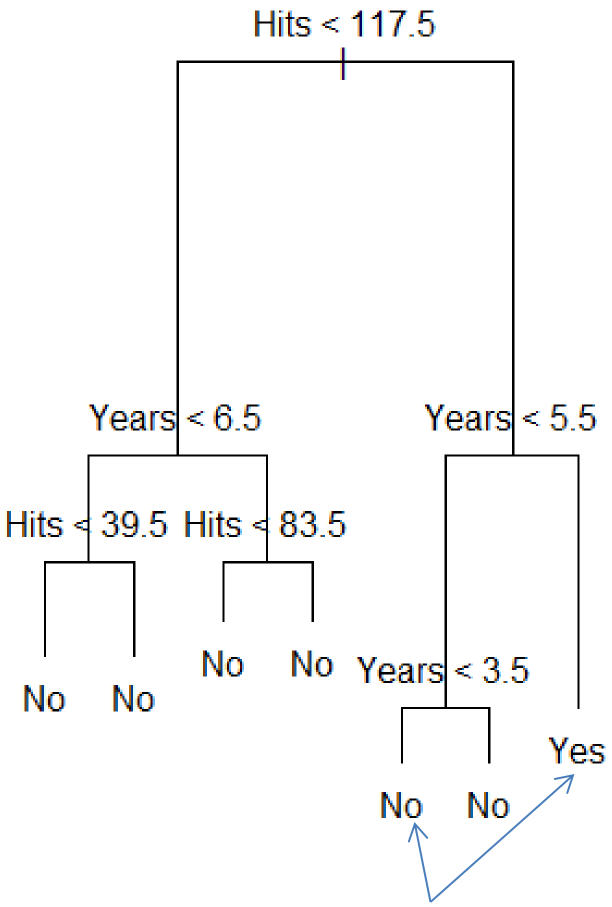
Classification tree is similar to Regression tree.

It is used to predict a qualitative response rather that quantitative response

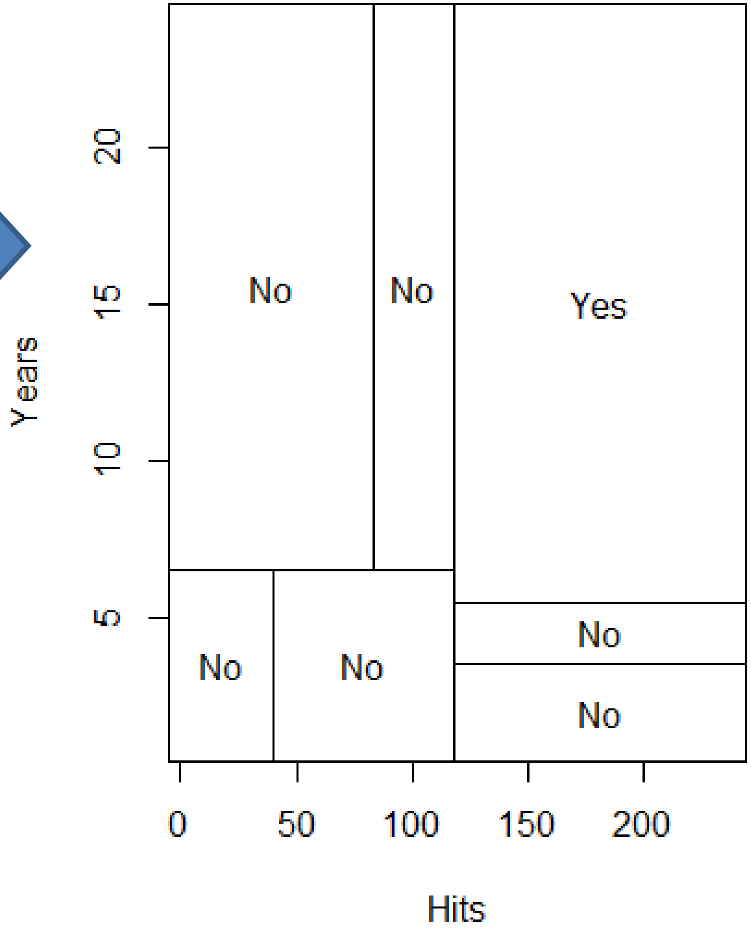
For classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.

Binary splitting is done on classification error rate which is minimized.

Classification Trees



Qualitative Response



Growing Classification Trees

Classification error rate is the fraction of the training observations in that region that do not belong to the most common class

Classification error rate is found to be not sufficiently sensitive for tree growing (making binary split) hence two other measures are preferred:

a) Gini Index
b) Cross Entropy

These will take near zero value if all the given class proportions has value near zero or one.

These are measure of node purity. A small value indicates that a node contains predominantly observations from a single class.

When building a classification tree, either the Gini Index or the Cross Entropy are typically used to evaluate the quality of a particular split

Classification Trees - Notes

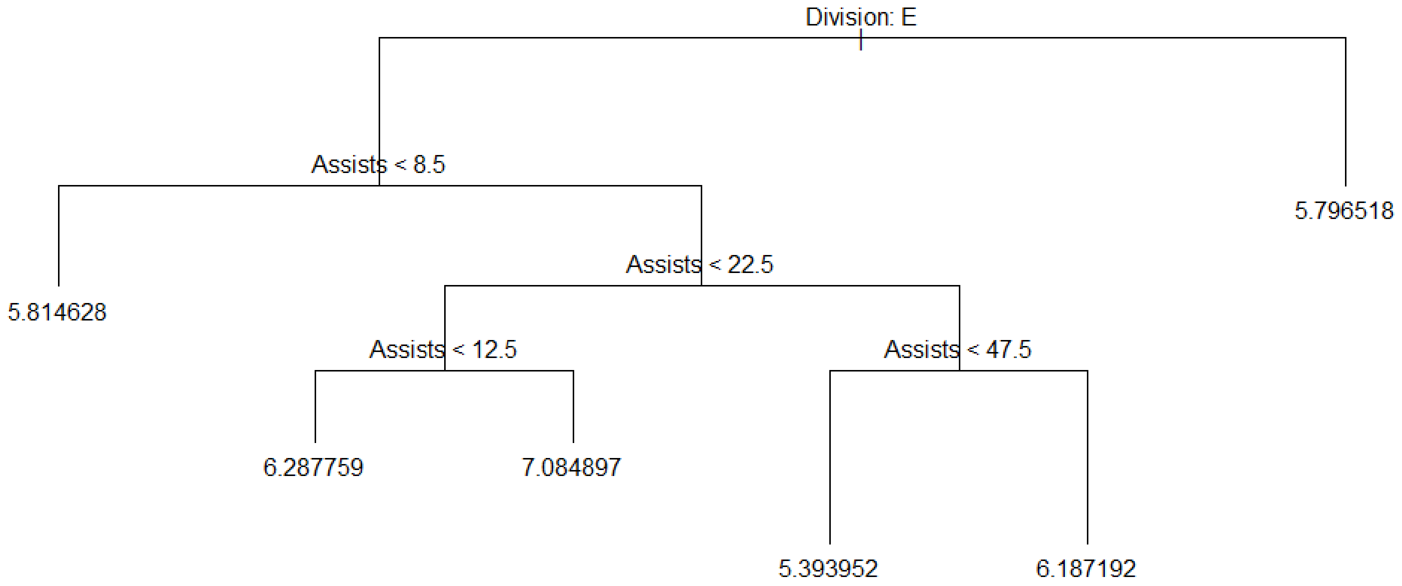
Some of the splits yield two terminal nodes that have the same predicted value.

The split is performed because it leads to increased node purity.

Even though such split does not reduce the classification error, it improves the Gini Index and the cross-entropy which is more sensitive to node purity.

Trees with Qualitative Predictors

- Decision tree can be constructed even in presence of qualitative predictor variables.
- A split on one of these variables amounts to assigning some of the quantitative values to one branch and remaining to the other branch.



Advantage-Disadvantage of Decision Trees

Decision Trees are simple and useful for interpretation and easy to explain.

Decision Trees more closely mirror human decision making

It has a nice graphical representation and are easily interpreted by non experts.

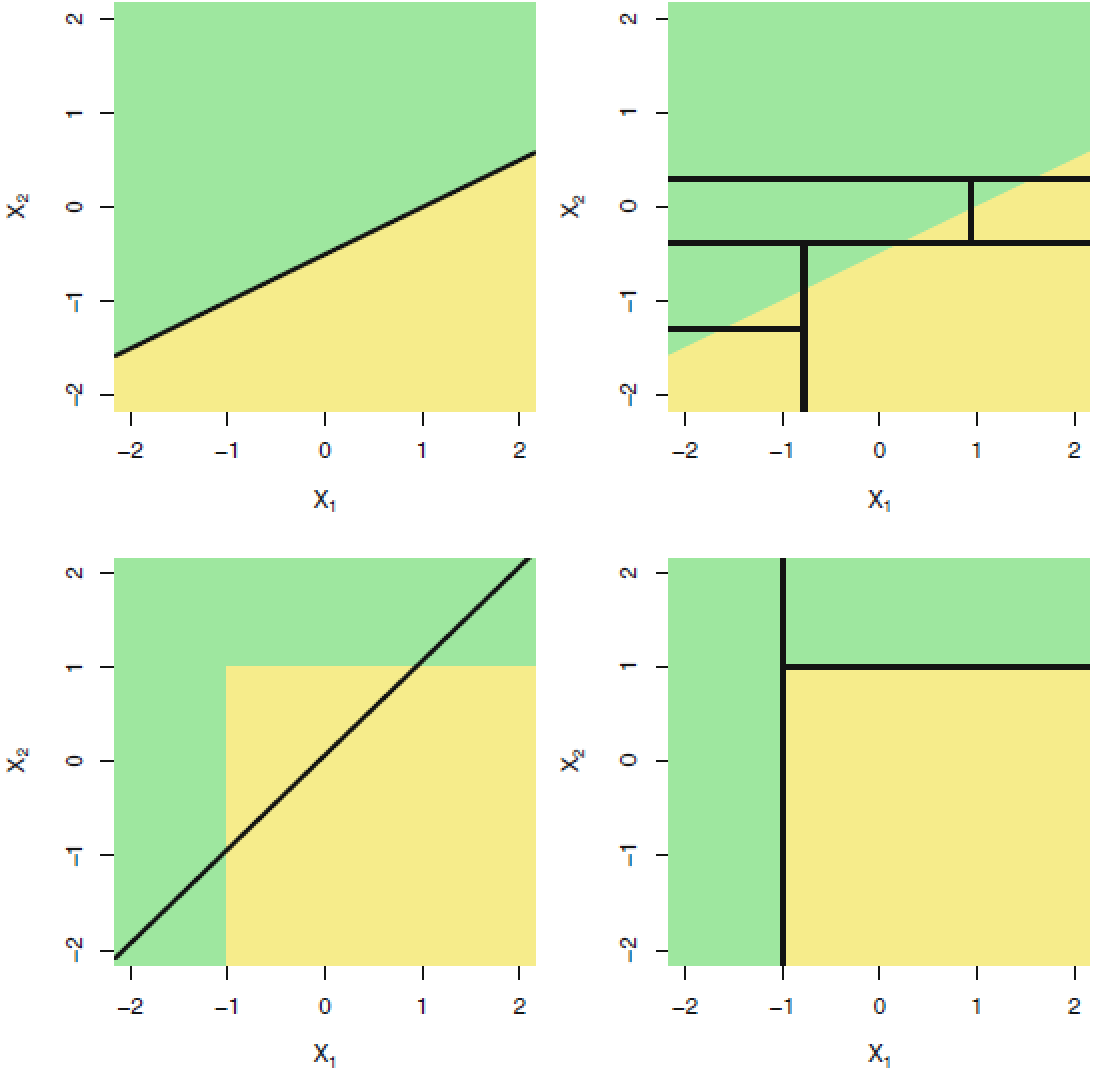
Trees can easily handle qualitative predictors.

However, they are usually not competitive with the best supervised learning approaches in terms of prediction accuracy.

Decision Tree vs Linear Model

- Which model is better: Regression and Classification Tree OR Regression and Classification Linear Model?
- It depends on problem at hand
- If the relationship between the predictors and response is well approximated by a liner model then the linear approach will work better and will outperform regression tree
- The reason being regression tree does not exploit the linear structure present in the relationship
- On the other hand, if there is highly non-linear and complex relationship between predictors and response then decision tree may outperform linear approaches
- The relative performance can be assessed by estimating the test error by validation set or cross validation approach

Decision Tree vs Linear Model


Linear Model

Tree Based Model

Linear Boundary

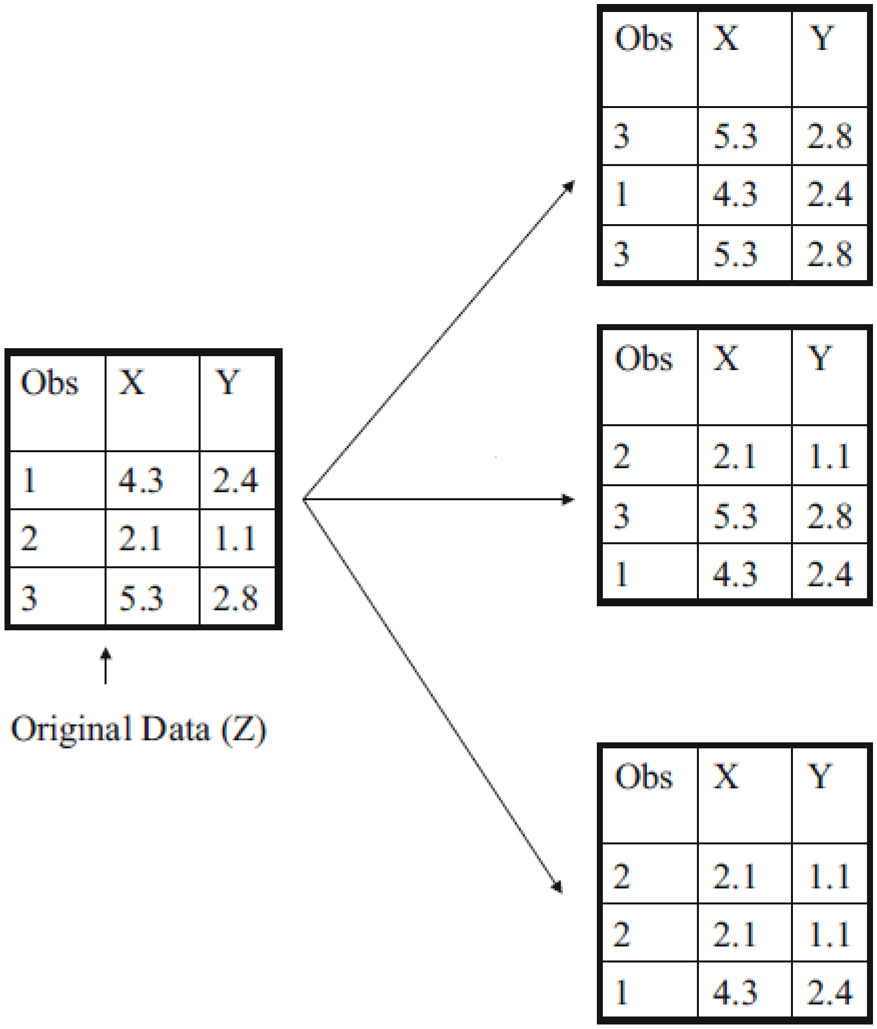
Non Linear Boundary

Prediction using multiple trees

- Alternate approach of combining large number of trees is adopted to improve prediction accuracy at the expense of some loss of interpretation
- This approach involves producing multiple trees and combining them to yield a single consensus prediction
- Here we simply construct B regression trees using B Bootstrapped training sets (because we generally do not have access to multiple training sets)
- Then we average the resulting predictions reducing the variance in prediction
- This approach is known as Bagging (Bootstrap Aggregation)

The Bootstrap

- In Bootstrap, rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set.
- The sampling is performed with replacement which means that the same observation can occur more than once in the bootstrap data set.
- It can be used to quantify the uncertainty associated with a given estimator or learning method (e.g. SE)



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

Random Forest

Suppose there is one strong predictor in data set along with some moderately strong predictors.

Then in the collection of multiple trees, most or all of the trees will use this strong predictor in top split.

Consequently all trees will look quite similar to each other.

Hence prediction from the such trees will be highly correlated

Random Forest

Here we build a number of trees on training samples

While building these trees, at each split, a random sample of m predictors is chosen as candidate amongst full set of p predictors.

Typically $m \sim \sqrt{p}$, and the split is allowed to use only one of these m predictors

A fresh sample of m predictors are chosen for next split

Thus in building a random forest, at each split of the tree, the algorithm is not even allowed to consider a majority of the available predictors thus de-correlating the trees

If random forest is built with $m=p$ then it amounts to Bagging

Out of Bag Error Estimate

It is straightforward way of estimating test error of Bagged/ Random Forest model without CV or Validation set approach

On an average each Bagged tree make use of 2/3 of observations. Remaining 1/3 of the observations not used to fit a given bagged tree are referred to as Out-of-bag (OOB) observations

We can predict the response for the *i*th observation using each of the trees in which that observation was OOB, yielding B/3 predictions. B is the number of Bootstrapped training sets, used to construct B trees.

We can average/take majority vote for these predicted responses which leads to a single OOB prediction for *i*th observation

Overall OOB MSE or Classification error can be computed using predicted responses for all n observations

The resulting OOB error is a valid estimate of the test error of the bagged model, since response for each observation is predicted using only the trees that were not fit using that observation

Variable Importance

However Bagging results in improved accuracy over prediction using a single tree, it can be difficult to interpret the resulting model

When we bag a large number of trees, it is no longer possible to represent the model using a single tree

Hence it is no longer clear which variables are most important. Thus bagging improves prediction accuracy at the cost of model interpretability

We can obtain an overall summary of the importance of each predictor, by averaging over all B trees:

a) Total amount that RSS decrease due to splits over a given predictor

b) Total amount that Gini index is decreased by splits over a given predictor

A large value indicates an important predictor

Formulae

Classification error rate E = 1 – $\max_k(\hat{p}_{mk})$

$\hat{p}_{mk}$ represents the proportion of training observations in the m^{th} region that are from k^{th} class

Deviance = $-2\ \sum_m\ \ \sum_k n_{mk} log\ \hat{p}_{mk}$

n_{mk} is the no of observations in the m_{th} terminal node that belongs to k_{th} class

A small deviance indicates a tree that provides a good fit to the (training) data.

Residual Mean Deviance = Deviance / (n – $|T_0|$)

n is total no of observations and T_0 is terminal nodes of the fully grown tree

Gini Index G = $\sum_{k=1}^{K} \hat{p}_{mk}\,(1 - \hat{p}_{mk})$

It is a measure of total variance across k classes

Cross Entropy D = - $\sum_{k=1}^{K} \hat{p}_{mk}\, log \hat{p}_{mk}$

Gini Index and Cross Entropy will take small values if m^{th} node is pure

Deviance (root) = RSS root =$\sum_{i=1}^{n}$ (yi – mean(y))^2

Deviance (final) = RSS final =$\sum_{i=1}^{n}$ (yi –$\hat{y}$)^2

Thank You

Abhinav Srivastava