Decision Tree & Random Forest Models

Presented by: Derek Kane

Overview of Topics

- Introduction to Decision Trees
- CART Models
- Conditional Inference Trees
- ID3 and C5.0
- Random Forest
- Model Evaluation Metrics
- Practical Example
  - Diabetes Detection
  - Customer Churn

Introduction to Decision Trees

- Decision Trees or recursive partitioning models are a decision support tool which uses a tree like graph of decisions and their possible consequences.

- A Decision Tree creates a type of flowchart which consists of nodes (referred to as "leafs") and a set of decisions to be made based off of node (referred to as "branches").

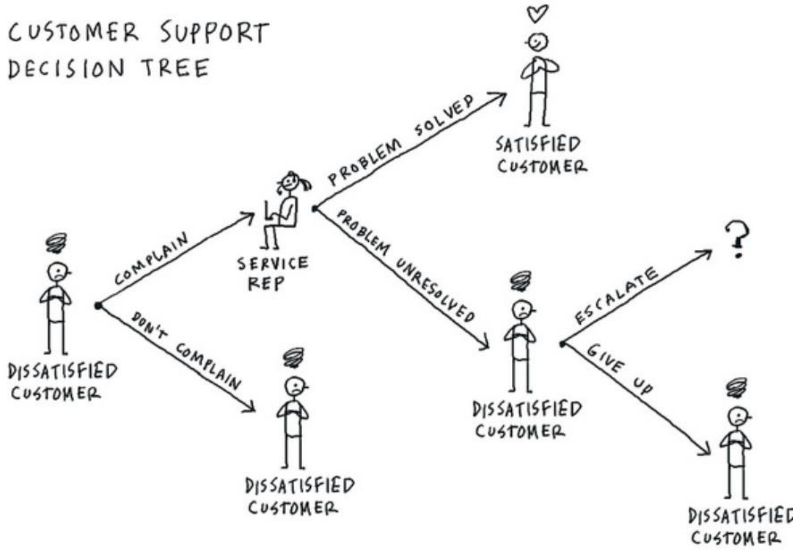
- The leaf and branch structure forms a hierarchical representation that mimic the form of a tree.

- Decision Tree Learning is one of the most widely used and practical methods for inductive inference and is an important tool in machine learning and predictive analytics.

Introduction to Decision Trees

Here are some applications of decision trees:

- Manufacturing- Chemical material evaluation for manufacturing/production.
- Production- Process optimization in electrochemical machining.
- Biomedical Engineering- Identifying features to be used in implantable devices.
- Astronomy- Use of decision trees for filtering noise from Hubble Space Telescope images.
- Molecular biology- Analyzing amino acid sequences in the Human Genome Project.
- Pharmacology- Developing an analysis of drug efficacy.
- Planning- Scheduling of printed circuit board assembly lines.
- Medicine- Analysis of the Sudden Infant Death Syndrome (SIDS).



Introduction to Decision Trees

Advantages of Decision Trees:

- They are simple to understand and interpret. People are able to understand decision tree models after a brief explanation.
- Have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities, and costs) and their preferences for outcomes.
- The algorithms are robust to noisy data and capable of learning disjunctive expressions.
- Help determine worst, best and expected values for different scenarios.

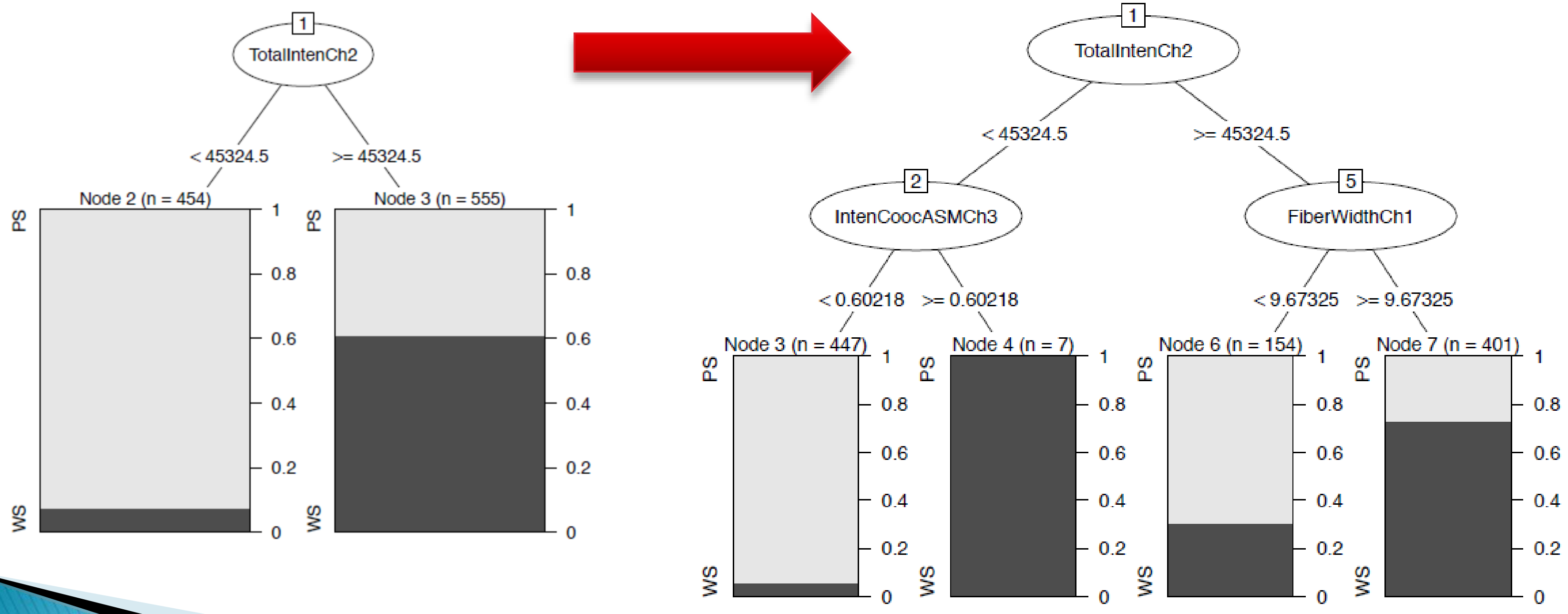
Disadvantages of Decision Trees:

- For data including categorical variables with different number of levels, information gain in decision trees are biased in favor of those attributes with more levels.
- Calculations can get very complex particularly if many values are uncertain and/or if many outcomes are linked.

Introduction to Decision Trees

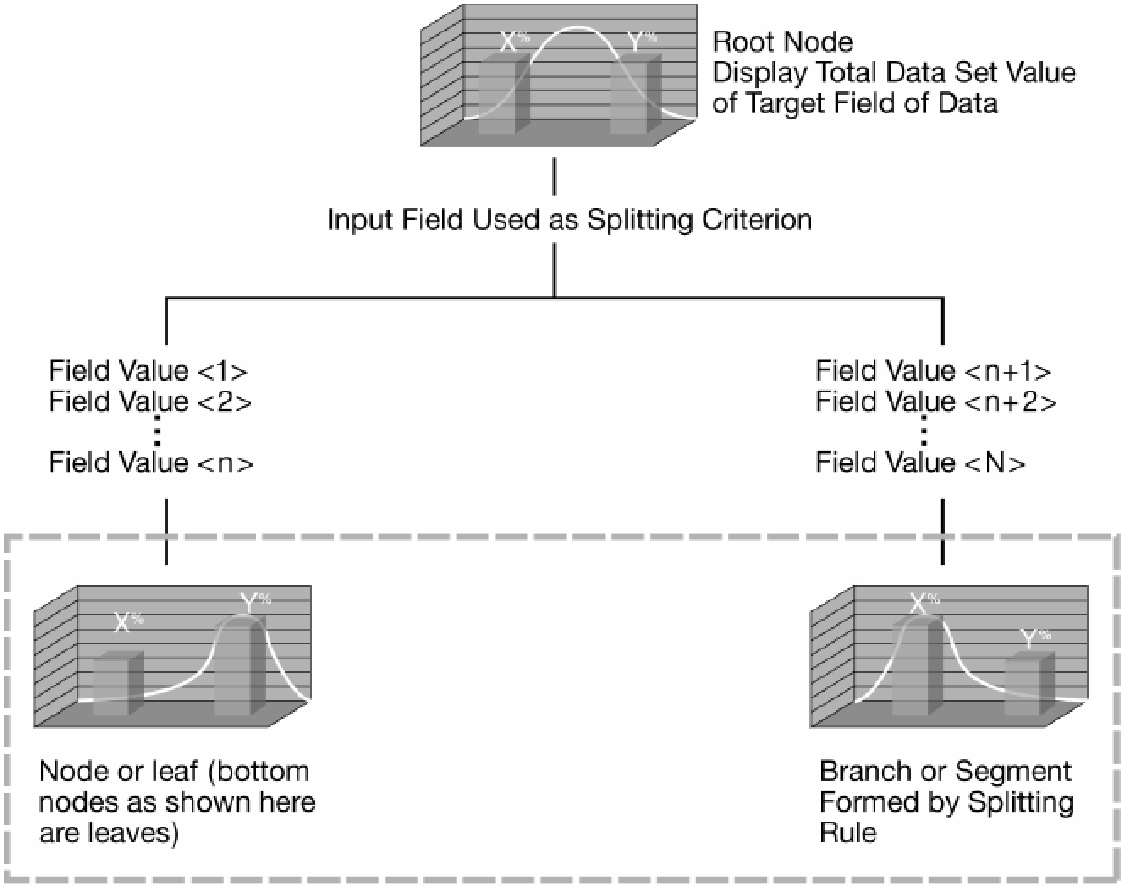
- A classification tree searches through each independent variable to find a value of single variable that best splits the data into 2 (or more) groups.

- Typically, the best split minimizes impurity of the outcome in the resulting data subsets. For these 2 resulting groups, the process is repeated until a stopping criteria is invoked (Ex. Minimum number of observations in a node).

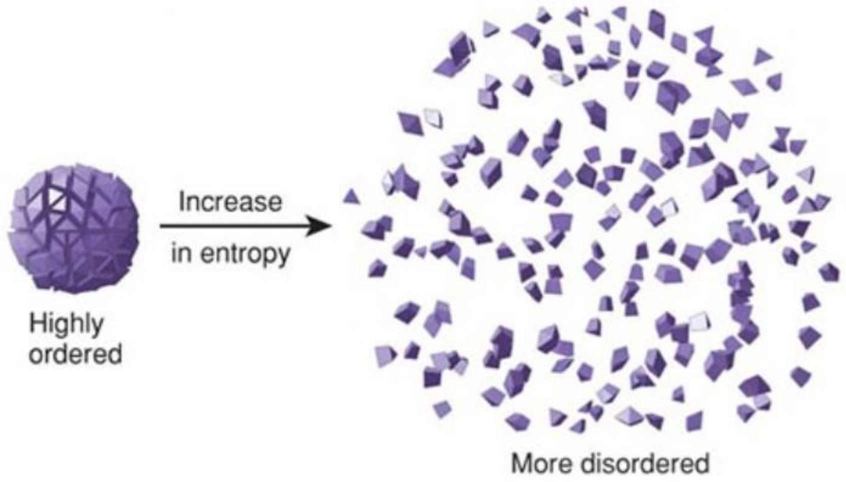


Introduction to Decision Trees

- This graphic depicts how a classification decision tree is constructed.



Introduction to Decision Trees



Examples of Splitting Criterion:

- Minimum number of observations in a node.
- Information Gain – Impurity based criterion that uses the entropy measure (information theory) as the impurity measure.
- Gini Index – Impurity based criterion that measures the divergences between the probability distributions of the target attribute's values.
- Gain Ratio – Normalizes the information gain by taking the information gain divided by the entropy.

Introduction to Decision Trees

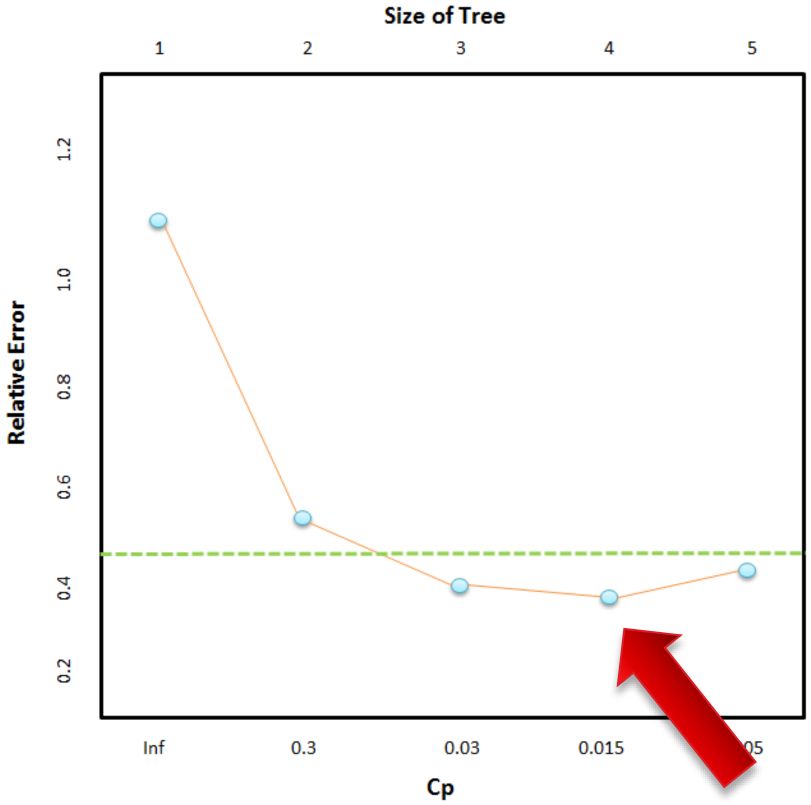
- Employing rigid stopping criteria tends to create small and under-fitted decision trees.

- On the other hand, using loosely stopping criteria tends to generate large decision trees that are over-fitted to training set.

- Pruning methods originally suggested by Brieman were developed to solve this dilemma.

- The methodology allows for a decision tree to first use a loose stopping criterion. After the tree is grown, then it is cut back into a smaller tree by removing sub branches that are not contributing to the generalization accuracy.

Introduction to Decision Trees



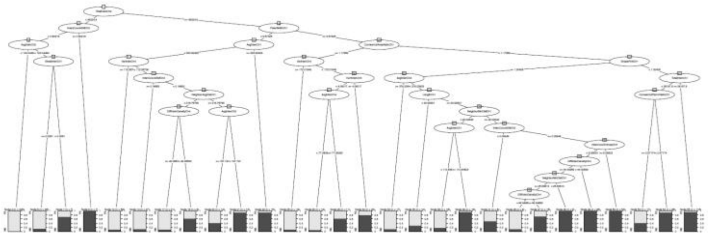
- Trees are typically indexed by their depth and the classical decision tree methodology uses the cost-complexity parameter (C_p) to determine the best tree depth.

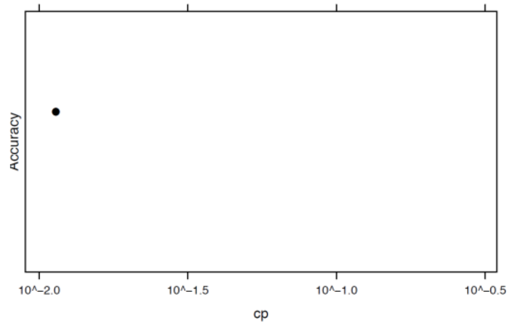
- The intention is to identify the lowest C_p value which guides us to the appropriate tree size.

- In this example, a tree of size 4 has the lowest C_p value. However, the principle of parsimony tells us that a tree of size 3 would also be adequate.

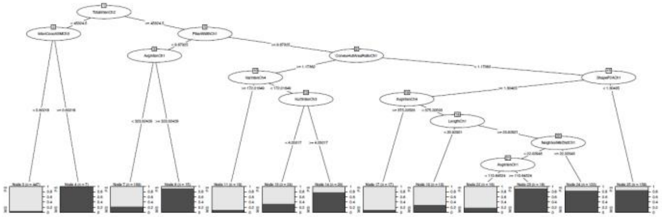
Introduction to Decision Trees

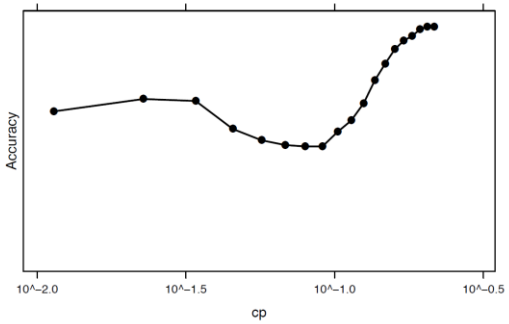
Full Decision Tree



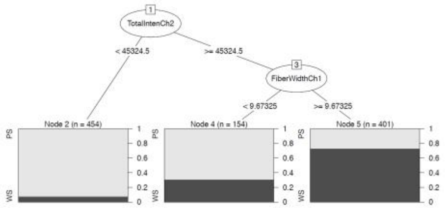


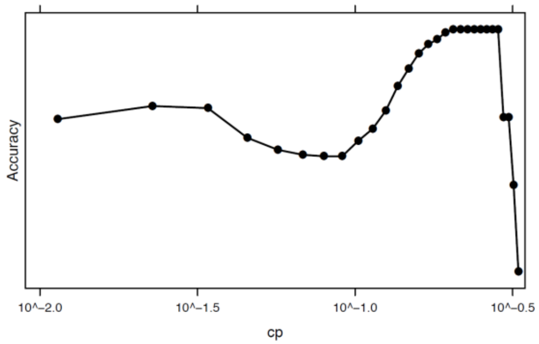
Start Pruning





Too Much!!!





Types of Decision Trees

- There are a number of different types of decision tree algorithms that can be used for machine learning. The calculation process utilizes recursive partitioning algorithms to create the splits.

- These types of models are commonly referred to as CART models or classification and regression trees.

- We need to first look at the characteristics of the dataset through our EDA procedure and assess the dependent variable.

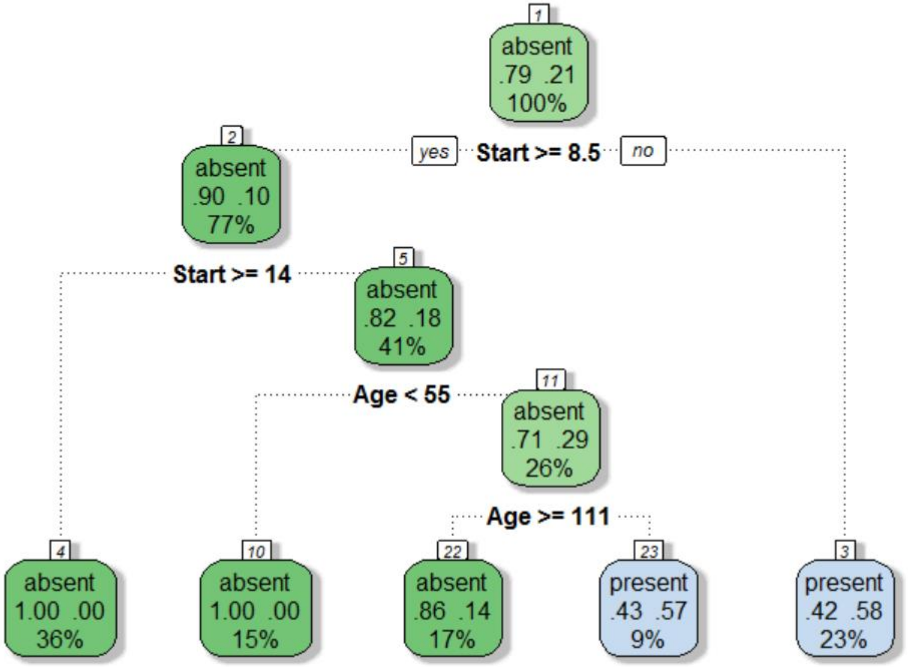
- Is this variable categorical or continuous in nature? If we are predicting a categorical (classification tree) or continuous (regression tree) outcome, then we have different algorithms which can be employed.

CART Modeling - Classification

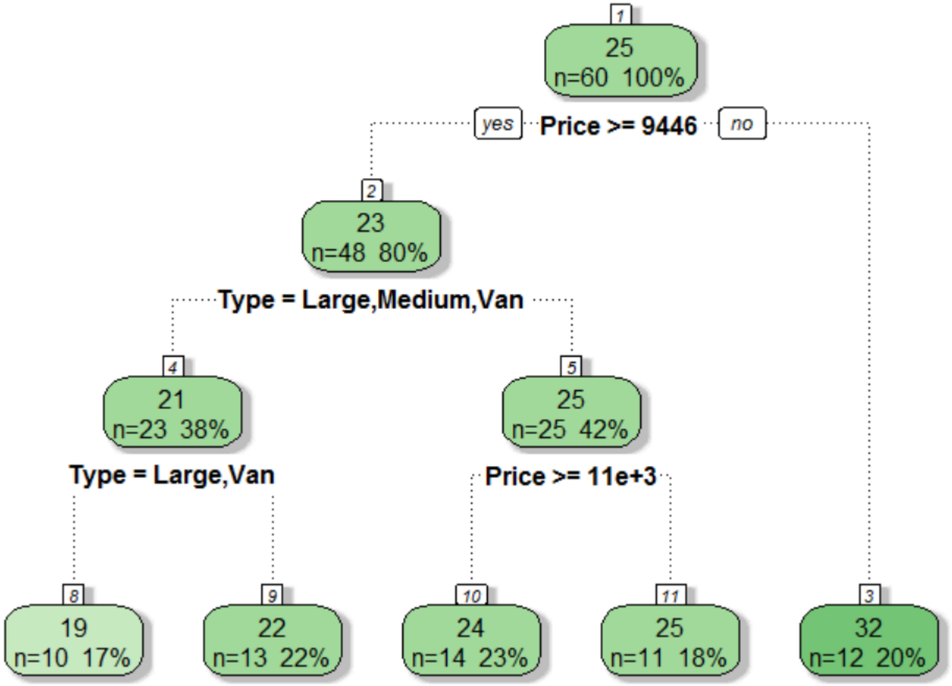
- This modeling approach is used to determine outcomes in a categorical nature.

- This could be a variable with a dichotomous outcome (Yes/No or 0/1) or with multiple category levels (high/medium/low).

- The example shows a CART model applied in the medical field that is detecting whether a type of deformation (kyphosis) is present or absent after surgery.



CART Modeling - Regression



- This modeling approach is used to determine outcomes in a continuous nature.
- This could be a dependent variable with a range of values from -∞ to +∞.
- The example shows a CART model applied in the automotive industry that is looking how variables such as price, car type, reliability, and country of origin are related to MPG.
- The MPG variable can take a value of 0 to ∞.

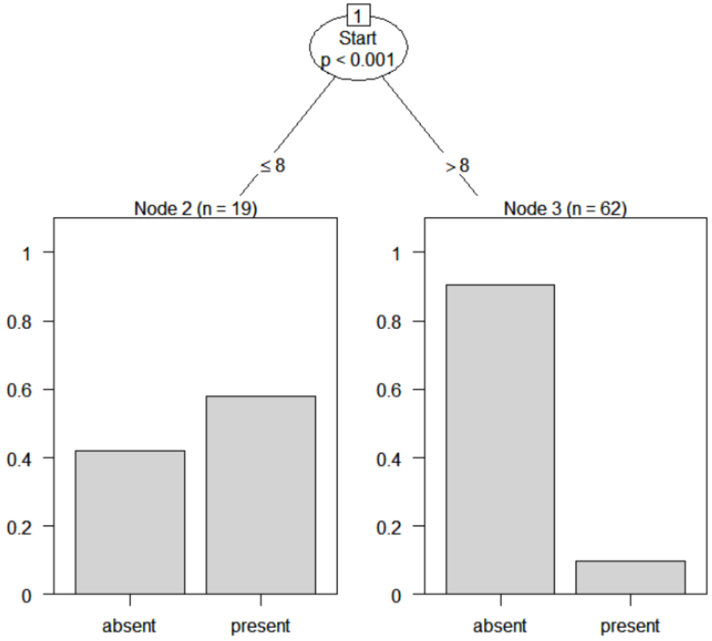
Conditional Inference Trees

- The manner in which the CART models decides which variable to include at each branch can potentially create a variable selection bias.

- A conditional inference tree uses a significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure.

- This conditional inference approach utilizes a covariate selection scheme that is based on statistical theory.

Conditional Inference Tree for Kyphosis



ID3 and C5.0

- The ID3 algorithm uses information gain as the splitting criteria.
  - The growing stops when all instances belong to a single value of target feature or when best information gain is not greater than zero.
  - ID3 does not apply any pruning procedures nor does it handle numeric attributes or missing values.

- C.50 is the evolution of the ID3 and was developed by the same author, Qunilin.
  - This approach uses the gain ratio as splitting criteria.
  - The splitting ceases when the number of instances to be split is below a certain threshold. Error-based pruning is performed after the growing phase.
  - The algorithm can handle numeric attributes.
  - It can induce from a training set that incorporates missing values by using corrected gain ratio criteria.
  - The approach contains built in features to handle resampling and boosting.

- The C.50 algorithm has been in used successfully in commerce for decades. Only in recent years had the proprietary source code been released to the general public.

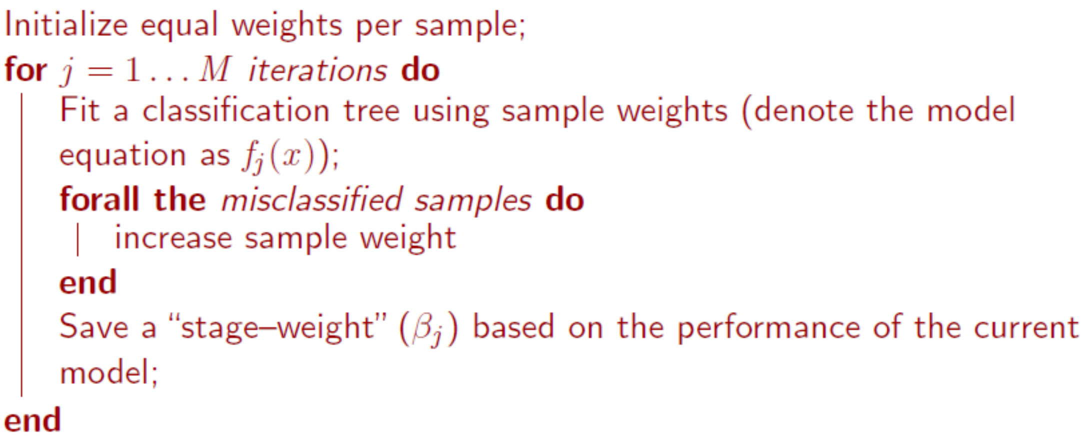
Pros and Cons of Single Trees

- Trees can be computed very quickly and have simple interpretations.
- Also, they have built-in feature selection; if a predictor was not used in any split, the model is completely independent of that data.
- Unfortunately, trees do not usually have optimal performance when compared to other methods.
- Additionally, small changes in the data can drastically affect the structure of a tree.
- This last point has been exploited to improve the performance of the trees via ensemble methods where many trees are fit and predictions are aggregated across the trees. Examples include bagging, boosting, and random forests.

Boosting Algorithms

- A method used to “boost” single trees into strong learning algorithms.
- Boosted trees try to improve the model fit over different trees by considering past fits.
- There are many different approaches to boosting including adaBoost (binary response) and stochastic gradient boosting.

The basic tree boosting algorithm:



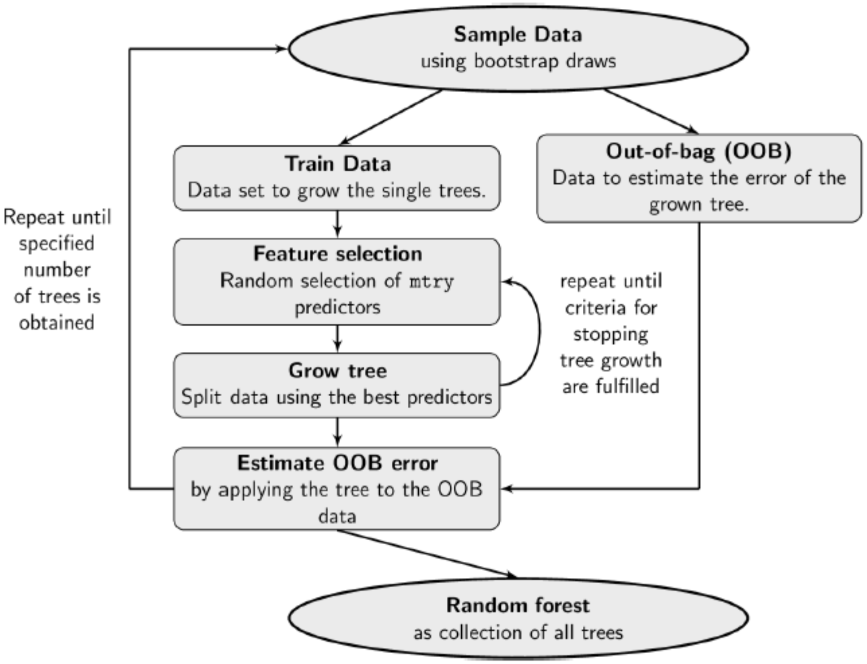
```
Initialize equal weights per sample;
for $j = 1 \ldots M$ iterations do
    Fit a classification tree using sample weights (denote the model
    equation as $f_j(x)$);
    forall the misclassified samples do
        increase sample weight
    end
    Save a “stage–weight” ($\beta_j$) based on the performance of the current
    model;
end
```

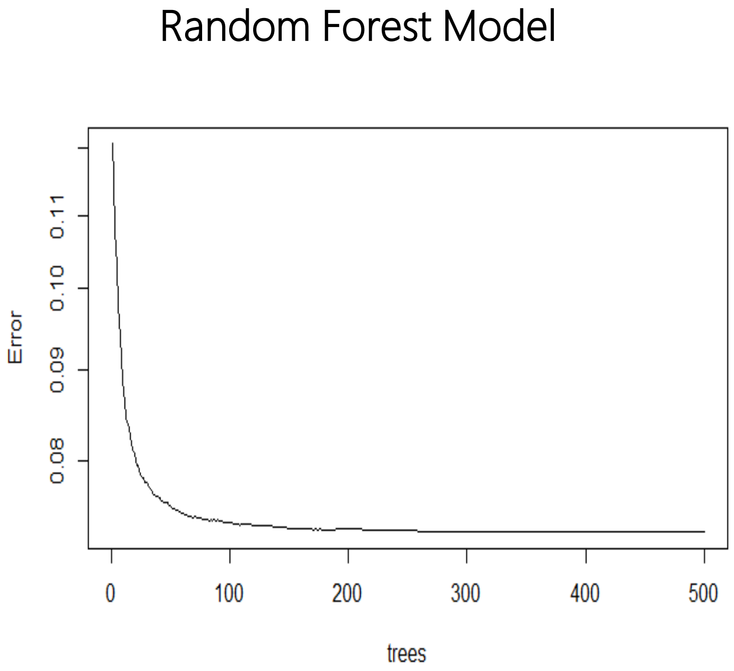
Random Forest Models

- A random forest algorithm takes the decision tree concept further by producing a large number of decision trees.
- The approach first takes a random sample of the data and identifies a key set of features to grow each decision tree.
- These decision trees then have their Out-Of-Bag error determined (error rate of the model) and then the collection of decision trees are compared to find the joint set of variables that produce the strongest classification model.

Random Forest Models

- An example of the process flow is depicted below.


Sample Data
using bootstrap draws
Train Data
Data set to grow the single trees.
Out-of-bag (OOB)
Data to estimate the error of the grown tree.
Repeat until specified number of trees is obtained
Feature selection
Random selection of mtry predictors
repeat until criteria for stopping tree growth are fulfilled
Grow tree
Split data using the best predictors
Estimate OOB error
by applying the tree to the OOB data
Random forest
as collection of all trees



Estimating Performance For Classification

- The most common application of decision trees is within classification. The manner in which we evaluate the performance of classification algorithms is distinctively different from continuous models (Ex. regression analysis).

- Because we know the correct class of the data, one common metric that is used is the overall predictive accuracy. However, this approach can sometimes be problematic when the classes are not balanced.

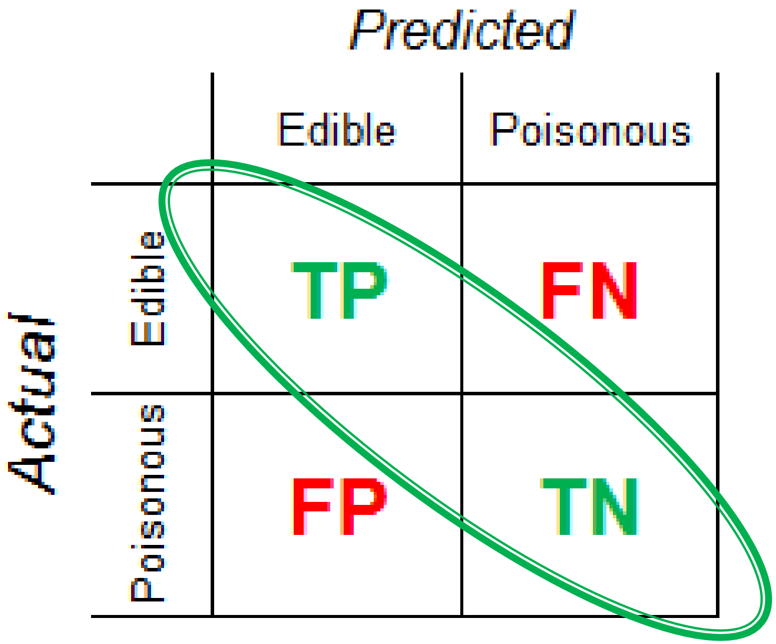
- The kappa statistic takes into account he expected error rate:

$$\kappa = \frac{O - E}{1 - E}$$

- where O is the observed accuracy, and E is the expected accuracy under chance agreement.

- For 2 class models, the Receiver Operating Characteristic (ROC) curves can be used to characterize model performance. (more on this later)

Confusion Matrix



		Predicted	
		Edible	Poisonous
Actual	Edible	TP	FN
	Poisonous	FP	TN

- A categorical variable (Ex. Dichotomous 0/1) can be represented in a cross tab which allows for us to see how well a model had performed.

- The left side of the matrix shows the actual scenario based on historical data and the top shows the predicted results of the model that we are evaluating.

- This is a handy way to show the True Positive (TP) , True Negative (TN), False Positive (FP), & False Negative (FN) rates of the models performance.

- The green circle is what is called the "off diagonal" and we want to see all of the values ordered on this plane.

Confusion Matrix, cont'd

- In order to perform an ROC analysis, we need to calculate some figures from the confusion matrix.

- It is highly unlikely that we will create perfect predictive models from the data we have available. There will be misclassifications and prediction errors which have to be considered in the performance evaluation.

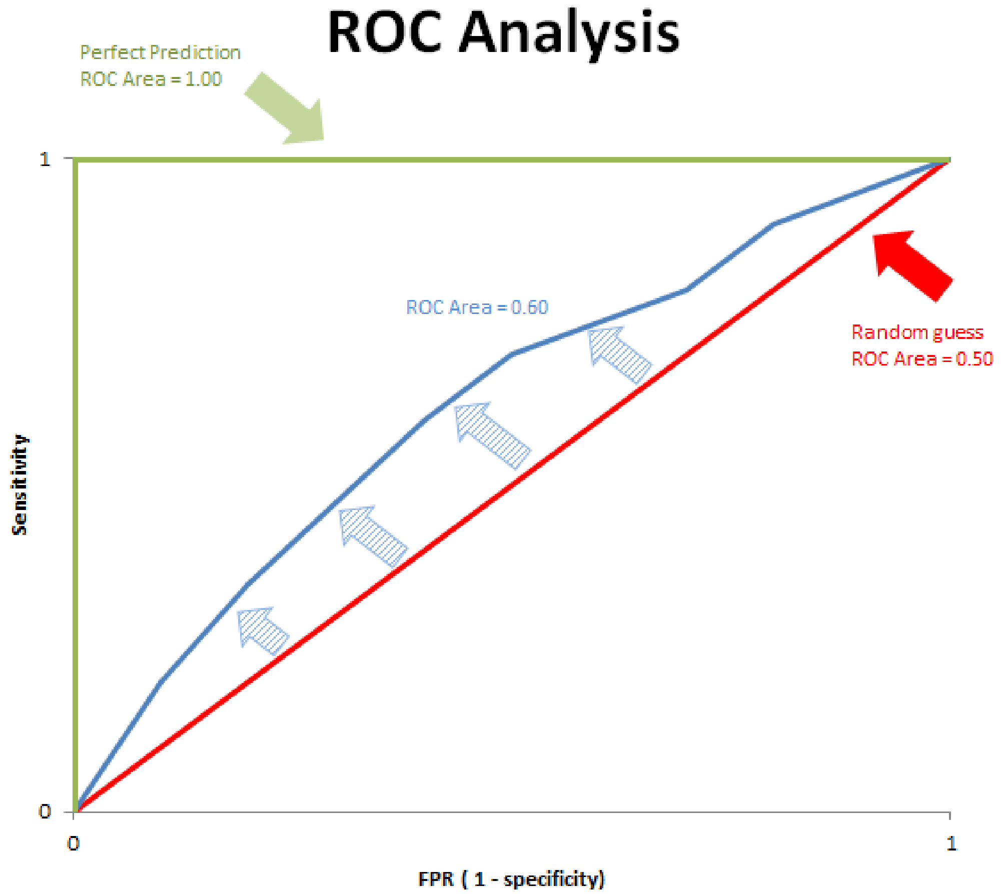
- Specificity and Sensitivity are statistical measures of the performance of a binary classification exercise and are a critical component to the ROC analysis. They represent aspects of the modeling error.

- Specificity = TN / (TN+FP)
  given that a result is not truly an event, what is the probability that the model will predict negative results.

- Sensitivity = TP / (TP + FN)
  given that a result is not truly an event, what is the probability that the model will predict event results.

- False Positive Rate = (1-Specificity)

Receiver Operator Curve (ROC)



- The ROC can be represented graphically in 2 dimensions as the relationship between the Sensitivity and FPR. This is typically referred to as the "lift".
- The AUC is a representation of the ROC in a single scalar.
- A perfect prediction will contain an AUC of 1.00 and a random guess will be 0.50. If your model is less than 0.50, then you are better off flipping a coin than using your model.

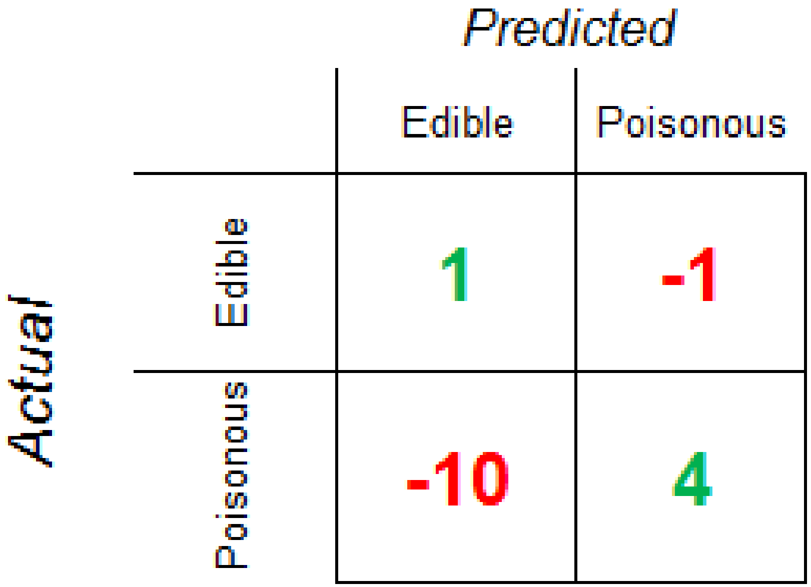
Cost Matrix

- Another interesting way to evaluate the strength of a model draws from the "confusion matrix" into what we call a "cost matrix".

- In real life, there is a cost for failure.

- If I eat a poisonous mushroom and think that it was edible, this may very well kill me. However, if I predict that a mushroom is poisonous when it is in fact edible, the cost is less severe.

- We can take the confusion/cost matrix and combine them to find models that extract the maximum value taking the cost of failure into account.



		Predicted	
		Edible	Poisonous
Actual	Edible	1	-1
	Poisonous	-10	4

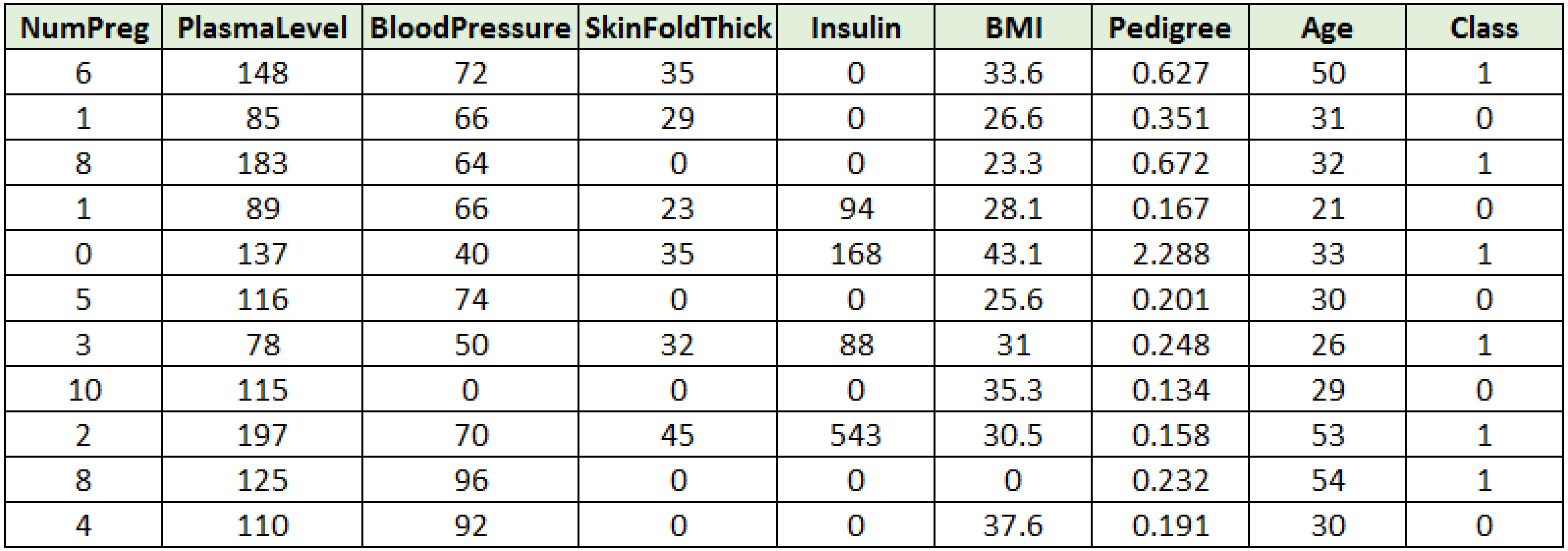
Practical Example
Diabetes Classification

Understanding the Data

- In the medical field, there are many applications of decision tree models which can aid in diagnosis and identification of treatment protocols.
- The dataset we will be working with contains information related to patients who have been diagnosed with Type II diabetes.
- The goal for this exercise will be to build two predictive models (CART and random forest) to assess the variables of related to predicting diabetes.

Understanding the Data

- Here is a view of the variables within the dataset:



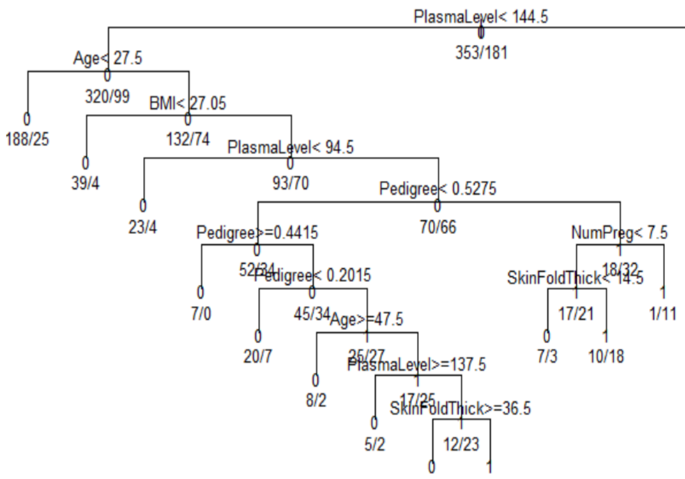
NumPreg	PlasmaLevel	BloodPressure	SkinFoldThick	Insulin	BMI	Pedigree	Age	Class
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0

- The dataset contains 768 observations and contains various measurements such as the number of pregnancies, BBMI, Age. The Class variable is the dependent variable and represents whether someone has diabetes or not.

CART Model

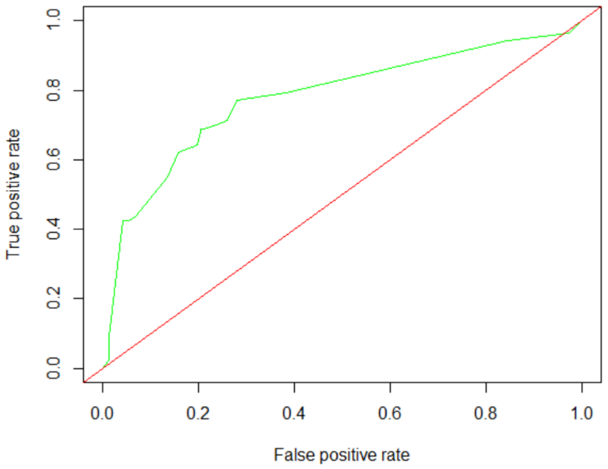
- After splitting the data into a training and testing set, we built a CART model without pruning on the training set and evaluated the performance on the test set.

Classification Tree for Diabetes



10 levels of depth; fairly complex

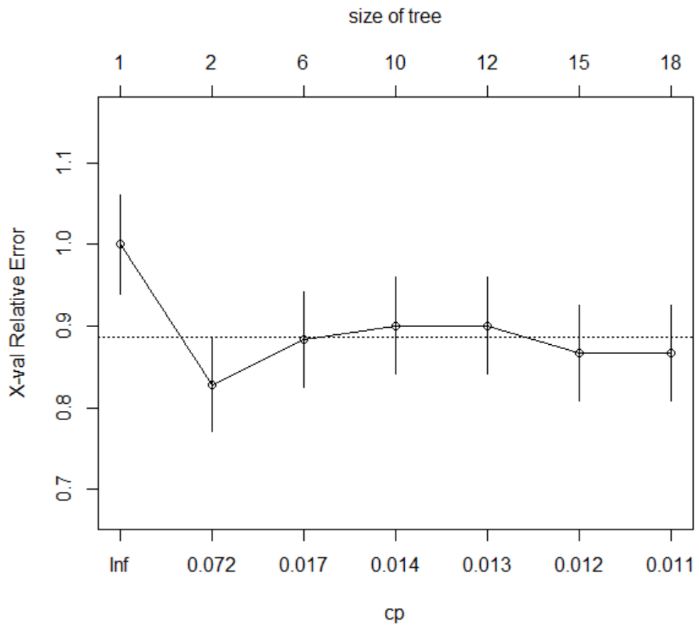
ROC Chart

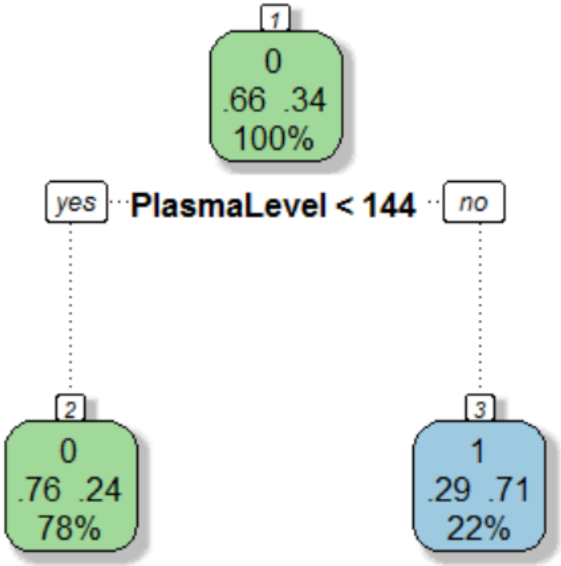


AUC: 0.7791

CART Model

- Now lets focus on pruning the tree. We calculated the C_p and determined that a tree of size 2 would be sufficient.

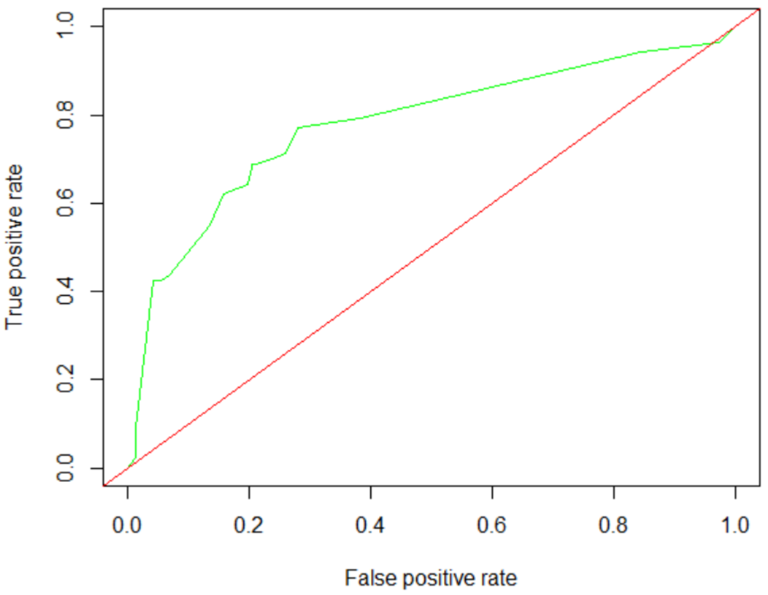




CART Model

- Interesting, the overall performance was unaffected after the pruning procedure.

ROC Chart



AUC: 0.7791

		Predicted	
		No Diabetes	Diabetes
Actual	No Diabetes	124	33
	Diabetes	23	54

```
              Accuracy : 0.7607
                95% CI : (0.7008, 0.8139)
   No Information Rate : 0.6282
   P-Value [Acc > NIR] : 1.055e-05

                 Kappa : 0.4754
Mcnemar's Test P-Value : 0.2291

           Sensitivity : 0.8435
           Specificity : 0.6207
        Pos Pred Value : 0.7898
        Neg Pred Value : 0.7013
            Prevalence : 0.6282
        Detection Rate : 0.5299
  Detection Prevalence : 0.6709
     Balanced Accuracy : 0.7321
```

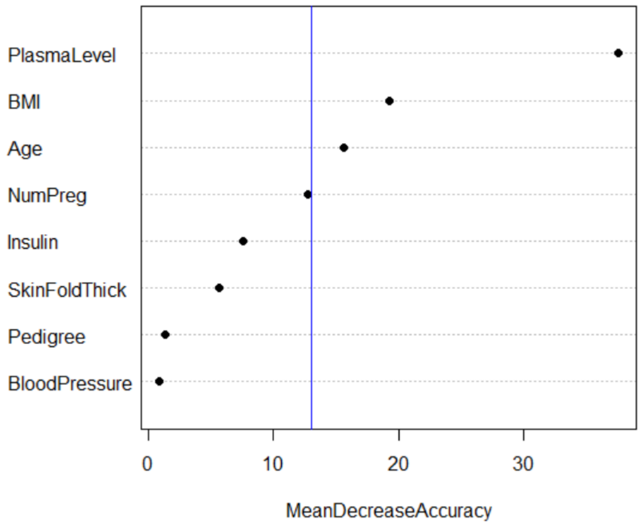
Random Forest Model

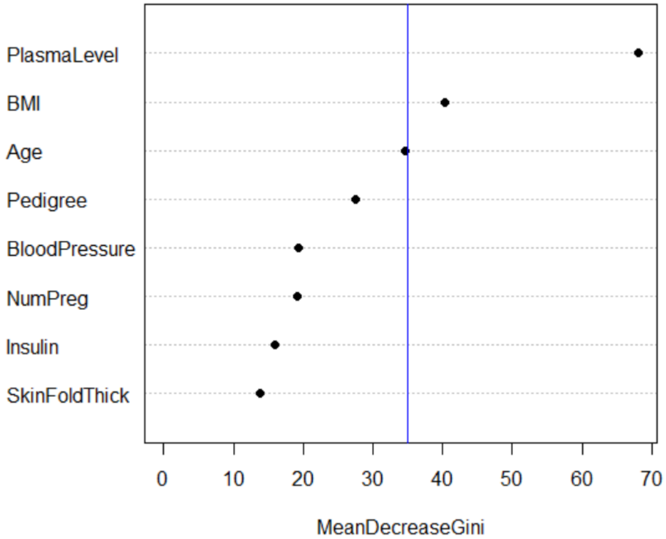
- Now lets build a random forest model. This model will build 500 separate decision trees with 5 variables defined at each split.

- The OOB estimate of the error rate is 26.03%.

- We can visualize the most importance variables in the final model through the Mean Decrease accuracy measurement or Mean Decrease Gini.

- The model has identified that plasma levels, BMI, and Age are the most important risk factors for diabetes.

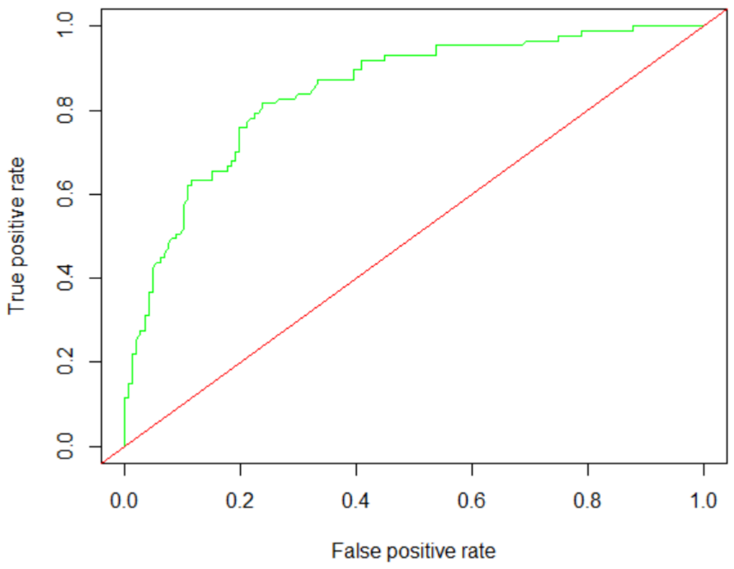




Random Forest Model

- The random forest model has an AUC of 0.8467 compared to the AUC of 0.7791 for the CART model.

ROC Chart



AUC: 0.8467

		Predicted	
		No Diabetes	Diabetes
Actual	No Diabetes	130	32
	Diabetes	17	55

```
               Accuracy : 0.7906
                 95% CI : (0.7328, 0.8409)
    No Information Rate : 0.6282
    P-Value [Acc > NIR] : 6.295e-08

                  Kappa : 0.5354
 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.8844
            Specificity : 0.6322
         Pos Pred Value : 0.8025
         Neg Pred Value : 0.7639
             Prevalence : 0.6282
         Detection Rate : 0.5556
   Detection Prevalence : 0.6923
      Balanced Accuracy : 0.7583
```

Practical Example
Cellular Customer Churn

Understanding the Data

- In the competitive business environment, a lot of effort is spent on attracting and maintaining customers.

- The cost to maintain an existing customer is much less than acquiring them. This makes it paramount for organizations to understand what factors contribute to the turnover of clients and enact preventative strategies to reduce turnover.

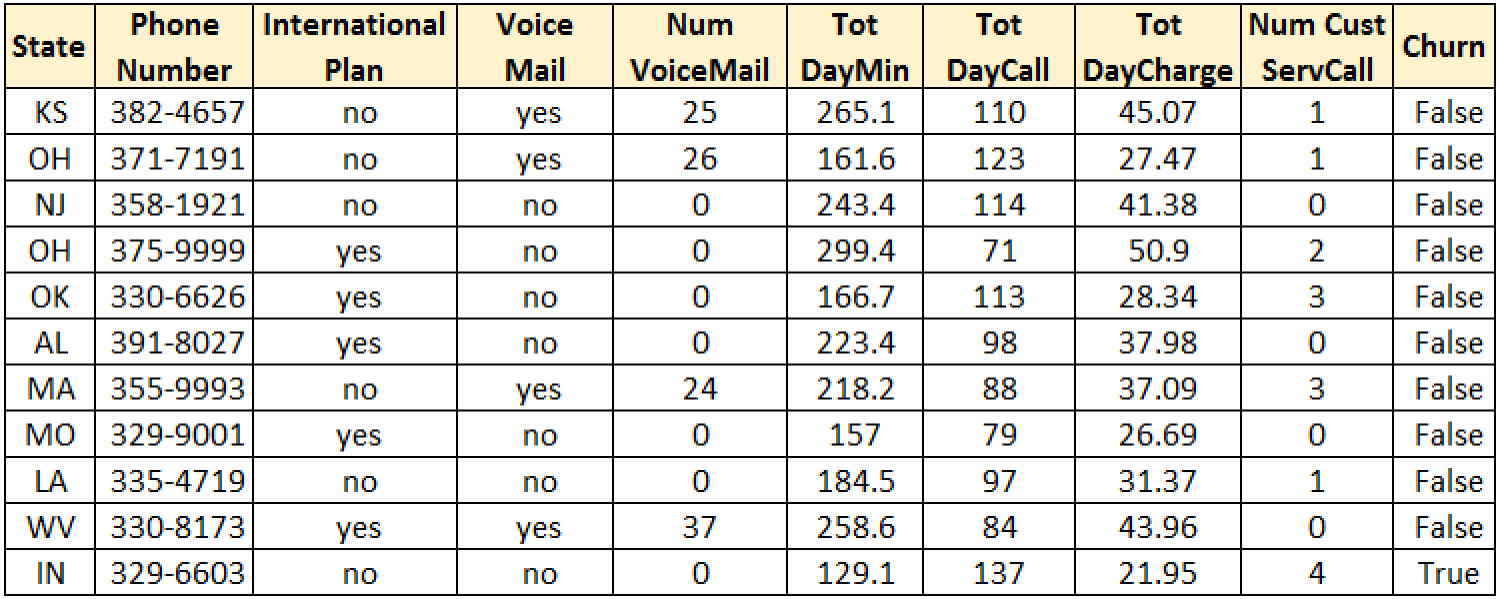
- This turnover is often referred to as the churn or churn rate.

- The dataset we will be working with is contains information related to cell phone subscribers and contains a variety of information related to customer churn.

- The goal for this exercise will be to provide a predictive model that can be leveraged to help understand and leveraged to mitigate customer churn.

Understanding the Data

- Here is a view of the variables within the dataset:

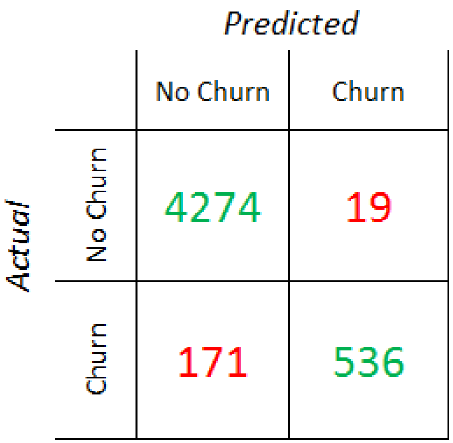


State	Phone Number	International Plan	Voice Mail	Num VoiceMail	Tot DayMin	Tot DayCall	Tot DayCharge	Num Cust ServCall	Churn
KS	382-4657	no	yes	25	265.1	110	45.07	1	False
OH	371-7191	no	yes	26	161.6	123	27.47	1	False
NJ	358-1921	no	no	0	243.4	114	41.38	0	False
OH	375-9999	yes	no	0	299.4	71	50.9	2	False
OK	330-6626	yes	no	0	166.7	113	28.34	3	False
AL	391-8027	yes	no	0	223.4	98	37.98	0	False
MA	355-9993	no	yes	24	218.2	88	37.09	3	False
MO	329-9001	yes	no	0	157	79	26.69	0	False
LA	335-4719	no	no	0	184.5	97	31.37	1	False
WV	330-8173	yes	yes	37	258.6	84	43.96	0	False
IN	329-6603	no	no	0	129.1	137	21.95	4	True

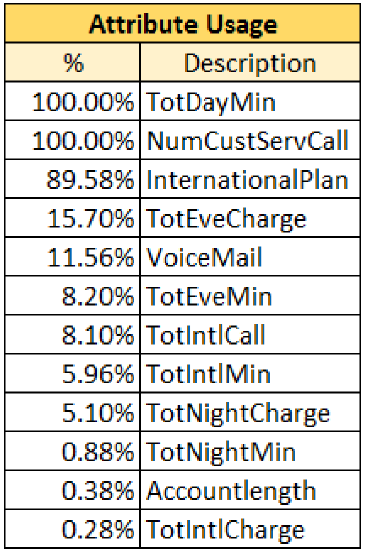
- The Total Minute, Call, and Charge variables are aggregated by Day, Evening, Night, & Intl.
- Our goal is to develop a C.50 model using the Churn variable as the dependent variable.

Model Performance

- When we apply the C5.0 algorithm in R, we are able to get a number of different performance metrics such as % of error, attribute usage, and a confusion matrix.



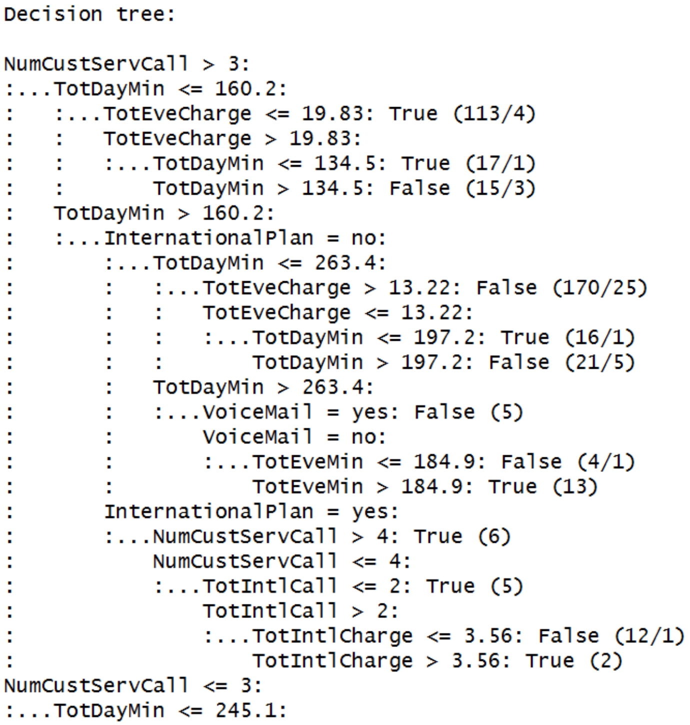
		Predicted	
		No Churn	Churn
Actual	No Churn	4274	19
	Churn	171	536



Attribute Usage	
%	Description
100.00%	TotDayMin
100.00%	NumCustServCall
89.58%	InternationalPlan
15.70%	TotEveCharge
11.56%	VoiceMail
8.20%	TotEveMin
8.10%	TotIntlCall
5.96%	TotIntlMin
5.10%	TotNightCharge
0.88%	TotNightMin
0.38%	Accountlength
0.28%	TotIntlCharge

Error Rate = 3.8%

Predictive Accuracy = 96.2%

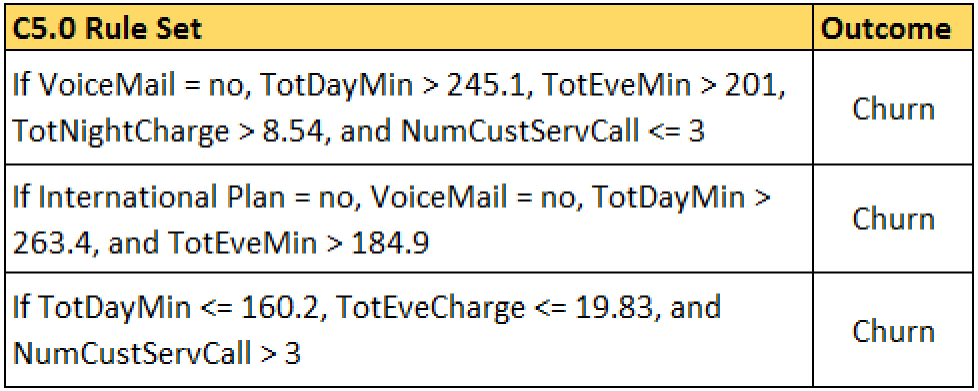


```
Decision tree:

NumCustServCall > 3:
:...TotDayMin <= 160.2:
:   :...TotEveCharge <= 19.83: True (113/4)
:   :   TotEveCharge > 19.83:
:   :   :...TotDayMin <= 134.5: True (17/1)
:   :       TotDayMin > 134.5: False (15/3)
:   TotDayMin > 160.2:
:   :...InternationalPlan = no:
:       :...TotDayMin <= 263.4:
:       :   :...TotEveCharge > 13.22: False (170/25)
:       :   :   TotEveCharge <= 13.22:
:       :   :   :...TotDayMin <= 197.2: True (16/1)
:       :   :       TotDayMin > 197.2: False (21/5)
:       :   TotDayMin > 263.4:
:       :   :...VoiceMail = yes: False (5)
:       :       VoiceMail = no:
:       :       :...TotEveMin <= 184.9: False (4/1)
:       :           TotEveMin > 184.9: True (13)
:       InternationalPlan = yes:
:       :...NumCustServCall > 4: True (6)
:           NumCustServCall <= 4:
:           :...TotIntlCall <= 2: True (5)
:               TotIntlCall > 2:
:               :...TotIntlCharge <= 3.56: False (12/1)
:                   TotIntlCharge > 3.56: True (2)
NumCustServCall <= 3:
:...TotDayMin <= 245.1:
```

Model Performance

- The C.50 algorithm contains a feature which allows for an export of the rule set generated by the model. Lets review some of the results for potential insights.



C5.0 Rule Set	Outcome
If VoiceMail = no, TotDayMin > 245.1, TotEveMin > 201, TotNightCharge > 8.54, and NumCustServCall <= 3	Churn
If International Plan = no, VoiceMail = no, TotDayMin > 263.4, and TotEveMin > 184.9	Churn
If TotDayMin <= 160.2, TotEveCharge <= 19.83, and NumCustServCall > 3	Churn

- There seems to be a pattern emerging that if a cell phone user exceeds 240 daytime minutes and 180 evening minutes that they are likely to churn. Perhaps the pricing of these rate plans is too high? Maybe all we need is to include a voicemail option in the package?

- Not surprisingly, we see that the a customer who makes 3 or more service calls is likely to discontinue the service.

Building off of the Model

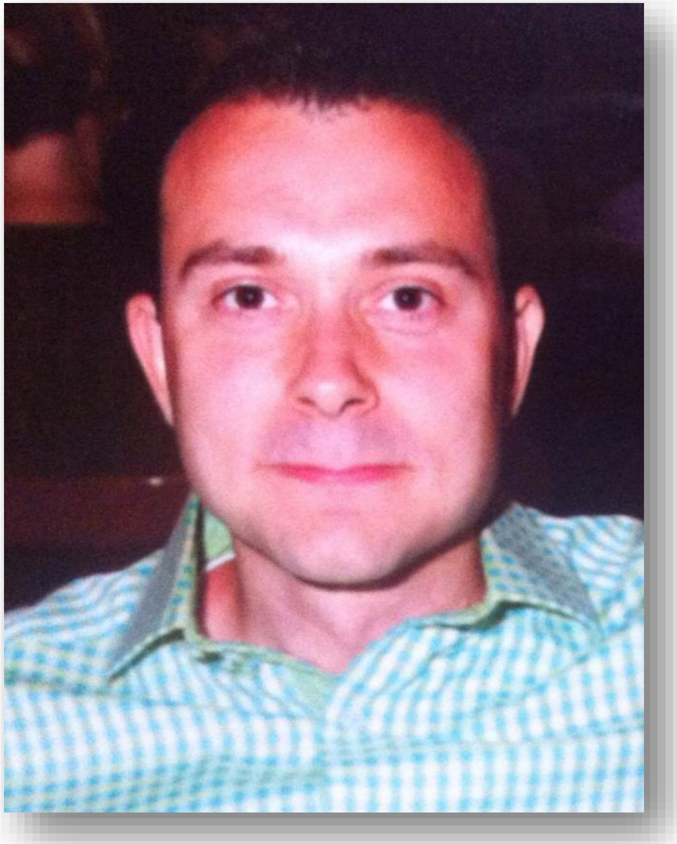
- This ruleset we had produced offers tangible evidence towards predicting behavior based off of product usage and consumption and drawing from machine learning techniques.

- To build off of this analysis, we can consider understanding the demographics of customers who fall within the rulesets.
  - What is their age range?
  - Income?
  - Marital Status? # of Children?
  - Education Level?
  - Blue-Collar or White Collar?
  - Ethnicity and Race?
  - Etc…

- Once we understand these characteristics and compare this against the population (perhaps even employing clustering algorithms), we can begin to frame the business strategy to drive the behavior we want to perform in the marketplace.

About Me

- Reside in Wayne, Illinois
- Active Semi-Professional Classical Musician (Bassoon).
- Married my wife on 10/10/10 and been together for 10 years.
- Pet Yorkshire Terrier / Toy Poodle named Brunzie.
- Pet Maine Coons' named Maximus Power and Nemesis Gul du Cat.
- Enjoy Cooking, Hiking, Cycling, Kayaking, and Astronomy.
- Self proclaimed Data Nerd and Technology Lover.



Fine

Acknowledgements

- http://trevorstephens.com/post/72923766261/titanic-getting-started-with-r-part-3-decision
- http://www.whrc.org/education/indonesia/pdf/DecisionTrees_RandomForest_v2.pdf
- http://www.rdatamining.com/examples/decision-tree
- http://www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf
- http://www.statmethods.net/advstats/cart.html
- http://machinelearningmastery.com/non-linear-classification-in-r-with-decision-trees/
- http://machinelearningmastery.com/non-linear-regression-in-r-with-decision-trees/
- http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.names
- http://stackoverflow.com/questions/24020666/how-to-make-a-tree-plot-in-caret-package
- http://en.wikipedia.org/wiki/Decision_tree
- https://www.cs.princeton.edu/courses/archive/spring07/cos424/papers/mitchell-dectrees.pdf
- http://support.sas.com/publishing/pubcat/chaps/57587.pdf
- http://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf