DublinR - Machine Learning 101

Introduction with Examples

Eoin Brazil - https://github.com/braz/DublinR-ML-treesandforests

Machine Learning Techniques in R

A bit of context around ML

How can you interpret their results?

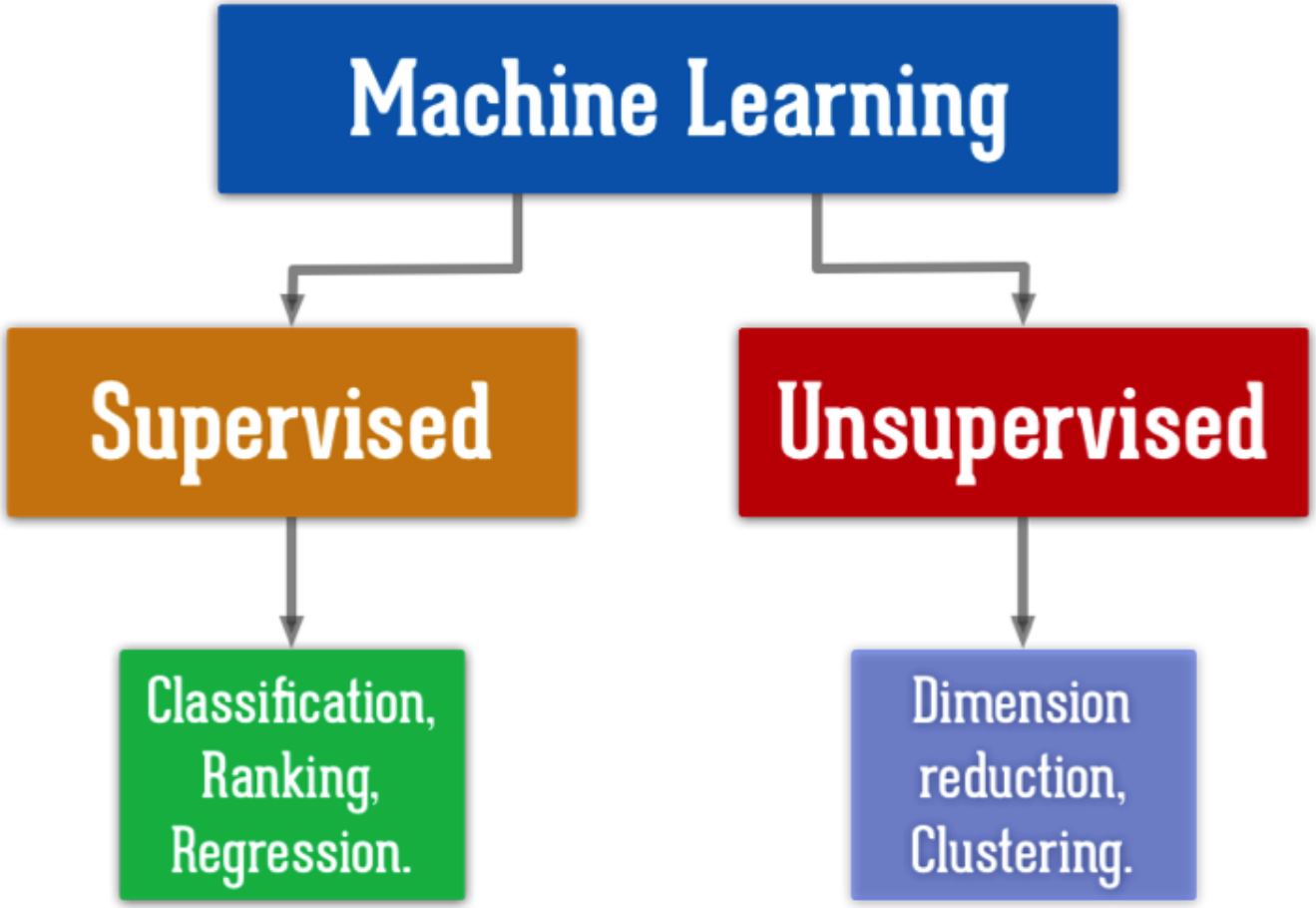
A few techniques to improve prediction / reduce over-fitting

Kaggle & similar competitions - using ML for fun & profit

Nuts & Bolts - 4 data sets and 6 techniques

A brief tour of some useful data handling / formatting tools

Types of Learning



A bit of context around ML

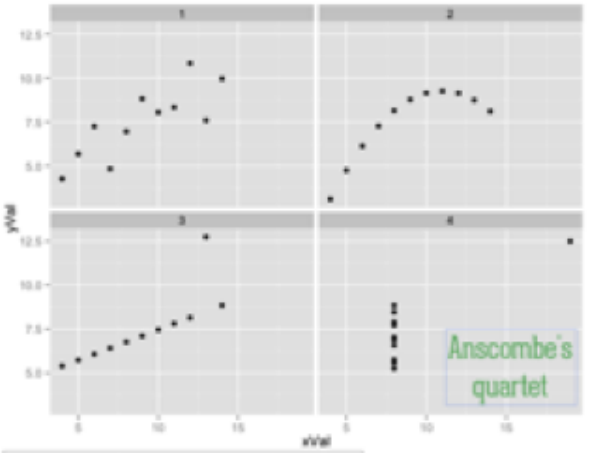
Clean

Simple

var1
var2
var3

} Pareto Principle

Visual




```
> aggregate(.~group,data=mydata,mean)
  group xVal     yVal
1     1    9 7.500909
2     2    9 7.500909
3     3    9 7.500000
4     4    9 7.500909
```

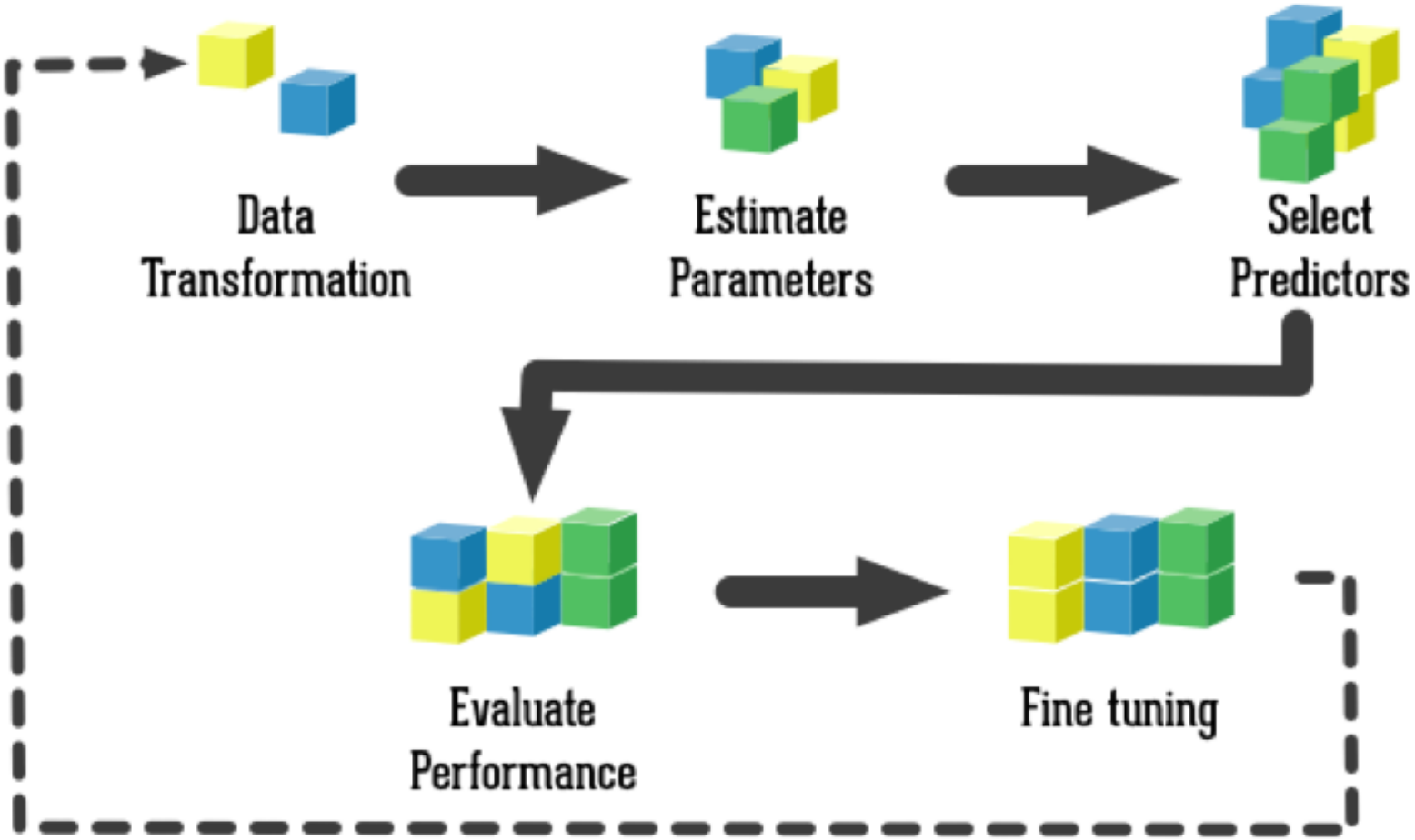
```
> aggregate(.~group,data=mydata,sd)
  group     xVal     yVal
1     1 3.316625 2.031568
2     2 3.316625 2.031657
3     3 3.316625 2.030424
4     4 3.316625 2.030579
```

Data Transformations

C	Zero Mean	}			
S	One SD		**1 - Predictors - M**	{	SS M-Dim Sphere
T	$\sqrt{}$ log INV	}			PCA
					PLS

Correlation, Dummy Variables, Filtering

Model Building Process



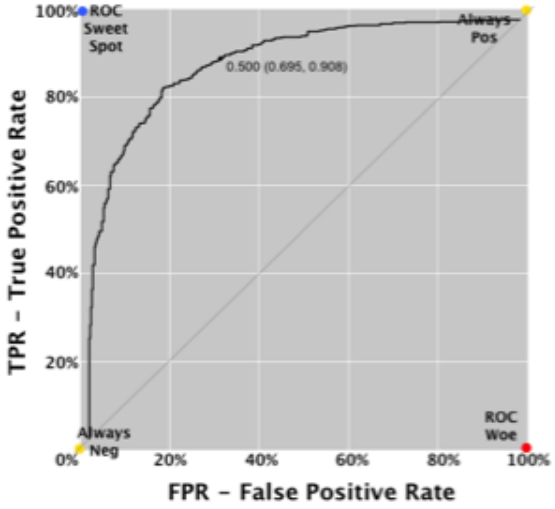
Model Selection and Model Assessment

Assessment [1 & 2]

1

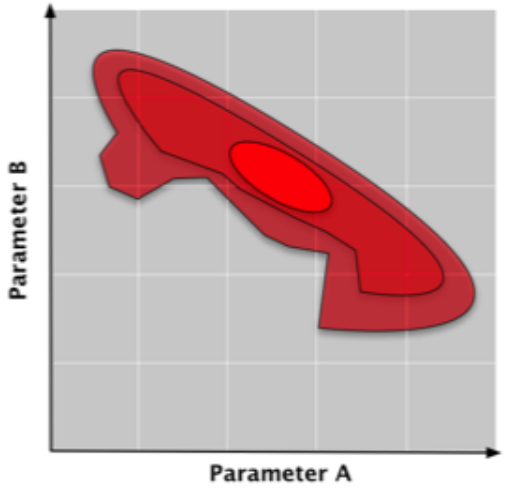
Confusion Matrix	REFERENCE (ACTUAL) HEALTHY (normal)	DISEASED (patient)
PREDICTED HEALTHY (normal)	Actual Healthy Patient and Predicted as Healthy Patient	Actual Diseased Patient but Predicted as Healthy Patient
PREDICTED DISEASED (patient)	Actual Healthy Patient but Predicted as Diseased Patient	Actual Diseased Patient and Predicted as Diseased Patient

2



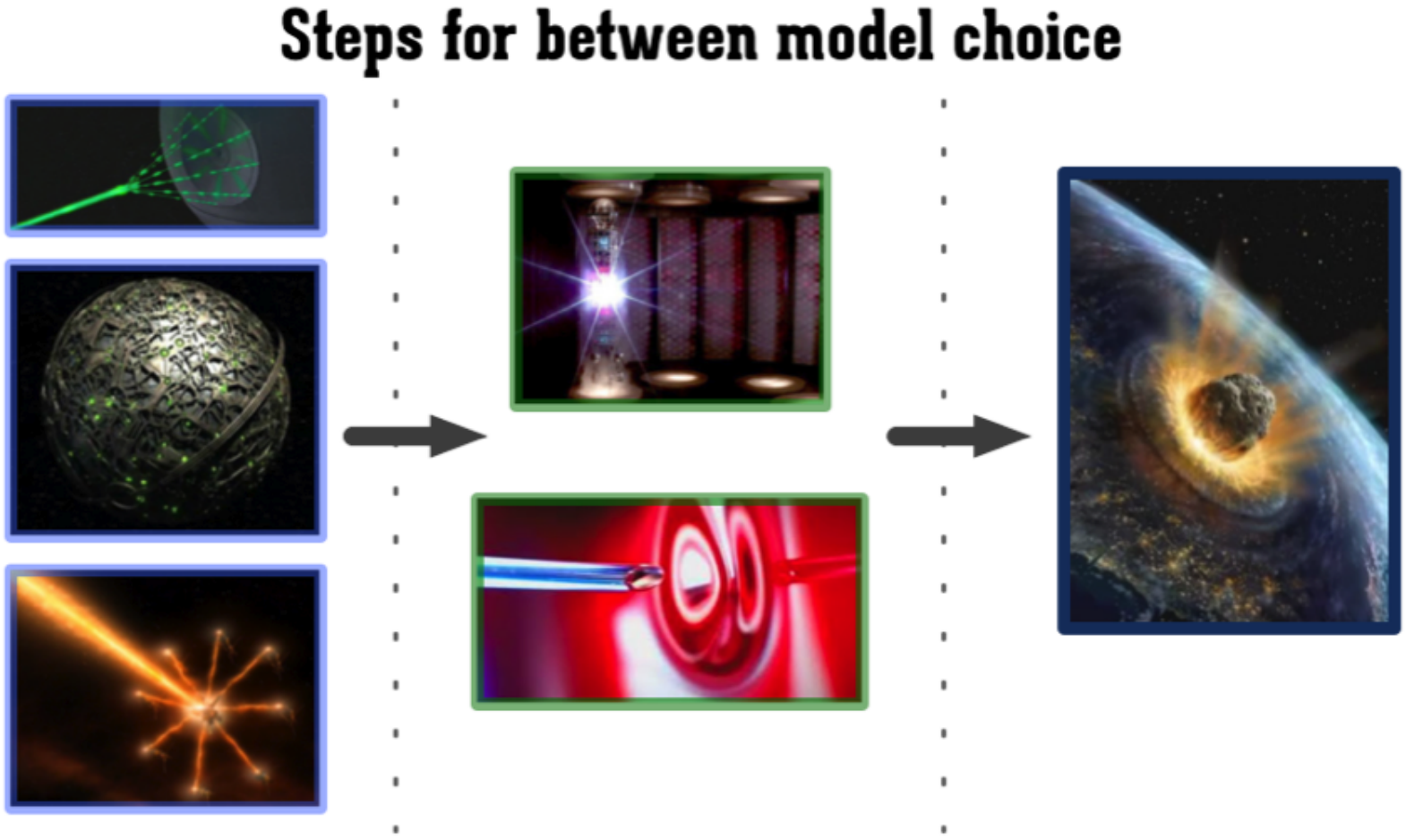
Selection [3 & 4]

3



4

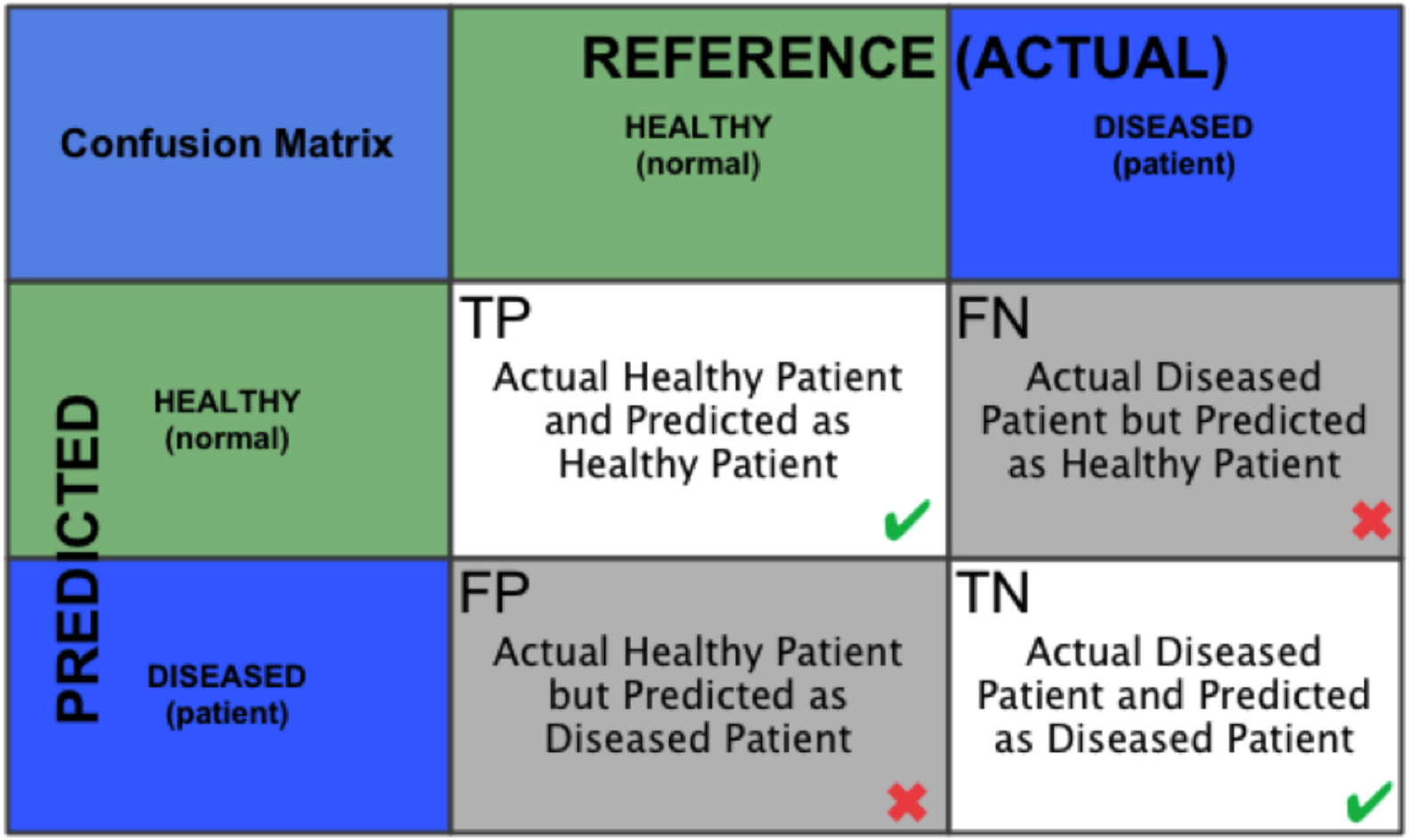
Model Choice - Move from Adaptability to Simplicity



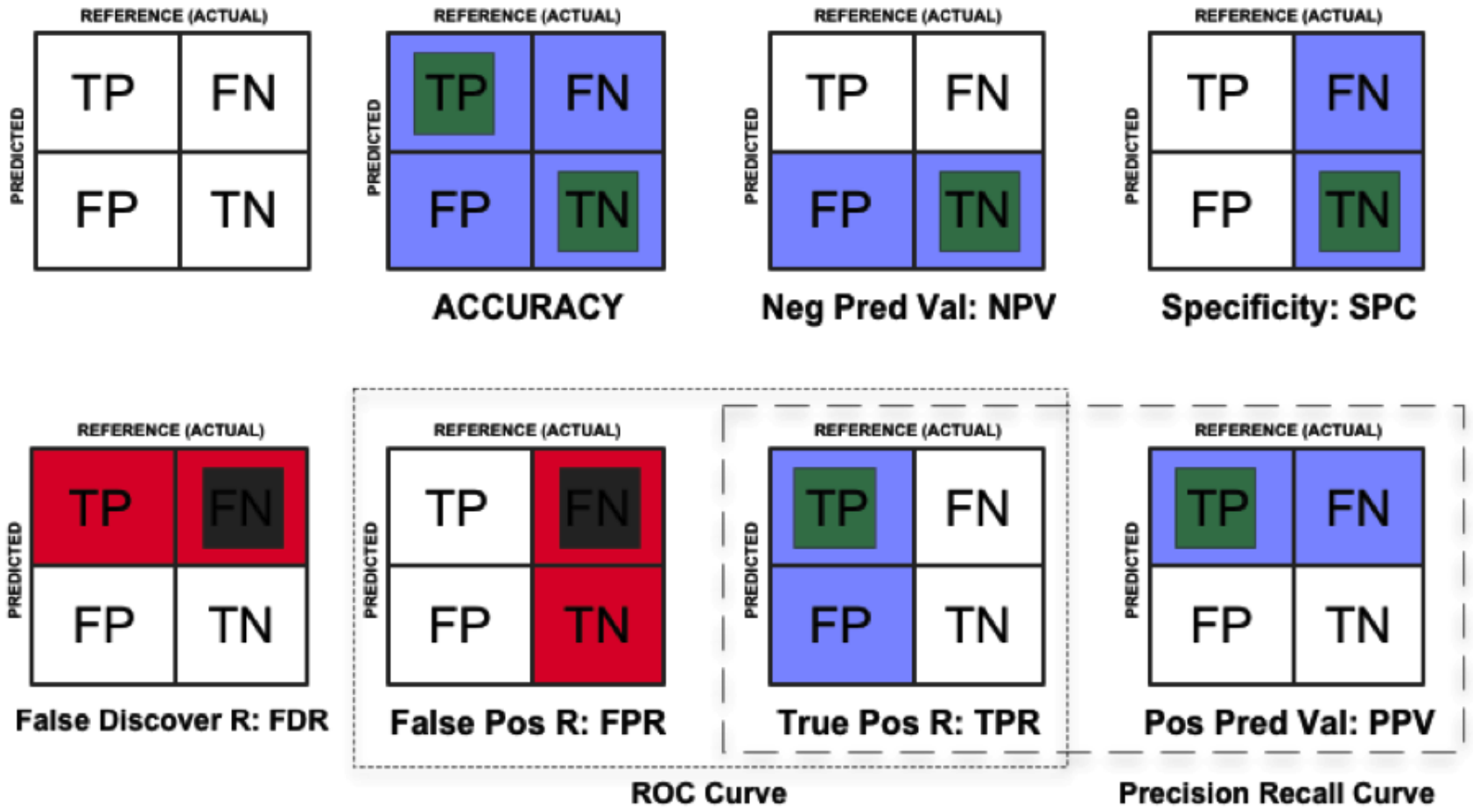
Interpreting A Confusion Matrix

Confusion Matrix		REFERENCE (ACTUAL)
	Predicted Positive	Predicted Negative
PREDICTED Positive Examples	TP true positive – a hit ✔	FN false negative – positive but classified as negative, Type II error ✖
PREDICTED Negative Examples	FP false positive – negative but classified as positive, Type I error ✖	TN true negative – a correct rejection ✔

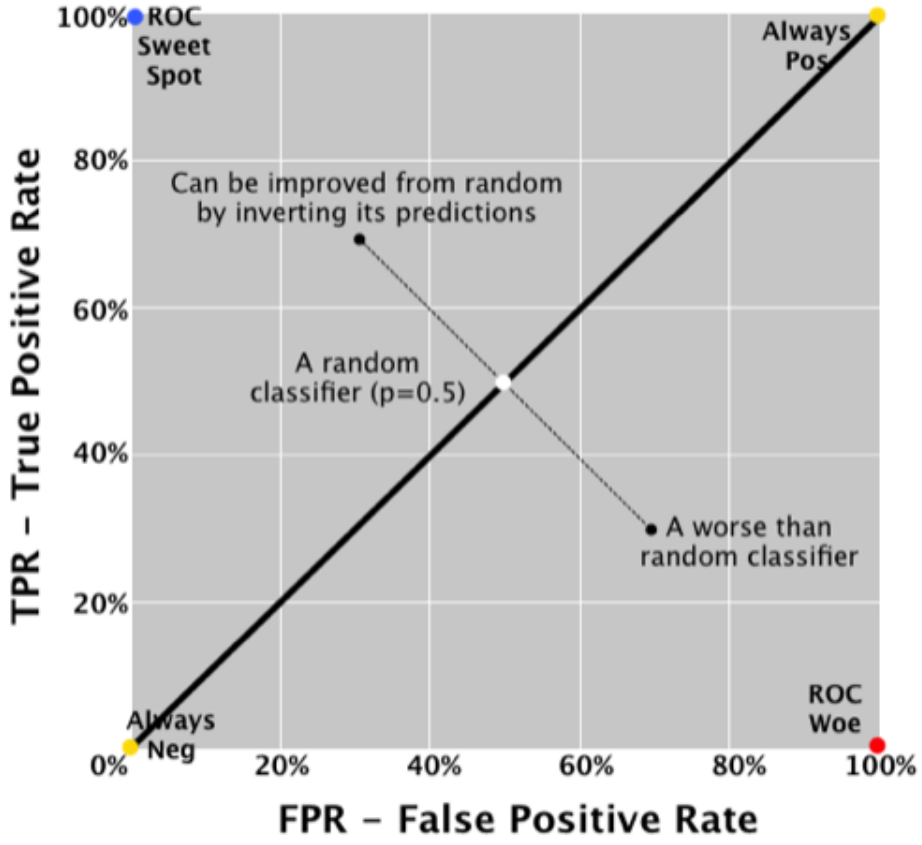
Interpreting A Confusion Matrix Example



Confusion Matrix - Calculations



Interpreting A ROC Plot



- A point in this plot is better than another if it is to the northwest (TPR higher / FPR lower / or both)
- ``Conservatives" - on LHS and near the X-axis - only make positive classification with strong evidence and making few FP errors but low TP rates
- ``Liberals" - on upper RHS - make positive classifications with weak evidence so nearly all positives identified however high FP rates

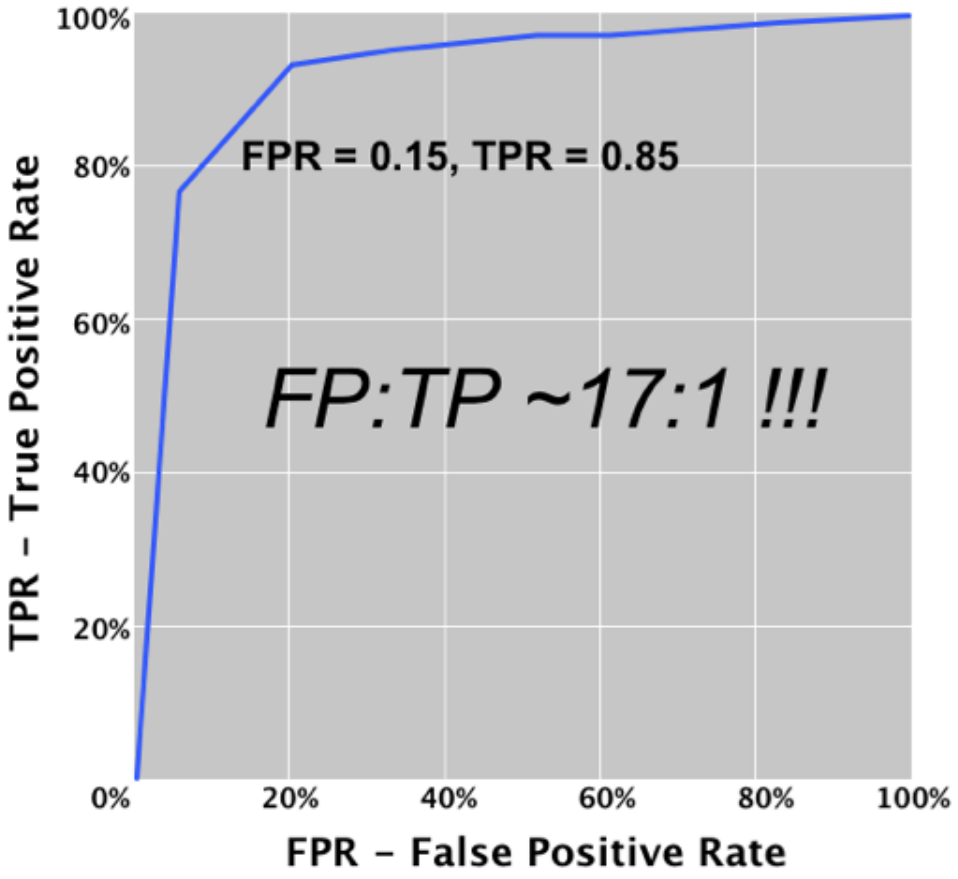
ROC Dangers

REFERENCE (ACTUAL)		
PREDICTED	TP	FN
	FP	TN

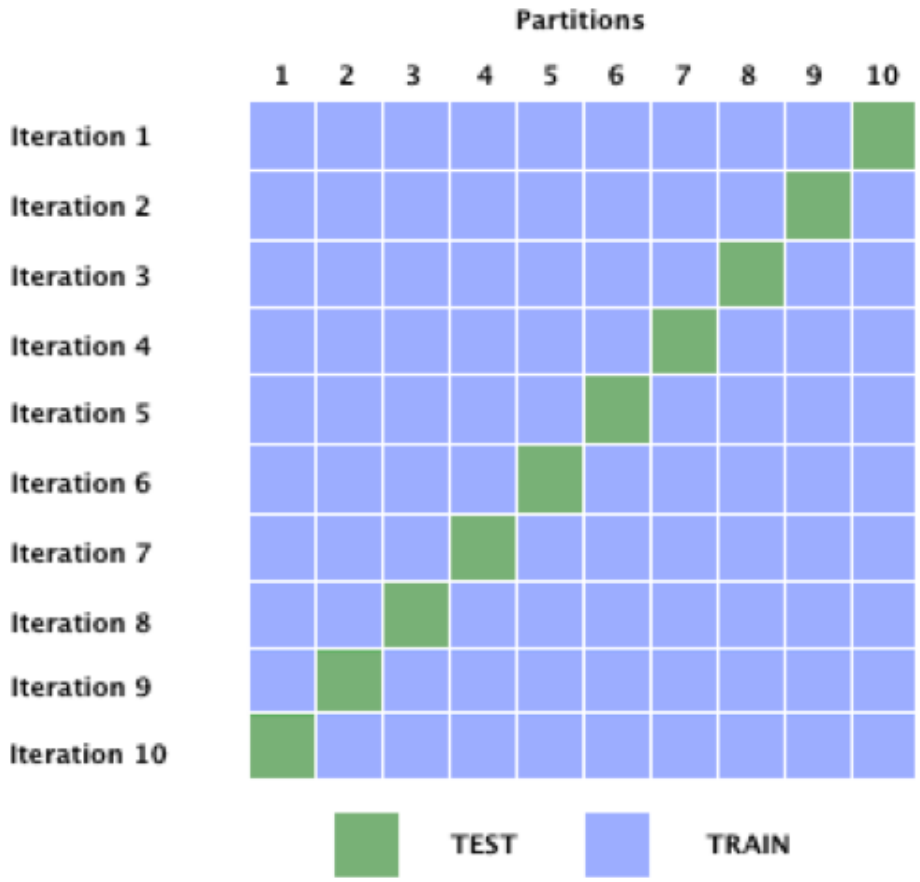
REFERENCE (ACTUAL)		
PREDICTED	85	1485
	15	8415

REFERENCE (ACTUAL)		
PREDICTED	0.85	14.85
	0.15	84.15

%

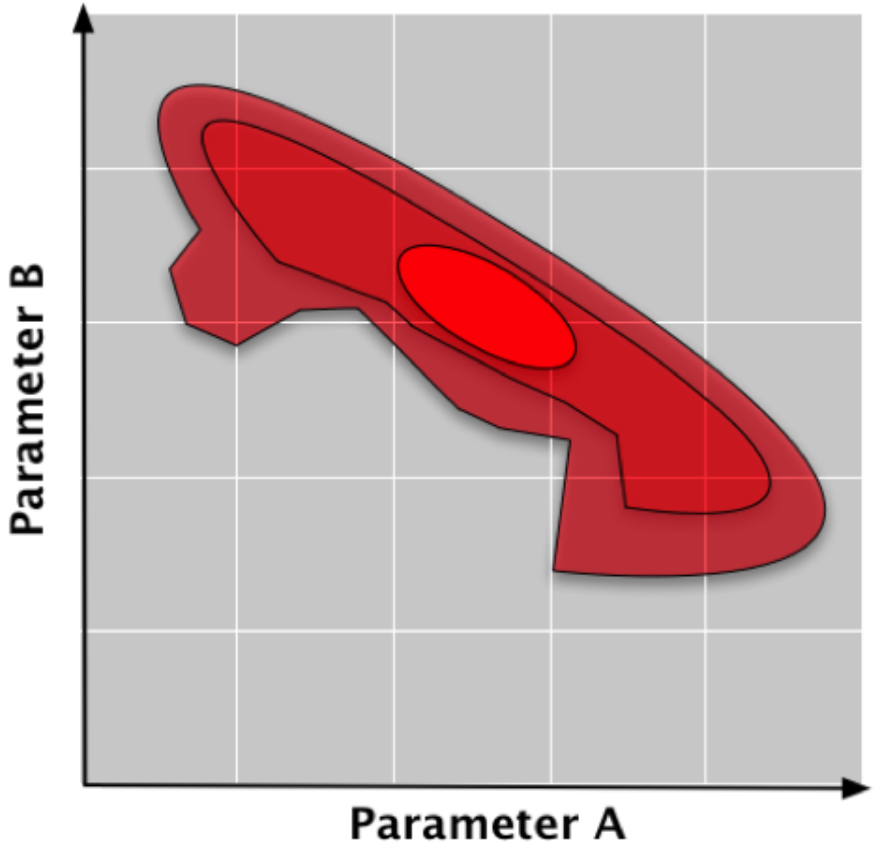


Addressing Prediction Error



- K-fold Cross-Validation (e.g. 10-fold)
  - Allows for averaging the error across the models
- Bootstrapping, draw B random samples with replacement from data set to create B bootstrapped data sets with same size as original. These are used as training sets with the original used as the test set.
- Other variations on above:
  - Repeated cross validation
  - The '.632' bootstrap

Addressing Feature Selection

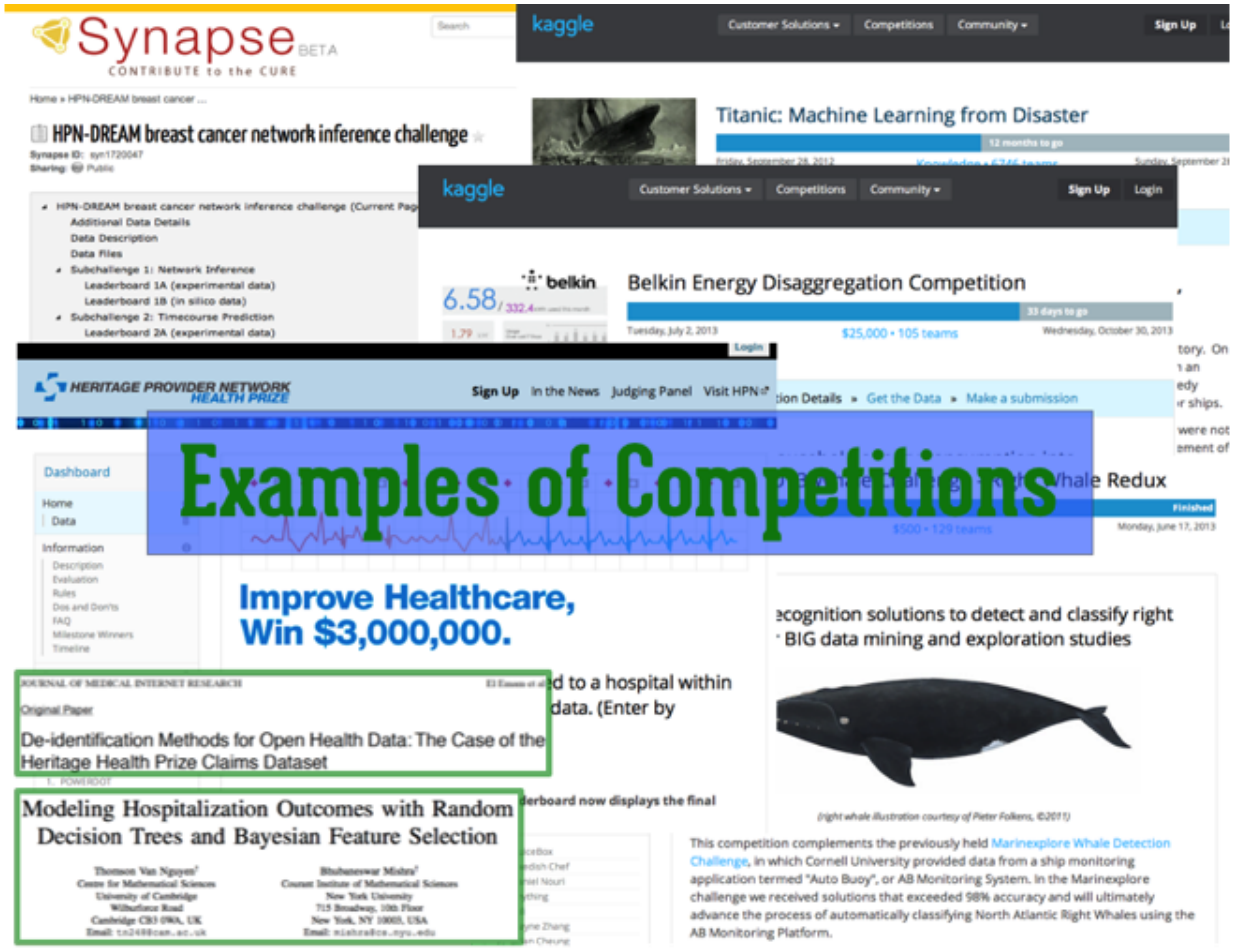


CARET and Feature Selection

- Recursive Feature Elimination Algorithm

```
> cvCtrl <- trainControl(method = "repeatedcv", repeats = 3,
    summaryFunction = twoClassSummary, classProbs = TRUE)
> rpartTune <- train(hadaffair ~ ., data = affairs.df.train,
    method = "rpart", tuneLength = 10, metric = "ROC", trControl = cvCtrl)
> rpartTune
481 samples
  9 predictors
  2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validation (10 fold, repeated 3 times)

Summary of sample sizes: 432, 433, 433, 433, 433, 433, ...

Resampling results across tuning parameters:

  cp       ROC    Sens   Spec    ROC SD  Sens SD  Spec SD
  0        0.654  0.911  0.225   0.0793  0.0597   0.11
  0.00417  0.652  0.911  0.222   0.0775  0.0597   0.106
  0.00833  0.594  0.932  0.161   0.121   0.0563   0.105
  0.0125   0.572  0.949  0.156   0.133   0.045    0.0972
  0.0167   0.562  0.958  0.119   0.133   0.0383   0.0839
  0.0208   0.548  0.955  0.119   0.134   0.044    0.092
  0.025    0.554  0.963  0.103   0.124   0.0474   0.092
  0.0292   0.559  0.976  0.0806  0.118   0.0417   0.0861
  0.0333   0.542  0.988  0.0528  0.101   0.037    0.0804
  0.0375   0.516  1      0       0.0692  0        0

ROC was used to select the optimal model using  the largest value.
The final value used for the model was cp = 0.
```

Kaggle - using ML for fun & profit



Nuts & Bolts - Data sets and Techniques

SCRIPTS

Complexity

1-Marketing-Survey-Dataset.R
Associative Rules

https://github.com/braz/DublinR-ML-treesandforests

2-Car-Dataset.R
Decision Trees [Simple]

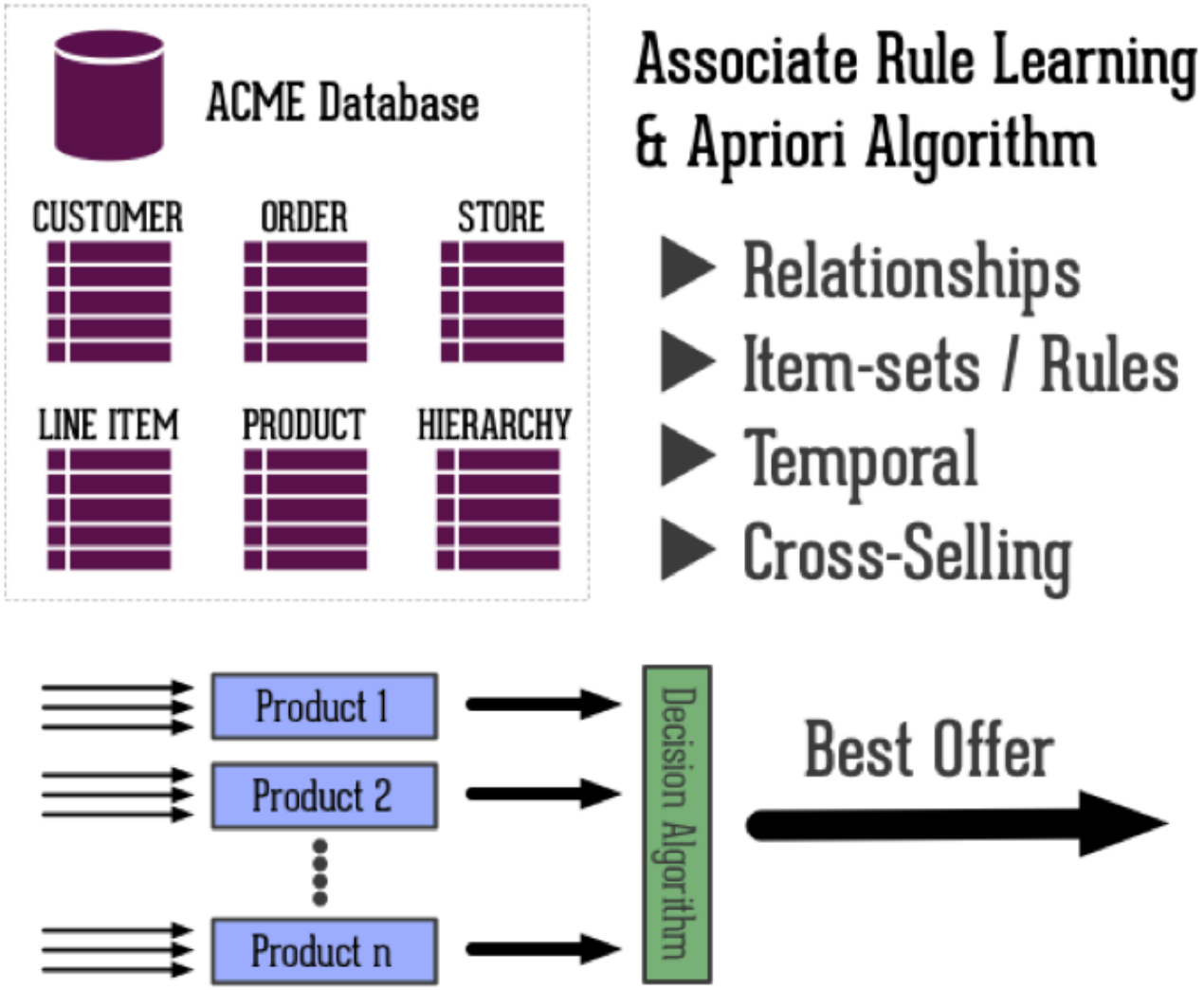
3-Image-Segmentation-Dataset.R
Decision Trees

4-Wine-Portugal-Dataset.R
Random Forests
k Nearest Neighbors
Neural Networks
Support Vector Machines

5-Affairs-Dataset.R
Random Forests
Naive Bayesian

Model comparison

Data Partitioning
Tuning [e.g. cross-fold validation]
Confusion Matrices & ROCs


ACME Database
CUSTOMER
ORDER
STORE
LINE ITEM
PRODUCT
HIERARCHY
Associate Rule Learning
& Apriori Algorithm
Relationships
Item-sets / Rules
Temporal
Cross-Selling
Product 1
Product 2
Product n
Decision Algorithm
Best Offer

Aside - How does associative analysis work ?

1

Customer	Items
1	Orange juice, Apples
2	Milk, orange juice, bread
3	Washing powder, milk
4	Orange juice, Milk
5	Milk, bread
6	Washing powder, orange juice
⋮	⋮

ITEMSET1 IF a *CUST* buys **MILK** then the *CUST* also buys **OJ**

ITEMSET2 IF a *CUST* buys **APPLES** then the *CUST* also buys **OJ**

2

	OJ	Apples	Milk	Bread	Washing Powder
OJ	4	1	2	1	1
Apples	1	1	0	0	0
Milk	2	0	4	1	0
Bread	1	0	2	2	0
Washing Powder	1	0	1	0	2

3 ***If customer buys bread, then the customer also buys milk***

Support: Num of trans containing all items in the rule

Confidence: Measure of predicting the RHS (after the THEN) of a rule

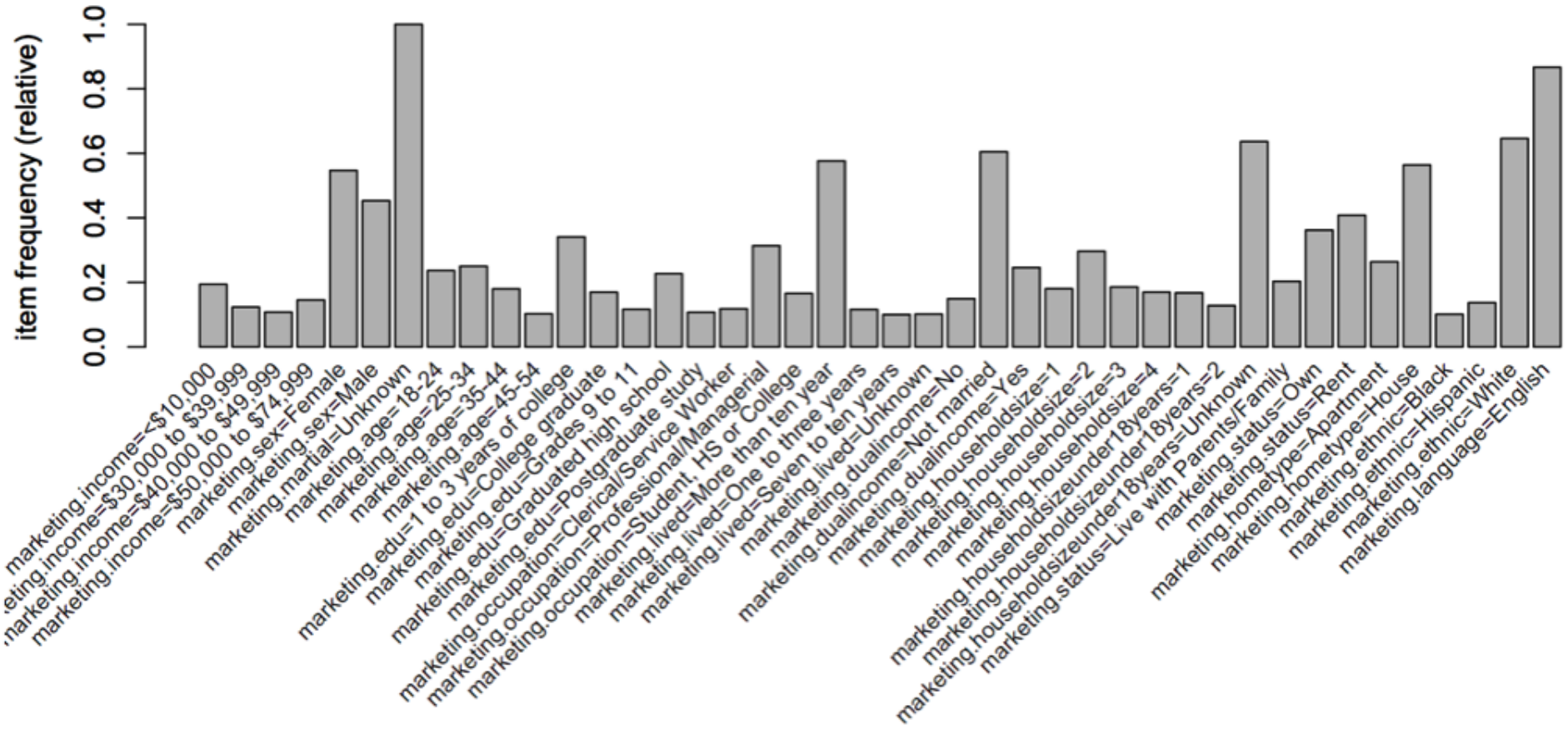
Lift: Measures the power of the rule to the full full by randomly guessing RHS

Chi-Square: Measures the probability that the contingency table behind the rule could be produced randomly

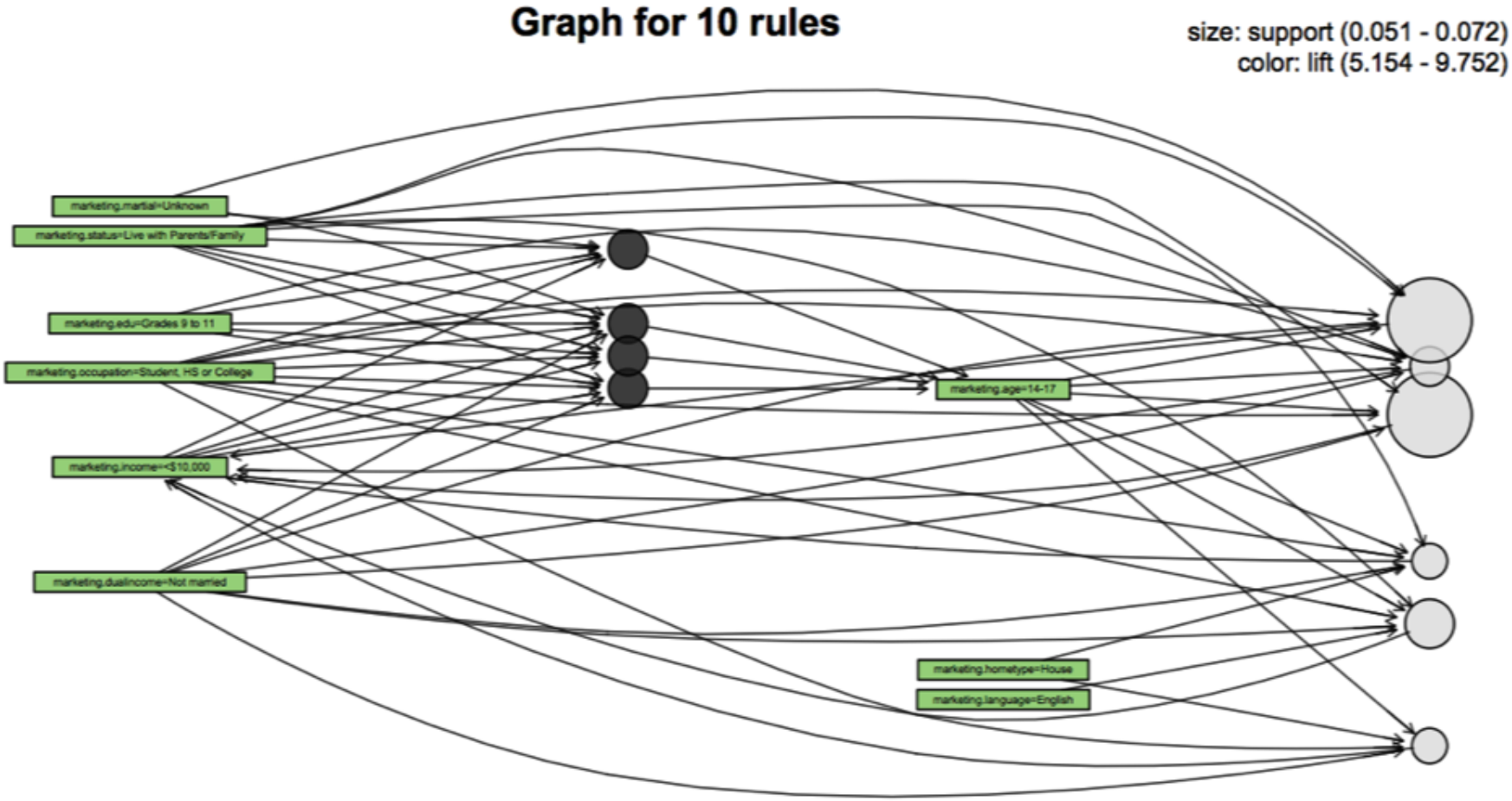
4 ***SKU, EAN, Hierarchies***

What are they good for ?

Marketing Survey Data - Part 1



Marketing Survey Data - Part 2

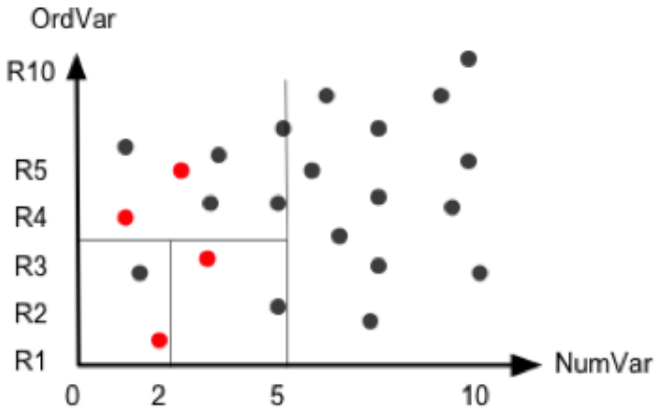


Aside - How do decision trees work ?

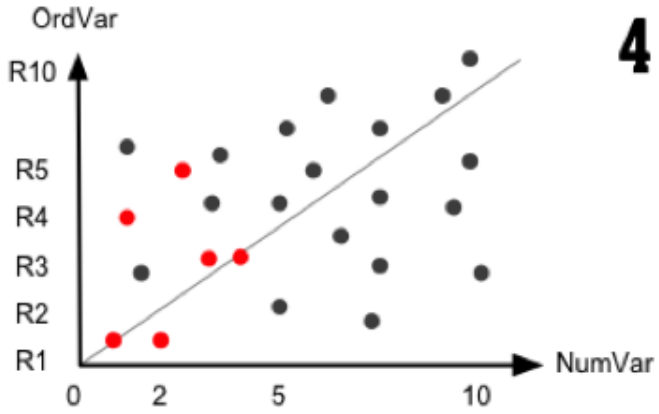
1

NumVar	OrdVar	CatVar
1	R2	Good
1.5	R4	Bad
2	R1	Bad
2.5	R5	Good
3	R3	Good
3.5	R3	Bad
⋮	⋮	⋮

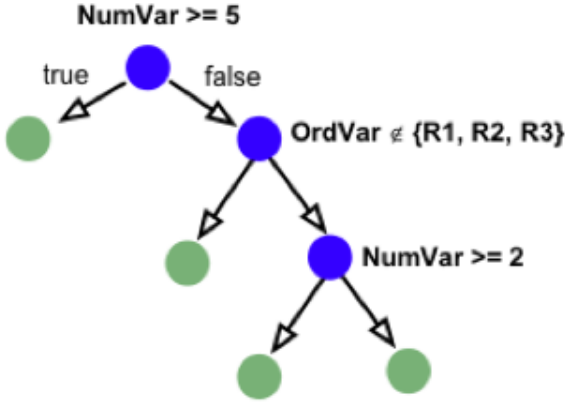
3



2

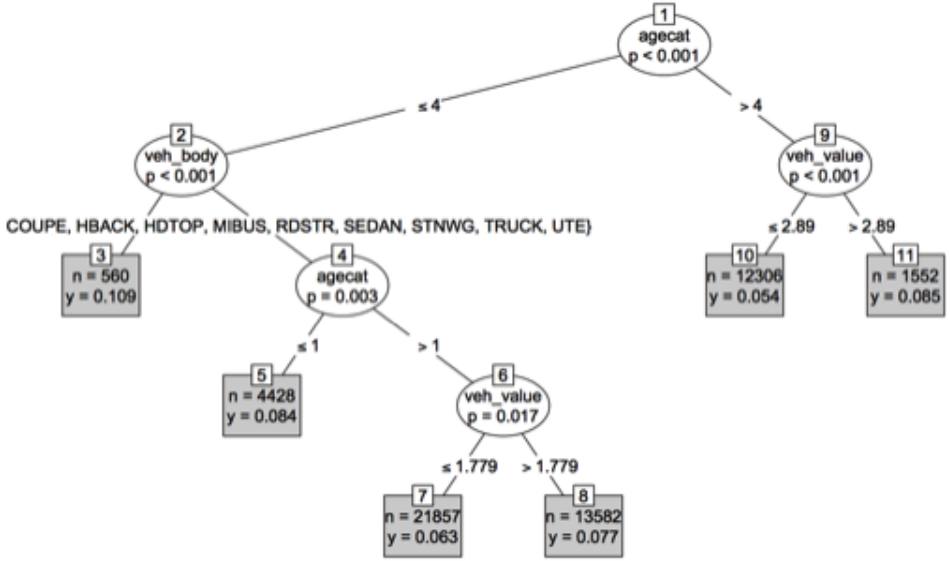


4



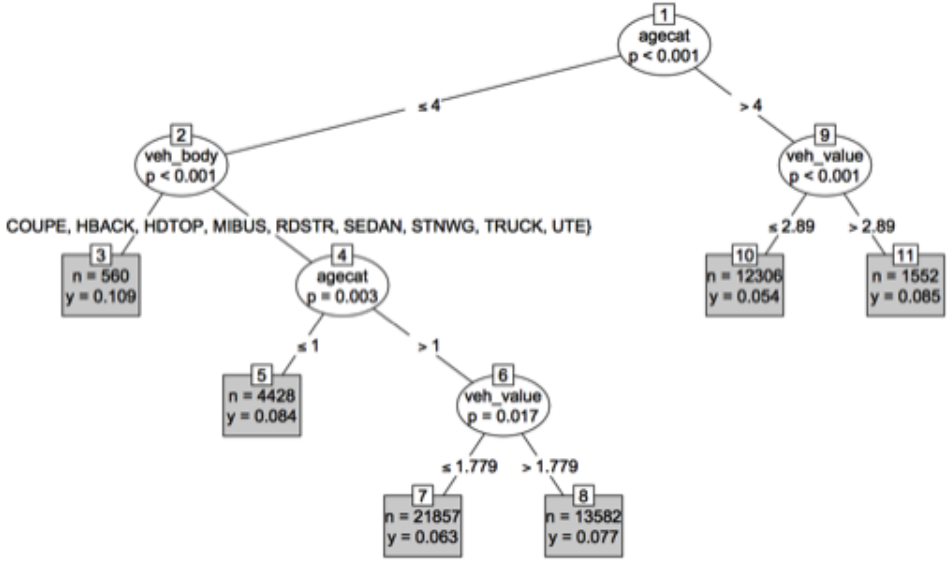
What are they good for ?

Car Insurance Policy Exposure Management - Part 1



- Analysing insurance claim details of 67856 policies taken out in 2004 and 2005.
- The model maps each record into one of X mutually exclusive terminal nodes or groups.
- These groups are represented by their average response, where the node number is treated as the data group.
- The binary claim indicator uses 6 variables to determine a probability estimate for each terminal node determine if a insurance policyholder will claim on their policy.

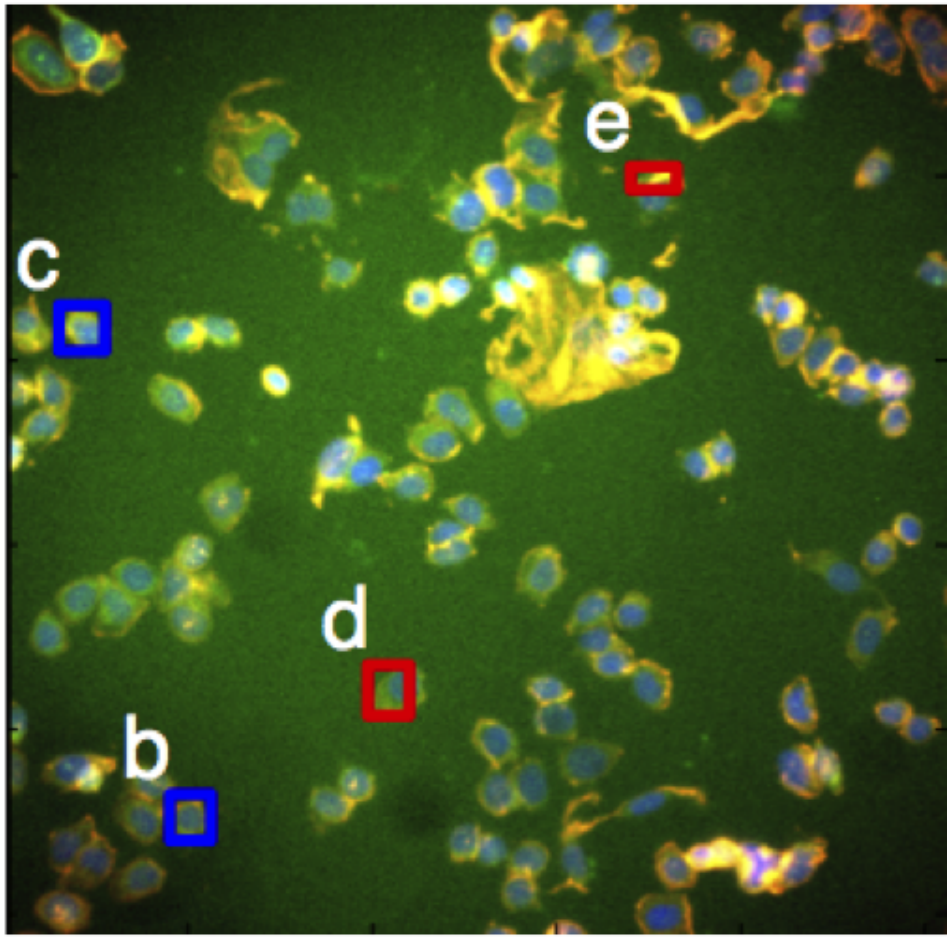
Car Insurance Policy Exposure Management - Part 2



- Root node, splits the data set on 'agecat'
- Younger drivers to the left (1-8) and older drivers (9-11) to right
- N9 splits on basis of vehicle value
- N10 <= $28.9k giving 15k records and 5.4% of claims
- N11 > $28.9k+ giving 1.9k records and 8.5% of claims
- Left Split from Root, N2 splits on vehicle body type, on age (N4), then on vehicle value (N6)
- The n value = num of overall population and the y value = probability of claim from a driver in that group

What are they good for ?

Cancer Research Screening - Part 1



- Hill et al (2007), models how well cells within an image are segmented, 61 vars with 2019 obs (Training = 1009 & Test = 1010).
  - "Impact of image segmentation on high-content screening data quality for SK-BR-3 cells, Andrew A Hill, Peter LaPan, Yizheng Li and Steve Haney, BMC Bioinformatics 2007, 8:340".
  - b, Well-Segmented (WS)
  - c, WS (e.g. complete nucleus and cytoplasmic region)
  - d, Poorly-Segmented (PS)
  - e, PS (e.g. partial match/es)

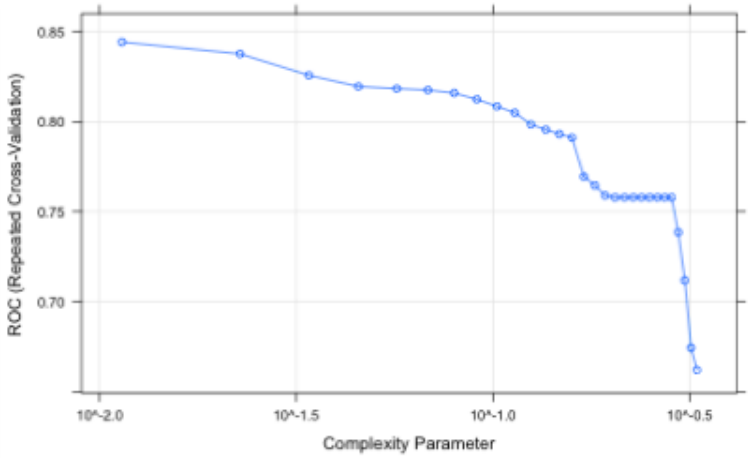
Cancer Research Screening Dataset - Part 2

Image Segmentation

```
> table(testing[,1])
 PS  WS
664 346
```

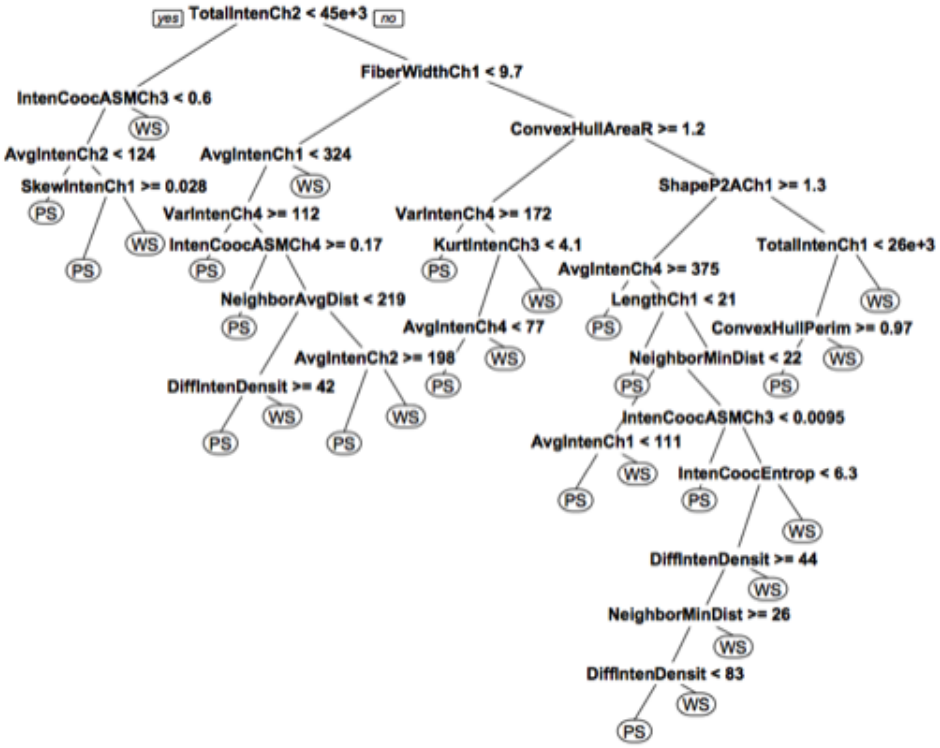
```
'data.frame':   1009 obs. of  6 variables:
$ Class      : Factor w/ 2 levels "PS","WS": 1 2 1 2 1 1 1 2 2 2 ...
$ AngleCh1   : num  133.8 106.6 69.2 109.4 104.3 ...
$ AreaCh1    : int  819 431 298 256 258 358 158 315 246 223 ...
$ AvgIntenCh1: num  31.9 28 19.5 18.8 17.6 ...
$ AvgIntenCh2: num  207 116 102 127 125 ...
$ AvgIntenCh3: num  69.9 63.9 28.2 13.6 22.5 ...
```

```
> head(training[1:2,])
  Class AngleCh1 AreaCh1 AvgIntenCh1 AvgIntenCh2 AvgIntenCh3 AvgIntenCh4 ConvexHullAreaRatioCh1 ConvexHullPerimRatioCh1 DiffIntenDensityCh1
      DiffIntenDensityCh3 DiffIntenDensityCh4 EntropyIntenCh1
2    PS 133.7520     819    31.92327    206.8785    69.91688    164.1535              1.263158              0.7970801           31.87500
              43.12228            79.30842         6.887592
3    WS 106.6464     431    28.03883    116.3155    63.94175    106.6966              1.053310              0.9354750           32.48771
              35.98577            51.35705         5.883557
  EntropyIntenCh3 EntropyIntenCh4 EqCircDiamCh1 EqEllipseLWRCh1 EqEllipseOblateVolCh1 EqEllipseProlateVolCh1 EqSphereAreaCh1 EqSphereVolCh1
      FiberAlign2Ch3 FiberAlign2Ch4 FiberLengthCh1 FiberWidthCh1
2        6.642761       7.880155      32.30550       1.558394             2232.9055             1432.8246       3270.726      17653.525
         1.487935       1.352374       64.28230      13.16708
3        6.683080       7.144601      23.44892       1.375386              802.1945              583.2504       1727.410       6750.985
         1.300522       1.522316       21.14115      21.14115
  IntenCoocASMCh3 IntenCoocASMCh4 IntenCoocContrastCh3 IntenCoocContrastCh4 IntenCoocEntropyCh3 IntenCoocEntropyCh4 IntenCoocMaxCh3
      IntenCoocMaxCh4 KurtIntenCh1 KurtIntenCh3 KurtIntenCh4 LengthCh1
2     0.028051061     0.012594975          8.227953            6.984046            6.822138            7.098988      0.15321477      0.
   07307141   -0.2487691   -0.3307839   -0.2652638  47.21855
3     0.006862315     0.006141691          14.446874           16.700843            7.580100            7.671478      0.02835052      0.
   02319588   -0.2934846    1.0512813    0.1506140  28.14303
  NeighborAvgDistCh1 NeighborMinDistCh1 NeighborVarDistCh1 PerimCh1 ShapeBFRCh1 ShapeLWRCh1 ShapeP2ACh1 SkewIntenCh1 SkewIntenCh3 SkewIntenCh4
      SpotFiberCountCh3 SpotFiberCountCh4 TotalIntenCh1
2           174.4442          30.11114          81.38063 154.8988   0.5397584    1.468181    2.255810    0.3987047    0.6197388    0.5272631
                    4                12         24964
3           158.4774          34.94477          98.43768  84.5646   0.7243116    1.328488    1.272193    0.4724871    0.9713788    0.3247065
                    2                 7         11552
  TotalIntenCh2 TotalIntenCh3 TotalIntenCh4 VarIntenCh1 VarIntenCh3 VarIntenCh4 WidthCh1 XCentroid YCentroid
2        160998         54675        128368    18.80923    56.71535   118.38814 32.16126       215       347
3         47511         26344         43959    17.29564    37.67105    49.47052 21.18553       371       252
```

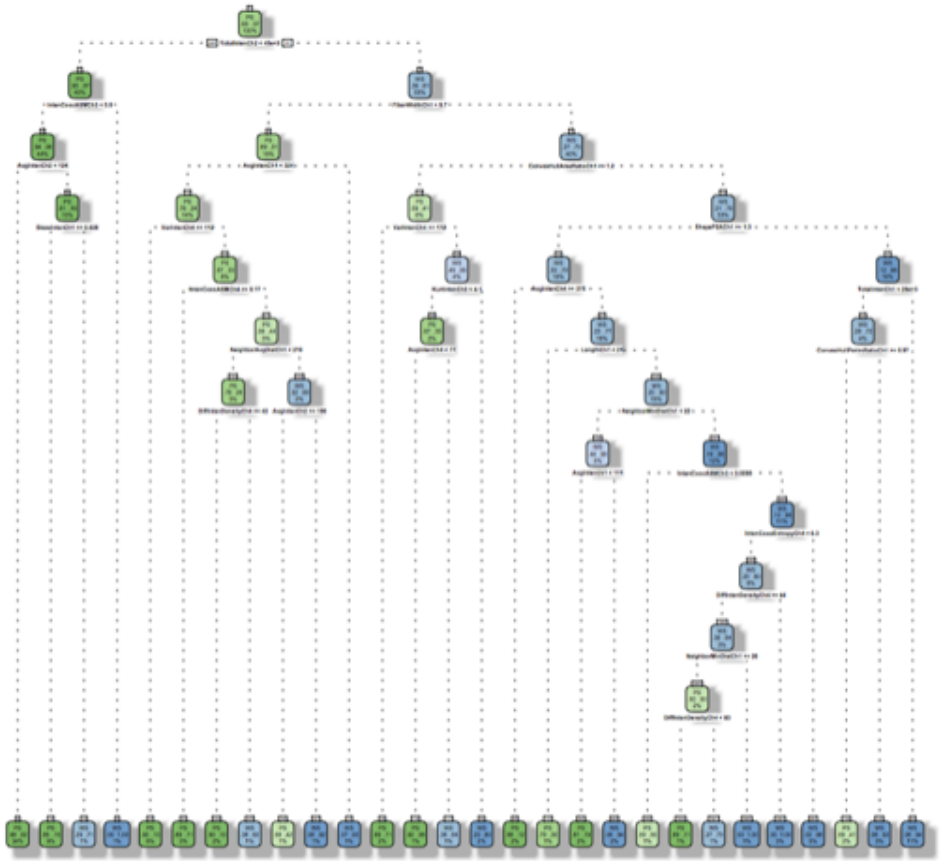


Cancer Research Screening - Part 3

"prp(rpartTune$finalModel)"



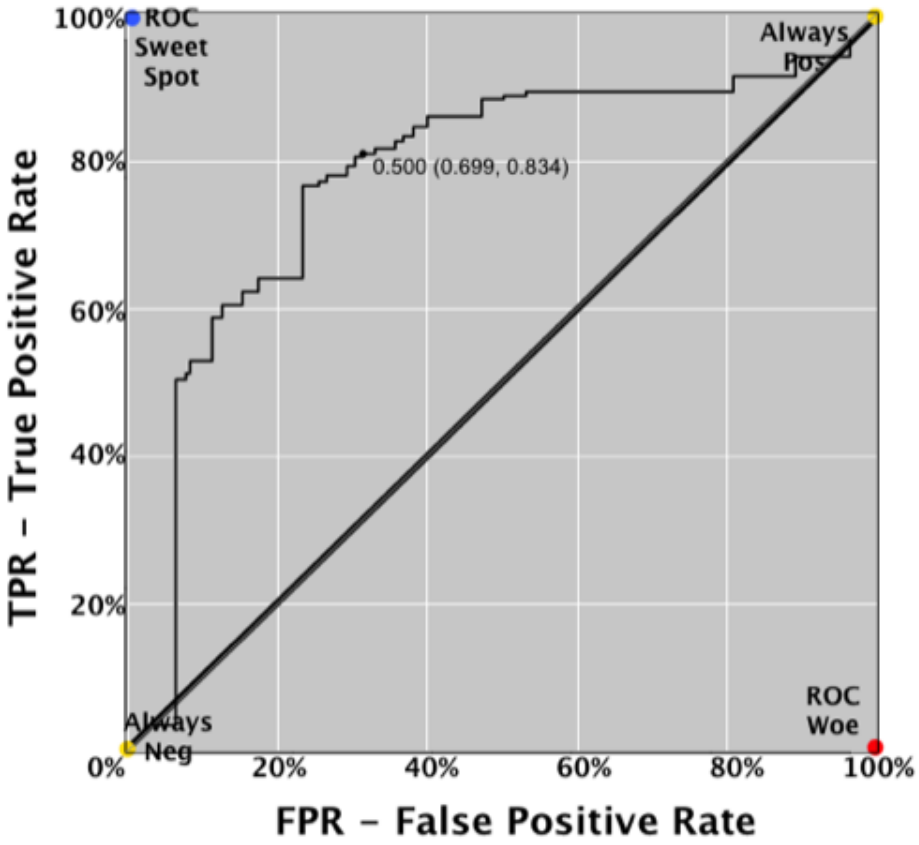
"fancyRpartPlot(rpartTune$finalModel)"



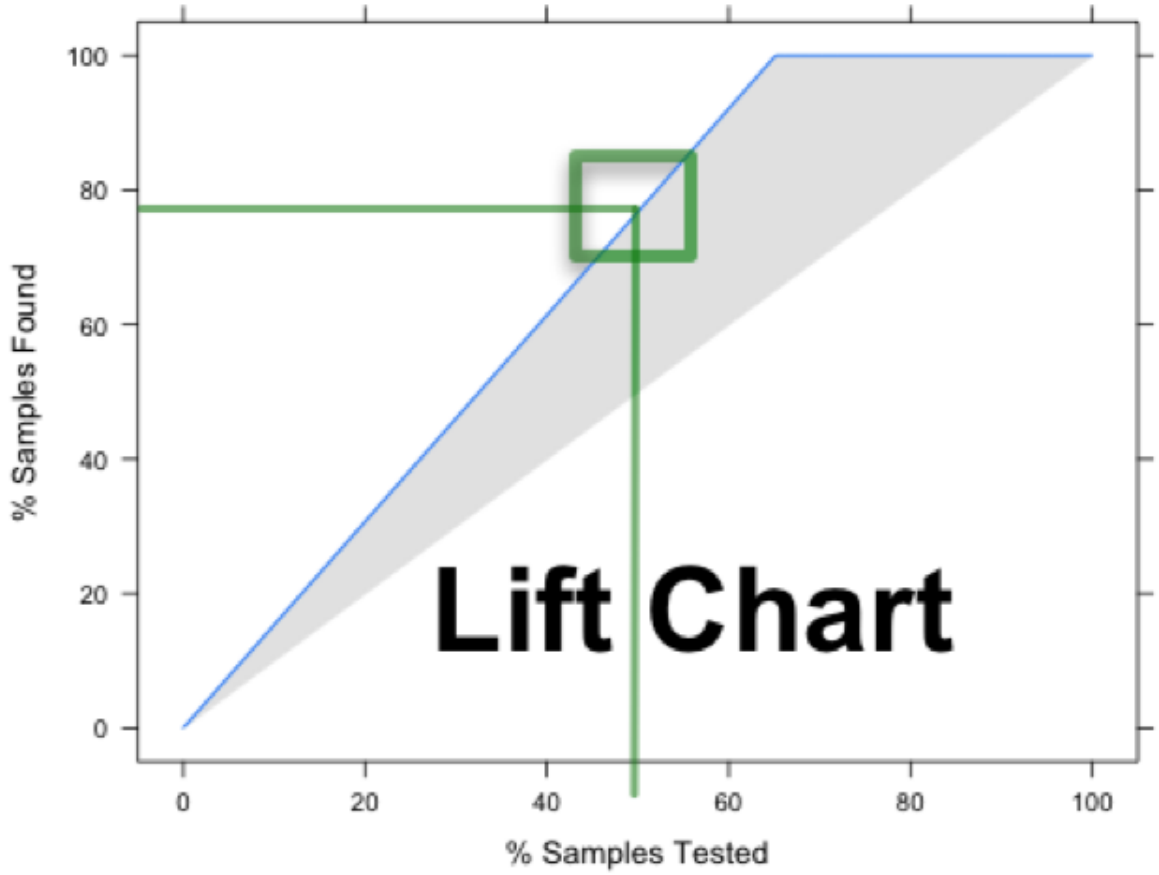
Cancer Research Screening - Part 4

Confusion Matrix	REFERENCE (ACTUAL) Predicted Positive PS	Predicted Negative WS	
PREDICTED Positive Examples PS	TP 554 / 55%	TN 104 / 10%	Pos
PREDICTED Negative Examples WS	FP 110 / 11%	FN 242 / 24%	Neg
	PPos	PNeg	N

Accuracy = 79%
(55+24)



Cancer Research Screening Dataset - Part 5



What are they good for ?

Predicting the Quality of Wine - Part 1

- Cortez et al (2009), models the quality of wines (Vinho Verde), 14 vars with 4898 obs (Training = 5199 & Test = 1298).
- "Modeling wine preferences by data mining from physicochemical properties, P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, Decision Support Systems 2009, 47(4):547-553".
  - Good (quality score is >= 6)
  - Bad (quality score is < 6)

```
##
##   Bad Good
##   476  822
```

Predicting the Quality of Wine - Part 2

Quality of Vinho Verde wines

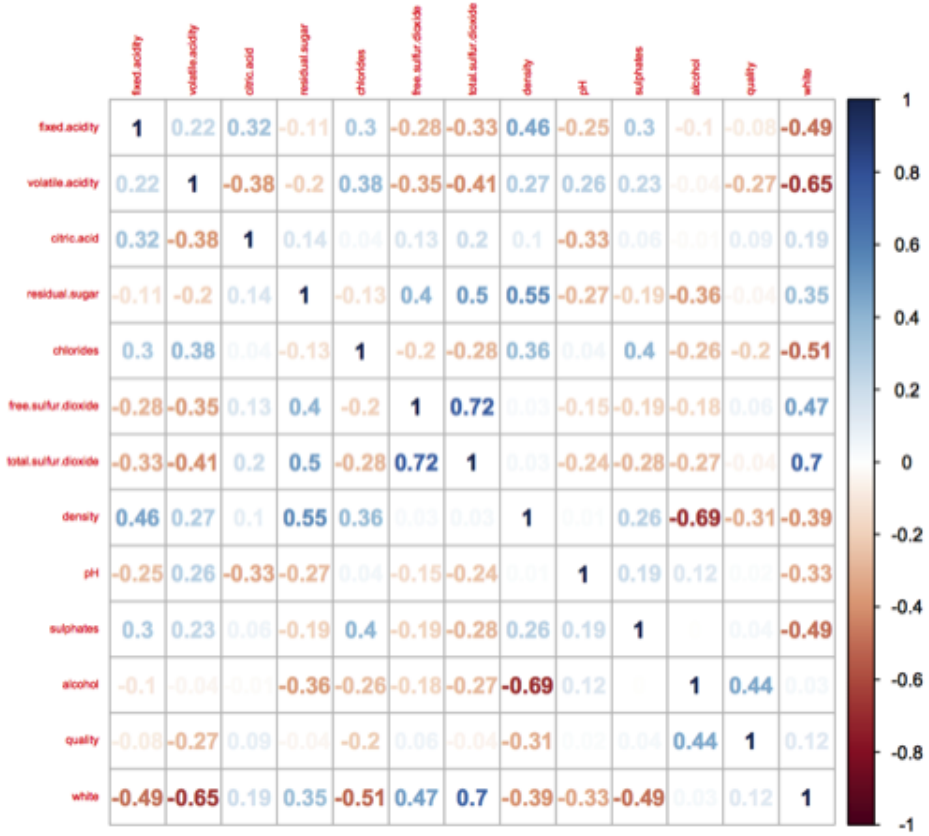
```
> table(wine.df.test$good)
 Bad Good
 476  822
```

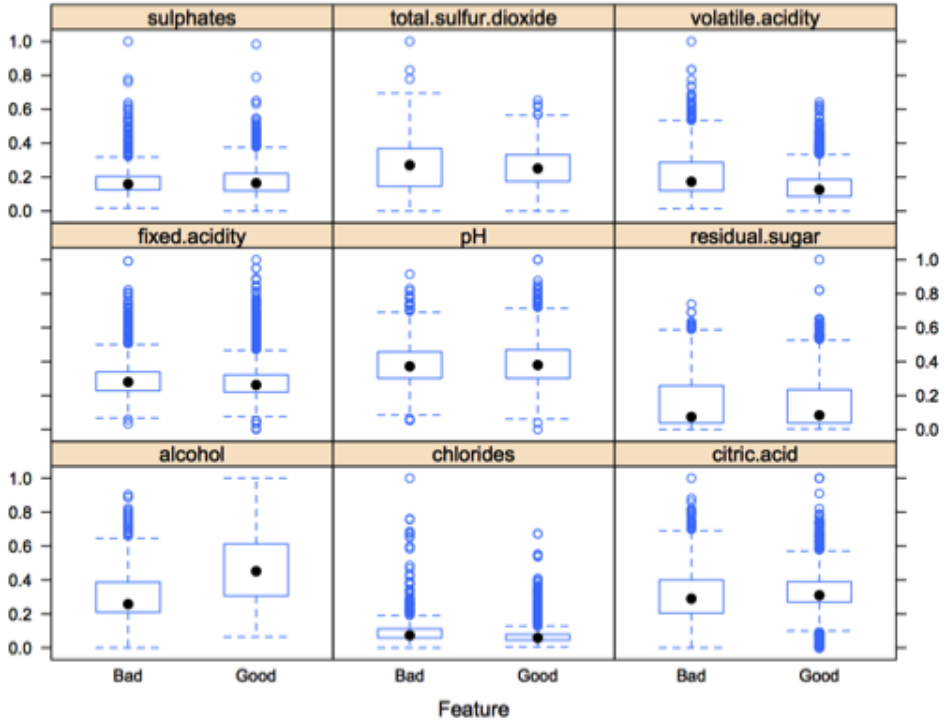
```
> str(wine.df)
'data.frame': 6497 obs. of  15 variables:
 $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
 $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
 $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
 $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
 $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
 $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
 $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
 $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
 $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
 $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
 $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
 $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
 $ color               : Factor w/ 2 levels "red","white": 1 1 1 1 1 1 1 1 1 1 ...
 $ white               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ good                : Factor w/ 2 levels "Bad","Good": 1 1 1 2 1 1 1 2 2 1 ...
```

```
> head(wine.df)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol quality color white good
1           7.4             0.70        0.00            1.9     0.076                 11                  34  0.9978 3.51      0.56     9.4       5   red     0  Bad
2           7.8             0.88        0.00            2.6     0.098                 25                  67  0.9968 3.20      0.68     9.8       5   red     0  Bad
3           7.8             0.76        0.04            2.3     0.092                 15                  54  0.9970 3.26      0.65     9.8       5   red     0  Bad
4          11.2             0.28        0.56            1.9     0.075                 17                  60  0.9980 3.16      0.58     9.8       6   red     0 Good
5           7.4             0.70        0.00            1.9     0.076                 11                  34  0.9978 3.51      0.56     9.4       5   red     0  Bad
6           7.4             0.66        0.00            1.8     0.075                 13                  40  0.9978 3.51      0.56     9.4       5   red     0  Bad
```

```
> wine.corr
                      fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides free.sulfur.dioxide total.sulfur.dioxide    density         pH   sulphates     alcohol     quality       white
fixed.acidity            1.00000000      0.21900826  0.32443573    -0.11198128  0.29819477        -0.28273543         -0.32905390  0.45890998 -0.25270047  0.299567744 -0.095451523 -0.07674321 -0.48673983
volatile.acidity         0.21900826      1.00000000 -0.37798132    -0.19681117  0.37712428        -0.35255731         -0.41447619  0.27129565  0.26145440  0.225983680 -0.037640386 -0.26569948 -0.65383559
citric.acid              0.32443573     -0.37798132  1.00000000     0.14245123  0.03899801         0.13312581          0.19524198  0.09615393 -0.32988819  0.056197300 -0.010493492  0.08553172  0.18739650
residual.sugar          -0.11198128     -0.19681117  0.14245123     1.00000000 -0.12894858         0.48287864          0.49548159  0.55251695 -0.26731984 -0.185927405 -0.359414771 -0.03698048  0.34882101
chlorides                0.29819477      0.37712428  0.03899801    -0.12894858  1.00000000        -0.19584479          0.36261466  0.04470798  0.39559331  0.395593307 -0.256915580 -0.20066550 -0.51267825
free.sulfur.dioxide     -0.28273543     -0.35255731  0.13312581     0.48287864 -0.19584479         1.00000000          0.72093488  0.82571684 -0.14585390 -0.188457249 -0.179838435  0.05546306  0.47164366
total.sulfur.dioxide    -0.32905390     -0.41447619  0.19524198     0.49548159 -0.27963845         0.72093488          1.00000000  0.83239451 -0.23841318 -0.275726820 -0.265739639 -0.04138545  0.70035716
density                  0.45890998      0.27129565  0.09615393     0.55251695  0.36261466         0.82571684          0.83239451  1.00000000  0.01168688  0.259478495 -0.686745422 -0.30585791 -0.39064532
pH                      -0.25270047      0.26145440 -0.32988819    -0.26731984  0.04470798        -0.14585390         -0.23841318  0.01168688  1.00000000  0.192123487  0.121248467  0.01950570 -0.32912865
sulphates                0.29956774      0.22598368  0.05619730    -0.18592741  0.39559331        -0.18845725         -0.27572682  0.25947850  0.19212341  1.000000000 -0.003829195  0.03848545 -0.48721797
alcohol                 -0.09545152     -0.03764839 -0.01049349    -0.35941477 -0.25691558        -0.17983843         -0.26573964 -0.68674542  0.12124847 -0.003829195  1.000000000  0.44431852  0.03296955
quality                 -0.07674321     -0.26569948  0.08553172    -0.03698048 -0.20066550         0.05546306         -0.04138545 -0.30585791  0.01950570  0.038485446  0.444318520  1.00000000  0.11932328
white                   -0.48673983     -0.65383559  0.18739650     0.34882101 -0.51267825         0.47164366          0.70035716 -0.39064532 -0.32912865 -0.487217970  0.032969551  0.11932328  1.00000000
```

Predicting the Quality of Wine - Part 3



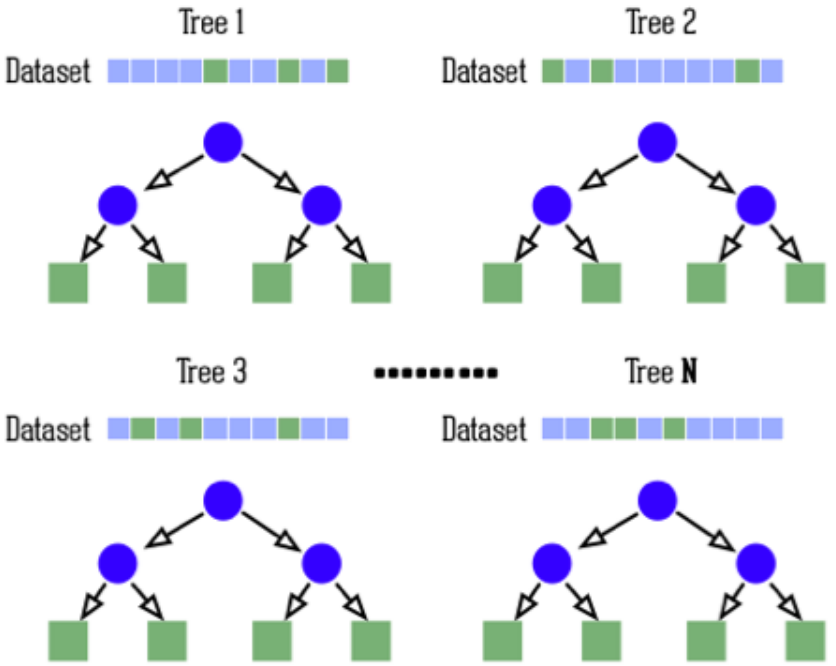


Predicting the Quality of Wine - Part 4 - Problems with Trees

- Deal with irrelevant inputs
- No data preprocessing required
- Scalable computation (fast to build)
- Tolerant with missing values (little loss of accuracy)
- Only a few tunable parameters (easy to learn)
- Allows for human understandable graphic representation

- Data fragmentation for high-dimensional sparse data set (over-fitting)
- Difficult to fit to a trend / piece-wise constant model
- Highly influenced by changes to the data set and local optima (deep trees might be questionable as the errors propagate down)

Aside - How does a random forest work ?



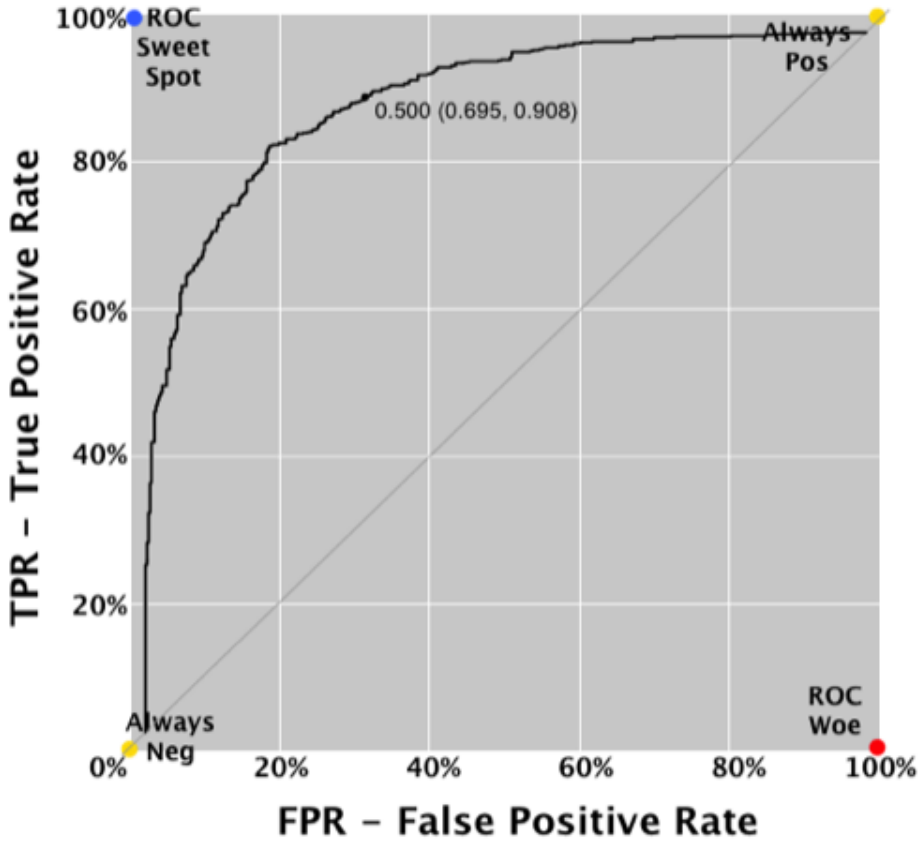
Traverse each tree and at each node in a tree:

i. Select **m** random predictor variables from available set
ii. Use the variable with best split [use objective function]
iii. Move to next node in tree

Predicting the Quality of Wine - Part 5 - Random Forest

Confusion Matrix	REFERENCE (ACTUAL) Predicted Positive Bad	Predicted Negative Good	
PREDICTED Positive Examples Bad	TP 331 / 25%	TN 76 / 6%	Pos
PREDICTED Negative Examples Good	FP 145 / 11%	FN 746 / 58%	Neg
	PPos	PNeg	N

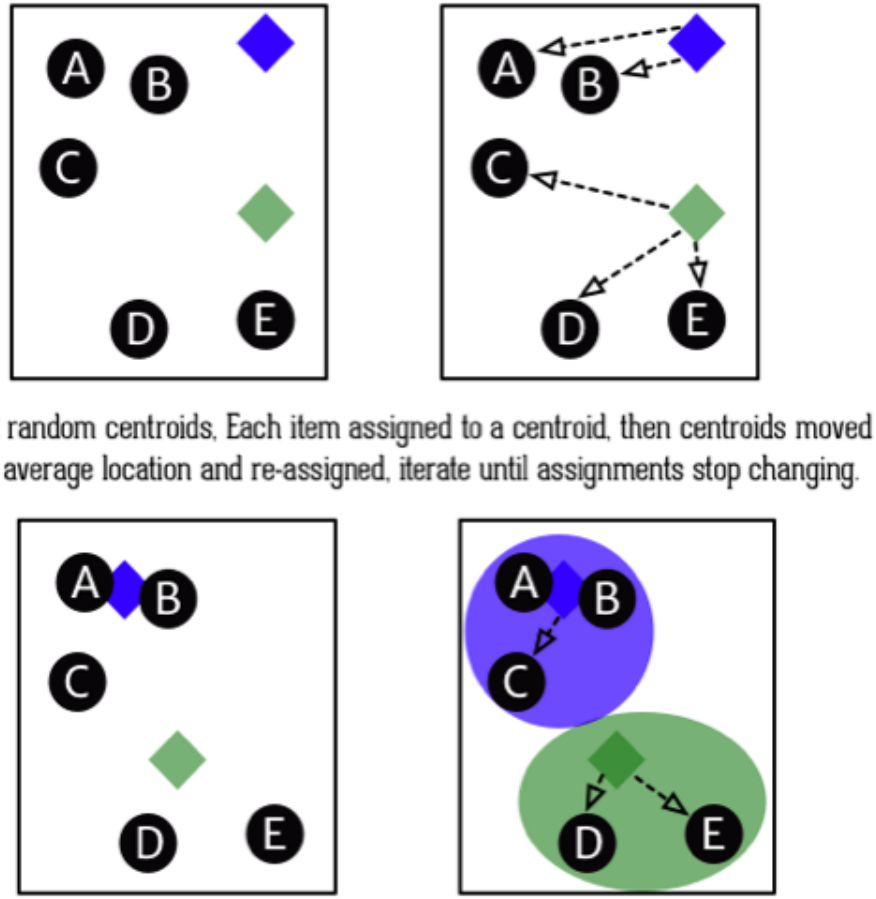
Accuracy = 83%
(25+58)



Predicting the Quality of Wine - Part 6 - Other ML methods

- K-nearest neighbors
  - Unsupervised learning / non-target based learning
  - Distance matrix / cluster analysis using Euclidean distances.
- Neural Nets
  - Looking at basic feed forward simple 3-layer network (input, 'processing', output)
  - Each node / neuron is a set of numerical parameters / weights tuned by the learning algorithm used
- Support Vector Machines
  - Supervised learning
  - non-probabilistic binary linear classifier / nonlinear classifiers by applying the kernel trick
  - constructs a hyper-plane/s in a high-dimensional space

Aside - How does k nearest neighbors work ?

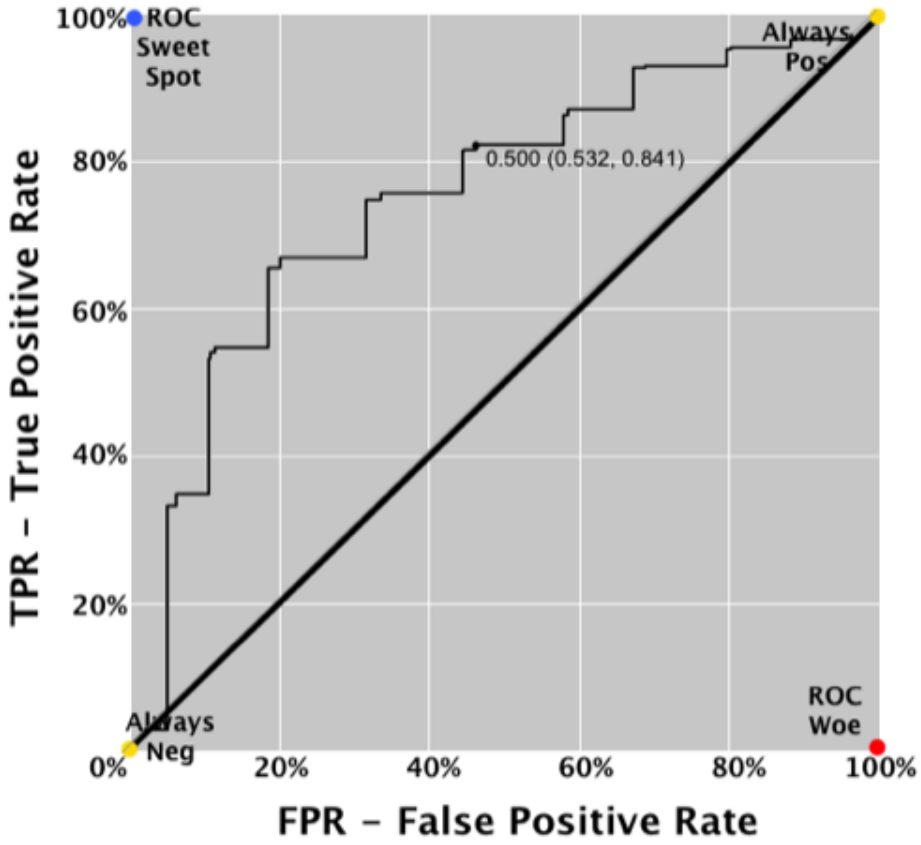


K=2 random centroids. Each item assigned to a centroid, then centroids moved to average location and re-assigned, iterate until assignments stop changing.

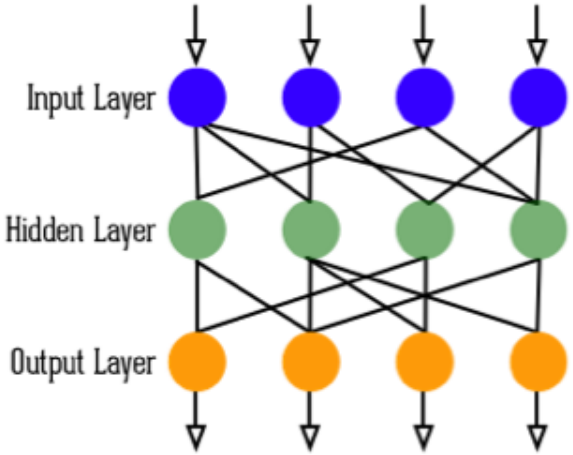
Predicting the Quality of Wine - Part 7 - kNN

Confusion Matrix	REFERENCE (ACTUAL) Predicted Positive Bad	Predicted Negative Good	
PREDICTED Positive Examples Bad	TP 257 / 20%	TN 132 / 10%	Pos
PREDICTED Negative Examples Good	FP 219 / 17%	FN 690 / 53%	Neg
	PPos	PNeg	N

Accuracy = 73% (20+53)



Aside - How do neural networks work ?

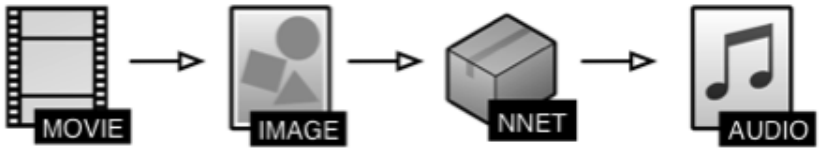


Single layer perceptron, feed forward network

Sum of product of weights and input calculated at each node, if above a threshold then the neuron fires and is activated.

tunepresto Beta
make music by magic

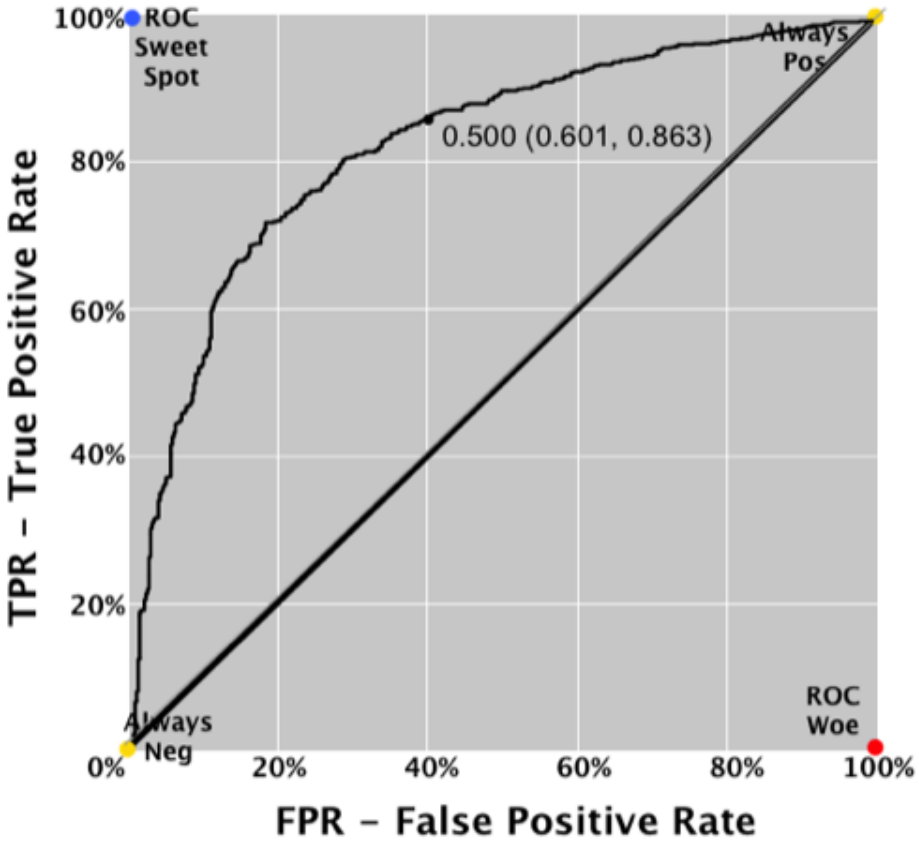
An example of how this can be used:



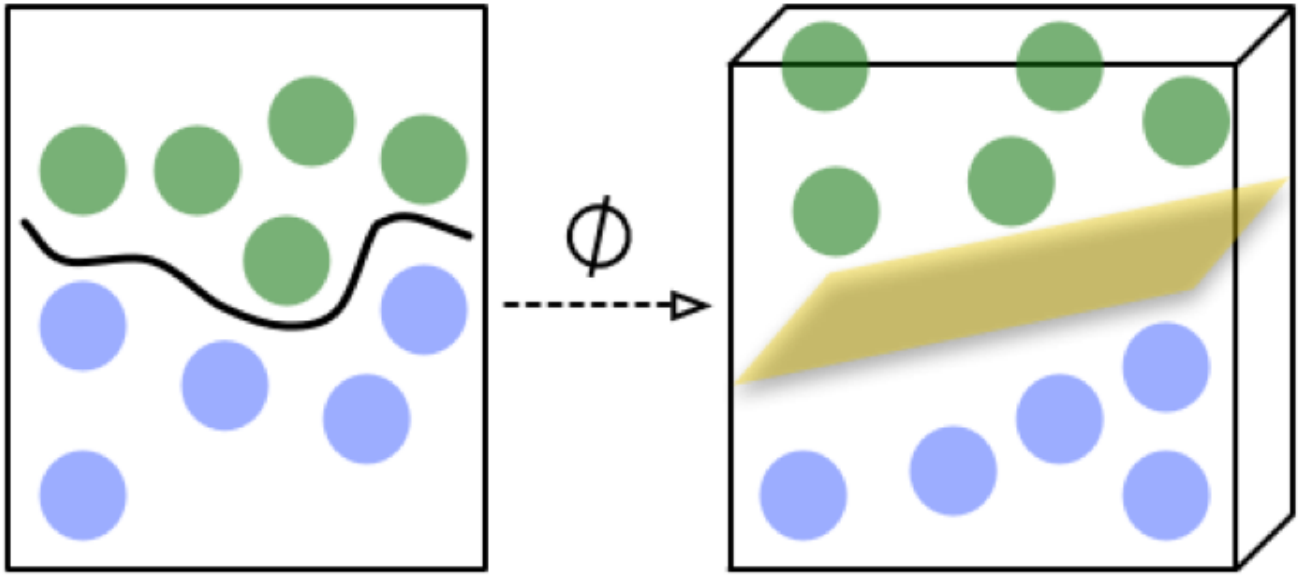
Predicting the Quality of Wine - Part 8 - NNET

Confusion Matrix	REFERENCE (ACTUAL) Predicted Positive Bad	Predicted Negative Good	
PREDICTED Positive Examples Bad	TP 286 / 22%	TN 113 / 9%	Pos
PREDICTED Negative Examples Good	FP 190 / 14%	FN 709 / 55%	Neg
	PPos	PNeg	N

Accuracy = 77%
(22+55)



Aside - How do support vector machines work ?

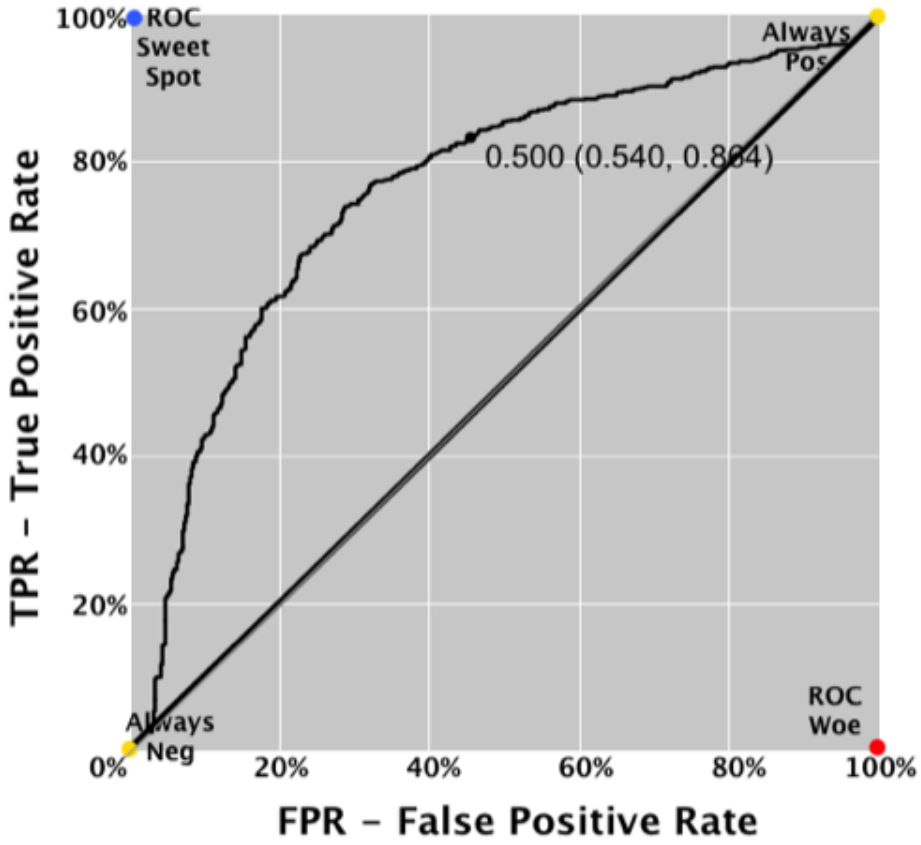


A list of number [a n-dimensional vector] and transform the points into higher dimensions so it is easier to separate them using a [n-1] dimensional hyperplane.

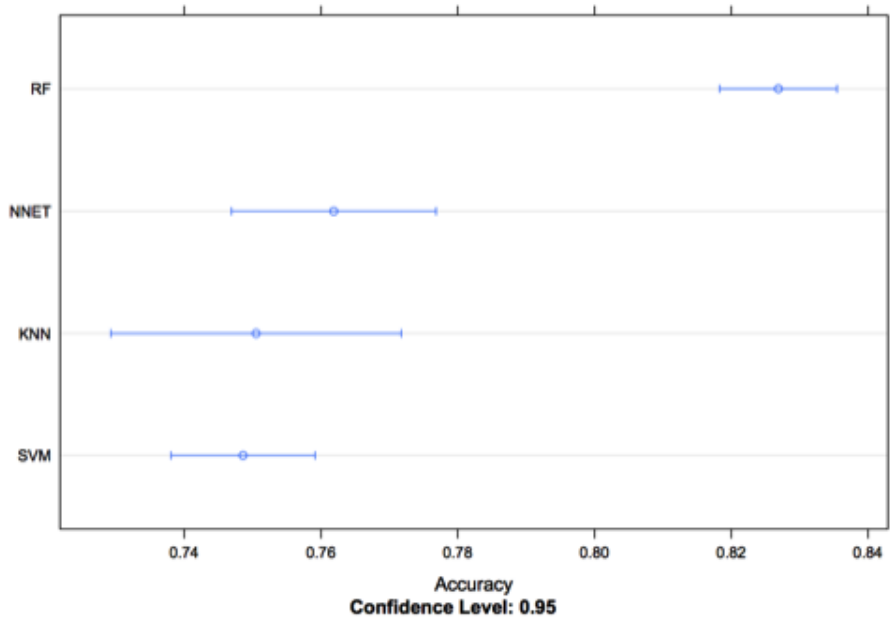
Predicting the Quality of Wine - Part 9 - SVN

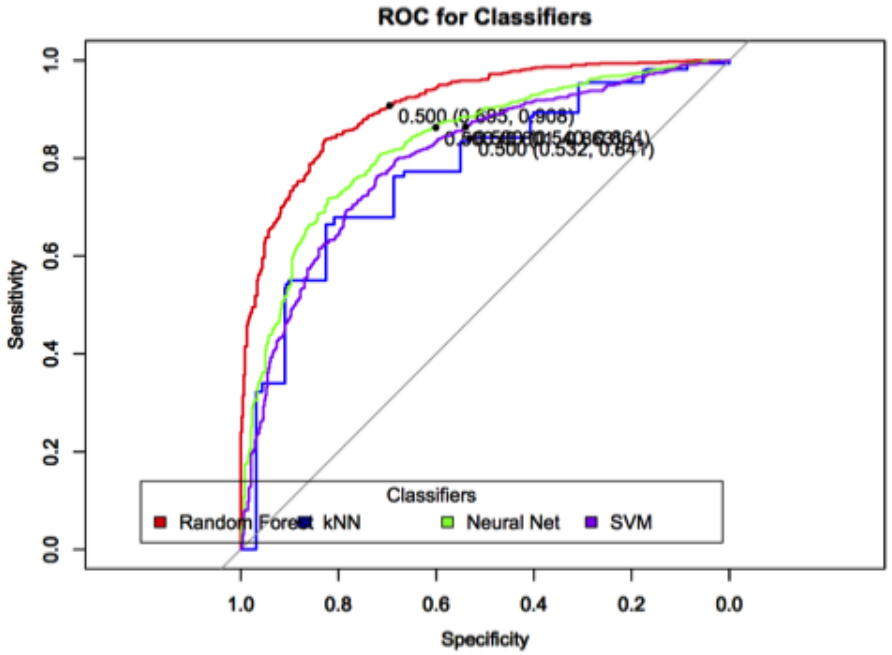
Confusion Matrix	REFERENCE (ACTUAL) Predicted Positive Bad	Predicted Negative Good	
PREDICTED Positive Examples Bad	TP 257 / 20%	TN 112 / 9%	Pos
PREDICTED Negative Examples Good	FP 219 / 16%	FN 710 / 55%	Neg
	PPos	PNeg	N

Accuracy = 75% (20+55)



Predicting the Quality of Wine - Part 10 - All Results





What are they not good for ?

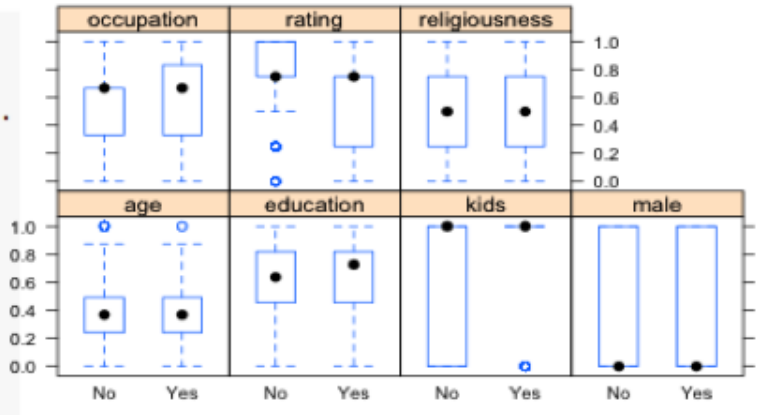
Predicting the Extramarital Affairs

- Fair, R.C. et al (1978), models the possibility of affairs, 9 vars with 601 obs (Training = 481 & Test = 120).
- "A Theory of Extramarital Affairs, Fair, R.C., Journal of Political Economy 1978, 86:45-61".
  - Yes (affairs is >= 1 in last 6 months)
  - No (affairs is < 1 in last 6 months)

```
##
##  No Yes
##  90  30
```

Extramarital Dataset

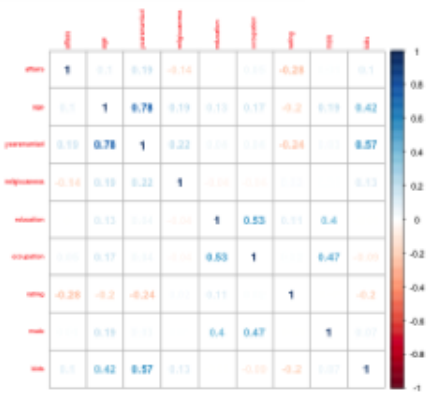
```
> str(Affairs)
'data.frame':   601 obs. of  12 variables:
 $ affairs       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ gender        : Factor w/ 2 levels "female","male": 2 1 1 2 2 1 1 2 1 2 ...
 $ age           : num  37 27 32 57 22 32 22 57 32 22 ...
 $ yearsmarried  : num  10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
 $ children      : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 2 2 1 ...
 $ religiousness: int  3 4 1 5 2 2 2 2 4 4 ...
 $ education     : num  18 14 12 18 17 17 12 14 16 14 ...
 $ occupation    : int  7 6 1 6 6 5 1 4 1 4 ...
 $ rating        : int  4 4 4 5 3 5 3 4 2 5 ...
 $ male          : num  1 0 0 1 1 0 0 1 0 1 ...
 $ kids          : num  0 0 1 1 0 0 0 1 1 0 ...
 $ hadaffair     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```




```
> head(Affairs)
  affairs gender age yearsmarried children religiousness education occupation rating male kids hadaffair
4       0   male  37        10.00       no             3        18          7      4    1    0        No
5       0 female  27         4.00       no             4        14          6      4    0    0        No
11      0 female  32        15.00      yes             1        12          1      4    0    1        No
16      0   male  57        15.00      yes             5        18          6      5    1    1        No
23      0   male  22         0.75       no             2        17          6      3    1    0        No
29      0 female  32         1.50       no             2        17          5      5    0    0        No
```

```
> table(affairs.df.test$hadaffair)

 No Yes
 90  30
```

```
> affairs.corr
                 affairs         age yearsmarried religiousness    education  occupation       rating         male         kids
affairs      1.000000000 0.0952372   0.18684169   -0.144501345 -0.002437441 0.04961176 -0.279512403 0.011736251  0.104010057
age          0.095237204 1.0000000   0.77754585    0.193776931  0.134596015 0.16641254 -0.198999899 0.190641080  0.421930815
yearsmarried 0.186841686 0.7775458   1.00000000    0.218260672  0.040002716 0.04459201 -0.243118827 0.030282521  0.572857364
religiousness -0.144501345 0.1937769 0.21826067   1.000000000 -0.042571079 -0.03972232  0.024295777 0.007679445  0.129351259
education    -0.002437441 0.1345960   0.04000272   -0.042571079 1.000000000 0.53360524  0.109303473 0.397504680 -0.006985882
occupation   0.049611758 0.1664125   0.04459201   -0.039722324  0.533605242 1.00000000  0.017422273 0.467923152 -0.092727118
rating       -0.279512403 -0.1989999 -0.24311883   0.024295777  0.109303473 0.01742227  1.000000000 -0.007523748 -0.196275616
male         0.011736251 0.1906411   0.03028252    0.007679445  0.397504680 0.46792315 -0.007523748 1.000000000  0.069222338
kids         0.104010057 0.4219308   0.57285736    0.129351259 -0.006985882 -0.09272712 -0.196275616 0.069222338  1.000000000
```



Predicting the Extramarital Affairs - RF & NB

Random Forest

```
##          Reference
## Prediction No Yes
##        No  90  30
##        Yes  0   0
```

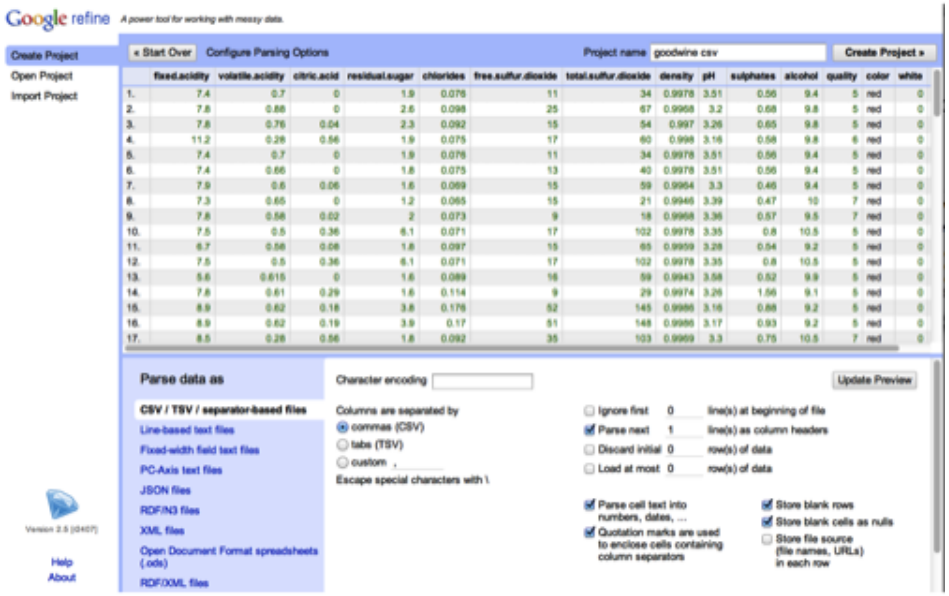
```
## Accuracy
##     0.75
```

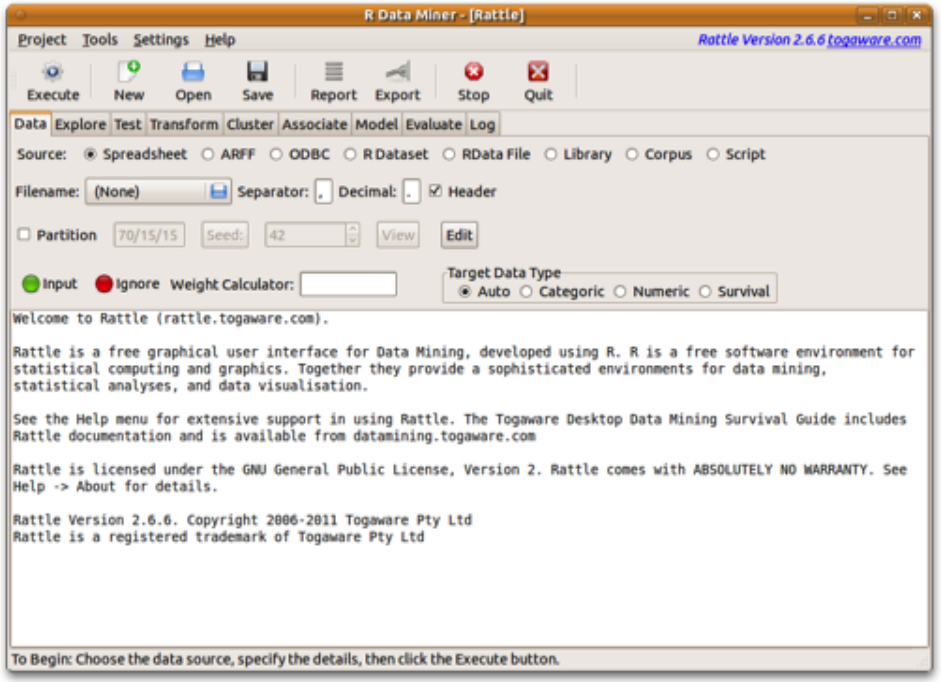
Naive Bayes

```
##          Reference
## Prediction No Yes
##        No  88  29
##        Yes  2   1
```

```
## Accuracy
##     0.75
```

Other related tools: OpenRefine (formerly Google Refine) / Rattle





Other related tools: Command Line Utilities

- http://www.gregreda.com/2013/07/15/unix-commands-for-data-science/
  - sed / awk
  - head / tail
  - wc (word count)
  - grep
  - sort / uniq
- http://blog.comsysto.com/2013/04/25/data-analysis-with-the-unix-shell/
  - join
  - Gnuplot
- http://jeroenjanssens.com/2013/09/19/seven-command-line-tools-for-data-science.html
  - http://csvkit.readthedocs.org/en/latest/
  - https://github.com/jehiah/json2csv
  - http://stedolan.github.io/jq/
  - https://github.com/jeroenjanssens/data-science-toolbox/blob/master/sample
  - https://github.com/bitly/data_hacks
  - https://github.com/jeroenjanssens/data-science-toolbox/blob/master/Rio
  - https://github.com/parmentf/xml2json

A (incomplete) tour of the packages in R

- caret
- party
- rpart
- rpart.plot
- AppliedPredictiveModeling
- randomForest
- corrplot
- arules
- arulesViz
- C50
- pROC
- corrplot
- kernlab
- rattle
- RColorBrewer
- corrgram
- ElemStatLearn
- car

In Summary

An idea of some of the types of classifiers available in ML.

What a confusion matrix and ROC means for a classifier and how to interpret them

An idea of how to test a set of techniques and parameters to help you find the best model for your data

Slides, Data, Scripts are all on GH:
https://github.com/braz/DublinR-ML-treesandforests