Using R's Caret Package for Machine Learning

Earl F Glynn

Kansas City R Users Group

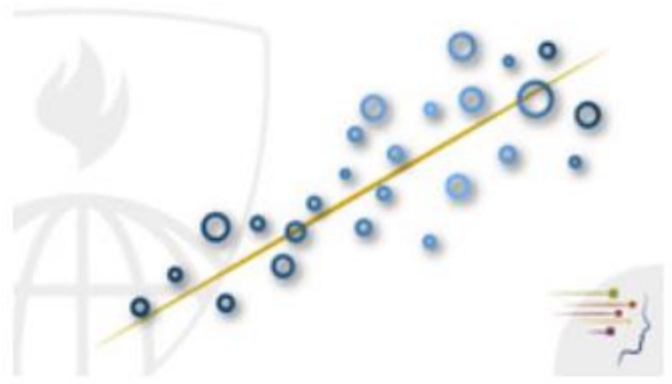
9 September 2017

Using R's Caret Package for Machine Learning

- Regressions Models vs Machine Learning
- Caret Package Overview
  - Visualize Data
  - Preprocess / Transform Data
  - Partition Data into Train and Test Sets
  - Train and Tune Model
  - Evaluate Model Performance
  - Estimate Variable Importance
  - Parallel Processing
- Caret Machine Learning: Forensic Glass Dataset
- Toy Shiny App to Compare Machine Learning Models
- Summary

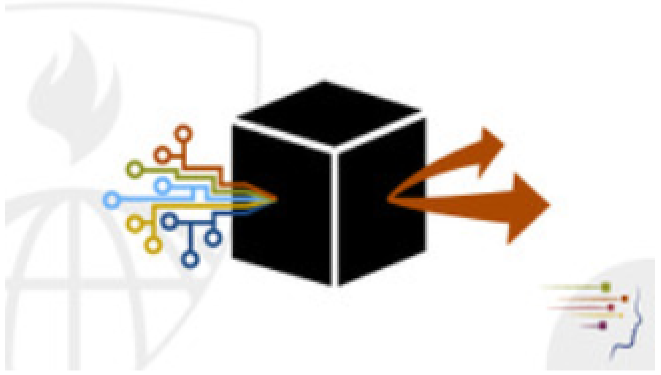
Regression Models vs. Practical Machine Learning

coursera



Johns Hopkins University
Regression Models

Focus: Interpretability

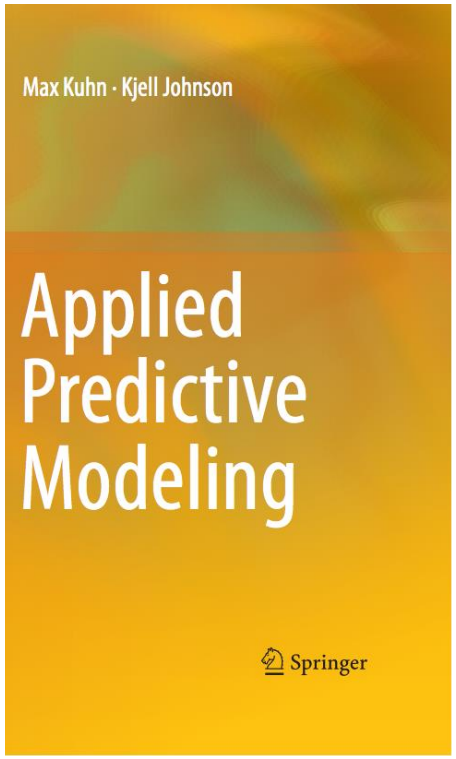


Johns Hopkins University
Practical Machine Learning

Focus: Accurate Predictions

https://www.coursera.org/specializations/jhu-data-science

Caret Package:
Classification And Regression Training



2013

http://appliedpredictivemodeling.com/
http://appliedpredictivemodeling.com/blog/

Max Kuhn
topepo

https://github.com/topepo/caret

Hadley Wickham
@hadleywickham
Following

Super excited to announce that Max Kuhn is joining my team at @rstudio: appliedpredictivemodeling.com/blog/2016/11/2... 🎉🎉😀 #rstats

Working at RStudio
I've joined Hadley's team at RStudio. Unsurprisingly, I'll be working on some modeling related R packages and infrastructure. It is very exciting and I'm looking forward to learning a lot and...
appliedpredictivemodeling.com

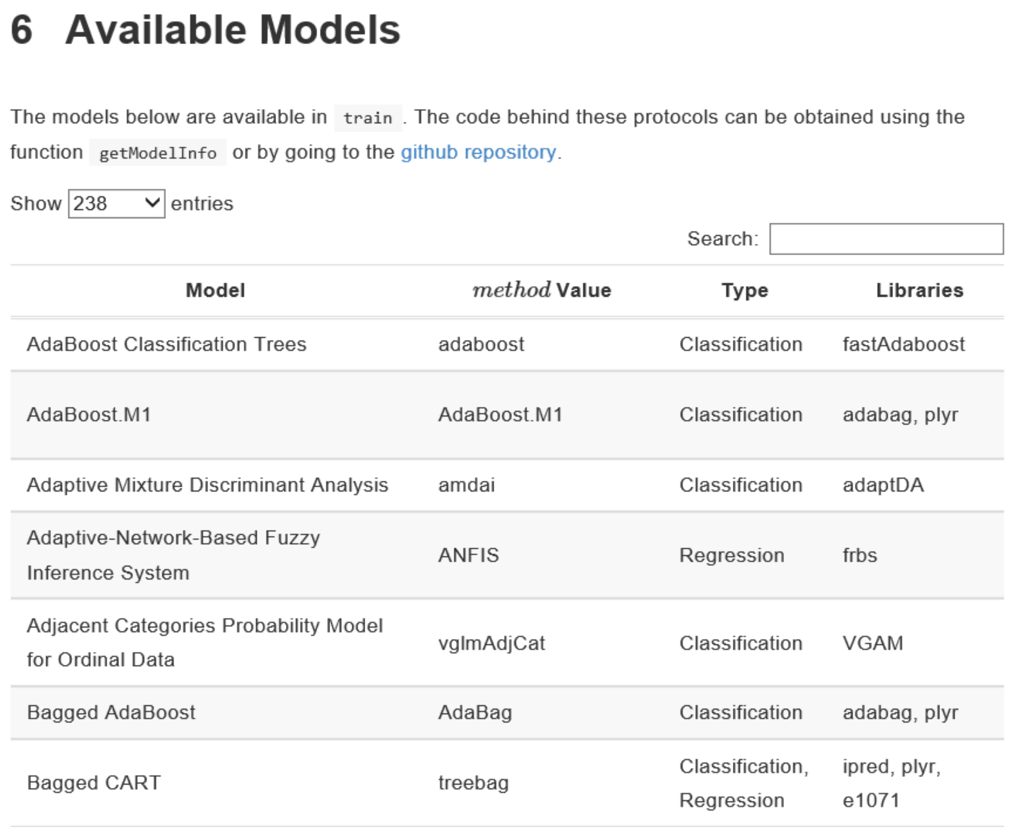
9:23 PM - 28 Nov 2016

Caret Package Features

- Provides uniform interface to machine learning models
- Streamlines model training and tuning
- Standardizes common tasks
- Uses dozens of R packages
- Provides parallel processing

Building Predictive Models in R Using the caret Package
https://www.jstatsoft.org/article/view/v028i05/v28i05.pdf
Max Kuhn, Pfizer Global R&D
Journal of Statistical Software, Nov. 2008

Caret Package Models



6 Available Models

The models below are available in `train`. The code behind these protocols can be obtained using the function `getModelInfo` or by going to the github repository.

Show 238 entries

Search:

Model	*method* Value	Type	Libraries
AdaBoost Classification Trees	adaboost	Classification	fastAdaboost
AdaBoost.M1	AdaBoost.M1	Classification	adabag, plyr
Adaptive Mixture Discriminant Analysis	amdai	Classification	adaptDA
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs
Adjacent Categories Probability Model for Ordinal Data	vglmAdjCat	Classification	VGAM
Bagged AdaBoost	AdaBag	Classification	adabag, plyr
Bagged CART	treebag	Classification, Regression	ipred, plyr, e1071

http://topepo.github.io/caret/modelList.html

Files: **caret-overview.Rmd** and **caret-overview.html**

Forensic Glass Dataset

https://sha.org/bottle/links.htm

fgl dataset in MASS package

- RI = refractive index
- Percentages by weight of oxides: Na, Mg, Al, Si, K, Ca, Ba, Fe
- type
  - window float glass (WinF: 70),
  - window non-float glass (WinNF: 76),
  - vehicle window glass (Veh: 17),
  - containers (Con: 13),
  - tableware (Tabl: 9)
  - vehicle headlamps (Head: 29).

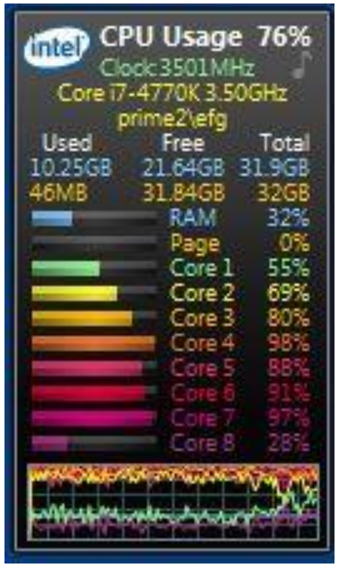
```
   RI    Na   Mg   Al    Si    K   Ca Ba   Fe type
 3.01 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00 WinF
-0.39 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00 WinF
```

Also available through UCI Repository.
Discussed in book Data Mining and Business Analytics with R.

Caret: Parallel Processing

On a PC …

```
# Setup parallel processing
# Let's use 6 cores
library(doParallel)
rCluster <- makePSOCKcluster(6)
registerDoParallel(rCluster)
…
stopCluster(rCluster)
```

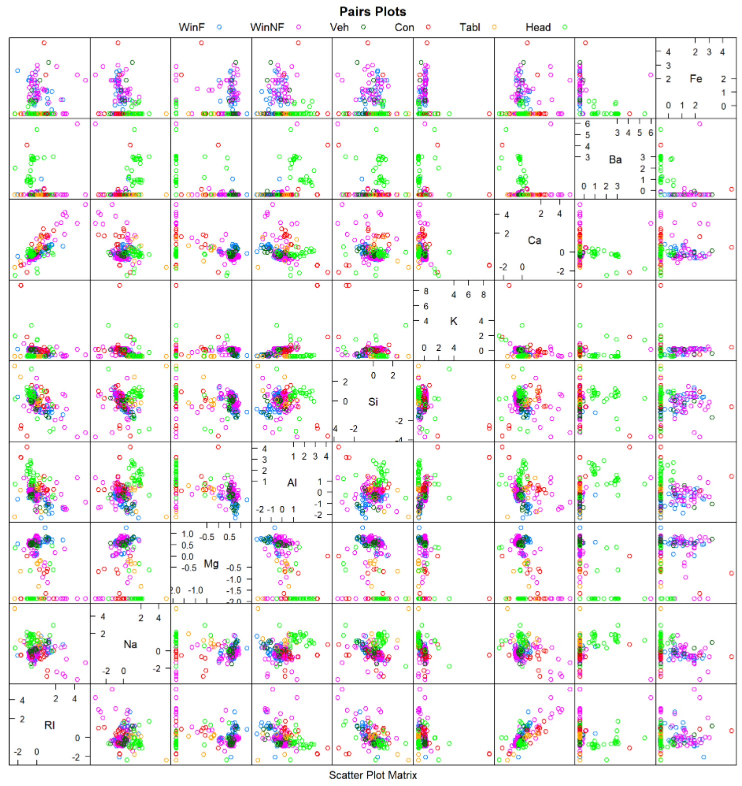


```
CPU Usage 76%
Clock 3501MHz
Core i7-4770K 3.50GHz
prime2\efg
Used     Free     Total
10.25GB  21.64GB  31.9GB
46MB     31.84GB  32GB
RAM      32%
Page     0%
Core 1   55%
Core 2   69%
Core 3   80%
Core 4   98%
Core 5   88%
Core 6   91%
Core 7   97%
Core 8   28%
```

Caret: Visualize Data

Pairs Plots

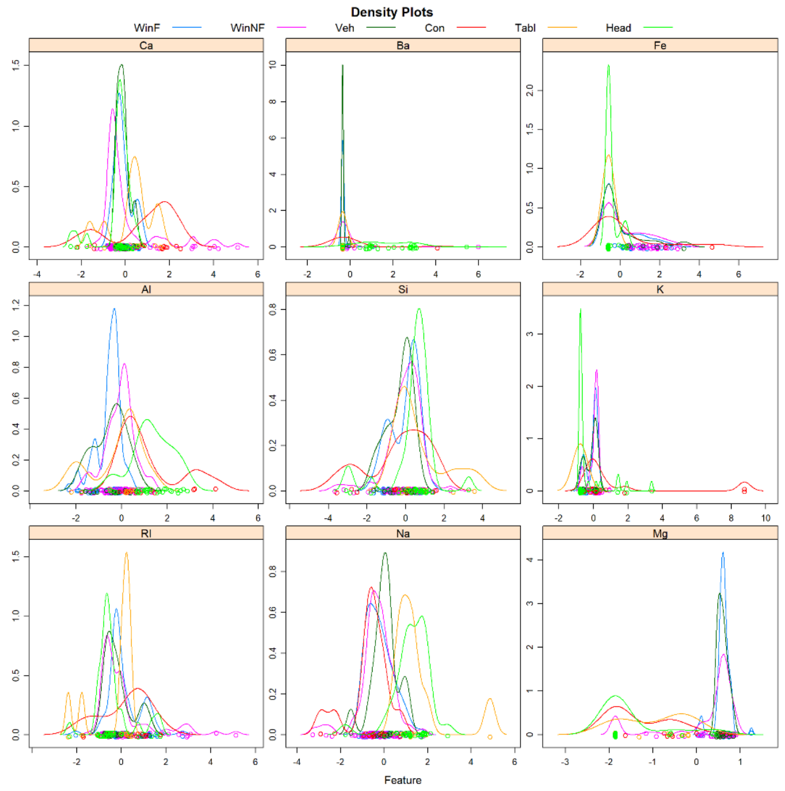
```
featurePlot(centeredScaled, type, "pairs", auto.key=list(columns=6), main="Pairs Plots")
```



Density Plots

See Density estimation.

```
featurePlot(centeredScaled, type, "density",
            scales=list(x=list(relation="free"),
                        y=list(relation="free")),
            auto.key=list(columns=6),
            main="Density Plots")
```



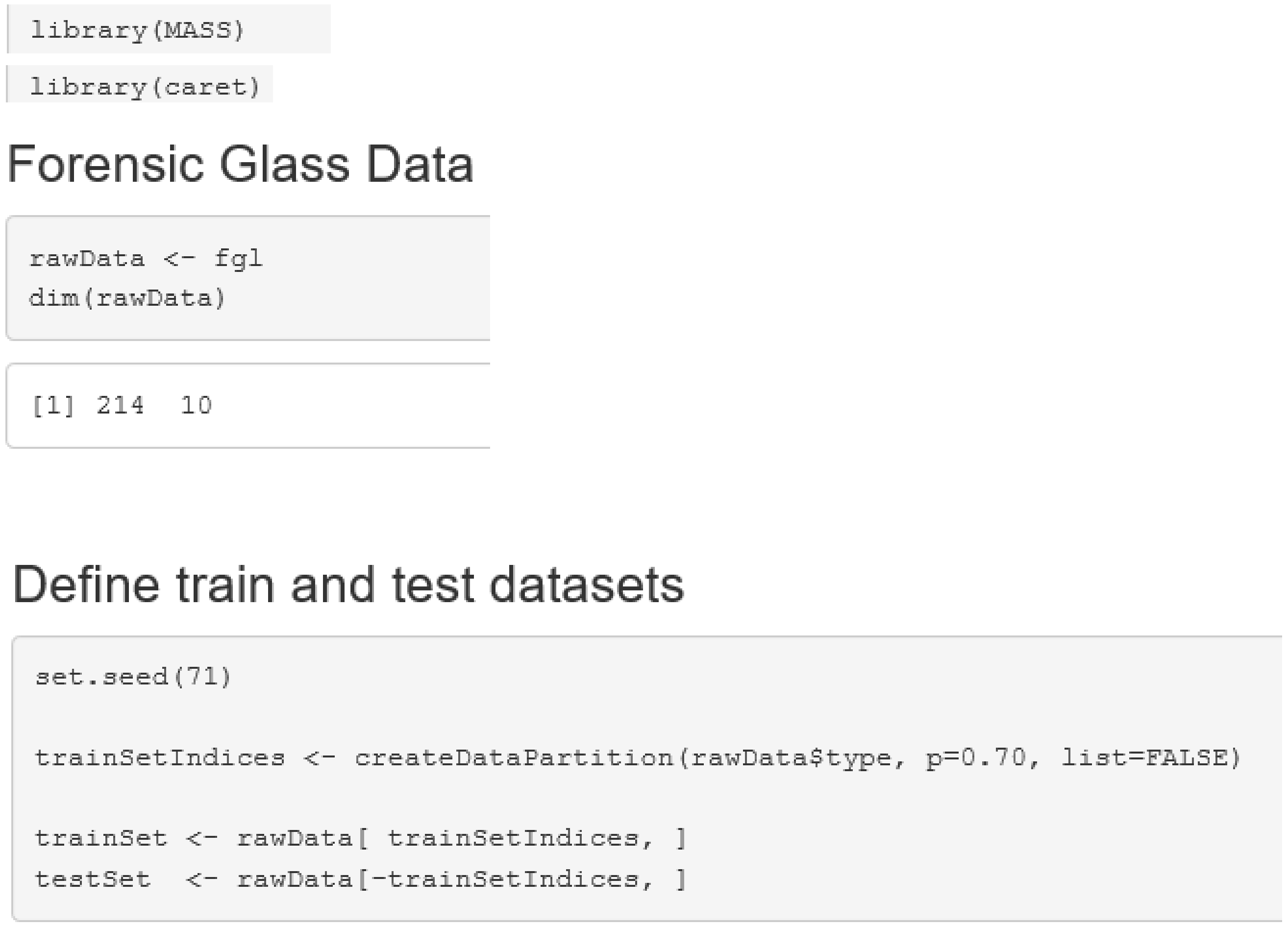
Files: Forensic-Glass-Exploratory.Rmd or **Forensic-Glass-Exploratory.html**

Data Visualization with the Caret R package

https://machinelearningmastery.com/data-visualization-with-the-caret-r-package/

Caret: Partition Data into Train & Test

Use function **createDataPartition** to create splits of the data



```
library(MASS)
```

```
library(caret)
```

Forensic Glass Data

```
rawData <- fgl
dim(rawData)
```

```
[1] 214  10
```

Define train and test datasets

```
set.seed(71)

trainSetIndices <- createDataPartition(rawData$type, p=0.70, list=FALSE)

trainSet <- rawData[ trainSetIndices, ]
testSet  <- rawData[-trainSetIndices, ]
```

Caret: Partition Data into Train & Test

Use function **createDataPartition** to create splits of the data.

http://topepo.github.io/caret/data-splitting.html

Approach Using Forensic Glass Dataset

214 glass samples each with 9 predictors

Split original dataset: 70% training, 30% final test

153 samples
training

61 samples
final test

Use for training and tuning

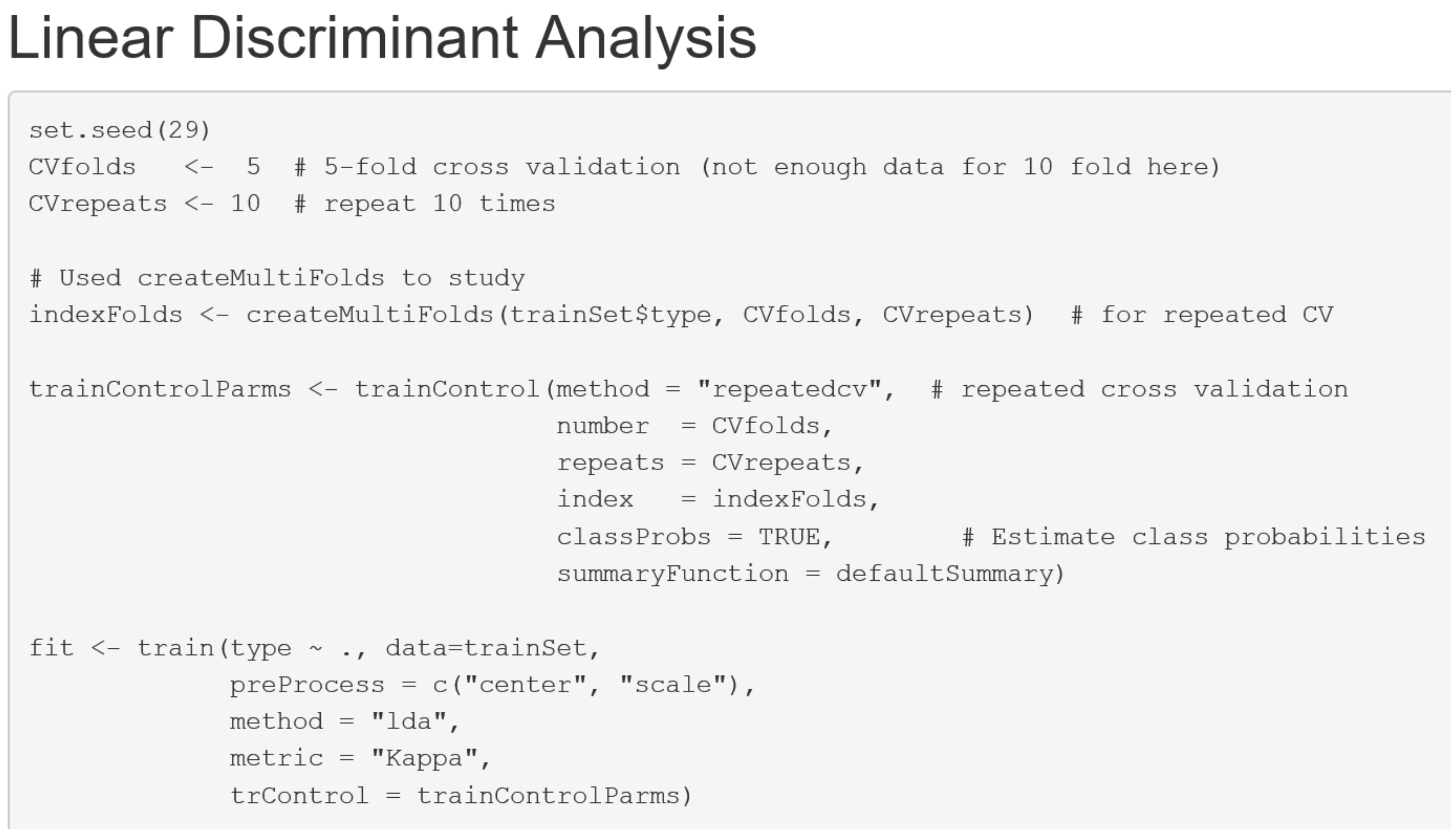
Use once and only once for final test of model. Why?

Caret: Preprocess / Transform Data

- Impute Missing Values
- Create Dummy Variables
- Remove Zero- / Near Zero-Variance Predictors
- Remove Correlated Predictors
- Remove Linear Dependencies
- Center / Scale / Standardize Data
- Transformations (BoxCox, YeoJohnson)

http://topepo.github.io/caret/pre-processing.html

Caret: Train Model

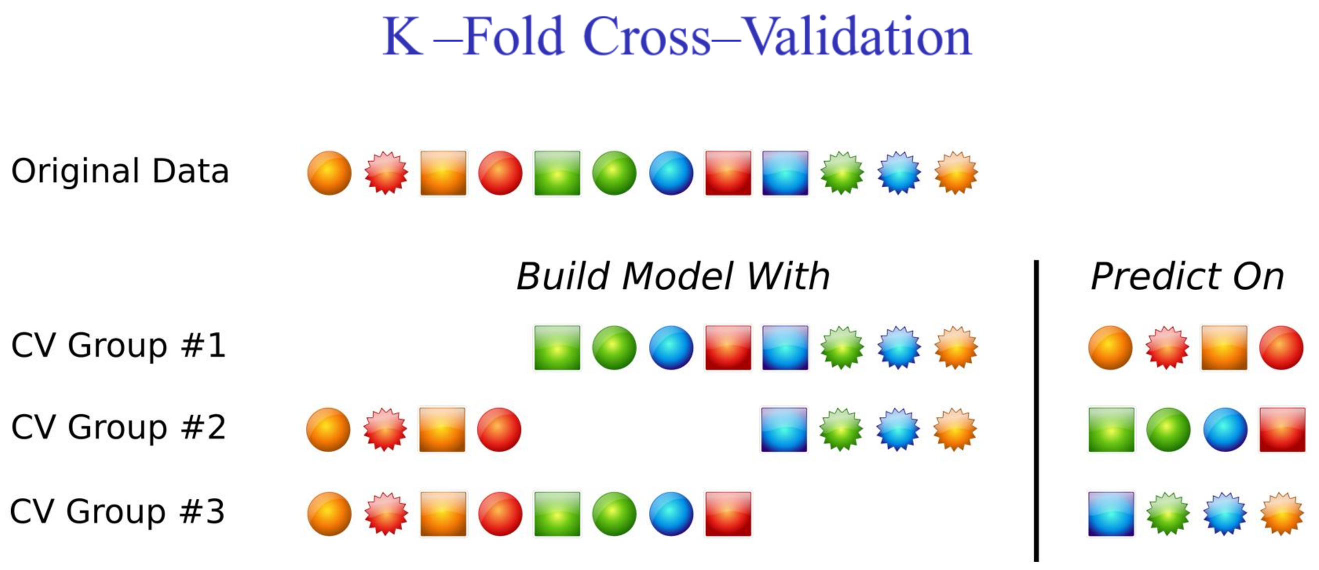


Linear Discriminant Analysis

```
set.seed(29)
CVfolds   <-  5  # 5-fold cross validation (not enough data for 10 fold here)
CVrepeats <- 10  # repeat 10 times

# Used createMultiFolds to study
indexFolds <- createMultiFolds(trainSet$type, CVfolds, CVrepeats)  # for repeated CV

trainControlParms <- trainControl(method = "repeatedcv",   # repeated cross validation
                                  number   = CVfolds,
                                  repeats = CVrepeats,
                                  index    = indexFolds,
                                  classProbs = TRUE,        # Estimate class probabilities
                                  summaryFunction = defaultSummary)

fit <- train(type ~ ., data=trainSet,
             preProcess = c("center", "scale"),
             method = "lda",
             metric = "Kappa",
             trControl = trainControlParms)
```

Files: **Forensic-Glass-caret-rf.Rmd** and **Forensic-Glass-caret-rf.html**

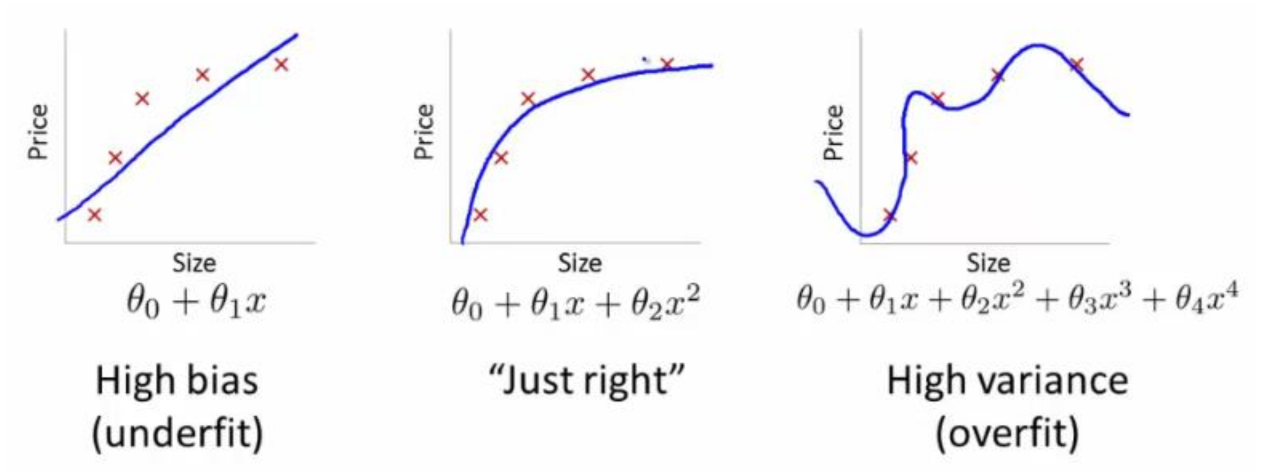
Caret: Train Model



https://opendatascience.com/conferences/predictive-modeling-workshop-max-kuhn/

Underfit vs Overfit

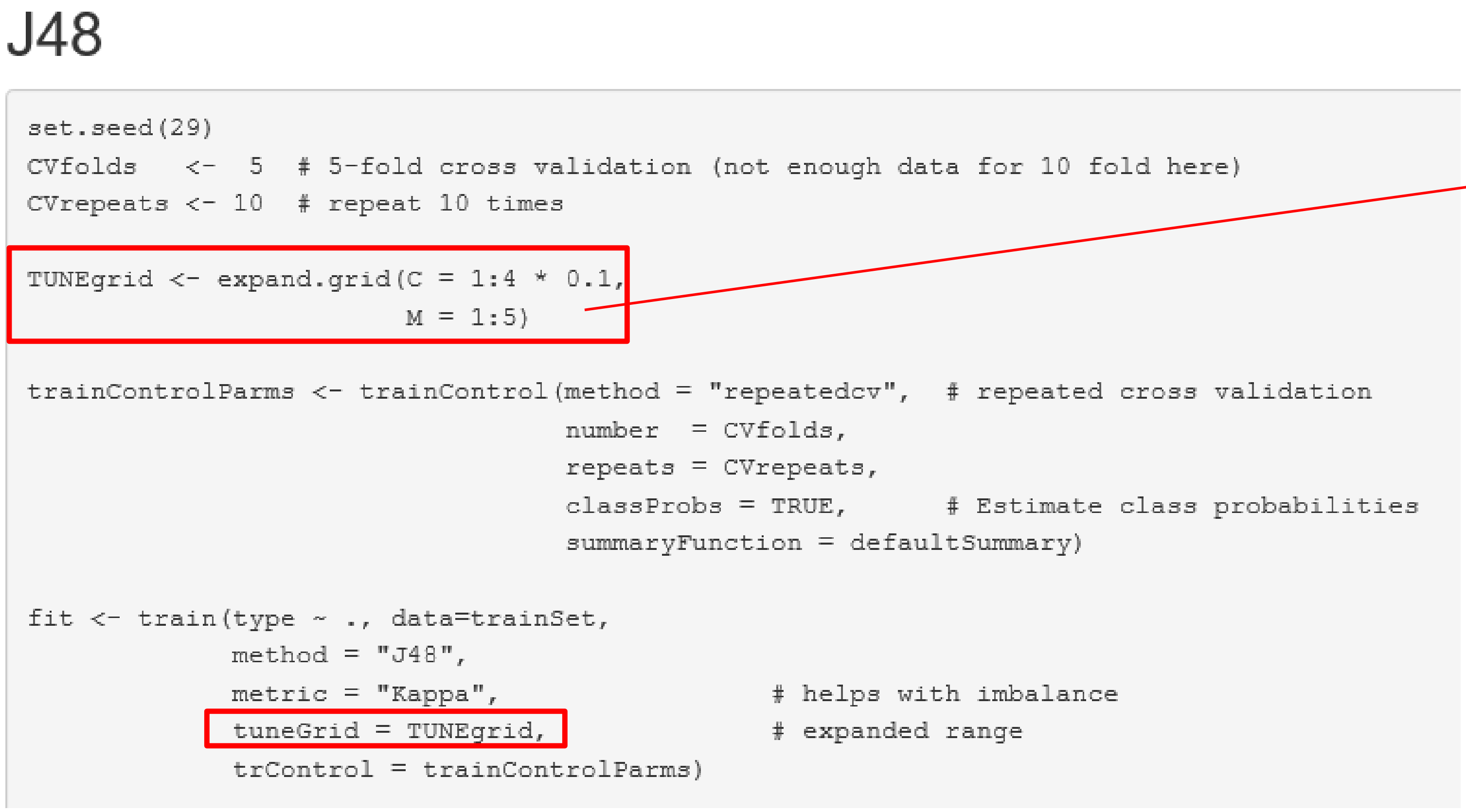
Bias-Variance Tradeoff



https://datascience.stackexchange.com/questions/361/when-is-a-model-underfitted

Caret: Tune Model

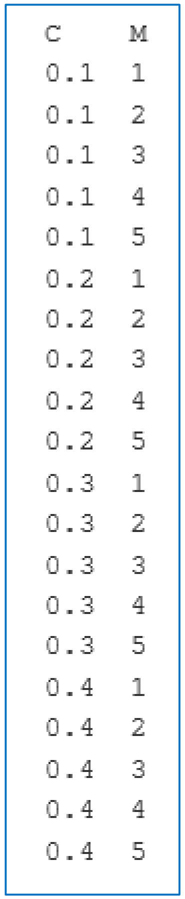
Often default tuning is adequate, but when not …



J48

```
set.seed(29)
CVfolds   <-  5  # 5-fold cross validation (not enough data for 10 fold here)
CVrepeats <- 10  # repeat 10 times

TUNEgrid <- expand.grid(C = 1:4 * 0.1,
                        M = 1:5)

trainControlParms <- trainControl(method = "repeatedcv",   # repeated cross validation
                                  number  = CVfolds,
                                  repeats = CVrepeats,
                                  classProbs = TRUE,      # Estimate class probabilities
                                  summaryFunction = defaultSummary)

fit <- train(type ~ ., data=trainSet,
             method = "J48",
             metric = "Kappa",                # helps with imbalance
             tuneGrid = TUNEgrid,             # expanded range
             trControl = trainControlParms)
```

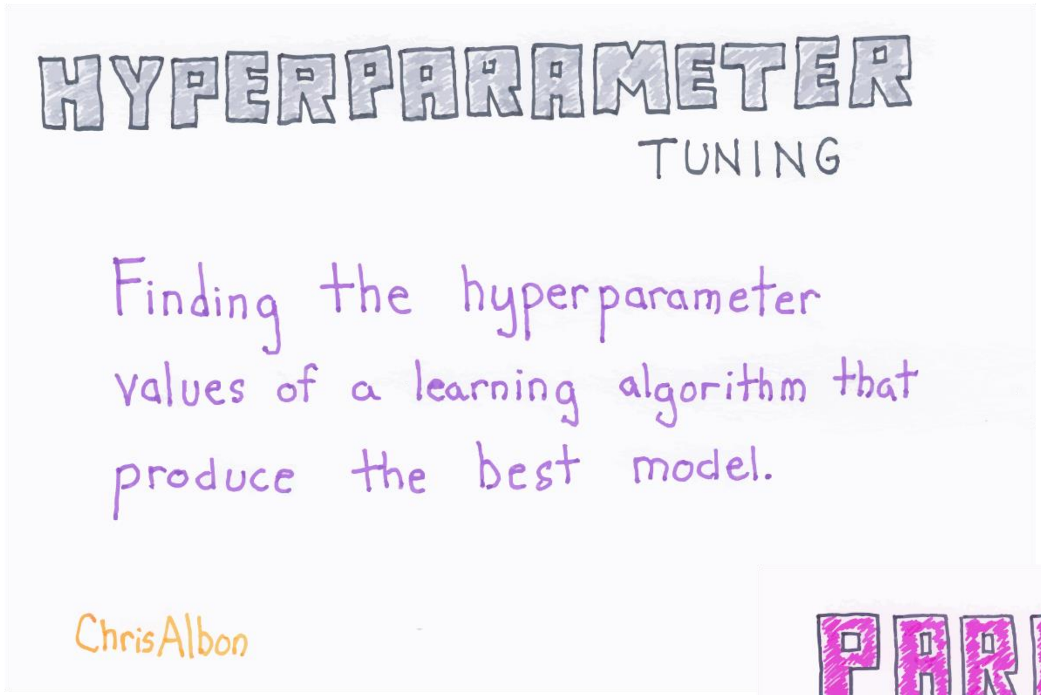
Hyperparameters



C	M
0.1	1
0.1	2
0.1	3
0.1	4
0.1	5
0.2	1
0.2	2
0.2	3
0.2	4
0.2	5
0.3	1
0.3	2
0.3	3
0.3	4
0.3	5
0.4	1
0.4	2
0.4	3
0.4	4
0.4	5

Files: **Forensic-Glass-caret-J48.Rmd** and **Forensic-Glass-caret-J48f.html**

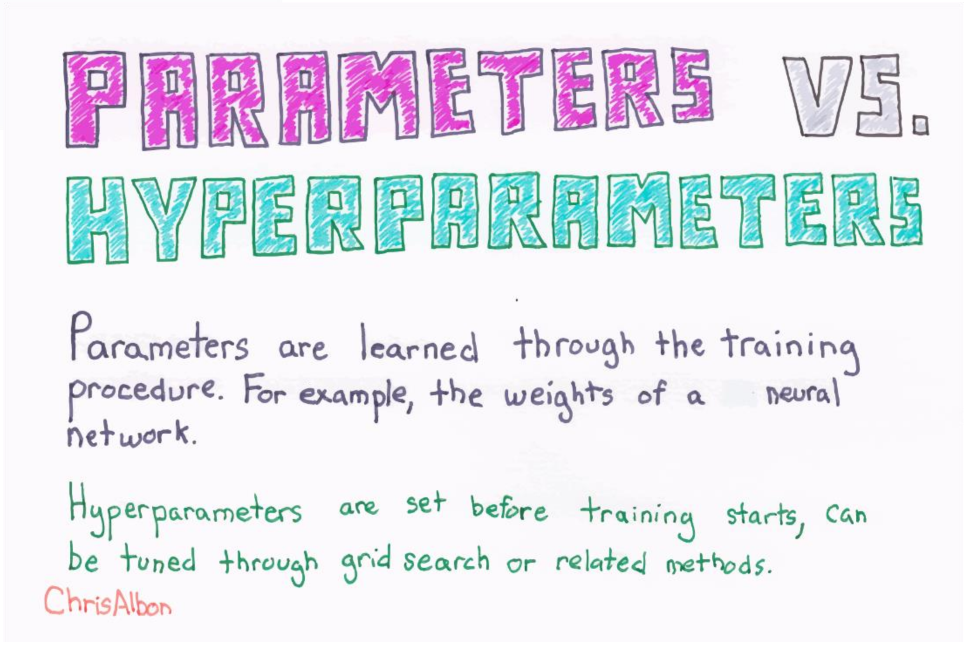
Tune Machine Learning Algorithms in R (random forest case study)
https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/



HYPERPARAMETER TUNING

Finding the hyperparameter values of a learning algorithm that produce the best model.

ChrisAlbon



PARAMETERS VS. HYPERPARAMETERS

Parameters are learned through the training procedure. For example, the weights of a neural network.

Hyperparameters are set before training starts, can be tuned through grid search or related methods.

ChrisAlbon

@ChrisAlbon
MachineLearningFlashcards.com

Caret: Evaluate Model Performance

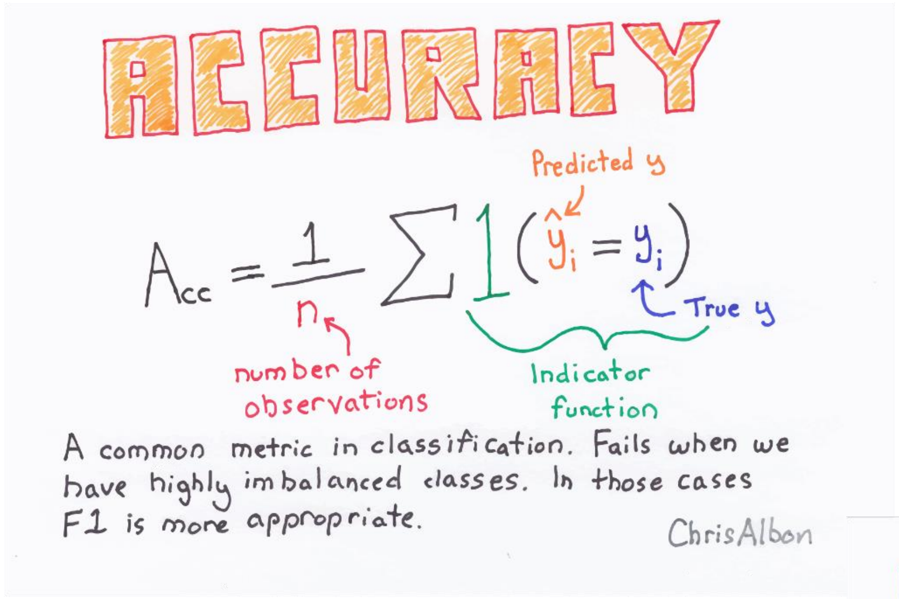
Classification Metrics

- Accuracy
- Area Under Curve (AUC) [2 classes]
- Kappa (better than Accuracy when groups not balanced)
- LogLoss

Regression Metrics

- RMSE (Root Mean Squared Error)
- R^2

Caret: Evaluate Model Performance



ACCURACY

$$Acc = \frac{1}{n} \sum \underbrace{1(\hat{y}_i = y_i)}_{\text{Indicator function}}$$

A common metric in classification. Fails when we have highly imbalanced classes. In those cases F1 is more appropriate.

ChrisAlbon

Kappa

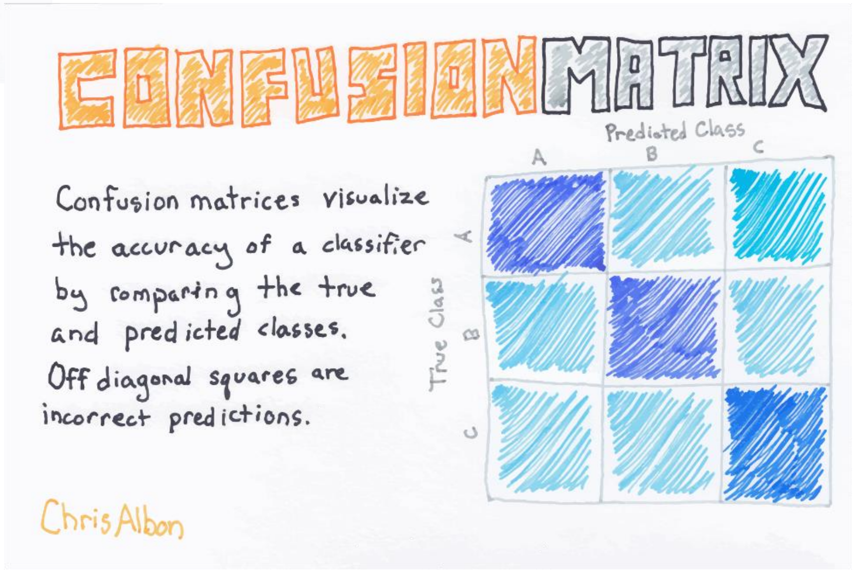
For classification models:

- **overall accuracy** can be used, but this may be problematic when the classes are not balanced.
- the **Kappa statistic** takes into account the expected error rate:

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

where O is the observed accuracy and E is the expected accuracy under chance agreement (psych:::cohen.kappa, vcd:::Kappa, . . .)

https://www.r-project.org/conferences/useR-2013/Tutorials/kuhn/user_caret_2up.pdf



CONFUSION MATRIX

Confusion matrices visualize the accuracy of a classifier by comparing the true and predicted classes. Off diagonal squares are incorrect predictions.

ChrisAlbon

@ChrisAlbon
MachineLearningFlashcards.com

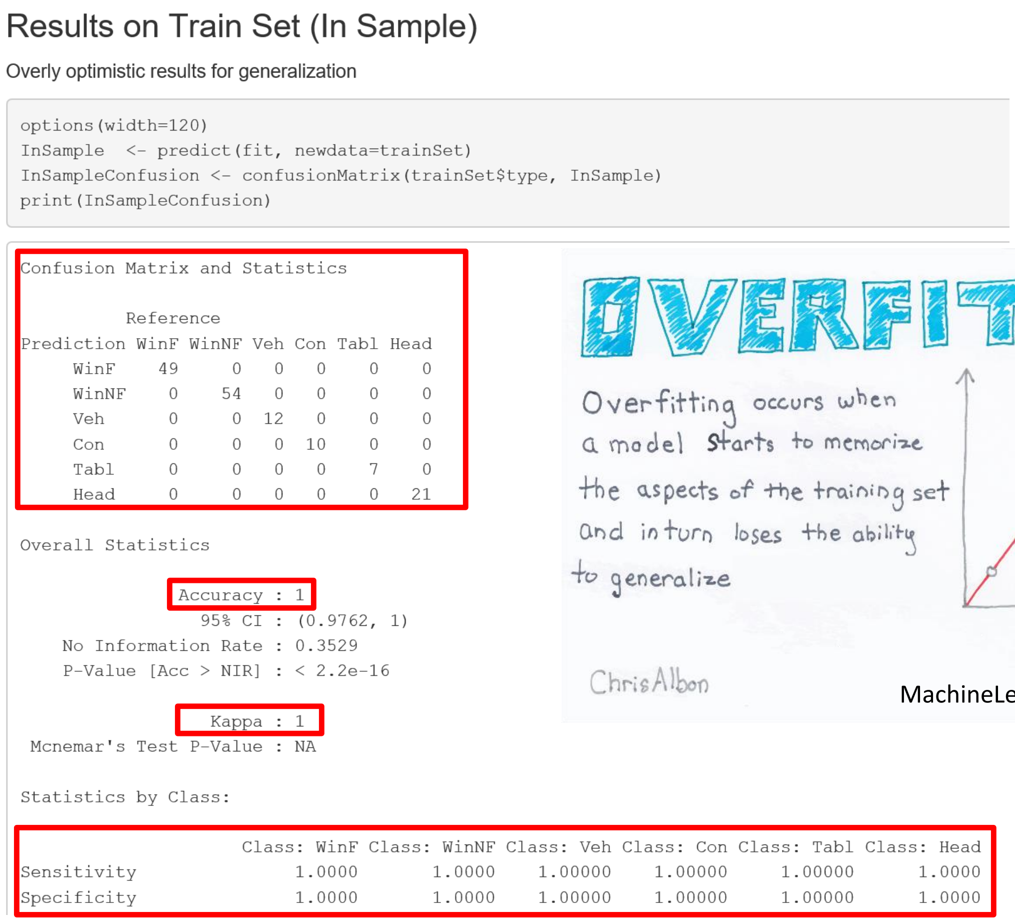
Caret: Evaluate Model Performance

- **Sensitivity**: given that a result is truly an event, what is the probability that the model will predict an event results?
- **Specificity**: given that a result is truly not an event, what is the probability that the model will predict a negative results?

https://www.r-project.org/conferences/useR-2013/Tutorials/kuhn/user_caret_2up.pdf

Caret: Evaluate Model Performance

Train: Overly Optimistic Results for Generaliztion

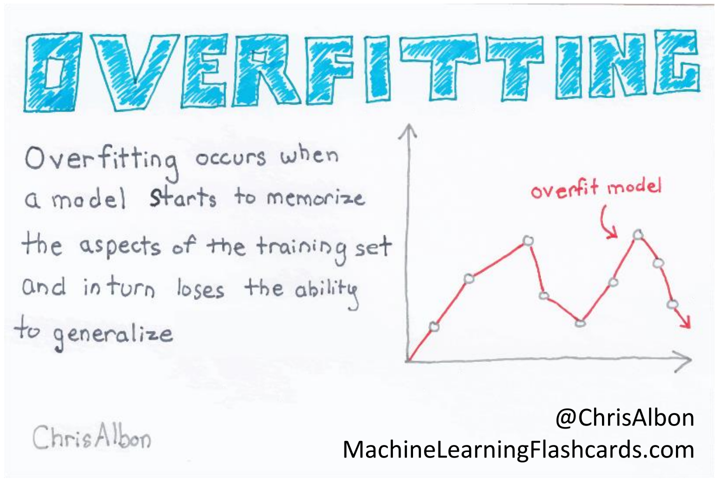


Results on Train Set (In Sample)

Overly optimistic results for generalization

```
options(width=120)
InSample   <- predict(fit, newdata=trainSet)
InSampleConfusion <- confusionMatrix(trainSet$type, InSample)
print(InSampleConfusion)
```

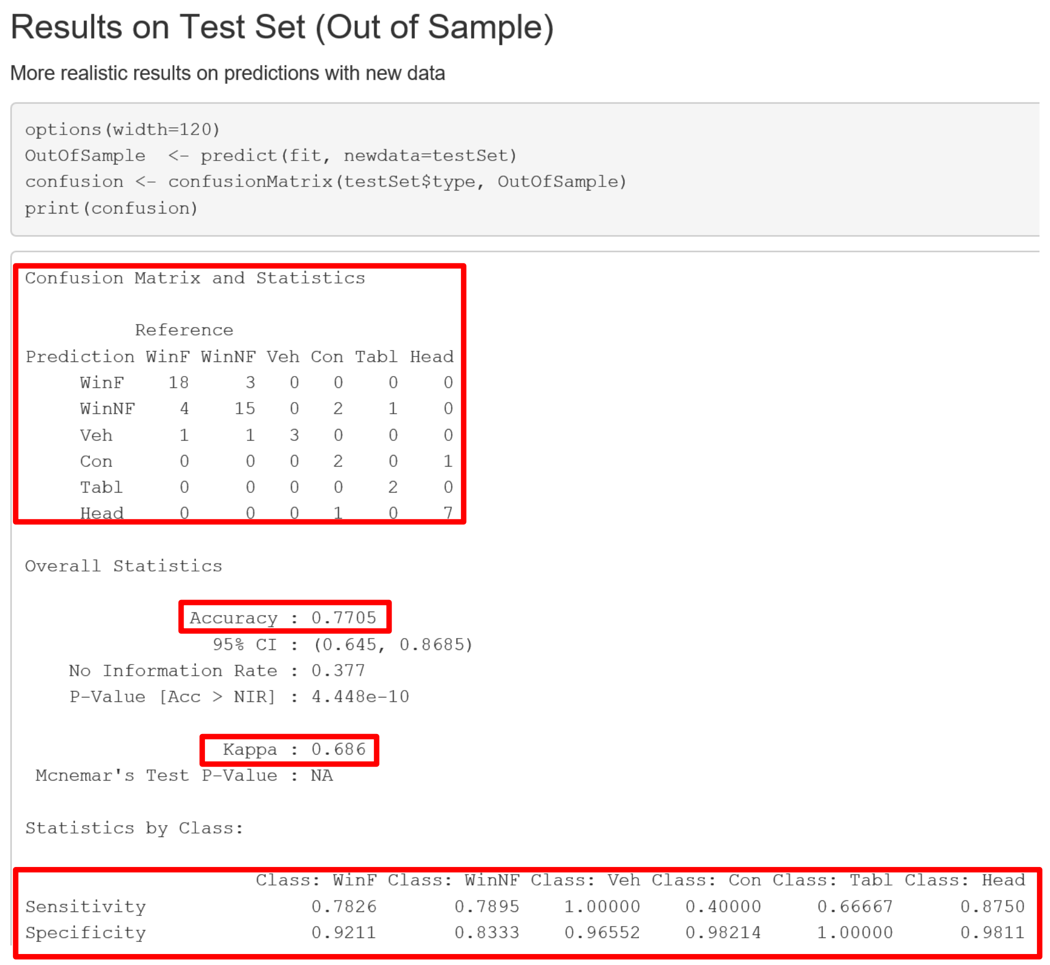
```
Confusion Matrix and Statistics

          Reference
Prediction WinF WinNF Veh Con Tabl Head
     WinF    49     0   0   0    0    0
     WinNF    0    54   0   0    0    0
     Veh      0     0  12   0    0    0
     Con      0     0   0  10    0    0
     Tabl     0     0   0   0    7    0
     Head     0     0   0   0    0   21

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9762, 1)
    No Information Rate : 0.3529
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: WinF Class: WinNF Class: Veh Class: Con Class: Tabl Class: Head
Sensitivity               1.0000       1.0000    1.00000    1.00000     1.00000      1.0000
Specificity               1.0000       1.0000    1.00000    1.00000     1.00000      1.0000
```


OVERFITTING
Overfitting occurs when a model starts to memorize the aspects of the training set and in turn loses the ability to generalize
overfit model
ChrisAlbon
@ChrisAlbon
MachineLearningFlashcards.com

Files: **Forensic-Glass-caret-rf.Rmd** and **Forensic-Glass-caret-rf.html**

Caret: Evaluate Model Performance

Test: More Realistic Results on Predictions with New Data



Results on Test Set (Out of Sample)

More realistic results on predictions with new data

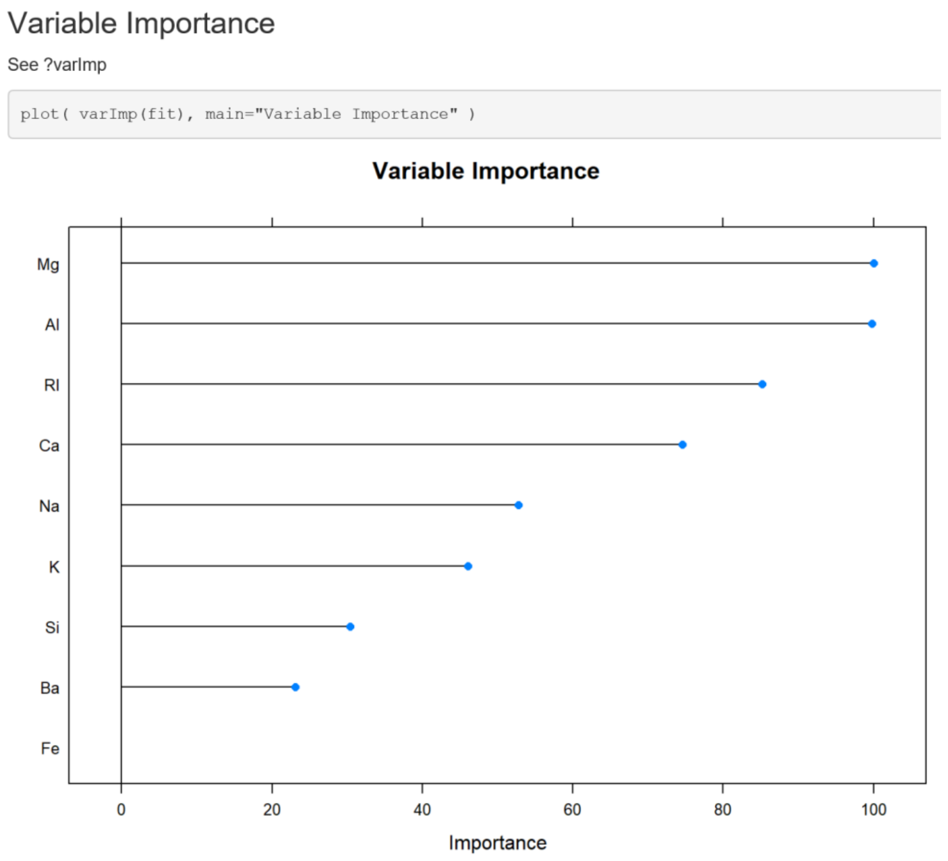
```
options(width=120)
OutOfSample  <- predict(fit, newdata=testSet)
confusion <- confusionMatrix(testSet$type, OutOfSample)
print(confusion)
```

```
Confusion Matrix and Statistics

          Reference
Prediction WinF WinNF Veh Con Tabl Head
     WinF    18     3   0   0    0    0
     WinNF    4    15   0   2    1    0
     Veh      1     1   3   0    0    0
     Con      0     0   0   2    0    1
     Tabl     0     0   0   0    2    0
     Head     0     0   0   1    0    7

Overall Statistics

               Accuracy : 0.7705
                 95% CI : (0.645, 0.8685)
    No Information Rate : 0.377
    P-Value [Acc > NIR] : 4.448e-10

                  Kappa : 0.686
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: WinF Class: WinNF Class: Veh Class: Con Class: Tabl Class: Head
Sensitivity               0.7826       0.7895    1.00000    0.40000     0.66667      0.8750
Specificity               0.9211       0.8333    0.96552    0.98214     1.00000      0.9811
```

Files: **Forensic-Glass-caret-rf.Rmd** and **Forensic-Glass-caret-rf.html**

Caret: Estimate Variable Importance

Random Forest: Single Prediction Model


Variable Importance

See ?varImp

```
plot( varImp(fit), main="Variable Importance" )
```

Files: **Forensic-Glass-caret-rf.Rmd** and **Forensic-Glass-caret-rf.html**

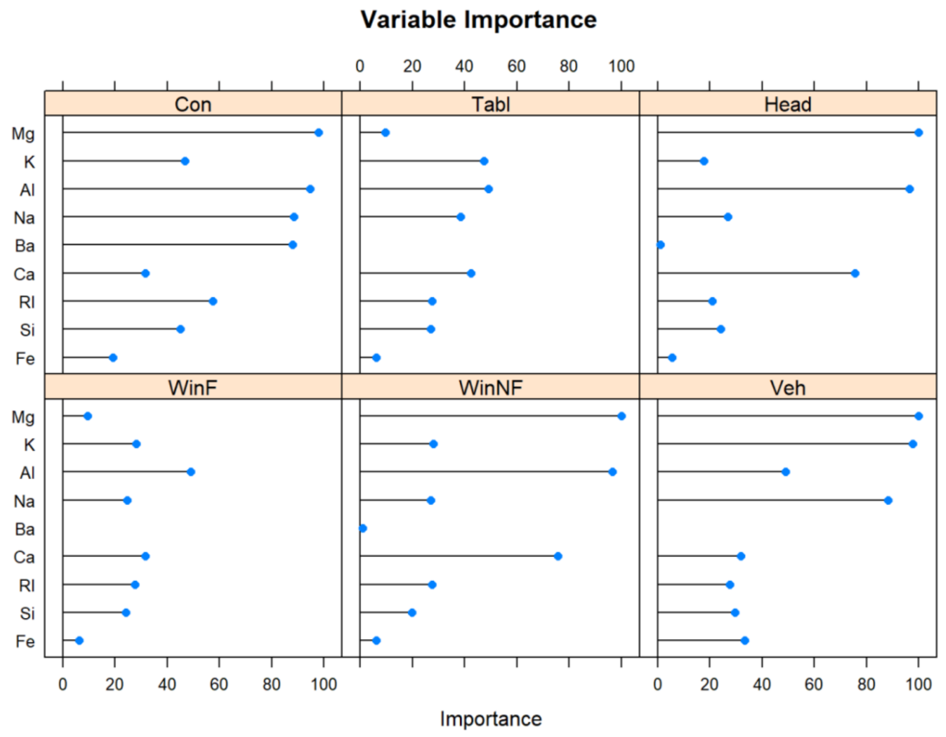
Caret: Estimate Variable Importance

J48: Separate Model for Each Type

Variable Importance

See ?varImp

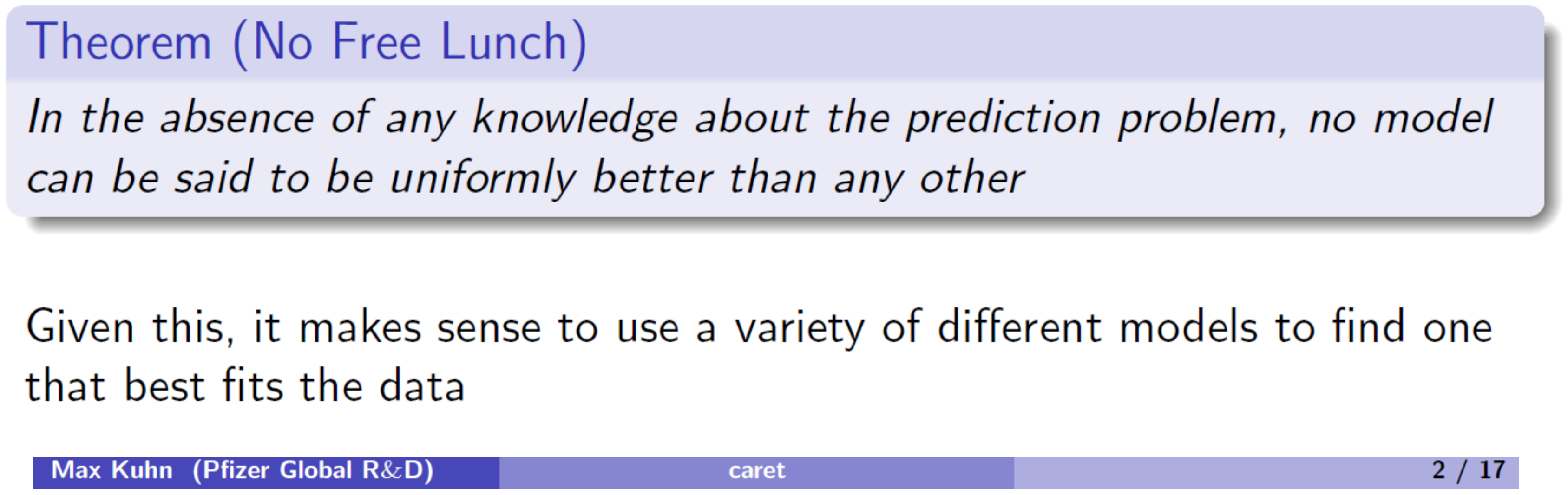
```
plot( varImp(fit), main="Variable Importance" )
```



Files: **Forensic-Glass-caret-J48.Rmd** and **Forensic-Glass-caret-J48.html**

Caret Machine Learning

Which Machine Learning Algorithms to use?



Theorem (No Free Lunch)

In the absence of any knowledge about the prediction problem, no model can be said to be uniformly better than any other

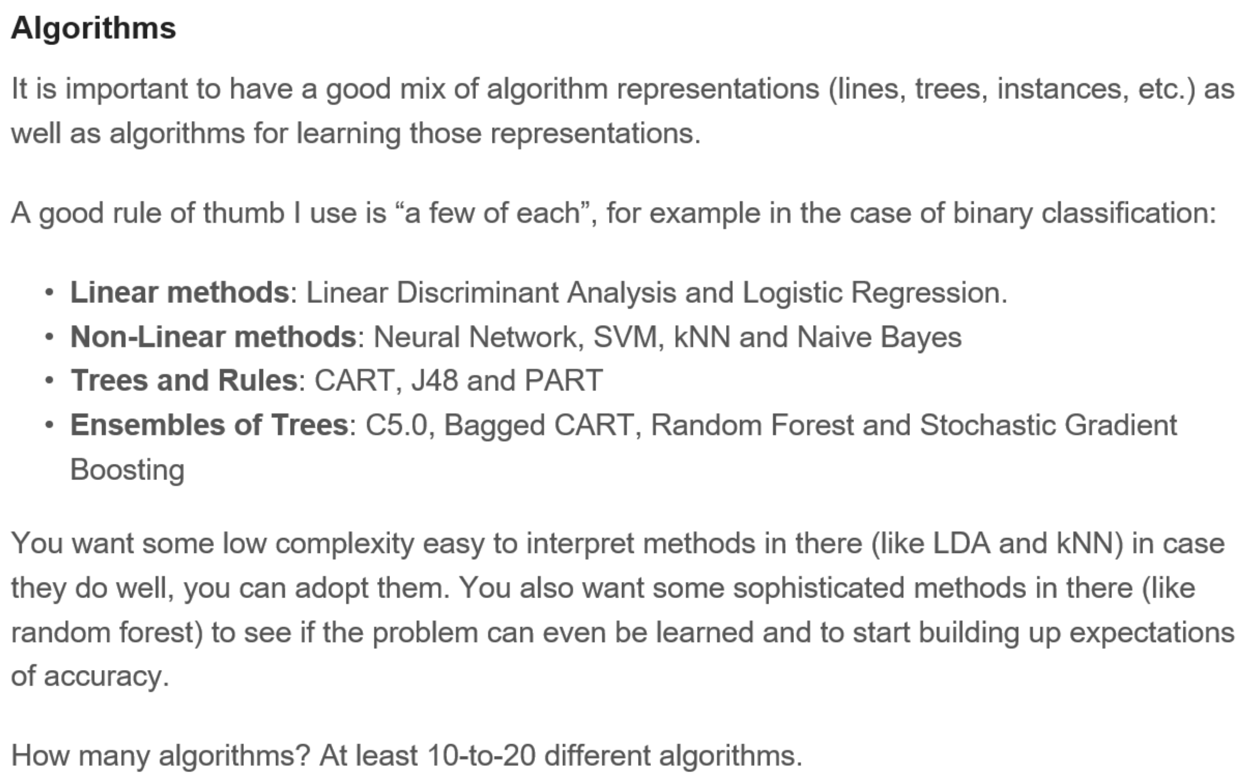
Given this, it makes sense to use a variety of different models to find one that best fits the data

Max Kuhn (Pfizer Global R&D) caret 2 / 17

https://www.r-project.org/conferences/useR-2010/slides/Kuhn.pdf

Caret Machine Learning

Which Machine Learning Algorithms to use?



Algorithms

It is important to have a good mix of algorithm representations (lines, trees, instances, etc.) as well as algorithms for learning those representations.

A good rule of thumb I use is “a few of each”, for example in the case of binary classification:

- **Linear methods**: Linear Discriminant Analysis and Logistic Regression.
- **Non-Linear methods**: Neural Network, SVM, kNN and Naive Bayes
- **Trees and Rules**: CART, J48 and PART
- **Ensembles of Trees**: C5.0, Bagged CART, Random Forest and Stochastic Gradient Boosting

You want some low complexity easy to interpret methods in there (like LDA and kNN) in case they do well, you can adopt them. You also want some sophisticated methods in there (like random forest) to see if the problem can even be learned and to start building up expectations of accuracy.

How many algorithms? At least 10-to-20 different algorithms.

https://machinelearningmastery.com/evaluate-machine-learning-algorithms-with-r/, Jason Brownlee

Caret Machine Learning Examples

Group	Algorithms	Caret Model	FILE
Linear Methods	Linear Discriminant Analysis	`lda`	`lda`
	Linear Discriminant Analysis with YeoJohnson preprocessing	`lda w/YeoJohnson`	`lda-YeoJohnson`
	LASSO, Ridge, and Elastic Net	`glmnet`	`glmnet`
	LASSO, Ridge, and Elastic Net with Synthetic Minority Over-Sampling Technique (SMOTE)	`glmnet w/SMOTE`	`glmnet-SMOTE`
Non-Linear Methods	Neural Network	`nnet`	`nnet`
	Support Vector Machine with Radial Basis Function Kernel	`svmRadial`	`svmRadial`
	Naïve Bayes	`nb`	`nb`
	Naïve Bayes with Independent Component Analysis (ICA)	`nb w/ICA`	`nb-ica-transform`
	k Nearest Neighbors	`knn`	`knn`
Trees and Rules	J48	`J48`	`J48`
	Classification and Regression Trees	`rpart`	`rpart`
Ensembles of Trees	C5.0	`C5.0`	`C50`
	Random Forest	`rf`	`rf`
	Random Forest with SMOTE	`rf w/SMOTE`	`rf-SMOTE`

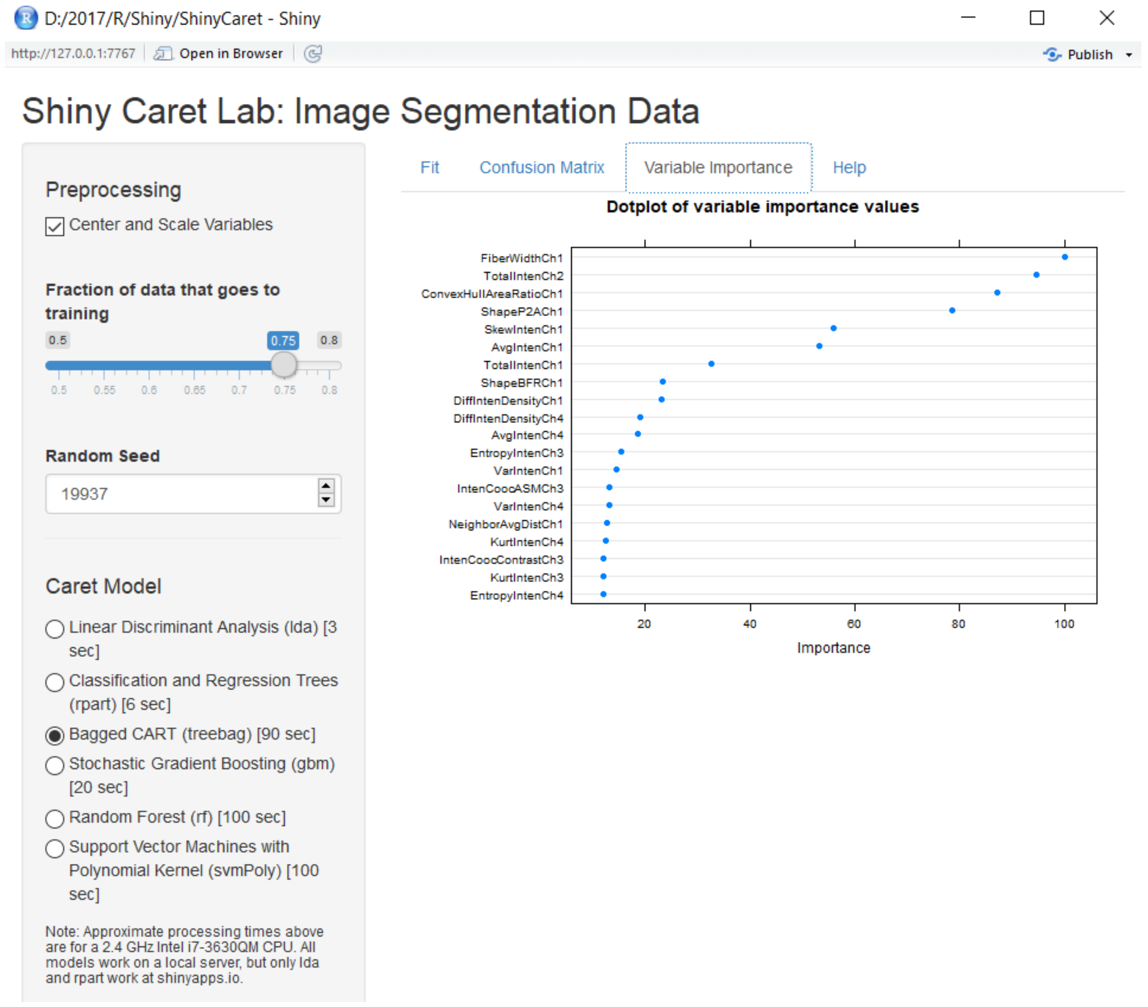
Files: **Forensic-Glass-caret-FILE.Rmd** and **Forensic-Glass-caret-FILE.html**

Summary of Results From Variety of Machine Learning Methods

Hold Out Test Dataset Results

Group	Model	Accuracy	Kappa
Linear Methods	lda	0.541	0.369
	lda w/YeoJohnson	0.623	0.486
	glmnet	0.639	0.498
	glmnet w/SMOTE	0.672	0.547
Non-Linear Methods	nnet	0.672	0.555
	svmRadial	0.639	0.486
	nb	0.607	0.459
	nb w/ICA	0.508	0.314
	knn	0.689	0.582
Trees and Rules	J48	0.672	0.560
	rpart	0.574	0.373
Ensembles of Trees	C5.0	0.689	0.573
	rf	**0.771**	**0.686**
	rf w/SMOTE	0.754	0.666

Toy Shiny App To Compare Models



D:/2017/R/Shiny/ShinyCaret - Shiny

http://127.0.0.1:7767 | Open in Browser | Publish

Shiny Caret Lab: Image Segmentation Data

Preprocessing

[✓] Center and Scale Variables

Fraction of data that goes to training: 0.75 (range 0.5 – 0.8)

Random Seed: 19937

Caret Model

- ○ Linear Discriminant Analysis (lda) [3 sec]
- ○ Classification and Regression Trees (rpart) [6 sec]
- ● Bagged CART (treebag) [90 sec]
- ○ Stochastic Gradient Boosting (gbm) [20 sec]
- ○ Random Forest (rf) [100 sec]
- ○ Support Vector Machines with Polynomial Kernel (svmPoly) [100 sec]

Note: Approximate processing times above are for a 2.4 GHz Intel i7-3630QM CPU. All models work on a local server, but only lda and rpart work at shinyapps.io.

Open ShinyCaret Folder as project in RStudio. Open file server.R. Select Run App.

Machine Learning Algorithms Using R's Caret Package

Summary

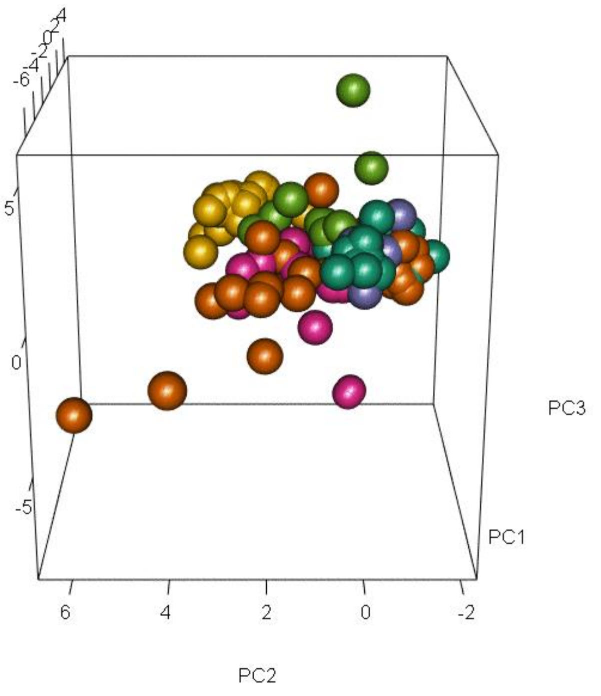
- Caret provides uniform approach to using many classification and regression algorithms
- Many machine algorithms can be explored quickly using Caret.
- Caret provides many useful tools for machine learning

Question

Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?

http://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf

Forensic Glass Dataset: PCA



Files: **Forensic-Glass-PCA.Rmd** and **Forensic-Glass-PCA.html**