Machine Learning with Caret in R

An overview of machine learning and guide to the Caret Package

Melissa Zhao
Research Scientist, Ogino Lab

Machine learning

" A method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. "

- coined by Arthur Samuel in 1959
- closely related to computational statistics
- focuses on making predictions / classifications
- supervised or unsupervised learning
- Goals: produce reliable and repeatable decisions, uncover hidden patterns/insights within subjects of interest

/

Types of Machine learning

Supervised

- Linear regression
- Decision trees
- Support vector machines
- Neural networks

Unsupervised

- Clustering Hierarchical clustering Mixture models
- Neural Networks Self-organizing map Generative Adversarial Networks
- Late variable models Expectation-maximization
- Blind signal separation Principal component analysis Singular value decomposition

/

Models in Caret

```
library(caret)
names(getModelInfo())
```

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.

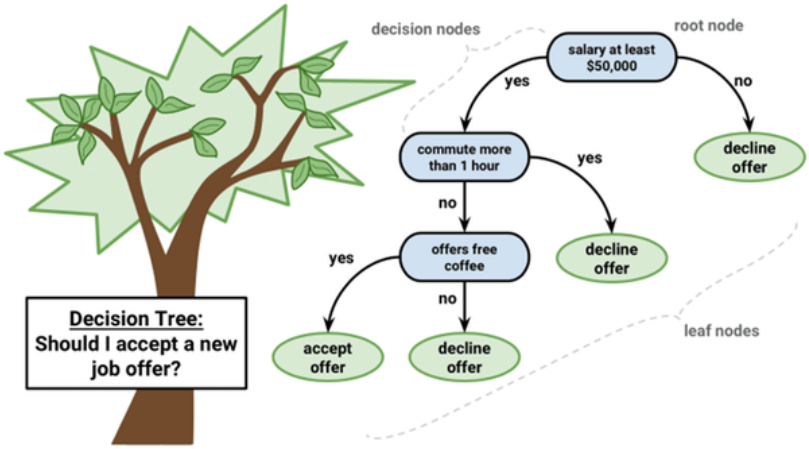
Show 237 entries

Search:

Model	*method* Value	Type	Libraries	Tuning Parameters
AdaBoost Classification Trees	adaboost	Classification	fastAdaboost	nIter, method
AdaBoost.M1	AdaBoost.M1	Classification	adabag, plyr	mfinal, maxdepth, coeflearn
Adaptive Mixture Discriminant Analysis	amdai	Classification	adaptDA	model
Adaptive-Network-Based Fuzzy Inference System	ANFIS	Regression	frbs	num.labels, max.iter

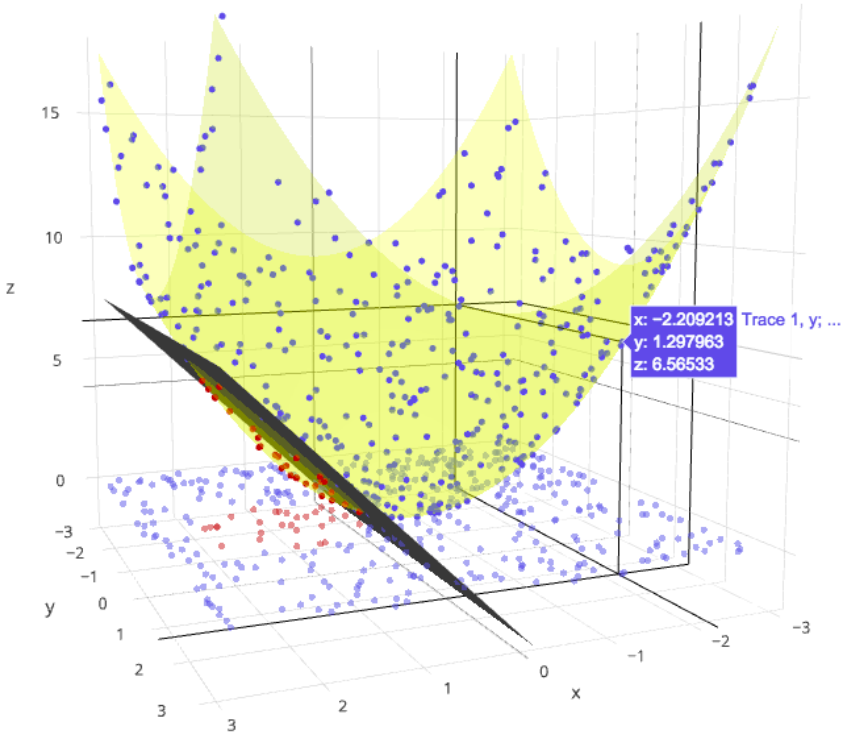
/

Decision trees



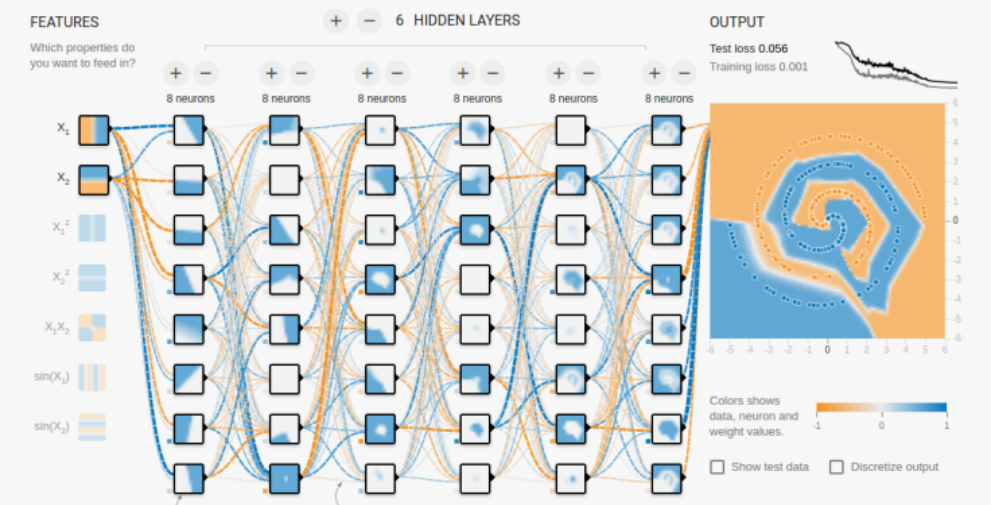
/

Support vector machines



/

Neural networks



/

Which model to choose?

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲*= good,* ◆*=fair, and* ▼*=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

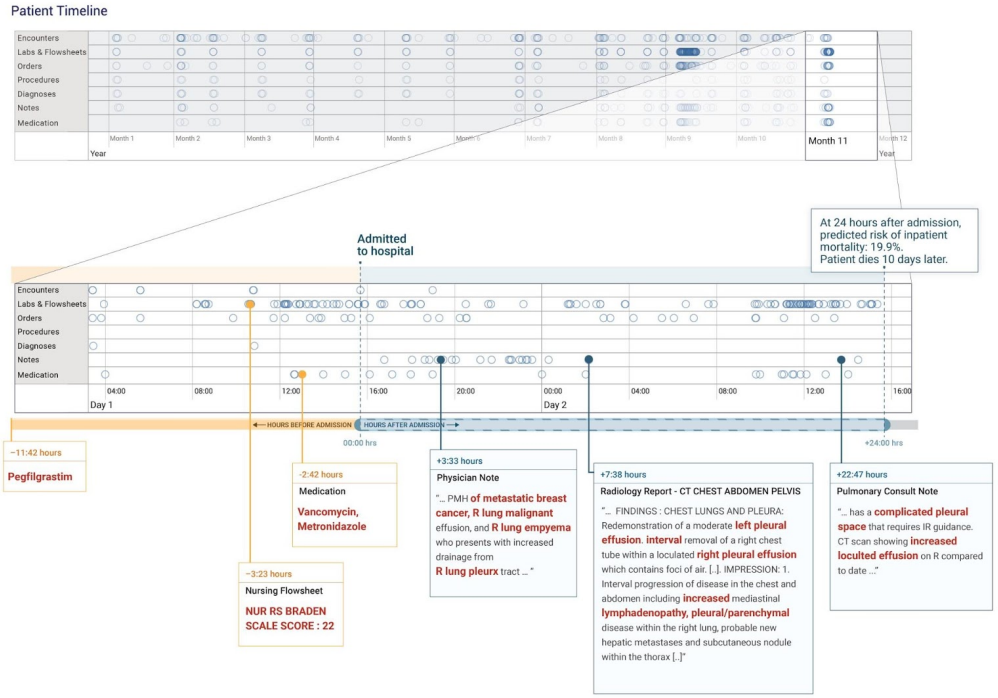
Source: ESL (Hastie)

/

Google AI: Deep Learning for Electronic Health Records

"When patients get admitted to a hospital, they have many questions about what will happen next. When will I be able to go home? Will I get better? Will I have to come back to the hospital? Having precise answers to those questions helps doctors and nurses make care better, safer, and faster - if a patient's health is deteriorating, doctors could be sent proactively to act before things get worse."

/


Patient Timeline
Encounters
Labs & Flowsheets
Orders
Procedures
Diagnoses
Notes
Medication
Month 1
Month 2
Month 3
Month 4
Month 5
Month 6
Month 7
Month 8
Month 9
Month 10
Month 11
Month 12
Year
Year
Admitted to hospital
At 24 hours after admission, predicted risk of inpatient mortality: 19.9%. Patient dies 10 days later.
04:00
08:00
12:00
16:00
20:00
00:00
04:00
08:00
12:00
16:00
Day 1
Day 2
HOURS BEFORE ADMISSION
HOURS AFTER ADMISSION
00:00 hrs
+24:00 hrs
−11:42 hours
Pegfilgrastim
-2:42 hours
Medication
Vancomycin, Metronidazole
−3:23 hours
Nursing Flowsheet
NUR RS BRADEN SCALE SCORE : 22
+3:33 hours
Physician Note
"... PMH of metastatic breast cancer, R lung malignant effusion, and R lung empyema who presents with increased drainage from R lung pleurx tract ... "
+7:38 hours
Radiology Report - CT CHEST ABDOMEN PELVIS
"... FINDINGS : CHEST LUNGS AND PLEURA: Redemonstration of a moderate left pleural effusion. interval removal of a right chest tube within a loculated right pleural effusion which contains foci of air. [..]. IMPRESSION: 1. Interval progression of disease in the chest and abdomen including increased mediastinal lymphadenopathy, pleural/parenchymal disease within the right lung, probable new hepatic metastases and subcutaneous nodule within the thorax [..]"
+22:47 hours
Pulmonary Consult Note
"... has a complicated pleural space that requires IR guidance. CT scan showing increased loculted effusion on R compared to date ..."

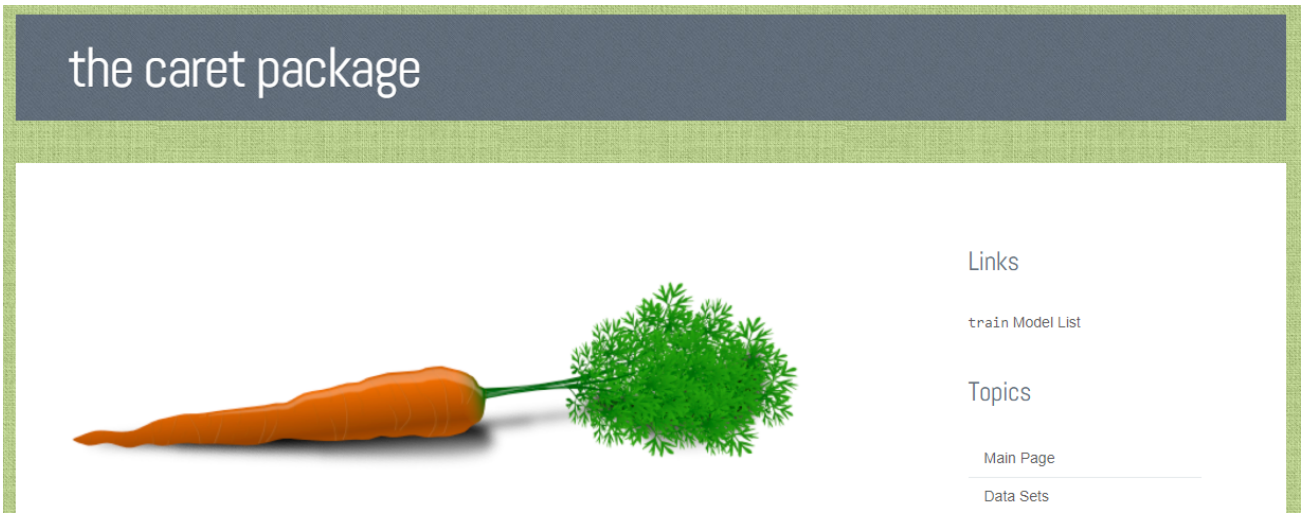
/

Caret package in R

Classification And Regression Training

A wrapper package that contains a set of functions to streamline creation of predictive models

Includes tools for data splitting, pre-processing, feature selection, model tuning, etc.



/

Steps

1. Getting started
2. Training/testing split
3. Pre-processing
4. Feature selection
5. Train models
6. Parameter tuning
7. Variable importance estimation
8. Model performance

/

Demonstration

For this demo, we will use the Abalone dataset from an original study on Abalone population in Australia.

"The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait" Nash (1994)

/

1. Getting started

Sex / nominal / -- / M, F, and I (infant)
Length / continuous / mm / Longest shell measurement
Diameter / continuous / mm / perpendicular to length
Height / continuous / mm / with meat in shell
Whole weight / continuous / grams / whole abalone
Shucked weight / continuous / grams / weight of meat
Viscera weight / continuous / grams / gut weight (after bleeding)
Shell weight / continuous / grams / after being dried
Rings / integer / -- / +1.5 gives the age in years

```
library(caret)
```

```
data = read.csv("B:/OneDrive - Harvard University/CBQG fall 2018/Rcaret/abalone.data", header = F)
colnames(data) = c("Sex", "Length", "Diameter", "Height", "Whole_Weight",
                   "Shuckled_Weight", "Viscera_Weight", "Shell_Weight", "Rings")
```

/

```
dim(data)
```

```
## [1] 4177    9
```

```
sum(is.na(data))
```

```
## [1] 0
```

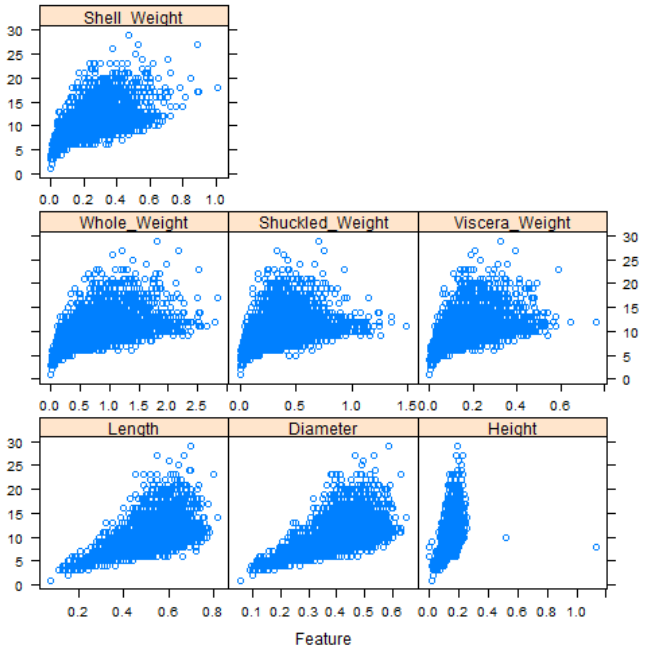
/

```
summary(data)
```

```
##  Sex          Length         Diameter          Height
##  F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000
##  I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150
##  M:1528   Median :0.545   Median :0.4250   Median :0.1400
##           Mean   :0.524   Mean   :0.4079   Mean   :0.1395
##           3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650
##           Max.   :0.815   Max.   :0.6500   Max.   :1.1300
##   Whole_Weight    Shuckled_Weight  Viscera_Weight    Shell_Weight
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.0005   Min.   :0.0015
##  1st Qu.:0.4415   1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300
##  Median :0.7995   Median :0.3360   Median :0.1710   Median :0.2340
##  Mean   :0.8287   Mean   :0.3594   Mean   :0.1806   Mean   :0.2388
##  3rd Qu.:1.1530   3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290
##  Max.   :2.8255   Max.   :1.4880   Max.   :0.7600   Max.   :1.0050
##      Rings
##  Min.   : 1.000
##  1st Qu.: 8.000
##  Median : 9.000
##  Mean   : 9.934
##  3rd Qu.:11.000
##  Max.   :29.000
```

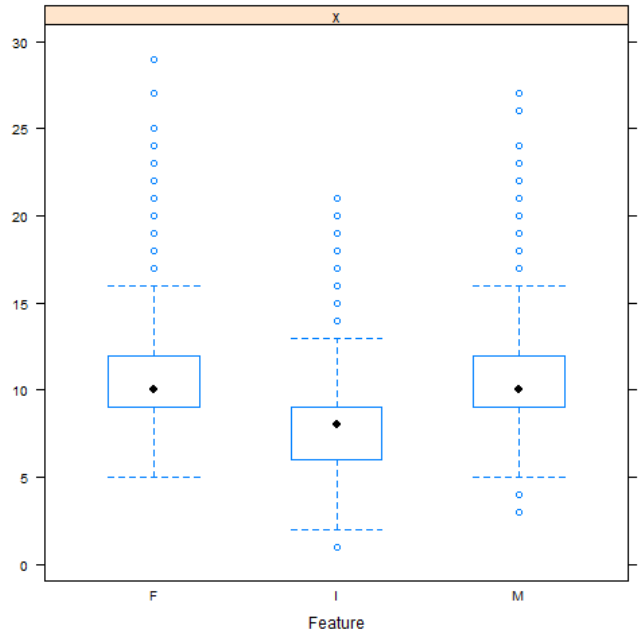
/

```
featurePlot(x = data[, 2:8],
            y = data$Rings,
            plot = "scatter")
```



/

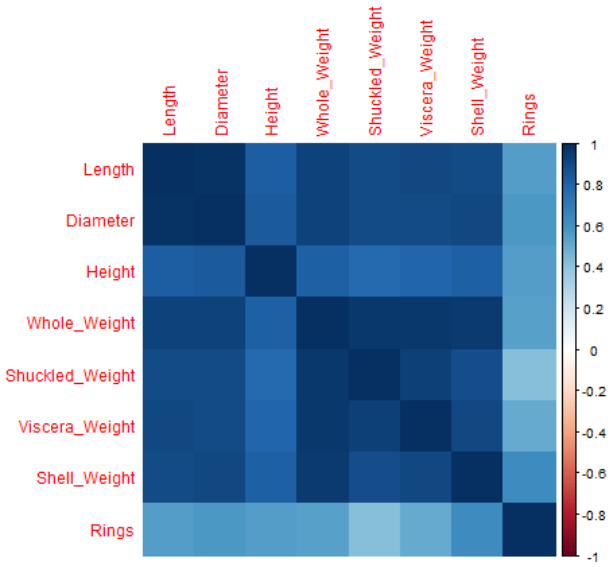
```
featurePlot(x = data$Rings,
            y = data$Sex,
            plot = "box")
```



/

```
library(corrplot)
```

```
M = cor(data[,-1])
corrplot(M, method = "color")
```



/

2. Training/testing split

Splitting into 80% training set and 20% testing set.

- Splitting based on predictors
- Splitting based on outcome

```
intrain = createDataPartition(data$Rings, p=0.8, list=FALSE)
train = data[ intrain,]
test = data[-intrain,]
```

/

3. Pre-processing

- Create dummy variables
- Zero- and Near Zero-variance analysis
- Correlated predictors
- Linear Dependencies
- Imputation
- Center and Scale

```
preProcValues = preProcess(train, method = c("knnImpute")) #automatically centers and scales
train = predict(preProcValues, train)
test = predict(preProcValues, test)
```

/

```
summary(train)
```

```
##  Sex          Length          Diameter          Height
##  F:1046   Min.   :-3.7474   Min.   :-3.5692   Min.   :-3.58170
##  I:1045   1st Qu.:-0.6226   1st Qu.:-0.5906   1st Qu.:-0.62956
##  M:1252   Median : 0.1690   Median : 0.1666   Median : 0.01221
##           Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##           3rd Qu.: 0.7523   3rd Qu.: 0.7220   3rd Qu.: 0.65398
##           Max.   : 2.4188   Max.   : 2.4385   Max.   : 9.63876
##   Whole_Weight      Shuckled_Weight   Viscera_Weight     Shell_Weight
##  Min.   :-1.68770   Min.   :-1.6125   Min.   :-1.6411   Min.   :-1.7082
##  1st Qu.:-0.78834   1st Qu.:-0.7728   1st Qu.:-0.7957   1st Qu.:-0.7856
##  Median :-0.05879   Median :-0.1011   Median :-0.0954   Median :-0.0317
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.65806   3rd Qu.: 0.6367   3rd Qu.: 0.6570   3rd Qu.: 0.6199
##  Max.   : 4.05416   Max.   : 5.0466   Max.   : 5.2442   Max.   : 5.4969
##      Rings
##  Min.   :-2.7741
##  1st Qu.:-0.6031
##  Median :-0.2930
##  Mean   : 0.0000
##  3rd Qu.: 0.3273
##  Max.   : 5.9098
```

/

```
summary(test)
```

```
##  Sex         Length           Diameter           Height
##  F:261   Min.   :-3.28912   Min.   :-3.16535   Min.   :-3.068288
##  I:297   1st Qu.:-0.66431   1st Qu.:-0.64111   1st Qu.:-0.757915
##  M:276   Median : 0.14813   Median : 0.11616   Median : 0.012210
##          Mean   :-0.03052   Mean   :-0.03105   Mean   :-0.001026
##          3rd Qu.: 0.74184   3rd Qu.: 0.72198   3rd Qu.: 0.653980
##          Max.   : 2.04382   Max.   : 1.98410   Max.   :25.426311
##   Whole_Weight      Shuckled_Weight    Viscera_Weight
##  Min.   :-1.67041   Min.   :-1.59684   Min.   :-1.6229
##  1st Qu.:-0.80639   1st Qu.:-0.82659   1st Qu.:-0.8297
##  Median :-0.08929   Median :-0.14813   Median :-0.1226
##  Mean   :-0.03224   Mean   :-0.03839   Mean   :-0.0422
##  3rd Qu.: 0.62577   3rd Qu.: 0.61876   3rd Qu.: 0.5981
##  Max.   : 3.49391   Max.   : 3.45684   Max.   : 3.4673
##   Shell_Weight          Rings
##  Min.   :-1.69388   Min.   :-2.46396
##  1st Qu.:-0.81163   1st Qu.:-0.60312
##  Median :-0.07299   Median :-0.29298
##  Mean   :-0.02103   Mean   :-0.01705
##  3rd Qu.: 0.66027   3rd Qu.: 0.32730
##  Max.   : 3.48650   Max.   : 4.04899
```

/

4. Feature selection

Many models have built-in feature selection methods, often based on error minimization / likelihood maximization. For models without this intrinsic method, feature selection can be performed via a variety of approaches in caret.

```
control <- rfeControl(functions = rfFuncs,
                      method = "repeatedcv",
                      repeats = 3,
                      verbose = F)
outcome = 'Rings'
predictors = names(train)[!names(train) %in% outcome]
rfe(train[,predictors], train[,outcome],
                        rfeControl = control)

predictors = c("Shuckled_Weight", "Shell_Weight", "Sex", "Height", "Viscera_Weight")
```

/

5. Train models

```
#set seed
set.seed(123)

#Gradient boosting
model_gbm<-train(train[,predictors],train[,outcome],method='gbm')

#Random forest
model_rf<-train(train[,predictors],train[,outcome],method='rf')

#Support vector machine
model_glm<-train(train[,predictors],train[,outcome],method='svmLinear')

#Neural networks
model_nnet<-train(train[,predictors],train[,outcome],method='nnet')#feed-forward, single hidden-layer network
```

/

6. Parameter tuning

Tuning can be performed via the specification of a tuning grid or tuning length.

```
fitControl <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 5)

modelLookup(model='gbm')
```

```
##   model         parameter                   label forReg forClass
## 1   gbm           n.trees   # Boosting Iterations   TRUE     TRUE
## 2   gbm interaction.depth          Max Tree Depth   TRUE     TRUE
## 3   gbm         shrinkage               Shrinkage   TRUE     TRUE
## 4   gbm    n.minobsinnode Min. Terminal Node Size   TRUE     TRUE
##   probModel
## 1      TRUE
## 2      TRUE
## 3      TRUE
## 4      TRUE
```

/

```
#Creating grid
grid <- expand.grid(n.trees=c(10,20,50,100,500,1000),
                    shrinkage=c(0.01,0.05,0.1,0.5),
                    n.minobsinnode = c(3,5,10),
                    interaction.depth=c(1,5,10))

# training the model
#using tune grid
model_gbm<-train(train[,predictors],train[,outcome],
                 method='gbm',trControl=fitControl,tuneGrid=grid)
#using tune length
model_gbm<-train(train[,predictors],train[,outcome],
                 method='gbm',trControl=fitControl,tuneLength=10)
```

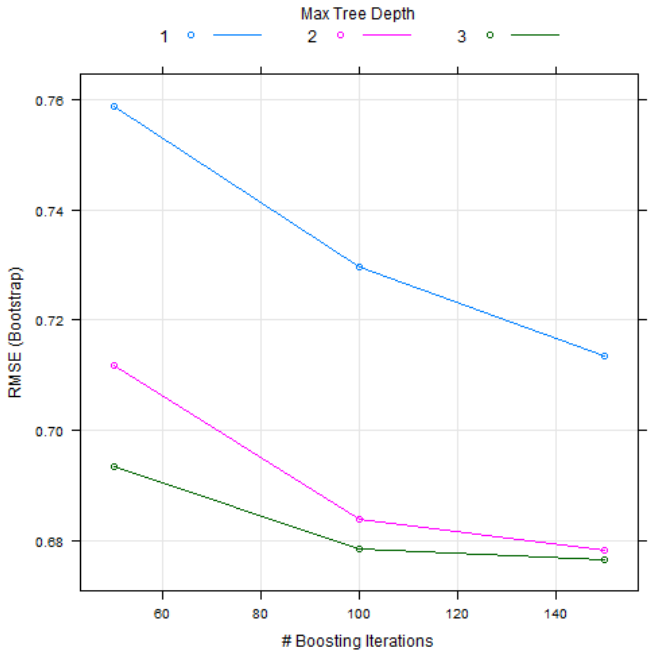
/

```
print(model_gbm)
```

```
## Stochastic Gradient Boosting
##
## 3343 samples
##    8 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3343, 3343, 3343, 3343, 3343, 3343, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE       Rsquared   MAE
##   1                   50      0.7658889  0.4292350  0.5506145
##   1                  100      0.7371362  0.4694133  0.5289367
##   1                  150      0.7216469  0.4892545  0.5173045
##   2                   50      0.7199629  0.4942853  0.5154158
##   2                  100      0.6927822  0.5261471  0.4911339
##   2                  150      0.6882157  0.5312498  0.4861801
##   3                   50      0.7022243  0.5160358  0.4991297
##   3                  100      0.6878071  0.5319263  0.4848958
##   3                  150      0.6849501  0.5357355  0.4825321
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

/

```
plot(model_gbm)
```



/

7. Variable importance estimation

```
#Variable Importance
library(gbm)
```

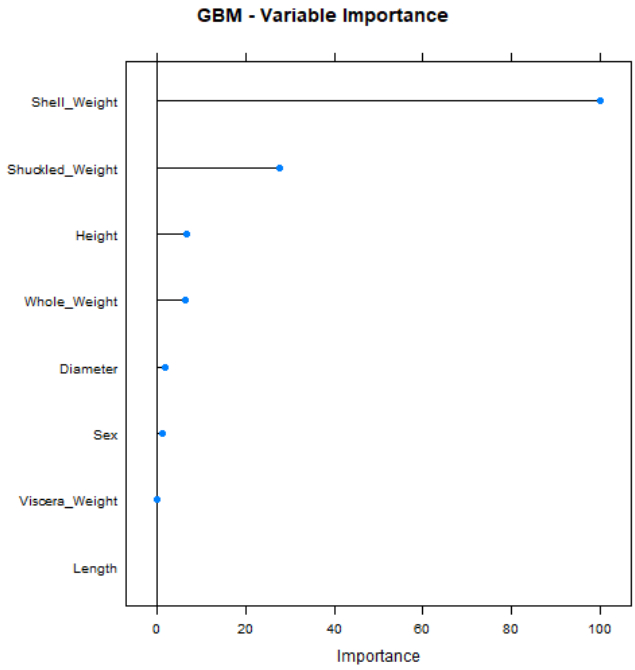
```
## Loaded gbm 2.1.4
```

```
varImp(object=model_gbm)
```

```
## gbm variable importance
##
##                 Overall
## Shell_Weight    100.0000
## Shuckled_Weight  27.6247
## Height           12.3579
## Whole_Weight      7.2000
## Viscera_Weight    0.8404
## Sex               0.8226
## Diameter          0.5509
## Length            0.0000
```

/

```
#Plotting Varianble importance for GBM
plot(varImp(object=model_gbm),main="GBM - Variable Importance")
```



/

8. Model performance

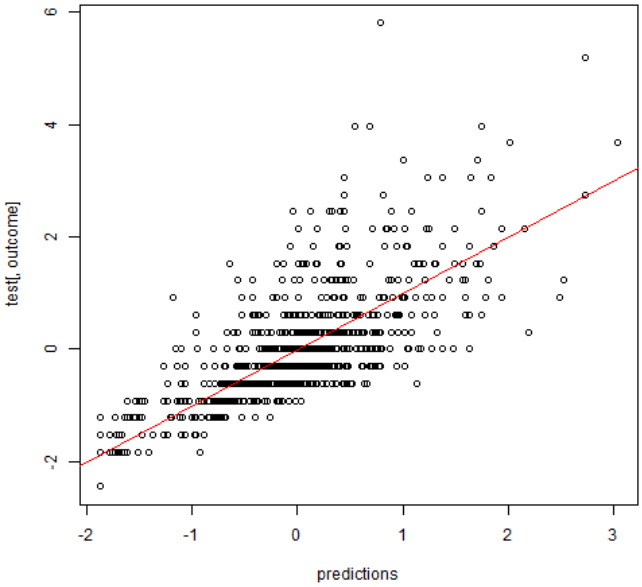
Classification: ROC, Accuracy, Sensitivity, Specificity (confusion matrix)
Regression: RMSE, R-squared, MAE

```
#Predictions
predictions<-predict.train(object=model_gbm,test[,predictors],type="raw")

#Performance
#Measures for regression
postResample(pred = predictions, obs = test[,outcome])

#Measures for classification
confusionMatrix(predictions,test[,outcome])
```

```
##      RMSE  Rsquared       MAE
## 0.6752421 0.5452949 0.4818521
```

/

```
plot(predictions, test[,outcome])
abline(a = 0 , b = 1, col = "red")
```



/

References and Resources

- Caret user guide: http://topepo.github.io
- Caret tutorial: https://datascienceplus.com/machine-learning-with-r-caret-part-1/
- Coursera: https://www.coursera.org/lecture/practical-machine-learning/caret-package-Bu9ns
- Google AI for EHR Example: https://ai.googleblog.com/2018/05/deep-learning-for-electronic-health.html
- Machine learning in Python: https://www.kaggle.com/ragnisah/eda-abalone-age-prediction
- Mathematical concepts: Introduction to Statistical Learning (Gareth James), Elements of Statistical Learning (Trevor Hastie)

/