

MAD-CB



Machine Learning

- Dr. Sharin Glander, Univ. de Münster, Alemanha
 - ▶ Webinar excelente recente
 - ▶ “Building meaningful machine learning models for disease prediction”
 - ▶ <https://github.com/ShirinG>
- Dados
 - ▶ UCI Machine Learning Repository
 - ▶ U. de Wisconsin dados sobre câncer de mama
 - ▶ Arquivo “breast-cancer-wisconsin-data.txt”

Machine Learning em Modelagem das Doenças

- Tipicamente, projetos com “big data”
- Modelo pode fornecer informação rapidamente e corretamente
 - ▶ Médicos podem usar a informação para desenhar tratamentos ou diagnósticos
- Aplicação para medicina personalizada de precisão
- Exemplo:
 - ▶ Diagnostico de câncer de mama com ajuda de modelo informatizado

Podemos Ter Confiança nos Modelos de Machine Learning?

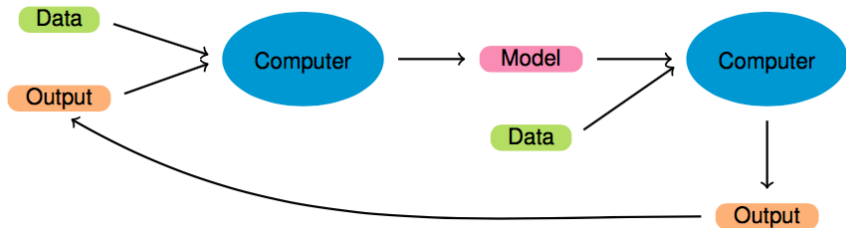
- Algoritmos de ML modelam interações de alto grau entre as variáveis
- Interpretação dos resultados de ML pode ser difícil
- A “caixa preta” dos algoritmos de ML escondem como eles fazem escolhas
- Assim, *precisamos modelos que significam algo* para os
 - ▶ Arquitetos
 - ▶ Usadores
- “Meaningful Models”

O Que Faz um Modelo um “Meaningful Model”

- Poder generalizar baseado no modelo
- Responde à pergunta original
- ... com suficiente precisão para ser confiável
- Grau de precisão depende no problema

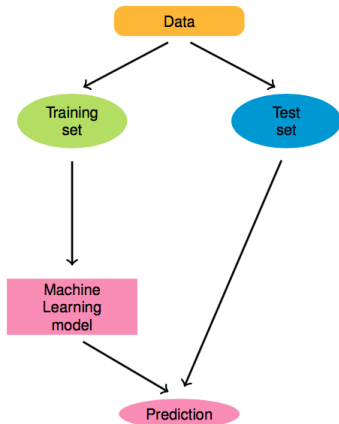
- Inteligência artificial (“AI”)
- Modelo orientado a dados
- Algoritmos **aprendem** por treinamento com dados observados
- E **prever casos desconhecidos**
- Computadores de hoje capazes de tratar essas bases de dados
 - ▶ Mesmo laptops

Machine Learning – 2



Machine Learning – Supervisionada vs. Não-Supervisionada

Supervised



Unsupervised

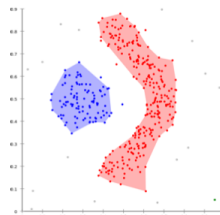
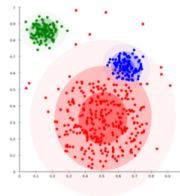
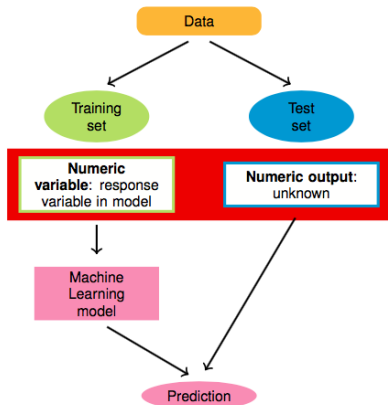


Image Source: Wikipedia

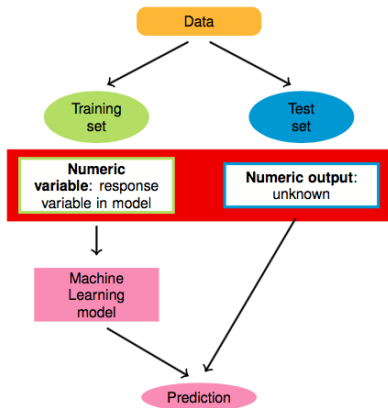
Supervisionada – Classificação vs. Regressão

Regression
e.g. weight loss

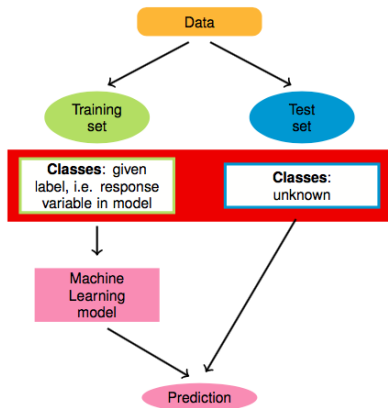


Supervisionada – Classificação vs. Regressão

Regression
e.g. weight loss



Classification
e.g. healthy vs disease



Features – Covariáveis

- Variáveis para treinar o modelo
- Selecionar as variáveis certas – **crucial**
- Mais features não necessariamente bom
 - ▶ Perigo de “overfitting”

Overfitting

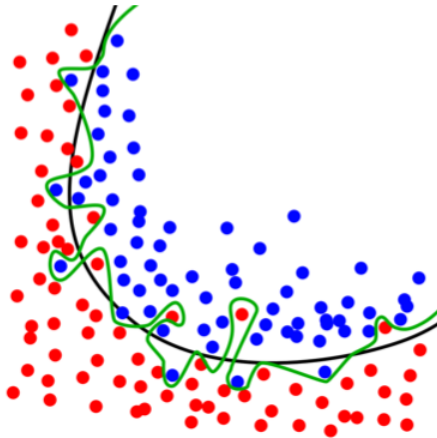
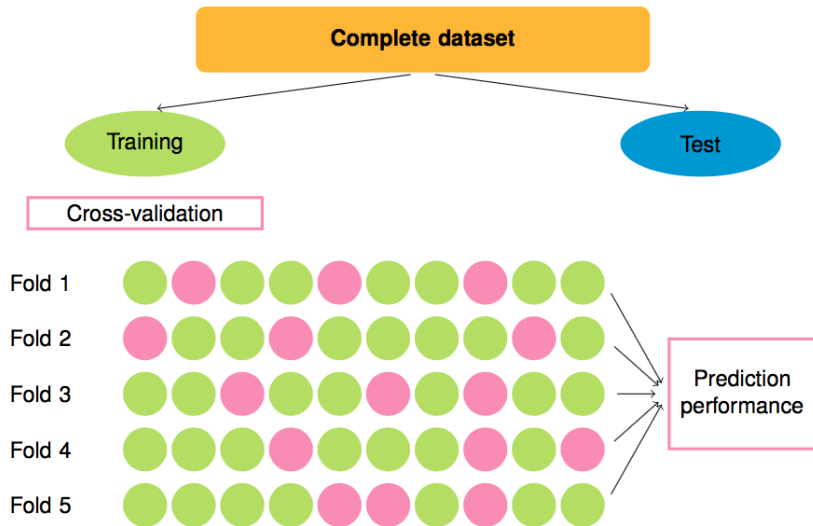


Image Source: Wikipedia

Treinamento, Testes & Cross Validation



- Divide os dados aleatoriamente em dois grupos
 - ▶ Treinamento
 - ▶ Testes
- Proporções pode variar entre
 - ▶ 50 - 50 – se você tem uma base de dados muito grande
 - ▶ 70 - 30 – em outros casos

**NUNCA, JAMAIS, USE OS MESMOS DADOS PARA TESTES
QUE VOCÊ USOU PARA TREINAMENTO**

Cross-Validation (*k-fold*)

- Uma de uma serie de técnicas usadas para fortalecer a capacidade do modelo para prever resultados
 - ▶ Bootstrap - reamostragem
- Com os dados de treinamento só
- Divide os dados em k grupos (“folds”) aleatórios de tamanho igual
- Construir o seu modelo usando todos fora de um grupo
- Testar o modelo nos dados no grupo que você reservou
 - ▶ Calcular o erro entre as previsões com o modelo e os valores observados
- Repetir e fazer a média dos erros
- O modelo (entre os k que você construiu) com a média menor é o modelo melhor

Vamos Pôr as Mãos na Massa

- Vêm de Wisconsin dados sobre câncer de mama
- Características dos tumores de mama
- Variável dependente: diagnose (diag)

- Vem de análise de imagens
 - ▶ Aspiração com agulha fina Características
 - ▶ Sample ID (code number)
 - ▶ Clump thickness
 - ▶ Uniformity of cell size
 - ▶ Uniformity of cell shape
 - ▶ Marginal adhesion
 - ▶ Single epithelial cell size
 - ▶ Number of bare nuclei
 - ▶ Bland chromatin
 - ▶ Number of normal nuclei
 - ▶ Mitosis

Carregar Dados

```
bc_data <- read.table("breast-cancer-wisconsin-data.txt",
  header = FALSE,
  sep = ",",
  na.strings = "?")
colnames(bc_data) <- c("sample_code_number",
  "clump_thickness",
  "uniformity_of_cell_size",
  "uniformity_of_cell_shape",
  "marginal_adhesion",
  "single_epithelial_cell_size",
  "bare_nuclei",
  "bland_chromatin",
  "normal_nucleoli",
  "mitosis",
  "diag")

bc_data$diag <- ifelse(bc_data$diag == "2", "benign",
  ifelse(bc_data$diag == "4", "malignant", NA))
```

Dados

```
glimpse(bc_data)
```

```
## Observations: 699
## Variables: 11
## $ sample_code_number      <int> 1000025, 1002945, 1015425, 1016277...
## $ clump_thickness         <int> 5, 5, 3, 6, 4, 8, 1, 2, 2, 4, 1, 2...
## $ uniformity_of_cell_size <int> 1, 4, 1, 8, 1, 10, 1, 1, 1, 2, 1, ...
## $ uniformity_of_cell_shape <int> 1, 4, 1, 8, 1, 10, 1, 2, 1, 1, 1, ...
## $ marginal_adhesion       <int> 1, 5, 1, 1, 3, 8, 1, 1, 1, 1, 1, 1...
## $ single_epithelial_cell_size <int> 2, 7, 2, 3, 2, 7, 2, 2, 2, 2, 1, 2...
## $ bare_nuclei             <int> 1, 10, 2, 4, 1, 10, 10, 1, 1, 1, 1...
## $ bland_chromatin         <int> 3, 3, 3, 3, 3, 9, 3, 3, 1, 2, 3, 2...
## $ normal_nucleoli         <int> 1, 2, 1, 7, 1, 7, 1, 1, 1, 1, 1, 1...
## $ mitosis                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1...
## $ diag                    <chr> "benign", "benign", "benign", "ben..."
```

Análise de NAs – Decisão sobre o Que Fazer com Eles

- Quantas NAs estão nos dados?

```
length(which(is.na(bc_data)))
```

```
## [1] 16
```

- Quantas amostras perdemos se retirarmos os NAs?

```
nrow(bc_data[is.na(bc_data), ])
```

```
## [1] 16
```

Imputar Valores de NAs

- Pacote e função `mice`
 - ▶ Multivariate Imputation by Chained Equations
- Cria dados imputados para dados incompletos multivariados
 - ▶ Gibbs Sampling (técnica bayesiana)
 - ▶ Gera valores plausíveis sintéticos dado as outras colunas no dataset
- Imputação introduza mais incerteza no modelo


```
summary(bc_data$bare_nuclei)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##      1.000   1.000   1.000   3.545   6.000  10.000    16
```

```
bc_data[,2:10] <- apply(bc_data[, 2:10], 2, function(x)  
  X = as.numeric(as.character(x)))  
dataset_impute <- mice(bc_data[, 2:10], print = FALSE)  
bc_data <- cbind(bc_data[, 11, drop = FALSE], mice::complete(dataset_impute, 1))  
summary(bc_data$bare_nuclei)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##      1.000   1.000   1.000   3.531   6.000  10.000
```

Resumo das Diagnoses

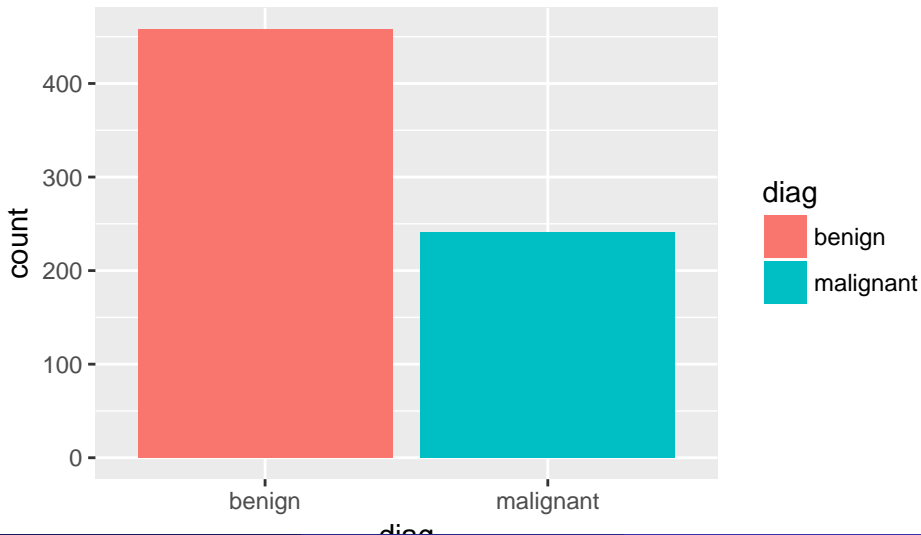
- Converter diag para um factor
- Quantos casos de benign e malignant têm?

```
bc_data$diag <- as.factor(bc_data$diag)
summary(bc_data$diag)
```

```
##      benign malignant
##      458         241
```

Gráfico das Diagnoses

```
brgr1 <- ggplot(bc_data, aes(x = diag, fill = diag)) + geom_bar()  
brgr1
```



Classes de diag Desequilibradas

- Normalmente precisa um ajuste para tratar dessa desequilibrade
- Não vamos fazer isso aqui

Exploração de Algumas das Covariáveis

```
Desc(bc_data$clump_thickness, plotit = FALSE)
```

```
## -----  
## bc_data$clump_thickness (numeric)  
##  
##      length      n      NAs  unique      Os  mean  meanCI  
##      699      699      0      10      0  4.42  4.21  
##           100.0%  0.0%           0.0%           4.63  
##  
##      .05      .10      .25  median  .75  .90      .95  
##      1.00      1.00      2.00    4.00  6.00  9.00    10.00  
##  
##      range      sd  vcoef      mad  IQR  skew      kurt  
##      9.00      2.82  0.64      2.97  4.00  0.59    -0.63  
##  
##  
##      level  freq  perc  cumfreq  cumperc  
## 1      1    145  20.7%    145    20.7%  
## 2      2     50   7.2%    195    27.9%  
## 3      3    108  15.5%    303    43.3%  
## 4      4     80  11.4%    383    54.8%  
## 5      5    130  18.6%    513    73.4%  
## 6      6     34   4.9%    547    78.3%  
## 7      7     23   3.3%    570    81.5%  
## 8      8     46   6.6%    616    88.1%  
## 9      9     14   2.0%    630    90.1%  
## 10     10     69   9.9%    699   100.0%
```

bland_chromatin

```
Desc(bc_data$bland_chromatin, plotit = FALSE)
```

```
## -----
## bc_data$bland_chromatin (numeric)
##
##   length      n    NAs  unique    Os  mean  meanCI
##   699      699      0      10      0  3.44   3.26
##   100.0%    0.0%          0.0%          3.62
##
##   .05    .10    .25  median   .75   .90    .95
##   1.00    1.00    2.00   3.00  5.00   7.00   8.00
##
##   range      sd  vcoef      mad   IQR  skew   kurt
##   9.00    2.44  0.71    1.48  3.00  1.10   0.17
##
##
##   level  freq  perc  cumfreq  cumperc
## 1      1   152  21.7%    152    21.7%
## 2      2   166  23.7%    318    45.5%
## 3      3   165  23.6%    483    69.1%
## 4      4    40   5.7%    523    74.8%
## 5      5    34   4.9%    557    79.7%
## 6      6    10   1.4%    567    81.1%
## 7      7    73  10.4%    640    91.6%
## 8      8    28   4.0%    668    95.6%
## 9      9    11   1.6%    679    97.1%
## 10     10    20   2.9%    699   100.0%
```

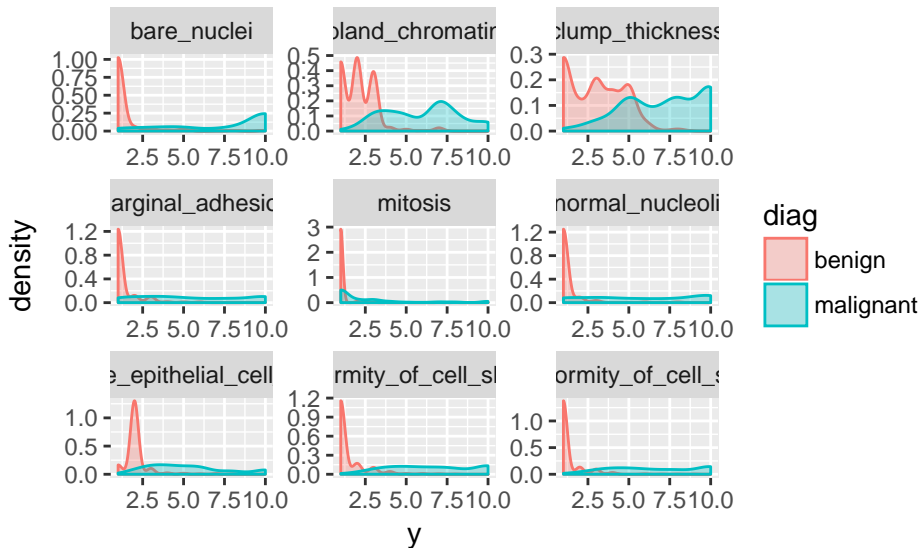
marginal_adhesion

```
Desc(bc_data$marginal_adhesion, plotit = FALSE)
```

```
## -----  
## bc_data$marginal_adhesion (numeric)  
##  
##      length      n      NAs  unique      Os  mean  meanCI  
##      699      699      0      10      0  2.81  2.59  
##      100.0%    0.0%      0.0%      3.02  
##  
##      .05      .10      .25  median  .75  .90      .95  
##      1.00      1.00      1.00      1.00  4.00  8.00  10.00  
##  
##      range      sd  vcoef      mad  IQR  skew      kurt  
##      9.00      2.86  1.02      0.00  3.00  1.52      0.96  
##  
##  
##      level  freq  perc  cumfreq  cumperc  
## 1      1    407  58.2%    407    58.2%  
## 2      2     58  8.3%     465    66.5%  
## 3      3     58  8.3%     523    74.8%  
## 4      4     33  4.7%     556    79.5%  
## 5      5     23  3.3%     579    82.8%  
## 6      6     22  3.1%     601    86.0%  
## 7      7     13  1.9%     614    87.8%  
## 8      8     25  3.6%     639    91.4%  
## 9      9      5  0.7%     644    92.1%  
## 10     10     55  7.9%     699   100.0%
```

Gráfico das Covariáveis com a Diagnose

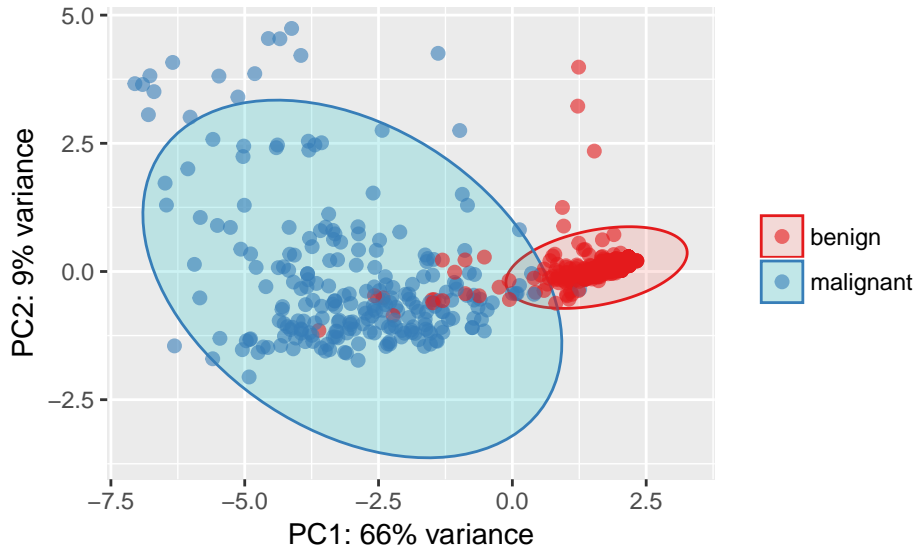
```
gather(bc_data, x, y, clump_thickness:mitosis) %>%  
  ggplot(aes(x = y, color = diag, fill = diag)) +  
    geom_density(alpha = 0.3) +  
    facet_wrap( ~ x, scales = "free", ncol = 3)
```

Análise de Componentes Principais (PCA)

- PCA – técnica para agrupar variáveis
- Neste caso
 - ▶ Mostra que os níveis de diagnose formam espaços coerentes
- PCA - assunto para uma aula futura

Gráfico de PCA



- Existem fortes ou fracas associações entre as covariáveis?
- Uso do pacote `corr`
 - ▶ Novo pacote associado com o tidyverse

```

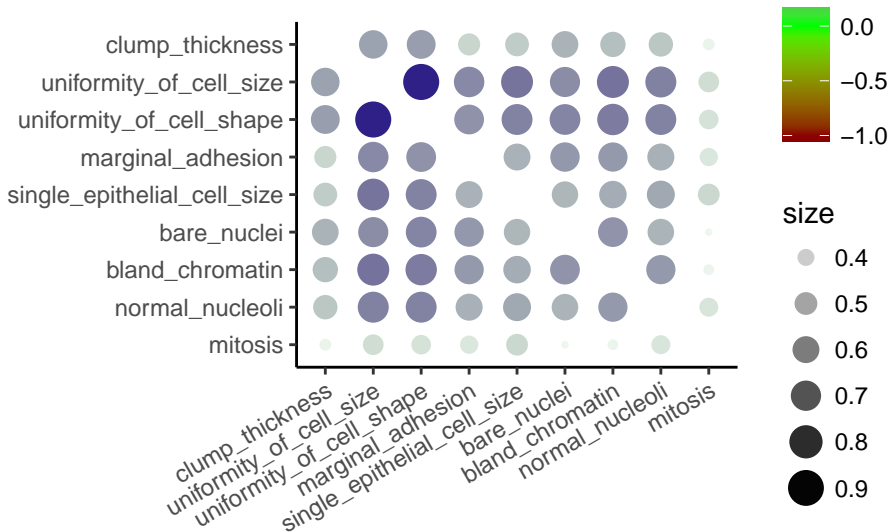
corrdf <- correlate(bc_data[,2:10])
cplot <- rplot(corrdf, legend = TRUE, colours =
               c("darkred", "green", "darkblue"))
cplot <- cplot + theme(axis.text.x =
                       element_text(angle = 30, hjust = 1, vjust = 1))

```

```

## # A tibble: 9 × 10
##           rowname clump_thickness uniformity_of_cell_size
##           <chr>         <dbl>                <dbl>
## 1           clump_thickness             NA             0.6449125
## 2    uniformity_of_cell_size             0.6449125             NA
## 3    uniformity_of_cell_shape             0.6545891             0.9068819
## 4      marginal_adhesion             0.4863562             0.7055818
## 5 single_epithelial_cell_size             0.5218162             0.7517991
## 6           bare_nuclei             0.5950137             0.6945537
## 7      bland_chromatin             0.5584282             0.7557210
## 8      normal_nucleoli             0.5358345             0.7228648
## 9           mitosis             0.3500339             0.4586931
## # ... with 7 more variables: uniformity_of_cell_shape <dbl>,
## #   marginal_adhesion <dbl>, single_epithelial_cell_size <dbl>,
## #   bare_nuclei <dbl>, bland_chromatin <dbl>, normal_nucleoli <dbl>,
## #   mitosis <dbl>

```



Treinamento e Teste – Dados Separados

- Funções para apoiar machine learning
- Pode conduzir toda a análise dentro de caret
- No grupos dos pacotes iniciais

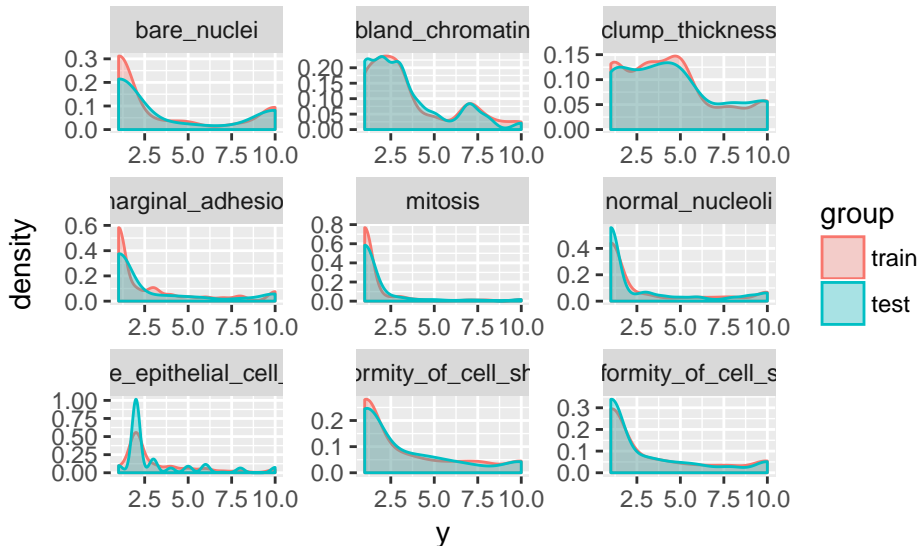
Separar Treinamento e Testes

- Utilizar função `caret::createDataPartition()` para criar bases separadas
 - ▶ 1 para treinamento do modelo
 - ▶ 1 para testes
- Especificar (p) porcentagem de dados colocado na base de treinamento
 - ▶ Entre 0.5 (50%) e 0.7 (70%)
- `createDataPartition()` estratifica os dados baseada nas proporções da variável y

Criar as Bases Treinamento e Testes

```
set.seed(42)
indice <- createDataPartition(bc_data$diag, p = 0.7, list = FALSE)
train_data <- bc_data[indice, ] # use os índices para o treinamento
test_data <- bc_data[-indice, ] # use os outros para testes
```

as Bases Refletem os Mesmos Dados?



Exemplos dos Tipos de Modelos

- Regressão Linear
 - ▶ Ex: GLM
 - ▶ com caret
- Classificação com Árvores
 - ▶ Árvores recursivas de particionamento e regressão (pacote rpart)
 - ▶ Florestas Aleatórias (“Random Forests”)
- Todos com caret

- Antes de iniciar o passo de treinar o modelos, precisamos decidir qual tipo de validação queremos usar
 - ▶ bootstrap, k-fold cross validation
- Especificar através da função `caret::trainControl()`
- Queremos usar *10-fold cross validation*
- Se pudermos repetir o processo de cross validation, faz a seleção do modelo ainda mais forte
 - ▶ Repetiremos 10 vezes

trainControl()

```
set.seed(42)
control <- trainControl(method = "repeatedcv",
                        number = 10,
                        repeats = 10,
                        savePredictions = TRUE,
                        verboseIter = FALSE)
```

Variável Dependente: *benign* ou *malignant*

- Qual tipo de análise mais relacionado?


Variável Dependente: *benign* ou *malignant*

- Qual tipo de análise mais relacionado?
- Regressão logística

Treinamento do Modelo – Regressão Logística

```
model_glm <- caret::train(diag ~ .,  
                           data = train_data,  
                           method = "glm",  
                           preProcess = c("scale", "center"),  
                           trControl = control)
```

Objeto de Modelo

 <code>model_glm</code>	<code>Large train (24 elements, 1 Mb)</code>
--	--

- R preserva todas as iterações do modelo
- Objeto grande (1MB)

Modelo

```
model_glm
```

```
## Generalized Linear Model
##
## 490 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## Pre-processing: scaled (9), centered (9)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 442, 441, 441, 441, 441, 441, ...
## Resampling results:
##
##   Accuracy   Kappa
## 0.9592182 0.9093487
##
##
```

Resumo dos Resultados do Modelo

```
summary(model_glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2955  -0.1322  -0.0727   0.0256   2.4606
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.1493     0.3497  -3.287 0.001013 **
## clump_thickness  1.3721     0.4251   3.228 0.001248 **
## uniformity_of_cell_size 0.1145     0.6609   0.173 0.862481
## uniformity_of_cell_shape 1.0012     0.7185   1.393 0.163482
## marginal_adhesion  0.9890     0.3697   2.675 0.007478 **
## single_epithelial_cell_size 0.1983     0.3720   0.533 0.594022
## bare_nuclei       1.2011     0.3609   3.328 0.000875 ***
## bland_chromatin    1.2178     0.4644   2.622 0.008738 **
## normal_nucleoli    0.3278     0.3685   0.890 0.373646
## mitosis           0.8129     0.5850   1.390 0.164653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 631.346  on 489  degrees of freedom
## Residual deviance:  85.958  on 480  degrees of freedom
## AIC: 105.96
##
## Number of Fisher Scoring iterations: 8
```

O Modelo Pode Predizer os Resultados de Treinamento e de Teste?

- Função `predict()`
 - ▶ com modelo e valores para ser usados para previsão
- Aplicado a base de `train` como exemplo
- Mais interessante – base de `test`
 - ▶ Modelo nunca viu esses dados antes
- **Teste ácido**

Previsões

```
predtr <- predict(model_glm, train_data)
predtest <- predict(model_glm, test_data)
prop.table(table(predtest))
```

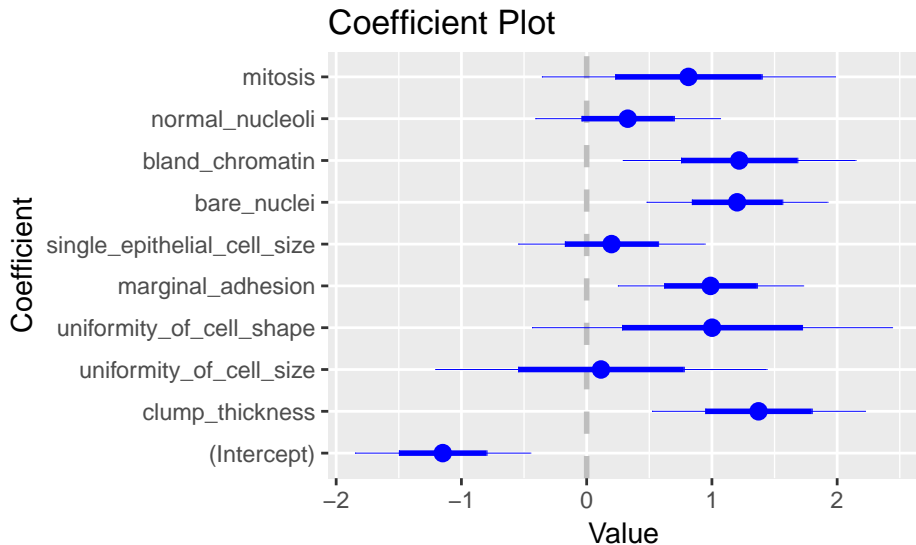
```
## predtest
##      benign malignant
## 0.6507177 0.3492823
```

```
prop.table(table(predtr))
```

```
## predtr
##      benign malignant
## 0.6510204 0.3489796
```

Gráfico de Coeficientes da Diagnose

```
coefplot(model_glm)
```



Previsões com os Dados de Teste – Matriz de Confusão

```
confusionMatrix(predtest, test_data$diag)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction  benign malignant
##   benign      133         3
##   malignant    4         69
##
##               Accuracy : 0.9665
##               95% CI : (0.9322, 0.9864)
##   No Information Rate : 0.6555
##   P-Value [Acc > NIR] : <2e-16
##
##               Kappa : 0.9261
##   Mcnemar's Test P-Value : 1
##
##               Sensitivity : 0.9708
##               Specificity : 0.9583
##               Pos Pred Value : 0.9779
##               Neg Pred Value : 0.9452
##               Prevalence : 0.6555
##               Detection Rate : 0.6364
##   Detection Prevalence : 0.6507
##               Balanced Accuracy : 0.9646
##
##   'Positive' Class : benign
##
```


Previsões com os Dados de Treinamento – Matriz de Confusão

```
confusionMatrix(predtr, train_data$diag)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  benign malignant
##   benign      313         6
##   malignant    8        163
##
##              Accuracy : 0.9714
##              95% CI : (0.9525, 0.9843)
##   No Information Rate : 0.6551
##   P-Value [Acc > NIR] : <2e-16
##
##              Kappa : 0.9369
##  Mcnemar's Test P-Value : 0.7893
##
##              Sensitivity : 0.9751
##              Specificity : 0.9645
##   Pos Pred Value : 0.9812
##   Neg Pred Value : 0.9532
##   Prevalence : 0.6551
##   Detection Rate : 0.6388
##   Detection Prevalence : 0.6510
##   Balanced Accuracy : 0.9698
##
##   'Positive' Class : benign
##
```