# Lista 4

César A. Galvão - 190011572

## Índice

# Questão 11

Pesquisar funções disponíveis em pacotes R para classificação utilizando a função logística. Apresentar um pequeno exemplo do uso das funções. Destacar vantagens e desvantagens em relação aos pacotes de Modelos Lineares Generalizados apresentados em aula.

Exemplos de pacotes para classificação no R: `caret`, `class`, `mlpack`.

---

A seguir são apresentadas as técnicas utilizadas em sala de aula e as funções dos pacotes `caret`. Não foi possível instalar o pacote `mlpack` e o pacote `class` não compreende funções para regressão logística.

A regressão logística é um Modelo Linear Generalizado, que pode ser descrito como

$$g\left[E(Y_i)\right] = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_p X_{pi}, \quad Y_i \overset{i.i.d.}{\sim} FE\left(g\left[E(Y_i)\right], \sigma^2\right),$$

em que $g$ é a função de ligação sobre o preditor linear $g\left[E(Y_i)\right] = \mathbf{X}\boldsymbol{\beta}$. A função de ligação mais comum para o modelo logístico é a função logit, dada por $g(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$, $\pi_i = P(Y_i = 1|\mathbf{X} = \mathbf{x}_i)$, e considera-se $Y_i \sim \text{Bernoulli}\left(p(\mathbf{x}|\omega_1)\right)$.

No contexto de classificação binária, temos que

$$\frac{p(\mathbf{x}|\omega_1)}{1 - p(\mathbf{x}|\omega_1)} = \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)} = \exp\left(\mathbf{X}\boldsymbol{\beta}\right),$$

considerando $\mathbf{X} = (\mathbb{1}^\top, \mathbf{X}^*)$, $\mathbf{X}^*$ a matriz de covariáveis.

A decisão de alocação de $\mathbf{x}_i$ a $\omega_1$ ocorre se $p(\mathbf{x}|\omega_1) > k$, constante que comumente é $0,5$.

Os modelos apresentados em aula compreendem regressão logística, múltipla, politômica e politômica ordenada. Para isso, diversos pacotes são utilizados como `stats`, `mlpack` e `VGAM`.

Para os modelos dicotômicos, são apresentados resultados de seleção de variáveis e medidas diagnósticas como medidas de influências, qualidade de ajuste com $G^2$, razão de verossimilhança e teste de Hosmer e Lemeshow.

Enquanto a implementação via ferramentas do pacote `stats` seja factível, o pacote `caret` apresenta um *framework* consistente para o fluxo de modelagem.

Por exemplo, o bloco a seguir apresenta a partição de uma base de dados em treino e teste com 80% dos dados destinados ao treino. O ajuste do modelo e a matriz de confusão são realizados funções do próprio pacote, mas é utilizado o `predict()` genérico do pacote `stats`:

```
data(iris)

iris$Class <- ifelse(iris$Species == "versicolor", 1, 0)

set.seed(123)
```

```r
trainIndex <- createDataPartition(iris$Class, p = .8, list = FALSE, times = 1)
trainData <- iris[trainIndex, ]
testData <- iris[-trainIndex, ]
```

Uma etapa simples de seleção de variáveis é exemplificada no bloco a seguir. A função `rfeControl` define como será feita a validação — aqui é feita validação cruzada com 10 partições da base de dados com tamanhos similares.

```r
control <- trainControl(method = "cv", number = 10)

# Train logistic regression model
logistic_model <- train(
  Class ~ .,                            # Formula for the model
  data = trainData,                     # Training data
  method = "glm",                       # Method: Generalized Linear Model
  family = "binomial",                  # Family: Binomial (logistic regression)
  trControl = control                   # Cross-validation control parameters
)
```

Os coeficientes, seus desvios e significâncias individuais são dados a seguir:

```r
coefficients <- summary(logistic_model$finalModel)$coefficients
print(coefficients)
```

```
                     Estimate Std. Error       z value  Pr(>|z|)
(Intercept)       -2.656607e+01   419598.9 -6.331301e-05 0.9999495
Sepal.Length       7.255362e-12   104971.0  6.911776e-17 1.0000000
Sepal.Width       -5.746443e-12   119016.7 -4.828264e-17 1.0000000
Petal.Length      -8.271818e-12   121523.0 -6.806794e-17 1.0000000
Petal.Width       -1.904911e-12   187285.1 -1.017118e-17 1.0000000
Speciesversicolor  5.313214e+01   312446.2  1.700521e-04 0.9998643
Speciesvirginica   2.286765e-11   430478.6  5.312147e-17 1.0000000
```

Usando o mesmo pacote, a seleção de variáveis poderia ser feita da seguinte forma, utilizando o método de seleção `recursive feature elimination`:

```r
ctrl <- rfeControl(functions = rfFuncs, method = "cv", number = 10)

rfe_model <- rfe(
  dplyr::select(trainData, -Species, -Class),
  trainData$Class,
  sizes = c(1:4),
  rfeControl = ctrl
)
```

```
rfe_model
```

```
Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables   RMSE Rsquared     MAE  RMSESD RsquaredSD   MAESD Selected
         1 0.2206   0.7447 0.09050 0.13513     0.2198 0.06772
         2 0.1541   0.8505 0.05990 0.11741     0.1693 0.05415        *
         3 0.1687   0.8550 0.08088 0.09827     0.1550 0.04936
         4 0.1829   0.8366 0.10219 0.08731     0.1332 0.05308

The top 2 variables (out of 2):
   Petal.Length, Petal.Width
```