

P2 - Teoria do Aprendizado Estatístico

Kauã Santana

26 de novembro de 2025

Contents

1	INTRODUÇÃO	3
2	OS DADOS	3
3	REGRESSÃO LINEAR	5
3.1	Códigos Implementados	5
3.1.1	Função para o Coeficiente Angular (β_1)	5
3.1.2	Função para o Intercepto (β_0)	6
3.1.3	Função para a Reta de Regressão	6
3.2	Resultados e Análise	7
3.2.1	Saída da função <code>lm()</code>	7
3.2.2	Saída das Funções Manuais	7
4	REGRESSÃO LOGÍSTICA	7
4.1	Códigos Implementados	7
4.1.1	Modelo Completo	7
4.1.2	Função Stepwise AIC (Otimização)	8
4.2	Resultados e Análise	8
4.2.1	Saída do Resumo do Modelo	9
4.2.2	Análise dos Coeficientes	9
5	K-NEAREST NEIGHBORS (KNN)	10
5.1	Códigos Implementados	10
5.1.1	Seleção e Predição	10
5.2	Resultados e Análise	10
5.2.1	Célula Testada e Predição	10

5.2.2	Análise	11
6	ÁRVORE DE DECISÃO	11
6.1	Códigos Implementados	11
6.1.1	Criação do Modelo	11
6.1.2	Predição	12
6.2	Resultados e Análise	12
7	AGRADECIMENTOS	13

1 INTRODUÇÃO

Para a segunda avaliação da matéria de Teoria do Aprendizado Estatístico, foram desenvolvidas 4 aplicações em R de diferentes modelos de aprendizado de máquina: Regressão Linear, Regressão Logística, KNN e Árvore de Decisão. Na aplicação da Regressão Linear, foi utilizado um conjunto de dados de incêndios no Parque Cultural de Montesinho, em Portugal. Nas demais aplicações de classificação, foi utilizado um conjunto de dados fictícios de autoria da Professora Dr.^a Cibele Russo, da Universidade de São Paulo. O objetivo deste trabalho é demonstrar a implementação e a análise dos resultados obtidos com cada um desses modelos.

2 OS DADOS

Foram utilizados dois conjuntos de dados distintos para as aplicações. O primeiro conjunto é focado na previsão de área queimada, enquanto o segundo é voltado para problemas de classificação de características de funcionários.

Table 1: Dicionário do Primeiro Conjunto (Incêndios Florestais)

Variável	Atributo	Tipo	Descrição
X	Feature	Inteiro	Coordenada espacial no eixo X (mapa do parque de Montesinho). Varia de 1 a 9.
Y	Feature	Inteiro	Coordenada espacial no eixo Y (mapa do parque de Montesinho). Varia de 2 a 9.
month	Feature	Nominal	Mês do ano (abreviação em inglês: jan–dec).
day	Feature	Nominal	Dia da semana (abreviação em inglês: mon–sun).
FFMC	Feature	Contínuo	Índice Fine Fuel Moisture Code do sistema FWI. Aproximadamente entre 18.7 e 96.20.
DMC	Feature	Contínuo	Índice Duff Moisture Code. Aproximadamente entre 1.1 e 291.3.
DC	Feature	Contínuo	Índice Drought Code. Intervalo de cerca de 7.9 até 860.6.
ISI	Feature	Contínuo	Índice Initial Spread Index. Varia de 0.0 até cerca de 56.10.
temp	Feature	Contínuo	Temperatura exterior (°C). Varia aproximadamente entre 2.2 e 33.3 °C.
RH	Feature	Contínuo	Umidade relativa do ar (%). Intervalo de 15% a 100%.
wind	Feature	Contínuo	Velocidade do vento (km/h). Aproximadamente entre 0.40 e 9.40 km/h.
rain	Feature	Contínuo	Quantidade de chuva (mm/m ²). Geralmente baixa, variando entre 0.0 e 6.4.
area	Target	Contínuo	Área queimada (hectares). Varia de 0.00 até cerca de 1090.84.

Fonte: adaptado de Cortez e Morais, 2007.

Table 2: Dicionário do Segundo Conjunto (Dados de Funcionários)

Variável	Atributo	Tipo	Descrição
Idade	Feature	Inteiro	Idade do funcionário (18-54)
Gênero	Target	Nominal	Gênero do funcionário (Masculino/Feminino)
Departamento	Feature	Nominal	Departamento de atuação do funcionário
Salário	Feature	Contínuo	Salário do funcionário
Horas_Trabalhadas	Feature	Contínuo	Horas de trabalho semanal do funcionário
Produtividade	Feature	Contínuo	Nível de produtividade do funcionário (67.74-112.06)
Satisfação	Feature	Contínuo	Nível de satisfação do funcionário (39.42-100)
Tempo_Empresa	Feature	Inteiro	Tempo de empresa do funcionário (0-30)
Cursos_Realizados	Feature	Inteiro	Quantidade de cursos realizados pelo funcionário (0-20)
Home_Office	Feature	Nominal	Regime exercido pelo funcionário (Sim/Não)

Fonte: adaptado de Russo, 2025.

3 REGRESSÃO LINEAR

A Regressão Linear foi aplicada ao primeiro conjunto de dados para prever a variável **area** (área queimada em hectares) com base nas demais features. Foram implementadas funções manuais em R para calcular os coeficientes angular (β_1) e o intercepto (β_0), com o objetivo de demonstrar a compreensão da formulação matemática do Modelo de Mínimos Quadrados Ordinários.

3.1 Códigos Implementados

O código a seguir apresenta as funções criadas para o cálculo dos coeficientes de forma manual.

3.1.1 Função para o Coeficiente Angular (β_1)

```
angular <- function(x, y) {  
  
  # variaveis  
  n <- length(x)
```

```

xiyi <- sum(x * y)
x_media <- mean(x)
y_media <- mean(y)
x_var <- var(x)

# formula
dividendo <- xiyi - (n * x_media * y_media)
divisor <- (n-1) * x_var

angular <- dividendo / divisor

return(angular)
}

```

3.1.2 Função para o Intercepto (β_0)

```

# função para calcular o intercepto
intercepto <- function(x, y, beta_1) {

  # variaveis
  y_media <- mean(y)
  x_media <- mean(x)

  # formula
  intercepto <- y_media - (beta_1 * x_media)

  return(intercepto)
}

```

3.1.3 Função para a Reta de Regressão

```

# função para calcular a reta
reta <- function(angular, intercepto, x) {
  return(angular + (intercepto * x))
}

```

3.2 Resultados e Análise

Para validação, os coeficientes calculados pelas funções manuais foram comparados com os resultados obtidos pela função nativa `lm()` do R, utilizando um conjunto de dados aleatório com distribuição normal.

3.2.1 Saída da função `lm()`

Call:

```
lm(formula = y ~ x, data = data.frame(x, y))
```

Coefficients:

```
(Intercept)          x  
-0.017484    -0.003221
```

3.2.2 Saída das Funções Manuais

```
[1] "b0: -0.0174840103550686 b1: -0.00322060644479855"
```

Os coeficientes calculados pelas funções implementadas (b_0 e b_1) são idênticos aos coeficientes retornados pela função `lm()`, confirmando a correção da implementação das fórmulas para o modelo de Regressão Linear Simples.

4 REGRESSÃO LOGÍSTICA

A Regressão Logística foi utilizada para um problema de classificação binária no segundo conjunto de dados (Funcionários). O objetivo foi prever a variável **Gênero** (codificada em `Gênero_Masculino`) com base nas demais características do funcionário (`Idade`, `Salário`, `Departamento`, etc.). Foi utilizado o método `glm` com a família `binomial` e a função de ligação (*link*) `logit`.

4.1 Códigos Implementados

4.1.1 Modelo Completo

```
modelo <- glm(  
  Gênero_Masculino ~ . - Gênero_Masculino,  
  data = treino,
```

```
family = binomial(link = "logit")
)
```

4.1.2 Função Stepwise AIC (Otimização)

Esta função é um procedimento de seleção de variáveis que utiliza o Critério de Informação de Akaike (AIC) para encontrar o modelo que melhor se ajusta aos dados com o menor número de variáveis preditoras.

```
stepwise_aic <- function(x, y, verbose = TRUE) {

  # regressoras, sem a target
  dados <- subset(x, select = -c(y))

  # distribuição da target
  formula_nula <- as.formula("y ~ 1")
  formula_completa <- as.formula(paste(
    "y ~", paste(names(x), collapse = " + ")
  ))

  # steps
  modelo_step <- step(
    glm(formula_nula, data = dados, family = binomial),
    scope = list(lower = formula_nula, upper = formula_completa),
    direction = "both", trace = ifelse(verbose, 1, 0)
  )

  # resultados
  if (verbose) {
    cat("\nModelo final:\n")
    print(summary(modelo_step))
  }

  return(modelo_step)
}
```

4.2 Resultados e Análise

O resumo do modelo revela a significância estatística das variáveis para a predição do Gênero_Masculino.

4.2.1 Saída do Resumo do Modelo

```
summary(modelo)
```

Call:

```
glm(formula = Gênero_Masculino ~ . - Gênero_Masculino,  
     family = binomial(link = "logit"),  
     data = treino)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.561e+00	2.856e+00	0.547	0.5846
Idade	-3.775e-02	3.010e-02	-1.254	0.2097
Salário	1.794e-04	4.372e-05	4.104	4.06e-05 ***
Horas_Trabalhadas	7.228e-02	1.117e-01	0.647	0.5177
Produtividade	-3.566e-02	5.591e-02	-0.638	0.5237
Satisfação	-6.316e-03	1.825e-02	-0.346	0.7293
Tempo_Empresa	-7.739e-02	4.514e-02	-1.715	0.0864 .
Cursos_Realizados	1.167e-02	5.327e-02	0.219	0.8266
Departamento_RH	-1.099e-01	5.180e-01	-0.212	0.8320
Departamento_TI	-8.815e-01	4.782e-01	-1.843	0.0653 .
Departamento_Vendas	6.264e-01	4.906e-01	1.277	0.2017
Home_Office_Sim	-3.855e-01	6.112e-01	-0.631	0.5282

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 254.29 on 183 degrees of freedom
Residual deviance: 229.95 on 172 degrees of freedom
AIC: 253.95

Number of Fisher Scoring iterations: 4

4.2.2 Análise dos Coeficientes

Os resultados mostram que a variável **Salário** é a mais estatisticamente significativa para a predição do gênero ($Pr(> |z|)$ de 4.06×10^{-5} - altamente significativo). O coeficiente positivo indica que um aumento no salário está associado a uma maior proba-

bilidade de o funcionário ser classificado como Masculino (assumindo que esta é a classe de referência positiva). As variáveis `Tempo_Empresa` e `Departamento_TI` são marginalmente significativas, apresentando um $p\text{-valor} < 0.1$.

5 K-NEAREST NEIGHBORS (KNN)

foi aplicado como um classificador para determinar o `Departamento` de um funcionário com base em suas características. O KNN é um método não-paramétrico que classifica um ponto de dados com base na maioria dos K vizinhos mais próximos. Neste caso, foi utilizado $K = 20$.

5.1 Códigos Implementados

5.1.1 Seleção e Predição

O código seleciona uma amostra aleatória do conjunto de dados para teste e aplica o modelo KNN, utilizando o restante dos dados como treino.

```
# linha teste (aleatoria)
n <- sample(1:nrow(x), 1)
linha <- x[n, ]

# modelo
knn_model <- knn(
  train = x[-n, ],
  test = linha,
  cl = y[-n],
  k = 20
)
```

5.2 Resultados e Análise

5.2.1 Célula Testada e Predição

```
[1] "Célula testada (dados originais):"
ID Idade    Gênero Departamento Salário Horas_Trabalhadas Produtividade
81    32 Masculino             TI 9382.24                37          85.18
Satisfação Tempo_Empresa Cursos_Realizados Home_Office
        68.31              5                0           Não
```

```
[1] "Departamento predito:"  
[1] "TI"  
[1] "Departamento real:"  
[1] "TI"
```

5.2.2 Análise

A predição foi bem-sucedida, com o modelo classificando corretamente o funcionário 81 no departamento TI. O sucesso do KNN depende fortemente da escolha do valor de K e do pré-processamento dos dados (normalmente, a padronização das features contínuas é crucial para garantir que todas as dimensões contribuam igualmente para o cálculo da distância euclidiana). Um $K = 20$ sugere que, dos 20 vizinhos mais próximos desta amostra, a maioria (ou todos) pertenciam ao departamento de TI.

6 ÁRVORE DE DECISÃO

O modelo de Árvore de Decisão (`rpart`) foi utilizado como outro classificador para a variável `Departamento`, usando o critério de impureza Gini. Este modelo particiona recursivamente o espaço de features em regiões para maximizar a homogeneidade dentro de cada nó.

6.1 Códigos Implementados

6.1.1 Criação do Modelo

```
# Criar o modelo de árvore de decisão  
modelo <- rpart(  
  Departamento ~ Idade  
+ Gênero + Home_Office  
+ Salário + Horas_Trabalhadas  
+ Produtividade + Satisfação  
+ Tempo_Empresa + Cursos_Realizados,  
  data = data,  
  method = "class",  
  parms = list(split = "gini"),  
  control = rpart.control(cp = 0.01, minsplit = 2)  
)
```

6.1.2 Predição

O modelo é testado com um novo funcionário fictício.

```
novo_cliente <- data.frame(

  Idade          = 19,
  Gênero = 'Masculino',
  Departamento = 'TI',
  Salário = 5650.89,
  Horas_Trabalhadas =      86,
  Produtividade = 87.1,
  Satisfação =      190.90,
  Tempo_Empresa      = 11,
  Cursos_Realizados = 6
)
previsao <- predict(modelo, novo_cliente, type = "class")
previsao

1: Sim
Levels:
'Não' 'Sim'
```

6.2 Resultados e Análise

O resultado da predição para o novo funcionário foi **Sim**, com base nos níveis de classificação binária encontrados.

O algoritmo `rpart` utiliza o índice de Gini, que mede a probabilidade de um elemento ser classificado incorretamente quando escolhido aleatoriamente, para determinar as melhores divisões (splits) na árvore. O fato de o resultado da predição ser **Sim** com níveis 'Não' 'Sim' sugere que, apesar de a variável **Departamento** ter sido utilizada na fórmula, a target real que a árvore aprendeu a prever, ou a forma como a variável **Departamento** foi processada, resultou em uma classificação binária (*e.g.*, se o departamento é 'TI' ou 'Não-TI'). O ponto de corte inicial da árvore seria a feature que fornecesse o maior ganho de informação (maior redução na impureza Gini).

7 AGRADECIMENTOS

Antes de finalizar, gostaria de agradecer ao senhor, prof. João, por ter insistido tanto nos menores detalhes para a excelência dos trabalhos, mesmo que em algo tão sucinto quanto a serifa de uma fonte. Venho aprendendo com o senhor desde o primeiro semestre e, talvez, o maior aprendizado que absorvi de ti foi a persistência e dedicação naquilo que é proposto, porque eu sei que posso sempre entregar mais. Infelizmente não pude me dedicar o tanto quanto gostaria nesta matéria, mas esses projetos, e outros futuros, serão sempre revisitados e aprimorados. Serão sempre uma fonte de inspiração e conhecimento. Obrigado.

Códigos em git: <https://github.com/0kauaa/TAE>