Teoria do Aprendizado Estatístico - P2

Ellen Ozores e Mel Yukari

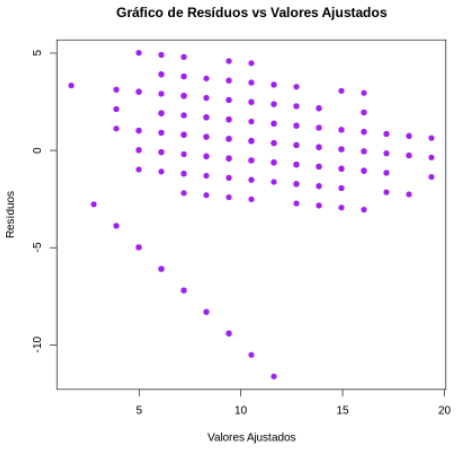
Para a realização desse trabalho, foi utilizado o mesmo dataset, “school performance”, disponível no UCI Machine Learning Repository, a fim de aplicar os conceitos dentro da Regressão Linear, Regressão Logística, KNN e Árvore de Decisão, aprendidos em aula.

Teste de Breusch-Pagan

O teste de Breusch-Pagan verifica se os resíduos do modelo têm variância constante (homocedasticidade) ou variância que muda dependendo das variáveis independentes (heterocedasticidade). Baseado no nosso dataset, a principal análise realizada é a relação da nota final (g3) com a nota do primeiro período (g1) e do segundo período (g2).

As figuras abaixo, mostram os resultados obtidos com o teste de Breusch-Pagan e o gráfico de resíduos x valores ajustados para cada regressão.

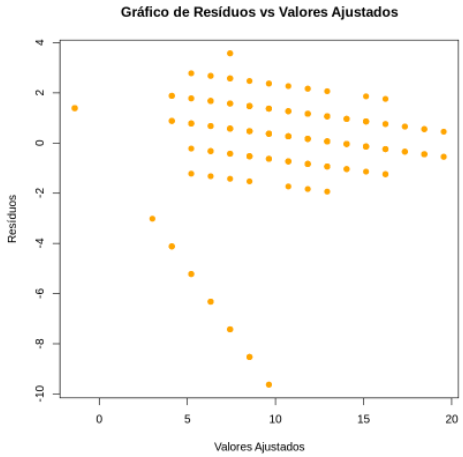
Figura 01 e 02 - Resultados para g3 X g1




```
    studentized Breusch-Pagan test

data:  modelo
BP = 25.761, df = 1, p-value = 3.865e-07
```

Fonte: Elaborado pelos autores.

Figura 03 e 04 - Resultados para g3 X g2




```
        studentized Breusch-Pagan test

data:  modelo
BP = 15.01, df = 1, p-value = 0.0001069
```

Fonte: Elaborado pelos autores.

Os gráficos gerados se mostraram similares, apresentando uma dispersão que não é homogênea (“funil” ou padrão de variância crescente/desigual dos resíduos conforme aumentam os valores ajustados) e um p-value muito abaixo de 0,05 nas duas situações. Todas essas características indicam uma heterocedasticidade no modelo, dessa forma, é necessário fazer o ajuste dos pesos para que observações com grande variância recebam um peso pequeno, e as mais confiáveis, recebam um peso maior.

Variância dos Erros

Tendo em vista o modelo heterocedástico, foi calculada a variância dos erros e o ajuste dos pesos, como mostram as figuras a seguir.

Figura 05 - Resultado da Variância dos erros g1

```
res <- resid(modelo1)
cor(res^2, dados$g1)
modelo_erros_1 <- lm(res^2 ~ g1, data = dados)
```

```
-0.255375727299539
```

Fonte: Elaborado pelos autores.

Figura 06 - Resultado da Variância dos erros g2

```
res <- resid(modelo2)
cor(res^2, dados$g2)
modelo_erros_2 <- lm(res^2 ~ g2, data = dados)
```

```
-0.19493682307559
```

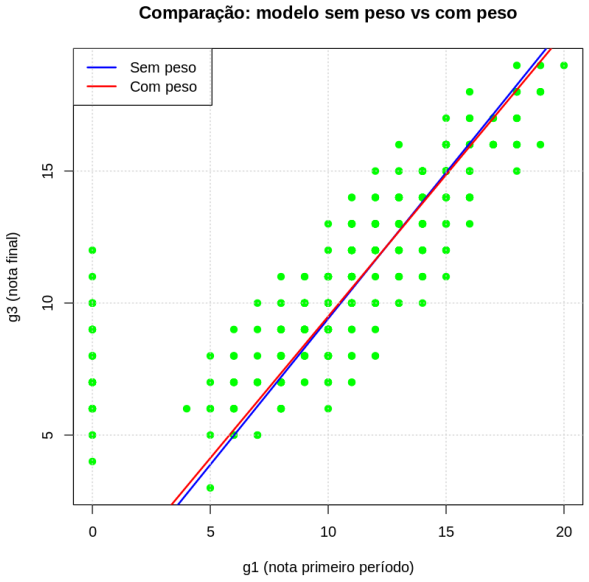
Fonte: Elaborado pelos autores.

Os resultados confirmam a heterocedasticidade nos modelos com g1 e g2 como preditores, onde os erros são maiores para notas menores. Desse modo, o ajuste com pesos é adequado porque reduz a influência das observações com maior erro, tornando o modelo estatisticamente mais confiável.

Regressão Linear - Função Lm ( )

Com as correções necessárias, utilizamos a função lm para ajustar os modelos lineares, comparando os dados com peso e sem, por meio dos gráficos de regressão. As figuras 07, 08, 09 e 10, mostram os resultados.

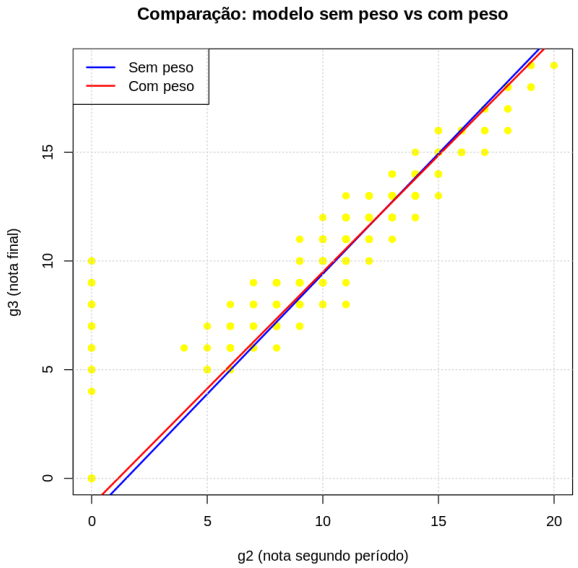
Figura 07 e 08 - Regressão linear para g3 X g1




```
Call:
lm(formula = y ~ x, data = dados)

Coefficients:
(Intercept)            x
     -1.653        1.106

Call:
lm(formula = y ~ x, data = dados, weights = weights1)

Coefficients:
(Intercept)            x
     -1.237        1.073
```

Fonte: Elaborado pelos autores.

Figura 09 e 10 - Regressão Linear para g3 x g2




```
Call:
lm(formula = y ~ x, data = dados)

Coefficients:
(Intercept)            x
     -1.393        1.102

Call:
lm(formula = y ~ x, data = dados, weights = weights2)

Coefficients:
(Intercept)            x
     -1.175        1.087
```

Fonte: Elaborado pelos autores.

As regressões entre g1 × g3 e g2 × g3 mostraram forte relação positiva, visto que os coeficientes ficaram próximos de 1,1 em ambos os casos. Após o ajuste com pesos para corrigir a heterocedasticidade, os valores mudaram pouco, mostrando que a correção não alterou o padrão da relação, então, essa estabilidade sugere que, embora houvesse alguma variação na dispersão dos erros, a correção por pesos não alterou a tendência central do modelo, apenas tornou as estimativas mais consistentes e confiáveis.

De modo geral, as notas do primeiro e segundo período demonstram forte influência sobre a nota final, indicando que o progresso acadêmico contínuo ao longo do curso está diretamente associado ao desempenho final.

Regressão Logística

A regressão logística é um método estatístico usado para prever a probabilidade de um resultado binário. Ela relaciona variáveis explicativas às chances de um evento acontecer, convertendo essa relação em probabilidades entre 0 e 1 por meio da função logística. É amplamente usada para classificação e interpretação de efeitos em termos de chances. Para isso, calculou-se o Odds Ratio (OR), que é uma medida utilizada para avaliar o efeito de cada variável sobre a

chance de ocorrência do desfecho, onde valores maiores que 1 indicam aumento das chances e menores que 1 indicam redução. Quanto mais distante de 1, mais intenso é o efeito observado.

Figura 11 - Resultados do Odds Ratio

A matrix: 9 × 5 of type dbl

	Estimate	Std. Error	z value	Pr(>|z|)	OddsRatio
(Intercept)	0.7901021	0.7297547	1.082696	2.789436e-01	2.2036215
failures	-0.7857145	0.1644509	-4.777807	1.772174e-06	0.4557939
absences	-0.2749865	0.1250856	-2.198386	2.792163e-02	0.7595824
schoolsupyes	-0.9981981	0.3952295	-2.525616	1.154956e-02	0.3685429
goout	-0.2361077	0.1383284	-1.706864	8.784742e-02	0.7896956
nurseryyes	-0.5800174	0.3769866	-1.538562	1.239112e-01	0.5598886
higheryes	1.0319142	0.7020119	1.469938	1.415785e-01	2.8064328
famsupyes	-0.4614626	0.2980745	-1.548145	1.215874e-01	0.6303610
studytime	0.2064402	0.1463806	1.410298	1.584516e-01	1.2292943

Fonte: Elaborado pelos autores.

No modelo ajustado, a variável *failures* (OR = 0,46) apresenta o impacto mais expressivo, reduzindo em mais de 50% as chances de bom desempenho, assim como a*bsences* (OR = 0,76), que também reduz significativamente. Variáveis de suporte, como *schoolsup* (OR = 0,37) e *famsup* (OR = 0,63), aparecem associadas à redução das chances, o que se explica pelo fato de que esses apoios são geralmente destinados aos alunos que já apresentam dificuldades acadêmicas.

Modelo Final

O modelo final estima a probabilidade de um estudante obter g3 ≥ 10 enquanto os coeficientes representam log-odds, que indicam a direção e intensidade do efeito de cada variável.

Figura 12 - Modelo final da regressão

```
Call:
glm(formula = y ~ failures + absences + schoolsup + goout + nursery +
    higher + famsup + studytime, family = binomial, data = dados)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.7901     0.7298   1.083   0.2789
failures      -0.7857     0.1645  -4.778 1.77e-06 ***
absences      -0.2750     0.1251  -2.198   0.0279 *
schoolsupyes  -0.9982     0.3952  -2.526   0.0115 *
goout         -0.2361     0.1383  -1.707   0.0878 .
nurseryyes    -0.5800     0.3770  -1.539   0.1239
higheryes      1.0319     0.7020   1.470   0.1416
famsupyes     -0.4615     0.2981  -1.548   0.1216
studytime      0.2064     0.1464   1.410   0.1585
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 389.72  on 315  degrees of freedom
Residual deviance: 326.75  on 307  degrees of freedom
AIC: 344.75

Number of Fisher Scoring iterations: 4
```

Fonte: Elaborado pelos autores.

A variável *failures* apresenta coeficiente negativo forte e altamente significativo, evidenciando que cada repetência reduz a probabilidade de bom desempenho, além disso, *absences* também possui efeito negativo significativo, indicando que o aumento nas faltas diminui a chance de alcançar nota alta e *schoolsup,* que reflete o fato de que esse suporte costuma ser direcionado a alunos que já apresentam dificuldades.

De forma geral, os preditores mais relevantes e estatisticamente confiáveis foram *failures*, *absences* e *schoolsup*. A redução da deviance (de 389,7 para 326,7) indica melhora substancial no ajuste, e o AIC de 344,75 demonstra um equilíbrio adequado entre qualidade do modelo e complexidade.

KNN - (Vizinhos mais Próximos)

O algoritmo K-Nearest Neighbors (KNN) foi aplicado para identificar, a partir das variáveis numéricas g1, g2, absences, studytime e failures, padrões que permitam classificar o desempenho final (g3), como alto ou baixo.

A figura a seguir, mostra o código usado e o resultado, onde o conjunto de dados está dividido em treino (70%) e teste (30%) e o modelo foi treinado com validação cruzada de 10 folds, testando diferentes valores de k (1 a 10).

Figura 13 e 14 - Código e resultado do KNN

```r
set.seed(123)

treino <- createDataPartition(dados_num$Desempenho, p = 0.7, list = FALSE)

train <- dados_num[treino, ]
test  <- dados_num[-treino, ]

knn_model <- train(
  Desempenho ~ .,
  data = train,
  method = "knn",
  trControl = trainControl(method = "cv"),  # validação cruzada (10-fold)
  tuneGrid = expand.grid(k = 1:10),         # testar k de 1 a 10
  metric = "Accuracy"
)

knn_model
```

```
k-Nearest Neighbors

278 samples
  5 predictor
  2 classes: 'Alto desempenho', 'Baixo desempenho'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 250, 251, 250, 250, 250, 249, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.9027413  0.8052683
   2  0.9096378  0.8187619
   3  0.9206167  0.8409830
   4  0.9207490  0.8410103
   5  0.9241972  0.8480870
   6  0.9206258  0.8412937
   7  0.9208812  0.8415910
   8  0.9136061  0.8269838
   9  0.9208812  0.8415910
  10  0.9174421  0.8353328

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Fonte: Elaborado pelos autores.

O algoritmo apresentou acurácia entre 0,90 e 0,92 ao longo dos diferentes valores de k, indicando boa capacidade preditiva. Ainda assim, o melhor desempenho sugerido pelo modelo foi obtido com k = 5, que alcançou acurácia de aproximadamente 92,4%, apresentando o melhor equilíbrio entre precisão e estabilidade na classificação do desempenho dos estudantes.

Para testar a acurácia do modelo, aplicou-se o modelo nos dados de teste com o uso da função *confusionMatrix* para calcular a matriz de confusão, como mostram as Figuras 15 e 16.

Figura 15 e 16 - Matriz de Confusão do modelo

```
Confusion Matrix and Statistics

                 Reference
Prediction         Alto desempenho Baixo desempenho
  Alto desempenho               58                5
  Baixo desempenho               4               50

               Accuracy : 0.9231
                 95% CI : (0.859, 0.9642)
    No Information Rate : 0.5299
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8454

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9355
            Specificity : 0.9091
         Pos Pred Value : 0.9206
         Neg Pred Value : 0.9259
             Prevalence : 0.5299
         Detection Rate : 0.4957
   Detection Prevalence : 0.5385
      Balanced Accuracy : 0.9223

       'Positive' Class : Alto desempenho
```

Fonte: Elaborado pelos autores.

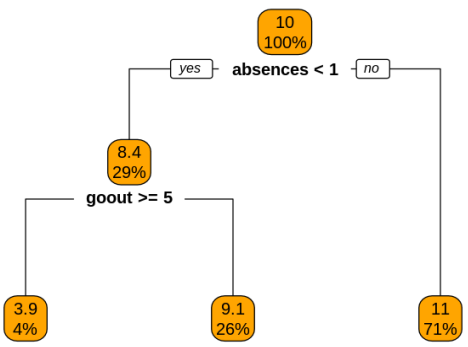
O modelo KNN apresentou acurácia de 92,3% na classificação do desempenho dos alunos, indicando excelente concordância entre as previsões e os valores reais, ademais, a sensibilidade para os alunos de alto desempenho foi de 93,5% e de baixo desempenho resultou em 90,9%, mostrando que o modelo classifica corretamente a grande maioria dos estudantes.

Árvore de Decisão

A árvore de decisão foi criada tendo como parâmetros às variáveis g3, absences, studytime e goout, para identificar os fatores que mais influenciam a nota final (g3), nesse caso, faltas, tempo de estudo e frequência de atividades sociais.

Para a aplicação, foi criado um estudante hipotético como entrada para o modelo, com 10 faltas, nível 3 de frequência em atividades sociais, onde 1 é muito baixo e 5 é muito alto, e nível 3 de tempo de estudo (5 - 10h estudadas). A figura 17 mostra a árvore encontrada com o cp ajustado pelo próprio R, através da função `best_cp <- modelo$cptable[which.min(modelo$cptable[, "xerror"]), "CP"]`.

Figura 17 - Árvore de decisão com cp definido pelo R



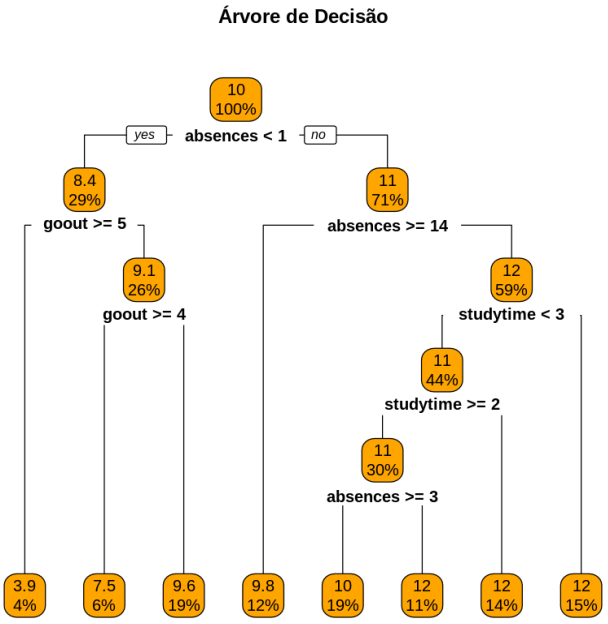
Fonte: Elaborado pelos autores.

A árvore de decisão, após o processo de poda, apresentou que absences (número de faltas) surgiu como o principal fator preditivo para a nota final (g3), sendo assim, alunos com menos de uma falta apresentaram média prevista próxima de 10, já aqueles com uma ou mais, a nota média sobe para aproximadamente 11.

Ainda dentro deste grupo, observou-se que quem sai com muita frequência (goout ≥ 5) tende a ter notas significativamente menores (≈ 3,9), enquanto valores mais baixos, estão associados a notas mais elevadas (≈ 9,1). Desse modo, a árvore de decisão indica que faltas e excesso de vida social são os fatores mais relevantes na predição do desempenho.

A figura 18, mostra o resultado obtido com o cp colocado manualmente (cp = 0.005). Essa árvore tem mais ramificações, o modelo fica mais completo e busca padrões mais específicos.

Figura 18 - Árvore de decisão para cp = 0.005



Fonte: Elaborado pelos autores.

A árvore resultante incorporou novas divisões, especialmente envolvendo as variáveis studytime, desse modo, ela identifica padrões mais específicos, como por exemplo, alunos com muitas faltas (absences ≥ 14) e pouco tempo de estudo

(studytime < 3) tendem a ter média prevista de 12. Apesar disso, a maior complexidade dificulta a interpretação geral e aumenta o risco de overfitting, isto é, o modelo se ajustar excessivamente aos dados de treino.

REFERÊNCIAS

CORTEZ, Paulo; SILVA, Alice. *Student Performance*. UCI Machine Learning Repository, 2008. Disponível em: https://archive.ics.uci.edu/dataset/320/student+performance

Código Colab: desempenhoAlunosR