Lição de Casa 3

James Hunter, Ph.D.

31 de outubro de 2023

Nesta lição da casa, vamos trabalhar com problemas de *machine learning*. No início de todos os problemas, fazer `set.seed(42)`. Senão, tudo mundo terá respostas diferentes.

Regressão Linear Múltipla

Problema 1 (`aml_dados.rds`)

O arquivo mostra dados para 2 grupos dos pacientes que morreram da leucemia mielóide aguda (AML). Os pacientes dos 2 grupos ou mostraram ou não a morfologia *AG* (hastes de Auer e/ou granulatura significativa das células leucêmicas na medula óssea). Leucemia está caraterizada por um excesso de células brancas no sangue. Maior a contagem das *WBC*, pior a doença. Queremos ver quantas semanas que os pacientes sobreviveram `semanas` dado a WBC `wbc` no tempo de diagnose.

a. Carregar os dados e fazer uma descrição das variáveis `semanas` e `wbc` utilizando `Hmisc::describe()` por cada condição de `ag_status` e fazer gráficos de dispersão. Calcular os coeficientes de correlação entre as variáveis nos 2 casos.

b. Montar uma regressão múltipla Avaliar os resultados? Fazer uma plotagem dos resíduos e de Q-Q.

c. Faça uma pouco de pesquisa. O que podemos fazer para melhorar os resultados e melhor cumprir as premissas da regressão linear?

Problema 2 (`melanoma_raw.rds`)

Voltaremos ao conjunto dos dados da Lição da Casa 3 - `melanoma`. Esta vez, queremos focar nas pessoas no estudo que morreram da melanoma. Então, precisa limpar os dados, tornar as variáveis categóricas em `factor` mesmo se elas têm valores númericos atualmente. Você pode tirar do conjunto a variável `year` porque não tem nada a ver com a questão que trataremos neste lição da casa.

a. Carregar os dados, fazer a limpeza como recommendada acima e criar um novo tibble com só esses casos que têm `status` de 1 - morreu da melanoma. Também, deve fazer uma descrição dos dados utilizando o pacote e função da sua preferência. A variável que queremos estudar é o tempo de sobrevivência.

b. Fazer uma regressão múltipla com todas as variáveis independentes utilizando `lm()` e mostrar o resumo de cada um (`summary()`). Extrair os coeficientes com `broom::tidy()`.

c. Fazer uma regressão múltipla no formato de machine learning com `caret` com validação cruzada de 3 folds e 10 repetições. Colocar 60% dos casos no grupo de treinamento. Fazer um `summary()`.

d. Utilizando `predict()`, calcular os valores para `time` previstos pelo modelo e os resíduos (diferenças dos valores certos). Determinar se os resíduos cumprem os resquisitos da regressão e se as previsões são úteis. Sugiro que desenhem um plotagem de histograma dos resíduos ou equivalente, fazer uma análise quantitativa deles e possívelmente uma plotagem Q-Q para testar normalidade.

A lição de tudo isso é que nem sempre nossos modelos e nossos dados produzem os resultados que desejamos.

Classificação

Esses problemas utilizarão o conjunto `amespt.rds` uma adaptação em português do conjunto *ames housing data* que aparece em muitos textos sobre R e *machine learning.*

O conjunto descreve a venda das casas em Ames, Iowa (EUA). O objectivo é de classificar as casas em dois grupos, aqueles com um valor alto e as outras com um valor baixo. Tratar o valor mediana de venda das casas como o ponto de divisão.

O conjunto original teve 73 varíaveis preditoras. Reduzi até 15. Pode usar todas ou algumas para montar os seus modelos. Mas, ainda tem perigo de *overfitting* com tantos casos. **Cuidado!** Traduzi os nomes de varíaveis para português. O diccionário dos dados segue.

Variable	Labels	Units	Levels	Class	Storage
preco	Preço de venda		0	integer	integer
terreno	Tamanho do terreno em sf	sf	0	integer	integer
area	Área construida da casa em sf	sf	0	integer	integer
quartos	Número de quartos		0	integer	integer
banheiros	Número de banheiros		0	integer	integer
garagem	Número de vagas na garagem		0	integer	integer
garagem_cond	Condição da garagem - 6 níveis		6		integer
ar_condicionado	Casa tem ar condicionadores centrais, SN		2		integer
ano_construcao	Ano da construção da casa	years	0	integer	integer
utilidades	Nível de serviço de agua e esgoto		3		integer
longitude	Longitude da casa	degN	0	numeric	double
latitude	Latitude da casa	degW	0	numeric	double
ano_construido			0		integer
condicao	Condição da casa		5		integer
salas	Número de salas públicas na casa		0	integer	integer
cozinha	Número das cozinhas na casa		0	integer	integer

Os dados vem do pacote `modeldata`.

Problema 3 - Estudar Dados

a. Estudar o conjunto de amespt. Fazer gráficos apropriados das variáveis chaves e prepara resumos das estatísticas delas.

b. Decidir o que seria o preço que vai usar para dividir entre valor alto e valor baixo.

Problema 4 - Regressão Logística

a. Dividir o conjunto em conjuntos `train` e `test`.

b. Usar `caret` para montar um modelo de regressão logística utilizando todas as variáveis preditoras.

c. Predizer os resultados para o conjunto `test`, montar uma matriz de confusão, e avaliar o modelo. Seu modelo consegue um nível de precisão confiável? Considerar quais variáveis contribuem ao resultado.

d. Fazer um segundo modelo com validação cruzada com 10 *folds*, repetido 10 vezes e repetir a previsão dos resultados de `test` e os parâmetros de precisão com uma matriz de confusão. Este modelo desempenhou melhor?

Problema 5 - Arvore de Decisão (`rpart`)

Trabalhar com os conjuntos test e train de Problema 4a acima.

a. Utilizando `caret` e `rpart`, repetir o processo de calcular e avaliar um modelo de *decision tree,* avaliar os valores previstos para `test` e mostrar um gráfico de arvore. Como foi o desempenho deste modelo?

Problema 6 - Random Forest (`ranger`)

Agora, voltará a conjunto completo de `amespt` e recalcular `train` e `test` com as funções de `tidymodels`.

a. Agora, siga os passos apropriados para montar um modelo de *random forests* utilizando receitas, um fluxo de trabalho e construindo um modelo em `parsnip`. Mostre os resultados do modelo e uma matriz de confusão. Como foi esta modelo? Dê o valor de 6 para hiperparâmetro `mtry`.

Problema 7 - Programação (Extra Credit)

Vamos terminar o curso com um problema de programação classico.

Parte A

Quero que vocês criam um vetor que contem os primeiros 100 elementos da série Fibonacci. A série Fibonacci começa com 0 e 1. Cada número subsequente é a soma dos 2 números anteriores. Esta sequência é muito famosa em biologia, matemática, finanças e até arte visual.

Parte B

Escrever um objeto em R que determina qual é o primeiro número na série que tem um valor acima de 1.000.000 (hum milhão). Mostre o índice desse valor e o valor exato e, claro, o código que usou para chegar neste resultado.