Machine Translated by Google

Wiley Series on Methods and Applications in Data Mining

DATA SCIENCE

USING PYTHON AND R

CHANTAL D. LAROSE | DANIEL T. LAROSE

WILEY

Machine Translated by Google

Sử dụng khoa học dữ liệu
Trăn và R

Machine Translated by Google

SERIES WILEY VỀ PHƯƠNG PHÁP VÀ ỨNG DỤNG
TRONG KHAI THÁC DỮ LIỆU

Biên tập sê-ri: Daniel T. Larose

Khai thác văn bản thực tế với Perl • Roger Bilisoly

Khám phá tri thức Máy vectơ hỗ trợ • Lutz Hamel

Khai thác dữ liệu cho Genomics và Proteomics: Phân tích dữ liệu biểu hiện gen và protein • Darius M. Dziuda

Khám phá tri thức trong dữ liệu: Giới thiệu về khai phá dữ liệu, ấn bản thứ hai • Daniel T. Larose và Chantal D. Larose

Khai thác dữ liệu và phân tích dự đoán • Daniel T. Larose và Chantal D. Larose

Khai thác dữ liệu và phân tích học tập: Ứng dụng trong nghiên cứu giáo dục • Samira ElAtia, Donald Ipperciel và Osmar R. Zaïane

Nhận dạng mẫu: Quan điểm về chất lượng dữ liệu • Władysław Homenda và Witold Pedrycz

Machine Translated by Google

Sử dụng khoa học dữ liệu Trăn và R

Chantal D. Larose Đại học
Bang Đông Connecticut Windham,
CT, Hoa Kỳ

Daniel T. Larose Central
Connecticut State University
New Britain, CT, USA

WILEY

Machine Translated by Google

Phiên bản này được xuất bản lần đầu năm 2019

© 2019 John Wiley & Sons, Inc.

Đã đăng ký Bản quyền. Không phần nào của ấn phẩm này có thể được sao chép, lưu trữ trong hệ thống truy xuất hoặc truyền đi, dưới bất kỳ hình thức nào hoặc bằng bất kỳ phương tiện nào, điện tử, cơ khí, sao chụp, ghi âm hoặc cách khác, trừ khi được pháp luật cho phép. Lời khuyên về cách xin phép sử dụng lại tài liệu từ tiêu đề này có sẵn tại http://www.wiley.com/go/permissions.

Quyền của Chantal D. Larose và Daniel T. Larose được xác định là tác giả của tác phẩm này đã được khẳng định theo luật.

Văn phòng đăng ký
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, Hoa Kỳ

tòa soạn
111 River Street, Hoboken, NJ 07030, Hoa Kỳ

Để biết chi tiết về các văn phòng biên tập toàn cầu, dịch vụ khách hàng của chúng tôi và biết thêm thông tin về các sản phẩm Wiley, hãy truy cập chúng tôi tại www.wiley.com.

Wiley cũng xuất bản sách của mình ở nhiều định dạng điện tử và in theo yêu cầu. Một số nội dung xuất hiện trong các phiên bản in tiêu chuẩn của cuốn sách này có thể không có sẵn ở các định dạng khác.

Giới hạn trách nhiệm/Từ chối bảo hành
Mặc dù nhà xuất bản và các tác giả đã cố gắng hết sức để chuẩn bị cho tác phẩm này, nhưng họ không đưa ra bất kỳ tuyên bố hay bảo đảm nào về tính chính xác hoặc tính đầy đủ của nội dung của tác phẩm này và đặc biệt từ chối mọi bảo đảm, bao gồm nhưng không giới hạn bất kỳ bảo đảm ngụ ý nào về tính thương mại hoặc tính phù hợp cho một mục đích cụ thể. Không có bảo hành nào có thể được tạo ra hoặc mở rộng bởi các đại diện bán hàng, tài liệu bán hàng bằng văn bản hoặc tuyên bố quảng cáo cho công việc này. Việc một tổ chức, trang web hoặc sản phẩm được nhắc đến trong tác phẩm này như một nguồn trích dẫn và/hoặc nguồn thông tin tiềm năng không có nghĩa là nhà xuất bản và tác giả xác nhận thông tin hoặc dịch vụ mà tổ chức, trang web hoặc sản phẩm đó có thể cung cấp hoặc khuyến nghị nó có thể thực hiện. Tác phẩm này được bán với sự hiểu biết rằng nhà xuất bản không tham gia vào việc cung cấp các dịch vụ chuyên nghiệp. Lời khuyên và chiến lược có trong tài liệu này có thể không phù hợp với tình huống của bạn. Bạn nên tham khảo ý kiến với một chuyên gia khi thích hợp. Hơn nữa, người đọc nên biết rằng các trang web được liệt kê trong tác phẩm này có thể đã thay đổi hoặc biến mất giữa thời điểm tác phẩm này được viết và khi nó được đọc. Nhà xuất bản cũng như tác giả sẽ không chịu trách nhiệm về bất kỳ tổn thất lợi nhuận hoặc bất kỳ thiệt hại thương mại nào khác, bao gồm nhưng không giới hạn đối với các thiệt hại đặc biệt, ngẫu nhiên, do hậu quả hoặc các thiệt hại khác.

Dữ liệu Biên mục của Thư viện Quốc hội

Tên: Larose, Chantal D., tác giả. | Larose, Daniel T., tác giả.
Tiêu đề: Khoa học dữ liệu sử dụng Python và R/Chantal D. Larose, Eastern Connecticut State University, Connecticut, USA, Daniel T. Larose, Central Connecticut State University, Connecticut, USA.
Mô tả: Hoboken, NJ : John Wiley & Sons, Inc, 2019. | Bao gồm chỉ số. | Định danh: LCCN 2019007280 (bản in) | LCCN 2019009632 (ebook) | ISBN 9781119526834 (Adobe PDF) | ISBN 9781119526841 (ePub) | ISBN 9781119526810 (bìa cứng)
Đối tượng: LCSH: Khai phá dữ liệu. | Python (Ngôn ngữ chương trình máy tính) | R (Ngôn ngữ chương trình máy tính) | Dữ liệu lớn. | Cấu trúc dữ liệu (Khoa học máy tính)
Phân loại: LCC QA76.9.D343 (ebook) | LCC QA76.9.D343 L376 2019 (bản in) | DDC 006.3/12–dc23

Hồ sơ LC có tại https://lccn.loc.gov/2019007280

Thiết kế bìa: Wiley
Ảnh bìa: © LumenGraphics/Shutterstock

Lấy bối cảnh 10/12pt Times bởi SPi Global, Pondicherry, Ấn Độ

In tại Hoa Kỳ

10 9 8 7 6 5 4 3 2 1

Machine Translated by Google

nội dung

lời nói đầu xi
GIỚI THIỆU VỀ TÁC GIẢ xv
Sự nhìn nhận xvii

Chương 1 Giới thiệu về Khoa học Dữ liệu 1

1.1 Tại sao lại là Khoa học Dữ liệu? 1 1.2 Khoa học dữ liệu là gì? 1
1.3 Phương pháp Khoa học Dữ liệu 2 1.4 Nhiệm vụ Khoa học Dữ liệu 5
1.4.1 Mô tả 6 1.4.2 Ước lượng 6
1.4.3 Phân loại 6
1.4.4 Phân cụm 7
1.4.5 Dự đoán 7
1.4.6 Hiệp hội 7
bài tập 8

Chương 2 CƠ BẢN VỀ PYTHON VÀ R 9

2.1 Tải xuống Python 9 2.2 Khái niệm cơ bản về mã hóa trong Python 9
2.2.1 Sử dụng Nhận xét trong Python 9
2.2.2 Thực thi các lệnh trong Python 10
2.2.3 Nhập các gói trong Python 11 2.2.4 Lấy dữ liệu vào Python 12 2.2.5 Lưu kết quả trong Python 13 2.2.6 Truy cập các bản ghi và biến trong Python 14 2.2.7 Thiết lập đồ họa trong Python 15 2.3 Tải xuống R và RStudio 17 2.4 Khái niệm cơ bản về mã hóa trong R 19 2.4.1 Sử dụng Nhận xét trong R 19 2.4.2 Thực thi lệnh trong R 20 2.4.3 Nhập gói trong R 20 2.4.4 Nhận Dữ liệu vào R 21 2.4.5 Lưu đầu ra trong R 23 2.4.6 Truy cập các bản ghi và biến trong R 24 Tài liệu tham khảo 26

bài tập 26

Machine Translated by Google

Chương 3 CHUẨN BỊ DỮ LIỆU 29

3.1 Bộ dữ liệu tiếp thị ngân hàng 29 3.2 Giai
đoạn tìm hiểu vấn đề 29 3.2.1 Trình bày rõ ràng các
mục tiêu của dự án 29 3.2.2 Chuyển các mục tiêu này thành
một bài toán khoa học dữ liệu 30 3.3 Giai đoạn chuẩn bị dữ liệu 31 3.4 Thêm
trường chỉ mục 31 3.4.1 Cách thực hiện
Thêm trường chỉ mục bằng Python 31
3.4.2 Cách thêm trường chỉ mục bằng R 32

3.5 Thay đổi giá trị trường gây hiểu lầm 33 3.5.1
Cách thay đổi giá trị trường gây hiểu lầm bằng Python 34 3.5.2 Cách thay đổi
giá trị trường gây hiểu lầm bằng R 34
3.6 Biểu thị lại dữ liệu phân loại dưới dạng số 36 3.6.1 Cách
biểu thị lại giá trị trường phân loại bằng Python 36 3.6.2 Cách biểu thị lại giá
trị trường phân loại bằng R 38 3.7 Chuẩn hóa trường số 39 3.7.1 Cách chuẩn
hóa trường số bằng Python 40 3.7 .2 Cách chuẩn hóa
trường số bằng R 40 3.8 Xác định giá trị ngoại lệ 40 3.8.1 Cách xác
định giá trị ngoại lệ bằng Python 41 3.8.2 Cách xác định giá trị
ngoại lệ bằng R 42 Tài liệu tham
khảo 43

bài tập 44

Chương 4 PHÂN TÍCH DỮ LIỆU KHÁM PHÁ 47

4.1 EDA so với HT 47
4.2 Biểu đồ thanh có lớp phủ phản hồi 47 4.2.1 Cách
tạo biểu đồ thanh có lớp phủ bằng Python 49 4.2.2 Cách tạo biểu đồ thanh có lớp
phủ bằng R 50 4.3 Bảng dự phòng 51 4.3.1 Cách tạo bảng dự phòng bằng Python
52 4.3.2 Cách xây dựng các bảng dự
phòng bằng R 53 4.4 Biểu đồ có lớp phủ phản hồi 53 4.4.1 Cách xây dựng
biểu đồ có lớp phủ bằng Python 55 4.4.2 Cách xây dựng biểu đồ có
lớp phủ bằng R 58 4.5 Tạo thành phần dựa trên giá trị
dự đoán 58 4.5. 1 Cách thực hiện Binning dựa trên giá trị dự đoán bằng Python 59
4.5.2 Cách thực hiện Binning dựa trên giá trị dự đoán bằng R 62 Tài liệu
tham khảo 63 Bài tập 63

Chương 5 CHUẨN BỊ MÔ HÌNH DỮ LIỆU 69

5.1 Câu chuyện cho đến nay 69
5.2 Phân vùng dữ liệu 69 5.2.1 Cách
phân vùng dữ liệu trong Python 70 5.2.2 Cách phân vùng
dữ liệu trong R 71

Machine Translated by Google

5.3 Xác thực phân vùng của bạn 72 5.4 Cân
bằng tập dữ liệu huấn luyện 73 5.4.1 Cách cân bằng
tập dữ liệu huấn luyện trong Python 74 5.4.2 Cách cân bằng tập dữ liệu
huấn luyện trong R 75
5.5 Thiết lập Hiệu suất Mô hình Cơ bản 77
Tài liệu tham khảo 78
bài tập 78

Chương 6 CÂY QUYẾT ĐỊNH 81

6.1 Giới thiệu về cây quyết định 81
6.2 Cây phân loại và hồi quy 83 6.2.1 Cách xây dựng cây
quyết định GIỎ HÀNG bằng Python 84 6.2.2 Cách xây dựng cây quyết định GIỎ
HÀNG bằng R 86
6.3 Thuật toán C5.0 để xây dựng cây quyết định 88
6.3.1 Cách xây dựng cây quyết định C5.0 bằng Python 89 6.3.2 Cách xây
dựng cây quyết định C5.0 bằng R 90 6.4 Rừng ngẫu nhiên 91

6.4.1 Cách xây dựng các khu rừng ngẫu nhiên trong Python 92
6.4.2 Cách xây dựng các khu rừng ngẫu nhiên trong R 92
Tài liệu tham khảo 93
bài tập 93

Chương 7 ĐÁNH GIÁ MÔ HÌNH 97

7.1 Giới thiệu về Đánh giá Mô hình 97
7.2 Các biện pháp đánh giá phân loại 97
7.3 Độ nhạy và Độ đặc hiệu 99 7.4
Độ chính xác, Thu hồi và Điểm Fβ 99 7.5
Phương pháp đánh giá mô hình 100 7.6 Ứng
dụng đánh giá mô hình 100 7.6.1 Cách thực hiện
đánh giá mô hình bằng R 103 7.7 Tính toán chi phí lỗi
không đồng đều 104 7.7.1 Tính toán cho sự
không đồng đều Chi phí lỗi khi sử dụng R 105 7.8 So
sánh các mô hình có và không có Chi phí lỗi không đồng đều 106 7.9
Chi phí lỗi do dữ liệu điều khiển
107 Bài tập 109

Chương 8 PHÂN LOẠI VỊNH NAÏVE 113

8.1 Giới thiệu về Naïve Bayes 113 8.2
Định lý Bayes 113 8.3 Giả
thuyết hậu nghiệm cực đại 114 Độc lập có
điều kiện của lớp 114 8.4
8.5 Ứng dụng Phân loại Naïve Bayes 115 8.5.1 Naïve Bayes trong
Python 121 8.5.2 Naïve Bayes trong R 123
Tài liệu tham khảo 125

bài tập 126

Machine Translated by Google

Chương 9 MẠNG Nơ-ron 129

9.1 Giới thiệu về Mạng nơ-ron 129

9.2 Cấu trúc mạng nơ-ron 129

9.3 Trọng số kết nối và Hàm kết hợp 131 9.4 Hàm kích hoạt sigmoid 133 9.5 Lan truyền ngược 134 9.6 Ứng dụng của mô hình mạng nơ-ron 134 9.7 Diễn giải các trọng số trong mô hình mạng nơ-ron 136 9.8 Cách sử dụng mạng nơ-ron trong R 137

Tài liệu tham khảo 138

bài tập 138

Chương 10 CỤM CỤM 141

10.1 Phân cụm là gì? 141 10.2 Giới thiệu về thuật toán phân cụm K-Means 142 10.3 Ứng dụng phân cụm K-Means 143 10.4 Xác thực cụm 144

10.5 Cách thực hiện phân cụm K-Means bằng Python 145 10.6 Cách thực hiện phân cụm K-Means bằng R 147 Bài tập 149

Chương 11 MÔ HÌNH HỒI QUY 151

11.1 Nhiệm vụ Ước tính 151

11.2 Lập mô hình hồi quy mô tả 151 11.3 Ứng dụng của mô hình hồi quy bội 152 11.4 Cách thực hiện mô hình hồi quy bội bằng Python 154 11.5 Cách thực hiện mô hình hồi quy bội bằng R 156 11.6 Đánh giá mô hình để ước lượng 157

11.6.1 Cách thực hiện đánh giá mô hình ước lượng bằng Python 159 11.6.2 Cách thực hiện đánh giá mô hình ước tính bằng R 160 11.7 Hồi quy từng bước 161 11.7.1 Cách thực hiện hồi quy từng bước bằng R 162 11.8 Mô hình cơ sở cho hồi quy 162 Tài liệu tham khảo 163 Bài tập 164

Chương 12 GIẢM KÍCH THƯỚC 167

12.1 Nhu cầu Giảm kích thước 167

12.2 Đa cộng tuyến 168 12.3 Xác định đa cộng tuyến bằng cách sử dụng hệ số lạm phát phương sai 171 12.3.1 Cách xác định đa cộng tuyến bằng Python 172 12.3.2 Cách xác định đa cộng tuyến trong R 173 12.4 Phân tích thành phần chính 175 12.5 Ứng dụng của phân tích thành phần chính 175 12.6 Chúng ta nên có bao nhiêu thành phần Trích xuất? 176

Machine Translated by Google

12.6.1 Tiêu chí Giá trị bản địa
176 12.6.2 Tỷ lệ phương sai được giải thích Tiêu chí 177
12.7 Thực hiện Pca với K = 4 178
12.8 Xác thực các thành phần chính 178 12.9
Cách thực hiện phân tích thành phần chính bằng Python 179 12.10 Cách thực hiện phân tích thành phần chính bằng R 181 12.11 Khi nào đa cộng tuyến không phải là vấn đề? 183
Tài liệu tham khảo 184
bài tập 184

Chương 13 MÔ HÌNH TUYẾN TÍNH TỔNG QUÁT 187

13.1 Tổng quan về các mô hình tuyến tính tổng quát 187
13.2 Hồi quy tuyến tính dưới dạng mô hình tuyến tính tổng quát 188 13.3 Hồi quy logistic dưới dạng mô hình tuyến tính tổng quát 188 13.4 Ứng dụng của mô hình hồi quy logistic 189
13.4.1 Cách thực hiện hồi quy logistic bằng Python 190
13.4.2 Cách thực hiện hồi quy logistic bằng R 191
13.5 Hồi quy Poisson 192 13.6 Ứng dụng của mô hình hồi quy Poisson 192
13.6.1 Cách thực hiện hồi quy Poisson bằng Python 193
13.6.2 Cách thực hiện hồi quy Poisson bằng R 194 Tham khảo 195
bài tập 195

Chương 14 QUY ĐỊNH HỘI 199

14.1 Giới thiệu Quy tắc Hiệp hội 199
14.2 Ví dụ đơn giản về khai thác quy tắc kết hợp 200
14.3 Hỗ trợ, độ tin cậy và thang máy 200 14.4 Khai thác quy tắc kết hợp
202 14.4.1 Cách khai thác quy tắc kết hợp bằng R 203
211 _ _ _ _ _ _ _ _ _ _ _ _ _

PHỤ LỤC TỔNG HỢP VÀ HÌNH ẢNH DỮ LIỆU 215

Phần 1: Tóm tắt 1: Xây dựng các khối phân tích dữ liệu 215
Phần 2: Trực quan hóa: Đồ thị và Bảng để Tóm tắt và Sắp xếp Dữ liệu 217
Phần 3: Tổng kết 2: Số đo Tâm, Độ biến thiên và Vị trí 222
Phần 4: Tóm tắt và Trực quan hóa Mối quan hệ hai biến 225

Machine Translated by Google

lời nói đầu

Khoa học dữ liệu sử dụng Python và R

Tại sao cuốn sách này là cần thiết

Lý do 1. Khoa học dữ liệu đang hot. Thực sự nóng. Bloomberg gọi nhà khoa học dữ liệu là "công việc hấp dẫn nhất ở Mỹ."1 Business Insider gọi đây là "Công việc tốt nhất ở Mỹ hiện nay."2 Glassdoor.com đánh giá đây là công việc tốt nhất thế giới năm 2018 trong năm thứ ba liên tiếp.3 Harvard Business Review gọi nhà khoa học dữ liệu là "Công việc hấp dẫn nhất trong thế kỷ 21."4

Lý do 2: Hai công cụ mã nguồn mở hàng đầu. Python và R là hai công cụ khoa học dữ liệu nguồn mở hàng đầu trên thế giới.5 Các nhà phân tích và lập trình viên từ khắp nơi trên thế giới làm việc chăm chỉ để xây dựng các gói phân tích mà sau đó người dùng Python và R có thể áp dụng miễn phí.

Khoa học dữ liệu Sử dụng Python và R sẽ đánh thức kiến thức chuyên môn của bạn trong lĩnh vực này-trường cạnh bằng cách sử dụng các công cụ phân tích mã nguồn mở phổ biến nhất trên thế giới. Trong Khoa học dữ liệu sử dụng Python và R, bạn sẽ tìm thấy các giải pháp thực hành từng bước cho các vấn đề kinh doanh trong thế giới thực, sử dụng các kỹ thuật tiên tiến nhất. Nói tóm lại, bạn sẽ học khoa học dữ liệu bằng cách làm khoa học dữ liệu.

Được viết cho người mới bắt đầu và người không bắt đầu như nhau

Khoa học dữ liệu sử dụng Python và R được viết cho người đọc phổ thông, không có kinh nghiệm lập trình hoặc phân tích trước đó. Chúng tôi biết rằng nền kinh tế thời đại thông tin đang khiến nhiều chuyên ngành tiếng Anh và chuyên ngành Lịch sử phải trang bị lại để tận dụng nhu cầu lớn đối với các nhà khoa học dữ liệu.6 Đây là lý do tại sao chúng tôi cung cấp các tài liệu sau đây để giúp những người mới bắt đầu làm quen với lĩnh vực này. .

[1] https://www.bloomberg.com/news/articles/2018-05-18/-sexiest-job-ignites-talent-wars-as-demand for-data-geeks-soars.

[2] https://www.businessinsider.com/what-its-like-to-be-a-data-scientist-best-job-in-america-2017-9.

[3] https://www.forbes.com/sites/louiscolumbus/2018/01/29/data-scientist-is-the-best-job-in-america theo-glassdoors-2018-ranks/#dd3f65055357.

[4] https://www.hbs.edu/faculty/Pages/item.aspx?num=43110.

5Xem ví dụ: https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science. html.

6Ví dụ: vào tháng 5 năm 2017, IBM đã dự đoán rằng nhu cầu hàng năm đối với "nhà khoa học dữ liệu, nhà phát triển dữ liệu và kỹ sư dữ liệu sẽ đạt gần 700.000 vị trí tuyển dụng vào năm 2020". Forbes, https://www.forbes.com/sites/louiscolumbus/2017/05/13/ibm-predicts-demand-for-data-scientists will-soar-28-by-2020/#6b6fde277e3b

Machine Translated by Google

- Toàn bộ chương dành riêng cho việc tìm hiểu kiến thức cơ bản về cách sử dụng Python và R, dành cho người mới bắt đầu. Nền tảng nào để sử dụng. Những gói để tải về. Mọi thứ bạn cần đã bắt đầu.
- Một phụ lục dành riêng để lấp đầy bất kỳ lỗ hổng nào mà bạn có thể có trong kiến thức nhập môn về phân tích dữ liệu của mình, được gọi là Tóm tắt và trực quan hóa dữ liệu.
- Hướng dẫn từng bước xuyên suốt. Mọi hướng dẫn cho mọi hành động.
- Mỗi chương đều có Bài tập để bạn có thể kiểm tra mức độ hiểu và tiến triển.

Những người có kinh nghiệm phân tích hoặc lập trình sẽ thích có một cửa hàng duy nhất để học cách làm khoa học dữ liệu bằng cả Python và R. Các nhà quản lý, CIO, CEO và CFO sẽ thích có thể giao tiếp tốt hơn với cơ sở dữ liệu và phân tích dữ liệu của họ các nhà phân tích. Cuốn sách này nhấn mạnh vào việc hạch toán chính xác chi phí mô hình sẽ giúp mọi người khám phá ra những kho kiến thức có lợi nhất từ dữ liệu, đồng thời tránh được những cạm bẫy tiềm ẩn có thể khiến công ty của bạn thiệt hại hàng triệu đô la.

Khoa học dữ liệu sử dụng Python và R bao gồm các chủ đề mới thú vị, chẳng hạn như sau:

- Rừng ngẫu nhiên,
- Mô hình tuyến tính tổng quát, và
- Chi phí lỗi dựa trên dữ liệu để nâng cao lợi nhuận.

Tất cả nhiều bộ dữ liệu được sử dụng trong cuốn sách đều có sẵn miễn phí trên trang web của bộ sách: DataMiningConsultant.com.

Khoa học dữ liệu sử dụng Python và R làm sách giáo khoa

Khoa học dữ liệu Sử dụng Python và R đương nhiên phù hợp với vai trò của sách giáo khoa cho một người-khóa học học kỳ hoặc trình tự hai học kỳ của các khóa học về khoa học dữ liệu sơ cấp và trung cấp. Các giảng viên của khoa sẽ đánh giá cao các bài tập ở cuối mỗi chương, tổng cộng có hơn 500 bài tập trong cuốn sách. Có ba loại bài tập, từ kiểm tra hiểu biết cơ bản đến phân tích thực hành nhiều hơn về các ứng dụng mới và đầy thách thức.

- Làm rõ các khái niệm. Những bài tập này kiểm tra hiểu biết cơ bản của học sinh về tài liệu, để đảm bảo rằng học sinh đã tiếp thu những gì họ đã đọc.
- Làm việc với Dữ liệu. Các bài tập ứng dụng này yêu cầu học sinh làm việc với Python và R, theo hướng dẫn từng bước đã được trình bày trong chương này.
- Phân tích Thực hành. Đây là nội dung thực sự của quá trình học tập dành cho sinh viên, nơi họ áp dụng kiến thức và kỹ năng mới tìm được để khám phá các mẫu và xu hướng trong các tập dữ liệu mới. Đây là nơi kiến thức chuyên môn của sinh viên được thử thách, trong điều kiện gần với thế giới thực. Hơn một nửa số bài tập trong cuốn sách bao gồm Phân tích Thực hành.

Machine Translated by Google

Các tài liệu hỗ trợ sau đây cũng có sẵn cho những người chấp nhận giảng viên của cuốn sách miễn phí.

- Hướng dẫn giải pháp đầy đủ, không chỉ cung cấp câu trả lời mà còn cung cấp cách đi đến những câu trả lời.
- Trình chiếu powerpoint của từng chương để bạn có thể giúp học sinh hiểu tài liệu hơn là chỉ giao cho học sinh đọc.

Để có quyền truy cập vào các tài liệu này, hãy liên hệ với đại diện Wiley tại địa phương của bạn và yêu cầu họ gửi email cho các tác giả xác nhận rằng bạn đã sử dụng cuốn sách này cho khóa học của mình.

Khoa học dữ liệu Sử dụng Python và R phù hợp cho các khóa học cấp đại học hoặc sau đại học nâng cao. Không yêu cầu thống kê trước đó, lập trình máy tính hoặc chuyên môn về cơ sở dữ liệu. Những gì được yêu cầu là một mong muốn học hỏi.

Cuốn sách được cấu trúc như thế nào

Khoa học dữ liệu sử dụng Python và R được cấu trúc dựa trên Phương pháp khoa học dữ liệu.

Phương pháp khoa học dữ liệu là một cách tiếp cận theo giai đoạn, thích ứng, lặp đi lặp lại để phân tích dữ liệu, trong khuôn khổ khoa học.

1. Giai đoạn tìm hiểu vấn đề. Đầu tiên, hãy trình bày rõ ràng các mục tiêu của dự án. Sau đó, chuyển các mục tiêu này thành công thức của một vấn đề có thể được giải quyết bằng khoa học dữ liệu.
2. Giai đoạn chuẩn bị dữ liệu. Làm sạch/chuẩn bị dữ liệu có lẽ là giai đoạn tốn nhiều công sức nhất trong toàn bộ quy trình khoa học dữ liệu.
   - Chương 3: Chuẩn bị dữ liệu.
3. Giai đoạn Phân tích Dữ liệu Khám phá. Hiểu rõ hơn về dữ liệu của bạn thông qua khám phá đồ họa.
   - Được trình bày trong Chương 4: Phân tích Dữ liệu Khám phá.
4. Giai đoạn thiết lập. Thiết lập hiệu suất mô hình cơ sở. Phân vùng dữ liệu. Cân bằng dữ liệu, nếu cần.
   - Đã có trong Chương 5: Chuẩn bị để lập mô hình dữ liệu.
5. Giai đoạn lập mô hình. Cốt lõi của quy trình khoa học dữ liệu. Áp dụng các thuật toán tiên tiến nhất để phát hiện ra một số mối quan hệ có lợi nghiêm trọng nằm ẩn trong dữ liệu.
   - Có trong Chương 6 và 8–14.
6. Giai đoạn Đánh giá. Xác định xem mô hình của bạn có tốt không. Chọn mô hình hoạt động tốt nhất từ một tập hợp các mô hình cạnh tranh.
   - Chương 7: Đánh giá mô hình.
7. Giai đoạn Triển khai. Giao diện với quản lý để điều chỉnh các mô hình của bạn để triển khai trong thế giới thực.

Machine Translated by Google

Giới thiệu về tác giả

Tiến sĩ Chantal D. Larose và Tiến sĩ Daniel T. Larose tạo thành một cặp cha con độc nhất vô nhị gồm các nhà khoa học dữ liệu. Đây là cuốn sách thứ ba của họ với tư cách là đồng tác giả. Trước đây, họ đã viết:

- Khai thác dữ liệu và phân tích dự đoán, Phiên bản thứ hai, Wiley, 2015.
  - Cuốn sách dày 800 trang này sẽ là người bạn đồng hành tuyệt vời với cuốn sách này, vì những người muốn tìm hiểu sâu hơn về lĩnh vực này.
- Khám phá tri thức trong dữ liệu: Giới thiệu về khai thác dữ liệu, Ấn bản thứ hai, Wiley, 2014.

Chantal D. Larose đã hoàn thành bằng Tiến sĩ về Thống kê tại Đại học Connecticut vào năm 2015, với luận án Phân cụm dữ liệu không đầy đủ dựa trên mô hình.
Với tư cách là Trợ lý Giáo sư Khoa học Quyết định tại SUNY, New Paltz, cô đã giúp phát triển chương trình Cử nhân Khoa học về Phân tích Kinh doanh. Giờ đây, với tư cách là Trợ lý Giáo sư về Thống kê và Khoa học Dữ liệu tại Đại học Bang Eastern Connecticut, cô ấy đang giúp phát triển chương trình giảng dạy khoa học dữ liệu của Khoa Khoa học Toán học.

Daniel T. Larose đã hoàn thành bằng Tiến sĩ về Thống kê tại Đại học Connecticut vào năm 1996, với luận án Phương pháp tiếp cận Bayesian đối với Phân tích tổng hợp. Ông là Giáo sư Thống kê và Khoa học Dữ liệu tại Đại học Bang Central Connecticut.
Năm 2001, ông đã phát triển chương trình Thạc sĩ Khoa học trực tuyến đầu tiên trên thế giới về Khai thác dữ liệu. Đây là cuốn sách giáo khoa 12 mà ông là tác giả hoặc đồng tác giả. Anh ấy điều hành một doanh nghiệp tư vấn nhỏ, DataMiningConsultant.com. Ông cũng chỉ đạo chương trình Thạc sĩ Khoa học Dữ liệu trực tuyến tại CCSU.

Machine Translated by Google

Sự nhìn nhận

Lời cảm ơn của Chantal

Xin gửi lời cảm ơn sâu sắc nhất đến cha Daniel của tôi, vì những câu châm biếm ngô nghê của ông khi hiệu đính. Sự hướng dẫn và niềm đam mê của anh ấy đối với nghề thủ công phản ánh và nâng cao niềm đam mê của tôi, đồng thời khiến việc làm việc với anh ấy trở thành một niềm vui. Rất cám ơn em gái Ravel của tôi, vì tình yêu vô bờ bến và những món quà khoa học và âm nhạc đáng kinh ngạc của em. Bạn đồng hành của tôi, cô ấy là một nguồn cảm hứng. Cảm ơn anh trai Tristan của tôi, vì anh ấy đã học hành chăm chỉ ở trường và để tôi đánh bại anh ấy đúng một lần ở Mario Kart. Cảm ơn mẹ Debra của tôi, vì thức ăn và những cái ôm. Ngoài ra, cà phê. Rất, rất nhiều nhờ cà phê.

Chantal D. Larose, Ph. D.

Trợ lý Giáo sư Thống kê & Khoa học Dữ liệu
Đại học Bang Đông Connecticut

Lời cảm ơn của Daniel

Đó là tất cả về gia đình. Tôi muốn cảm ơn con gái Chantal của tôi vì trí tuệ sáng suốt, sự hiện diện dịu dàng của con bé và niềm vui mà con bé mang đến mỗi ngày. Cảm ơn con gái Ravel của tôi, vì sự độc đáo của nó, và vì đã can đảm theo đuổi ước mơ của nó và trở thành một nhà hóa học. Cảm ơn Tristan, con trai tôi, vì kỹ năng tính toán và máy tính của nó, và vì sự giúp đỡ của nó khi di chuyển những tảng đá ở sân sau. Tôi cũng muốn gửi lời cảm ơn đến đứa con gái còn trong bụng mẹ Ellyriane Soleil của tôi. Làm thế nào chúng tôi bỏ lỡ những gì bạn sẽ trở thành. Cuối cùng, xin cảm ơn người vợ yêu dấu của tôi, Debra, vì tình yêu và sự quan tâm sâu sắc của cô ấy dành cho tất cả chúng tôi trong suốt những năm qua. Tôi yêu các bạn nhiều lắm.

Daniel T. Larose, Ph. D.

Giáo sư Thống kê và Khoa học Dữ liệu
Đại học Bang Trung tâm Connecticut
www.ccsu.edu/faculty/larose

Machine Translated by Google

chương 1

Giới thiệu về dữ liệu Khoa học

1.1 TẠI SAO PHẢI KHOA HỌC DỮ LIỆU?

Khoa học dữ liệu là một trong những lĩnh vực phát triển nhanh nhất trên thế giới, với số lượng việc làm trong năm 2017 tăng gấp 6,5 lần so với năm 2012.1 Nhu cầu về các nhà khoa học dữ liệu dự kiến sẽ tăng trong tương lai. Ví dụ: vào tháng 5 năm 2017, IBM đã dự đoán rằng nhu cầu hàng năm đối với "nhà khoa học dữ liệu, nhà phát triển dữ liệu và kỹ sư dữ liệu sẽ đạt gần 700.000 vị trí tuyển dụng vào năm 2020."2 http://InfoWorld.com đã báo cáo rằng "lý do số 1 khiến nhà khoa học dữ liệu vẫn là công việc hàng đầu ở Mỹ"3 là "tài năng không có nhiều". Đó là lý do tại sao chúng tôi viết cuốn sách này để giúp giảm bớt sự thiếu hụt các nhà khoa học dữ liệu có trình độ.

1.2 KHOA HỌC DỮ LIỆU LÀ GÌ?

Nói một cách đơn giản, khoa học dữ liệu là phân tích dữ liệu có hệ thống trong khuôn khổ khoa học. Đó là, khoa học dữ liệu là

- cách tiếp cận thích ứng, lặp đi lặp lại và theo giai đoạn để phân tích dữ liệu,
- được thực hiện trong khuôn khổ có hệ thống,
- phát hiện ra các mô hình tối ưu,
- bằng cách đánh giá và hạch toán chi phí thực sự của các lỗi dự đoán.

1Forbes, https://www.forbes.com/sites/louiscolumbus/2017/12/11/linkedins-fastest- Growing-jobs-today are-in-data-science-machine-learning/#5b3100f051bd

2Forbes, https://www.forbes.com/sites/louiscolumbus/2017/05/13/ibm-predicts-demand-for-data-scientists will-soar-28-by-2020/#6b6fde277e3b

3http://Infoworld.com, https://www.infoworld.com/article/3190008/big-data/3-reasons-why-data-scientist rest-the-top-job-in-america.html

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

Khoa học dữ liệu kết hợp các

- cách tiếp cận dựa trên dữ liệu của phân tích dữ liệu thống kê,
- sức mạnh tính toán và sự nhạy bén trong lập trình của khoa học máy tính, và
- kinh doanh thông minh theo miền cụ thể,

để phát hiện ra các mẫu thông tin có thể hành động và có lợi nhuận từ các cơ sở dữ liệu lớn.

Nói cách khác, khoa học dữ liệu cho phép chúng tôi trích xuất kiến thức có thể hành động từ cơ sở dữ liệu ít được sử dụng. Do đó, các kho dữ liệu đã thu thập bụi giờ đây có thể được tận dụng để phát hiện ra lợi nhuận tiềm ẩn và nâng cao lợi nhuận. Khoa học dữ liệu cho phép mọi người tận dụng một lượng lớn dữ liệu và sức mạnh tính toán để giải quyết các câu hỏi phức tạp. Các mẫu có thể phát sinh từ dữ liệu mà nếu không thì không thể phát hiện ra. Những khám phá này có thể dẫn đến những kết quả mạnh mẽ, chẳng hạn như điều trị hiệu quả hơn cho các bệnh nhân y tế hoặc mang lại nhiều lợi nhuận hơn cho một công ty.

1.3 PHƯƠNG PHÁP KHOA HỌC DỮ LIỆU

Chúng tôi tuân theo Phương pháp khoa học dữ liệu (DSM),4 giúp nhà phân tích theo dõi giai đoạn phân tích mà họ đang thực hiện. Hình 1.1 minh họa bản chất thích ứng và lặp lại của DSM, sử dụng các giai đoạn sau:

1. Giai đoạn tìm hiểu vấn đề. Các nhóm có thường xuyên làm việc chăm chỉ để giải quyết một vấn đề, để rồi sau đó phát hiện ra rằng họ đã giải quyết sai vấn đề không? Ngoài ra, nhóm tiếp thị và nhóm phân tích có thường xuyên không ở trên cùng một trang không? Giai đoạn này cố gắng tránh những cạm bẫy này.

   Một. Đầu tiên, trình bày rõ ràng các mục tiêu của dự án,

   b. Sau đó, chuyển các mục tiêu này thành công thức của một vấn đề có thể được giải quyết bằng cách sử dụng khoa học dữ liệu.

2. Giai đoạn chuẩn bị dữ liệu. Dữ liệu thô từ kho lưu trữ dữ liệu hiếm khi sẵn sàng cho các thuật toán ngay lập tức. Thay vào đó, nó cần được làm sạch hoặc "chuẩn bị để phân tích". Khi các nhà phân tích kiểm tra dữ liệu lần đầu tiên, họ phát hiện ra những vấn đề không thể tránh khỏi với chất lượng dữ liệu dường như luôn xảy ra. Chính trong giai đoạn này, chúng tôi khắc phục những vấn đề này. Làm sạch/chuẩn bị dữ liệu có lẽ là giai đoạn tốn nhiều công sức nhất trong toàn bộ quy trình khoa học dữ liệu. Phần sau đây là danh sách không đầy đủ các vấn đề đang chờ người chuẩn bị dữ liệu.

   Một. Xác định các ngoại lệ và xác định phải làm gì với chúng.

   b. Chuyển đổi và chuẩn hóa dữ liệu.

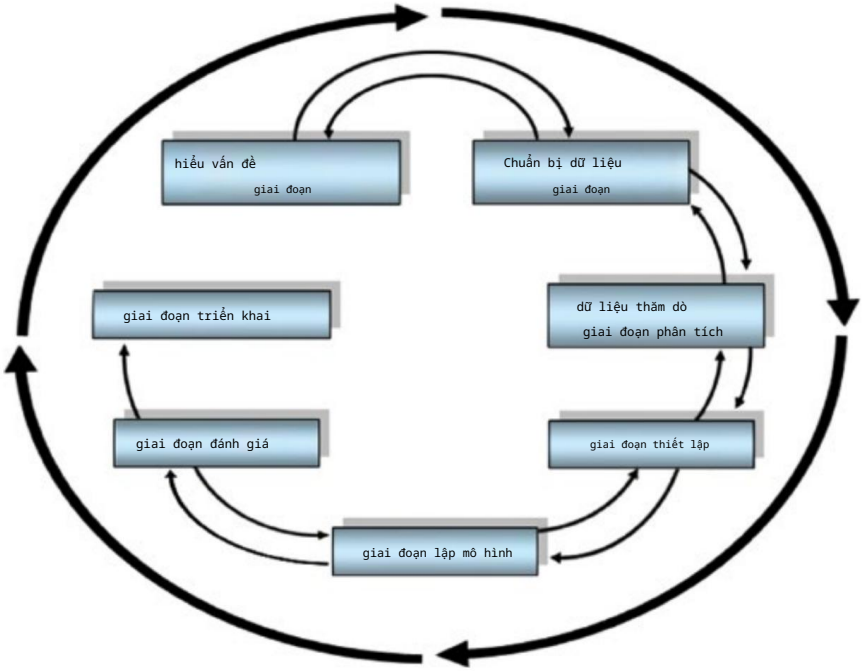
   c. Phân loại lại các biến phân loại.

   d. Binning các biến số.

   đ. Thêm một trường chỉ mục.

[4]Được điều chỉnh từ Thực hành tiêu chuẩn xuyên ngành để khai thác dữ liệu (CRISP-DM). Ví dụ, hãy xem Khai thác dữ liệu và Phân tích dự đoán, của Daniel T. Larose và Chantal D. Larose, John Wiley và Sons, Inc, 2015.

Machine Translated by Google



Hình 1.1 Phương pháp khoa học dữ liệu: bảy giai đoạn.

Giai đoạn chuẩn bị dữ liệu được trình bày trong Chương 3.

3. Giai đoạn Phân tích Dữ liệu Khám phá. Bây giờ dữ liệu của bạn đã đẹp và rõ ràng, chúng ta có thể bắt đầu khám phá dữ liệu và tìm hiểu một số thông tin cơ bản. Khám phá đồ họa là trọng tâm ở đây. Bây giờ không phải là lúc cho các thuật toán phức tạp.

   Thay vào đó, chúng tôi sử dụng các phương pháp khám phá đơn giản để giúp chúng tôi có được một số hiểu biết sơ bộ. Bạn có thể thấy rằng bạn có thể học được khá nhiều điều chỉ bằng cách sử dụng các phương pháp đơn giản này. Dưới đây là một số cách chúng ta có thể làm điều này.

   Một. Khám phá các mối quan hệ đơn biến giữa các yếu tố dự đoán và mục tiêu Biến đổi.

   b. Khám phá mối quan hệ đa biến giữa các biến.

   c. Binning dựa trên giá trị dự đoán để nâng cao mô hình của chúng tôi.

   d. Tạo ra các biến mới dựa trên sự kết hợp của các biến hiện có.

   Chúng tôi đề cập đến giai đoạn phân tích dữ liệu thăm dò trong Chương 4.

4. Giai đoạn thiết lập. Tại thời điểm này, chúng tôi gần như đã sẵn sàng để bắt đầu lập mô hình dữ liệu. Chúng ta chỉ cần quan tâm đến một số công việc quan trọng trước tiên, chẳng hạn như sau:

   Một. Xác thực chéo, gấp đôi hoặc gấp n lần. Điều này là cần thiết để tránh nạo vét dữ liệu. Ngoài ra, các phân vùng dữ liệu của bạn cần được đánh giá để đảm bảo rằng chúng thực sự là ngẫu nhiên.

   b. Cân bằng dữ liệu. Điều này nâng cao khả năng của các thuật toán nhất định để khám phá các mối quan hệ trong dữ liệu.

Machine Translated by Google

c. Thiết lập hiệu suất cơ bản. Giả sử chúng tôi đã nói với bạn rằng chúng tôi có một mô hình có thể dự đoán chính xác liệu một giao dịch thẻ tín dụng có phải là cho vay gian lận hay không trong 99% thời gian. Ấn tượng? Bạn không nên. không phải tỷ lệ giao dịch gian lận là 99,932%.5 Vì vậy, mô hình của chúng tôi có thể dự đoán một cách đơn giản rằng mọi giao dịch đều không gian lận và đúng 99,932%. Điều này minh họa tầm quan trọng của việc thiết lập hiệu suất cơ bản cho các mô hình của bạn để chúng tôi có thể hiệu chỉnh các mô hình của mình và xác định xem chúng có tốt không.

Giai đoạn thiết lập được trình bày trong Chương 5.

5. Giai đoạn lập mô hình. Giai đoạn lập mô hình thể hiện cơ hội áp dụng các thuật toán tiên tiến nhất để phát hiện ra một số mối quan hệ có lợi nghiêm trọng nằm ẩn trong dữ liệu. Giai đoạn lập mô hình là trung tâm của cuộc điều tra khoa học dữ liệu của bạn và bao gồm những điều sau:

Một. Lựa chọn và thực hiện các thuật toán mô hình thích hợp. Áp dụng các kỹ thuật không phù hợp sẽ dẫn đến kết quả không chính xác có thể khiến công ty của bạn phải trả giá đắt.

b. Đảm bảo rằng các mô hình của chúng tôi hoạt động tốt hơn các mô hình cơ bản.

c. Tinh chỉnh các thuật toán mô hình của bạn để tối ưu hóa kết quả. Cây quyết định của chúng ta nên rộng hay sâu? Mạng thần kinh của chúng ta nên có một hay hai lớp ẩn? Điều gì nên là điểm cắt của chúng tôi để tối đa hóa lợi nhuận? Các nhà phân tích sẽ cần dành thời gian tinh chỉnh các mô hình của họ trước khi đi đến giải pháp tối ưu.

Giai đoạn lập mô hình đại diện cho cốt lõi của nỗ lực khoa học dữ liệu của bạn và được đề cập trong Chương 6 và 8–14.

6. Giai đoạn Đánh giá. Đồng nghiệp của bạn tại nơi làm việc có thể nghĩ rằng anh ấy có khả năng đoán trước trận Super Bowl. Nhưng dự đoán của anh ấy có tốt không? Đó là câu hỏi. Bất cứ ai cũng có thể đưa ra dự đoán. Đó là cách các dự đoán thực hiện đối với dữ liệu thực là thử nghiệm thực sự. Trong giai đoạn đánh giá, chúng tôi đánh giá xem các mô hình của mình đang hoạt động như thế nào, liệu chúng có kiếm được tiền hay không hoặc liệu chúng tôi có cần quay lại và cố gắng cải thiện các mô hình dự đoán của mình hay không.

Một. Các mô hình của bạn cần được đánh giá dựa trên các phép đo hiệu suất cơ bản từ Giai đoạn thiết lập. Chúng ta có đang đánh bại mô hình khỉ dùng phi tiêu không? Nếu không, tốt hơn hãy thử lại.

b. Bạn cần xác định xem các mô hình của mình có thực sự giải quyết được vấn đề hiện tại hay không. Các mô hình của bạn có thực sự đạt được các mục tiêu đã đặt ra cho nó trong Giai đoạn Tìm hiểu Vấn đề không? Có phải một số khía cạnh quan trọng của vấn đề đã không được giải thích đầy đủ?

5Báo cáo gian lận Alaric, 2015, https://www.paymentscardsandmobile.com/wp-content/uploads/2015/03/PCM_Alaric_Báo cáo gian lận_2015.pdf

Machine Translated by Google

c. Áp dụng chi phí lỗi nội tại cho dữ liệu, bởi vì đánh giá chi phí dựa trên dữ liệu là cách tốt nhất để lập mô hình chi phí thực tế liên quan. Chẳng hạn, trong một chiến dịch tiếp thị, một kết quả dương tính giả không tốn kém bằng một kết quả âm tính giả. Tuy nhiên, đối với người cho vay thế chấp, thông tin dương tính giả sẽ tốn kém hơn nhiều.

d. Bạn nên lập bảng cho một bộ mô hình và xác định mô hình nào trên mỗi biểu mẫu là tốt nhất. Chọn một mô hình tốt nhất hoặc một số lượng nhỏ các mô hình để chuyển sang Giai đoạn triển khai.

Giai đoạn Đánh giá được trình bày trong Chương 7.

7. Giai đoạn Triển khai. Cuối cùng, các mô hình của bạn đã sẵn sàng cho thời gian chính! Báo cáo với ban quản lý về các mô hình tốt nhất của bạn và làm việc với ban quản lý để điều chỉnh các mô hình của bạn để triển khai trong thế giới thực.

Một. Viết báo cáo kết quả của bạn có thể được coi là một ví dụ đơn giản về triển khai. Trong báo cáo của bạn, hãy tập trung vào các kết quả mà ban quản lý quan tâm. Cho thấy rằng bạn đã giải quyết vấn đề và báo cáo về lợi nhuận ước tính, nếu có.

b. Hãy tham gia cùng với dự án! Tham gia vào các cuộc họp và quy trình liên quan đến triển khai mô hình để họ tập trung vào vấn đề hiện tại.

Cần nhấn mạnh rằng DSM có tính lặp lại và thích ứng. Bằng cách thích ứng, chúng tôi muốn nói rằng đôi khi cần phải quay lại giai đoạn trước để làm việc tiếp theo, dựa trên một số kiến thức thu được trong giai đoạn hiện tại. Đây là lý do tại sao có các mũi tên chỉ cả hai chiều giữa hầu hết các giai đoạn. Ví dụ: trong Giai đoạn Đánh giá, chúng tôi có thể thấy rằng mô hình mà chúng tôi tạo ra không thực sự giải quyết được vấn đề ban đầu và chúng tôi cần quay lại Giai đoạn Lập mô hình để phát triển một mô hình sẽ giải quyết được vấn đề đó.

Ngoài ra, DSM có tính lặp lại, trong đó đôi khi chúng ta có thể sử dụng kinh nghiệm của mình để xây dựng một mô hình hiệu quả cho một vấn đề tương tự. Nghĩa là, mô hình chúng tôi tạo ra đóng vai trò là đầu vào cho việc điều tra một vấn đề liên quan. Đây là lý do tại sao vòng mũi tên bên ngoài trong Hình 1.1 cho thấy sự tái chế liên tục các mô hình cũ hơn được sử dụng làm đầu vào để kiểm tra các giải pháp mới cho các vấn đề mới.

1.4 NHIỆM VỤ KHOA HỌC DỮ LIỆU

Các nhiệm vụ khoa học dữ liệu phổ biến nhất như sau:

- Mô tả
- Ước tính
- Phân loại
- Phân cụm
- Dự đoán
- Hiệp hội

Machine Translated by Google

Tiếp theo, chúng tôi mô tả ý nghĩa của từng nhiệm vụ này và các nhiệm vụ này được đề cập trong các chương nào.

1.4.1 Mô tả

Các nhà khoa học dữ liệu thường được yêu cầu mô tả các mẫu và xu hướng nằm trong dữ liệu. Ví dụ: một nhà khoa học dữ liệu có thể mô tả một nhóm khách hàng có nhiều khả năng rời bỏ dịch vụ của công ty chúng tôi nhất là những người có số phút sử dụng cao và số lượng cuộc gọi dịch vụ khách hàng cao. Sau khi mô tả cụm này, nhà khoa học dữ liệu có thể giải thích rằng số lượng lớn các cuộc gọi dịch vụ khách hàng cho thấy có lẽ khách hàng không hài lòng. Sau đó, làm việc với nhóm tiếp thị, nhà phân tích có thể đề xuất các biện pháp can thiệp khả thi để khám phá nhằm giữ chân những khách hàng như vậy.

Nhiệm vụ mô tả được sử dụng rộng rãi trên khắp thế giới bởi các chuyên gia cũng như những người không chuyên. Ví dụ: khi một phát thanh viên thể thao tuyên bố rằng một cầu thủ bóng chày có tỷ lệ đánh bóng trung bình trong đời (số lần đánh/số lần đánh bóng) là 0,350, thì người đó đang mô tả thành tích đánh bóng trong đời của cầu thủ này. Đây là một ví dụ mô tả 6 ví dụ khác có thể tìm thấy trong Phụ lục: Thống kê dữ liệu,
Tóm tắt và Trực quan hóa. Gần như mọi chương trong cuốn sách đều chứa các ví dụ về nhiệm vụ mô tả, từ các phương pháp EDA đồ họa ở Chương 4, đến mô tả các cụm dữ liệu ở Chương 10, đến các mối quan hệ hai biến ở Chương 11.

1.4.2 Ước lượng

Ước tính đề cập đến giá trị gần đúng của biến mục tiêu số bằng cách sử dụng tập hợp các biến dự đoán. Các mô hình ước tính được xây dựng bằng cách sử dụng các bản ghi trong đó giá trị mục tiêu đã biết, để mô hình có thể tìm hiểu giá trị mục tiêu nào được liên kết với giá trị dự đoán nào. Sau đó, các mô hình ước tính có thể ước tính các giá trị mục tiêu cho dữ liệu mới mà giá trị mục tiêu chưa biết. Ví dụ: nhà phân tích có thể ước tính số tiền thế chấp mà một khách hàng tiềm năng có thể chi trả, dựa trên một loạt các yếu tố cá nhân và nhân khẩu học. Ước tính này dựa trên một mô hình được xây dựng bằng cách xem xét các mô hình trước đây về số tiền mà những khách hàng trước đây có thể chi trả. Ước tính yêu cầu biến mục tiêu phải là số. Các phương pháp ước tính được đề cập trong Chương 9, 11 và 13.

1.4.3 Phân loại

Phân loại tương tự như ước tính, ngoại trừ biến mục tiêu là phân loại thay vì liên tục. Phân loại có lẽ là nhiệm vụ phổ biến nhất trong khoa học dữ liệu và mang lại nhiều lợi nhuận nhất. Chẳng hạn, một người cho vay thế chấp sẽ quan tâm đến việc xác định khách hàng nào của họ có khả năng vỡ nợ

6Ví dụ, xem Khám phá số liệu thống kê, của Daniel T. Larose, WH Freeman, 2016.

Machine Translated by Google

các khoản vay thế chấp. Tương tự, đối với các công ty thẻ tín dụng. Các mô hình phân loại được hiển thị rất nhiều bản ghi hoàn chỉnh có chứa trạng thái mặc định thực tế của khách hàng trong quá khứ. Sau đó, các mô hình sẽ tìm hiểu những thuộc tính nào được liên kết với những khách hàng không trả được nợ. Cuối cùng, các mô hình được đào tạo này sau đó được triển khai cho dữ liệu mới, những khách hàng đã đăng ký khoản vay hoặc thẻ tín dụng, với mong muốn rằng các mô hình sẽ giúp phân loại những khách hàng nào có nhiều khả năng không trả được khoản vay của họ nhất. Các phương pháp phân loại được trình bày trong các Chương 6, 8, 9 và 13.

1.4.4 Phân cụm

Nhiệm vụ phân cụm tìm cách xác định các nhóm bản ghi giống nhau. Ví dụ: trong tập dữ liệu về người đăng ký thẻ tín dụng, một cụm có thể đại diện cho những khách hàng trẻ hơn, có trình độ học vấn cao hơn, trong khi một cụm khác có thể đại diện cho những khách hàng lớn tuổi, ít học hơn. Ý tưởng là các bản ghi trong một cụm tương tự như các bản ghi khác trong cùng một cụm, nhưng khác với các bản ghi trong các cụm khác.

Việc tìm kiếm các cụm khả thi hữu ích ở ít nhất hai khía cạnh: (i) khách hàng của bạn có thể quan tâm đến cấu hình cụm, tức là các mô tả chi tiết về các đặc điểm của từng cụm và (ii) bản thân các cụm có thể được sử dụng làm đầu vào để phân loại hoặc các mô hình ước tính xuôi dòng. Các phương pháp phân cụm được trình bày trong Chương 10.

1.4.5 Dự đoán

Nhiệm vụ dự đoán tương tự như ước tính hoặc phân loại, ngoại trừ việc dự đoán các dự báo liên quan đến tương lai. Ví dụ, một nhà phân tích tài chính có thể quan tâm đến việc dự đoán giá cổ phiếu Apple trong ba tháng tới. Điều này sẽ đại diện cho ước tính, vì giá là một biến số và dự đoán, vì nó liên quan đến tương lai. Ngoài ra, một nhà hóa học khám phá thuốc có thể quan tâm đến việc liệu một phân tử cụ thể có dẫn đến một loại thuốc mới có lợi nhuận cho một công ty dược phẩm hay không. Điều này đại diện cho cả dự đoán và phân loại, vì biến mục tiêu là biến có/không, liệu thuốc có mang lại lợi nhuận hay không.

1.4.6 Hiệp hội

Nhiệm vụ liên kết liên quan đến việc xác định thuộc tính nào được liên kết với nhau, nghĩa là thuộc tính nào “đi cùng nhau”. Nhà khoa học dữ liệu sử dụng liên kết tìm cách khám phá các quy tắc để định lượng mối quan hệ giữa hai hoặc nhiều thuộc tính. Các luật kết hợp này có dạng, “Nếu tiền đề, thì hậu quả,” cùng với các độ đo về độ hỗ trợ và độ tin cậy của luật kết hợp. Ví dụ: các nhà tiếp thị đang cố gắng tránh việc khách hàng rời đi có thể phát hiện ra quy tắc kết hợp sau: “Nếu các cuộc gọi đến dịch vụ khách hàng lớn hơn ba lần, thì khách hàng sẽ rời bỏ”. Hỗ trợ đề cập đến tỷ lệ bản ghi mà quy tắc áp dụng; sự tự tin là tỷ lệ số lần quy tắc là chính xác. Chúng tôi đề cập đến nhiệm vụ liên kết trong Chương 14.

Machine Translated by Google

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Khoa học dữ liệu là gì?
2. Khoa học dữ liệu kết hợp những lĩnh vực nghiên cứu nào?
3. Mục tiêu của khoa học dữ liệu là gì?
4. Kể tên bảy giai đoạn của DSM.
5. Tại sao nên có Giai đoạn Tìm hiểu Vấn đề?
6. Tại sao chúng ta cần Giai đoạn Chuẩn bị Dữ liệu? Đặt tên cho ba vấn đề được xử lý trong này giai đoạn.
7. Giai đoạn nào nhà phân tích dữ liệu bắt đầu khám phá dữ liệu để học một số thao tác đơn giản thông tin?
8. Giải thích bằng ngôn từ của riêng bạn tại sao chúng ta cần thiết lập hiệu suất cơ sở cho các mô hình của mình. Điều này xảy ra trong giai đoạn nào?
9. Giai đoạn nào đại diện cho trọng tâm của cuộc điều tra khoa học dữ liệu của bạn? Tại sao chúng ta có thể áp dụng nhiều hơn một thuật toán để giải một bài toán?
10. Làm thế nào để xác định liệu dự đoán của chúng ta có chính xác hay không? Trong giai đoạn nào làm điều này xảy ra?
11. Đúng hay sai: Công việc của nhà khoa học dữ liệu được thực hiện với Giai đoạn Đánh giá. Giải thích.
12. Giải thích DSM thích ứng như thế nào.
13. Mô tả cách lặp lại của DSM.
14. Liệt kê các nhiệm vụ khoa học dữ liệu phổ biến nhất.
15. Nhiều người không chuyên đã làm những nhiệm vụ nào trong số những nhiệm vụ này?
16. Ước lượng là gì? Trong ước tính, điều gì phải đúng với biến mục tiêu?
17. Nhiệm vụ phổ biến nhất trong khoa học dữ liệu là gì? Đối với nhiệm vụ này, điều gì phải đúng với biến mục tiêu?
18. Cấu hình cụm là gì?
19. Đúng hay sai: Dự đoán chỉ có thể được sử dụng cho các biến mục tiêu phân loại. Giải thích.
20. Đối với luật kết hợp, độ hỗ trợ nghĩa là gì?

Machine Translated by Google

chương 2

CƠ BẢN CỦA PYTHON VÀ R

2.1 TẢI XUỐNG PYTHON

Để chạy mã Python, bạn cần sử dụng trình biên dịch Python. Trong văn bản này, chúng tôi sẽ sử dụng trình biên dịch Spyder, được bao gồm trong gói phần mềm Anaconda. Bằng cách tải xuống và cài đặt Anaconda, chúng tôi cũng sẽ tải xuống và cài đặt Python cùng một lúc.

Để tải xuống Anaconda, hãy truy cập trang cài đặt Spyder[1] và chọn liên kết Anaconda trong tùy chọn Windows hoặc MacOS X. Sau khi cài đặt hoàn tất, tìm chương trình Spyder và mở nó.

Khi mở Spyder lần đầu tiên, bạn sẽ thấy màn hình như Hình 2.1. Hộp bên trái là nơi bạn sẽ viết mã Python. Chiếc hộp đó là nơi chúng ta sẽ dành phần lớn thời gian của mình. Hộp trên cùng bên phải liệt kê các bộ dữ liệu và các mục khác đã được tạo bằng cách chạy mã Python. Hộp dưới cùng bên phải là nơi đầu ra của chúng tôi sẽ xuất hiện, cũng như mọi thông báo lỗi hoặc thông tin khác.

2.2 CƠ BẢN VỀ MÃ HÓA TRONG PYTHON

Trong Python, cũng như hầu hết các ngôn ngữ lập trình khác, bạn chạy mã để thực hiện một hành động. Một số hành động cũng tạo ra đầu ra. Có năm loại hành động mà chúng ta sẽ tập trung vào trong chương này: Sử dụng nhận xét, Nhập gói, Thực thi lệnh, Lưu đầu ra và Nhận dữ liệu vào Python.

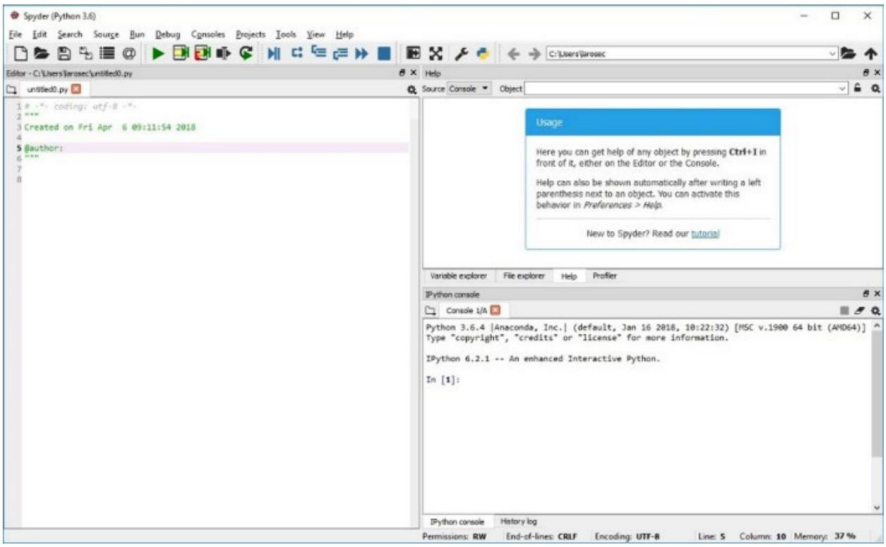
2.2.1 Sử dụng Nhận xét trong Python

Nhận xét là các đoạn mã không được trình biên dịch thực thi. Tại sao chúng ta bắt đầu chương lập trình của mình với các lệnh không chạy được? Bởi vì bình luận là một phần quan trọng của bất kỳ dự án lập trình nào.

[1] http://pythonhosted.org/spyder/installation.html

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google



Hình 2.1 Cửa sổ Spyder khi bạn mở chương trình lần đầu tiên.

Nhận xét là các dòng mã mà lập trình viên đặt ở đó để người khác hiểu mã tốt hơn. Ví dụ: nếu bạn muốn giải thích chức năng của một đoạn mã cụ thể, bạn có thể bắt đầu bằng một nhận xét giải thích nó làm gì và kết quả sẽ ra sao.

Làm cách nào để chúng tôi viết nhận xét bằng Python? Nhận xét là dòng mã bắt đầu với dấu thăng, #. Sau đây là một ví dụ về một bình luận.

```
# Đây là một bình luận!
```

Lưu ý rằng kiểu chữ cho bất kỳ mã nào, kể cả chú thích, sẽ được in đậm trong sách giáo khoa này. Điều này áp dụng cho cả mã Python và R.

2.2.2 Thực thi lệnh trong Python

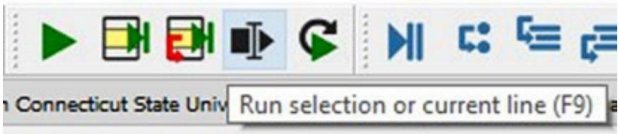
Bất kỳ mã nào bạn nhập cần phải được chạy hoặc thực thi trước khi mã đó hoạt động. Có một vài cách khác nhau để thực thi mã.

Thông thường, con trỏ của bạn sẽ nằm trên một dòng mã và bạn sẽ muốn chạy dòng đó. Có một nút trong Spyder để chạy một dòng mã. Ngoài ra còn có một phím tắt, được hiển thị nếu bạn di chuột qua nút đó. Cả nút và văn bản di chuột được hiển thị trong Hình 2.2.

Bạn có thể muốn chạy nhiều dòng mã cùng một lúc. Trong trường hợp đó, hãy đánh dấu các dòng có liên quan và nhấn nút “Chạy lựa chọn hoặc dòng hiện tại” hoặc nhấn phím tắt. Tất cả mã được đánh dấu sẽ được chạy cùng một lúc.

Bạn cũng có thể thử thực hiện nhận xét. Như đã thảo luận trước đây, các nhận xét sẽ không được biên dịch và sẽ không có đầu ra. Tuy nhiên, chúng sẽ xuất hiện trong

Machine Translated by Google



Hình 2.2 Nút và văn bản di chuột để chạy một lựa chọn hoặc dòng mã Python trong Spyder.

cửa sổ dưới cùng bên phải của Spyder. Điều này cho bạn biết rằng trình biên dịch Spyder đã xem xét bình luận, mặc dù Spyder không làm gì sau khi đọc bình luận.

2.2.3 Nhập các gói trong Python

Mặc dù nhiều thứ có thể được thực hiện bằng Python "ngay lập tức", nghĩa là ngay sau khi bạn tải xuống và cài đặt nó, nhưng hầu hết những gì chúng tôi muốn thực hiện đều yêu cầu nhập các gói ing. Các gói chứa mã được thiết kế đặc biệt cho phép chúng tôi thực hiện các nhiệm vụ khoa học dữ liệu phức tạp mà không cần tự viết mã. Ví dụ, trong Chương 6, chúng ta cần xây dựng mô hình cây phân loại và hồi quy (CART). Thay vì tìm cách xây dựng mô hình GIỎ HÀNG từ đầu, chúng tôi sẽ nhập một gói chứa mã đó. Sau khi gói được nhập, chúng ta có thể chạy mã để tạo mô hình GIỎ HÀNG.

Một số lệnh chuyên biệt, chẳng hạn như lệnh MultinomialNB() trong gói sklearn.naive_bayes (xem Chương 10). Mặt khác, có một số lệnh sẽ được sử dụng trong mỗi chương. Đây là những con gấu trúc và numpy
gói. Để nhập các gói này, bạn cần nhập và chạy hai dòng mã sau:

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
```

Lưu ý rằng chúng tôi nhập các gói bằng cách nhập. Còn mã as pd và as np sau đó thì sao ? Mã as đổi tên gói bằng tên mà chúng tôi có thể chỉ định. Chúng tôi đổi tên các gói để làm việc với chúng dễ dàng hơn.

Để sử dụng các lệnh có trong các gói pandas và numpy , chúng ta sẽ cần nêu tên gói trước tên lệnh. Ví dụ, trong Phần 2.2.4, chúng tôi sử dụng lệnh read_csv() từ pandas. Để sử dụng lệnh đó, chúng ta cần gõ pandas.read_csv(). Nếu bạn đang sử dụng một lệnh cụ thể nhiều lần hoặc sử dụng mã từ một gói có tên dài, bạn sẽ phải gõ rất nhiều! Để tiết kiệm một số thao tác gõ đó, chúng ta có thể đặt tên hiệu cho gói. Trong trường hợp trên, chúng tôi đổi tên gói gấu trúc thành pd

sử dụng như pd và đổi tên gói numpy thành np sử dụng như np. Đây là cách, trong Phần 2.2.4, chúng ta có thể sử dụng lệnh read_csv() bằng cách sử dụng mã pd.read_csv(). Đổi tên các gói sẽ giúp chúng tôi tiết kiệm rất nhiều việc gõ trong thời gian dài!

Machine Translated by Google

Chúng tôi cũng có thể nhập các phần cụ thể từ một gói mà không cần nhập toàn bộ gói. Ví dụ, trong Chương 6, chúng ta sẽ sử dụng các lệnh DecisionTreeClassifier() và export_graphviz từ sklearn.tree
bưu kiện. Để làm như vậy, chúng tôi sẽ sử dụng đoạn mã sau:

```
từ sklearn.tree nhập Quyết địnhTreeClassifier, xuất_
graphviz
```

Lưu ý rằng cú pháp đã thay đổi so với cách chúng tôi đã nhập một gói trước đây. Thay vì nói nhập sklearn.tree, bây giờ chúng ta bắt đầu bằng cách nói từ sklearn.
cây. Việc sử dụng from cho Python biết nơi tìm kiếm các lệnh mà chúng ta muốn. Sau from sklearn.tree, sau đó chúng tôi chỉ định những gì chúng tôi muốn nhập bằng cách nhập. Nếu chúng tôi chỉ nhập lệnh DecisionTreeClassifier , chúng tôi sẽ kết thúc dòng sau tên lệnh đó. Tuy nhiên, vì chúng tôi muốn nhập hai lệnh nên chúng tôi thêm dấu phẩy và tiếp tục với tên lệnh thứ hai, export_graphviz. Thực thi dòng này sẽ nhập cả hai lệnh.

2.2.4 Lấy dữ liệu vào Python

Bây giờ chúng ta sẽ thảo luận về cách đưa tập dữ liệu vào Python. Trong văn bản này, chúng tôi sẽ sử dụng lệnh read_csv , sử dụng cấu trúc sau:

```
your_name_for_the_data_set = pd.read_csv("the_path_to_
tập tin")
```

Lệnh read_csv xuất phát từ gói pandas . Theo mã trong Phần 2.2.3, chúng tôi đã nhập gói pandas dưới dạng pd. Sau khi nhập gấu trúc
gói, chúng ta có thể truy cập lệnh read_csv bằng cách gõ pd. Để sử dụng read_
csv() , nhập pd theo sau là dấu chấm, sau đó nhập lệnh read_csv.

Phần tiếp theo của mã là đường dẫn đến tệp dữ liệu, được đặt trong dấu ngoặc kép. Đối với nhiều người dùng Windows, đường dẫn sẽ bắt đầu bằng C:/và kết thúc bằng tên tệp. Ví dụ: trong Chương 4, bạn sẽ cần nhập bank_marketing_
tập dữ liệu huấn luyện . Mã để nhập tập dữ liệu được đưa ra dưới đây.

```
bank_train = pd.read_csv("C:/.../bank_marketing_
đào tạo")
```

Lập trình viên sẽ thay thế đường dẫn tệp của riêng họ cho phần “C:/./” của đường dẫn đã cho ở trên. Dưới đây là ví dụ về nhập tệp dữ liệu.

```
bank_train = pd.read_csv("C:/Users/Data Science/Data/
ngân hàng_tiếp thị_đào tạo")
```

Hướng dẫn Python trong Chương 4 yêu cầu người đọc “đọc trong bank_marketing_
tập dữ liệu huấn luyện là bank_train.” Điều này chỉ định không chỉ tập dữ liệu được nhập mà còn cả tên để gọi tập dữ liệu. Để làm theo hướng dẫn, bạn nên chỉ định

Machine Translated by Google

tên của tập dữ liệu là bank_train, như được minh họa trong mã ở trên. Điều quan trọng là phải nhớ tên mà bạn lưu tập dữ liệu và cố gắng giữ cho nó tương đối ngắn. Tên bạn lưu dữ liệu sẽ là tên bạn sẽ nhập bất cứ khi nào bạn muốn sử dụng dữ liệu đó trong phần còn lại của mã.

2.2.5 Tiết kiệm đầu ra trong Python

Một số lệnh tạo đầu ra có thể được sử dụng trong các dòng mã khác. Để sử dụng đầu ra trong mã sau này, bạn cần lưu đầu ra dưới dạng một đối tượng được đặt tên. Để lưu đầu ra, hãy làm theo cấu trúc sau:

```
your_name_for_the_output = the_command_that_generated_
đầu ra
```

Bạn có thể nhận thấy sự giống nhau của cấu trúc trên với cấu trúc mà chúng tôi đã sử dụng để nhập tập dữ liệu bank_marketing_training . Bạn có thể đã suy luận chính xác rằng lệnh mà chúng ta đã sử dụng, read_csv(), đã tạo ra một "đầu ra" của ngân hàng_ tập dữ liệu marketing_training và chúng tôi đã đặt tên cho ngân hàng đầu ra đó. Nhập tập dữ liệu sử dụng cùng một cú pháp mã hóa như lưu đầu ra dưới một tên cụ thể. Bây giờ hãy để chúng tôi minh họa sự khác biệt giữa việc chạy một lệnh và không lưu kết quả so với việc chạy một lệnh và lưu kết quả, sử dụng một bảng dự phòng.

Trong Chương 4, bạn cần tạo một bảng dự phòng và lưu nó để tạo biểu đồ thanh bằng cách sử dụng bảng đó. Mã để tạo một bảng dự phòng là một dòng mã, được đưa ra dưới đây và được hiển thị trong Hình 2.3.

```
pd.crosstab(bank_train['previous_outcome'], bank_
đào tạo ['phản hồi'])
```

Nếu bạn không lưu đầu ra được tạo bởi lệnh crosstab() , bảng dự phòng kết quả sẽ được hiển thị trong Spyder như trong Hình 2.3. Trong hình, "In" biểu thị mã chúng tôi đã chạy và "Out" biểu thị kết quả đầu ra.

Nếu chúng tôi chỉ muốn tạo một bảng và không sử dụng nó ở nơi khác trong mã của mình, thì điều này là đủ. Tuy nhiên, chúng tôi muốn sử dụng bảng này để tạo biểu đồ thanh. Vì vậy, chúng ta phải tiết kiệm nó. Để lưu nó, chúng tôi thêm tên cho mục đã lưu và dấu bằng ở bên trái của lệnh đã tạo đầu ra. Mã được đưa ra dưới đây và được hiển thị trong Hình 2.4.

```
In [9]: pd.crosstab(bank_train['previous_outcome'], bank_train['response'])
Out[9]:
response             no   yes
previous_outcome
failure            2390   385
nonexistent       21176  2034
success             320   569
```

Hình 2.3 Tạo bảng dự phòng trong Python mà không lưu đầu ra.

Machine Translated by Google


```
In [10]: crosstab_01 = pd.crosstab(bank_train['previous_outcome'], bank_train['response'])

In [11]:
```

Hình 2.4 Tạo bảng dự phòng trong Python và lưu đầu ra với tên crosstab_01.

```
In [11]: crosstab_01
Out[11]:
response              no   yes
previous_outcome
failure             2390   385
nonexistent        21176  2034
success              320   569
```

Hình 2.5 Trong Python, chúng tôi chạy tên của đầu ra đã lưu, crosstab_01, để xem đầu ra.

```
crosstab_01 = pd.crosstab(bank_train['previous_
kết quả'], bank_train['phản hồi'])
```

Tên có thể là bất kỳ thứ gì, miễn là bắt đầu bằng một chữ cái (không phải số hoặc ký hiệu) và không chứa dấu chấm hoặc các ký tự đặc biệt khác ngoài dấu gạch dưới. Trong Hình 2.4, chúng tôi đặt tên cho bảng là crosstab_01.

Lưu ý rằng không có đầu ra trong Hình 2.4. Các câu lệnh "Vào" nhảy từ "Vào [10]" sang "Vào [11]" mà không chỉ định "Ra [10]" ở giữa. Python bỏ qua câu lệnh "Out" vì đầu ra được tạo từ lệnh crosstab() đã được lưu dưới tên crosstab_01.

Để xem đầu ra, hãy chạy tên chúng tôi đã đặt cho đầu ra, trong trường hợp này crosstab_01, của chính nó. Kết quả được thể hiện trong Hình 2.5.

2.2.6 Truy cập Bản ghi và Biến trong Python

Trong cuộc phiêu lưu khoa học dữ liệu của mình, bạn có thể muốn kiểm tra một bản ghi cụ thể. Ví dụ: làm cách nào để chúng tôi truy cập một bản ghi bên trong tập dữ liệu bank_train ? chúng tôi sử dụng loc thuộc tính mà tất cả các khung dữ liệu gấu trúc đều có và cho biết bạn muốn xem phần nào của khung dữ liệu.

Python tham chiếu các bản ghi của nó bắt đầu từ bản ghi 0, vì vậy nếu chúng ta muốn xem bản ghi đầu tiên, chúng ta yêu cầu bản ghi 0. Tương tự, để xem bản ghi thứ hai, hãy yêu cầu bản ghi 1, v.v. Ví dụ: để xem bản ghi đầu tiên của bank_train, hãy sử dụng mã sau:

```
bank_train.loc[0]
```

Sử dụng thuộc tính .loc như trên để xem bản ghi đầu tiên sẽ trả về giá trị của tất cả các biến cho bản ghi đó. Hình 2.6 cho thấy bốn giá trị biến đầu tiên cho bản ghi đầu tiên trong tập dữ liệu bank_train .

Machine Translated by Google


```
In [30]: bank_train.loc[0]
Out[30]:
age                          56
job                   housemaid
marital                 married
education             basic.4y
```

Hình 2.6 Python cho chúng ta thấy các giá trị biến của bản ghi đầu tiên trong tập dữ liệu bank_train (bốn biến đầu tiên được hiển thị).

Nếu chúng ta muốn truy cập nhiều bản ghi thì sao? Chúng tôi sẽ sử dụng thuộc tính .loc và liệt kê các bản ghi mà chúng tôi muốn xem. Nếu bạn muốn các hàng đầu tiên, thứ ba và thứ tư, hãy sử dụng đoạn mã sau:

```
bank_train.loc[[0, 2, 3]]
```

Mặt khác, nếu bạn muốn bao gồm 10 hàng đầu tiên, bạn sẽ sử dụng đoạn mã sau:

```
bank_train[0:10]
```

Trong khi chúng tôi tham chiếu các hàng theo số, các cột của chúng tôi có tên. Điều này có nghĩa là nếu chúng ta muốn chỉ định biến nào chúng ta muốn xem, chúng ta đặt tên cho biến đó.

```
bank_train['tuổi']
```

Sử dụng một bộ dấu ngoặc và đặt tên biến trong dấu ngoặc đơn sẽ trả về toàn bộ biến. Mã và bốn giá trị tuổi đầu tiên được hiển thị trong Hình 2.7.

Nếu chúng ta muốn xem nhiều biến thì sao? Chúng tôi sẽ sử dụng thuộc tính .loc và liệt kê các biến mà chúng tôi muốn xem. Nếu chúng ta muốn xem các biến tuổi và công việc, chúng ta đặt từng tên biến đó trong dấu ngoặc đơn, bên trong dấu ngoặc kép, được phân tách bằng dấu phẩy.

```
bank_train[['tuổi', 'công việc']]
```

Đầu ra của lệnh này được hiển thị trong Hình 2.8.

2.2.7 Thiết lập đồ họa trong Python

Trước khi rời khỏi phần mã hóa Python, chúng ta cần giải quyết một điều nữa: cách lấy và chỉnh sửa đầu ra đồ họa trong Python.

Theo mặc định, Spyder hiển thị tất cả đồ họa trong bảng điều khiển IPython ở cấp thấp hơn cửa sổ bên phải. Một ví dụ về điều này được thể hiện trong Hình 2.9, sử dụng một biểu đồ. Những màn hình nhỏ này, không có tùy chọn chỉnh sửa, có thể chấp nhận được đối với các biểu đồ đơn giản. Tuy nhiên, chúng tôi sẽ tạo đồ họa phức tạp, sẽ yêu cầu chỉnh sửa thêm và hiển thị lớn hơn. Để cho phép chúng tôi xem và thao tác chi tiết với đồ thị, chúng tôi cần thay đổi cài đặt đồ họa của Spyder.

Machine Translated by Google


```
In [48]: bank_train['age']
Out[48]:
0        56
1        57
2        41
3        25
```

Hình 2.7 Python hiển thị các giá trị của biến tuổi (bốn giá trị đầu tiên được hiển thị).

```
In [49]: bank_train[['age', 'job']]
Out[49]:
       age           job
0       56     housemaid
1       57      services
2       41   blue-collar
3       25      services
```

Hình 2.8 Python hiển thị các giá trị của biến tuổi và công việc (bốn giá trị đầu tiên được hiển thị).

Các bước sau đây chỉ cần thực hiện một lần sẽ thiết lập đồ họa tùy chọn chúng ta cần:

1. Trong Spyder, nhấp vào Tools trên thanh menu, sau đó chọn Preferences.
2. Trong danh sách ở phía bên trái của cửa sổ Tùy chọn, nhấp vào Bảng điều khiển IPython.
3. Chọn tab Đồ họa ở trên cùng của cửa sổ bên tay phải.
4. Trong Phần phụ trợ đồ họa, hãy nhấp vào menu thả xuống Phần phụ trợ và chọn Tự động. Một ví dụ về cửa sổ tại thời điểm này trong quy trình được hiển thị trong Hình 2.10.
5. Sau khi chọn Tự động , nhấp vào nút Áp dụng và OK .

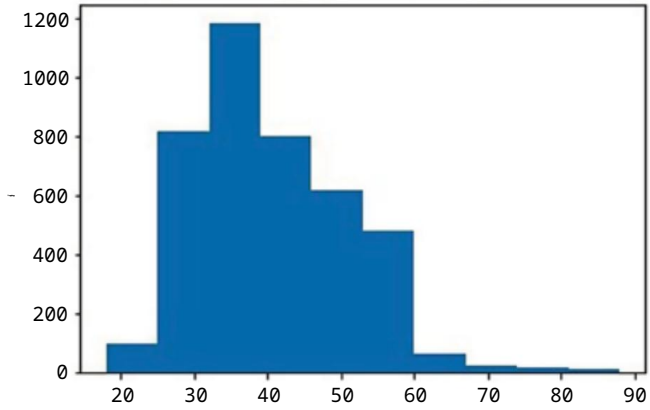
Khi các tùy chọn đồ họa được thay đổi theo các bước trên, hãy đóng Spyder và mở lại để cài đặt mới có hiệu lực.

Thay đổi phụ trợ đồ họa sẽ mở đầu ra đồ họa trong một cửa sổ mới. Cửa sổ, với đồ họa được hiển thị, được hiển thị trong Hình 2.11. Ngoài việc cho phép chúng ta xem đồ họa chi tiết hơn, cửa sổ còn cung cấp một số tùy chọn tùy chỉnh sẽ có ích trong suốt phần còn lại của cuốn sách này. Ví dụ, hãy xem xét nút Configure subplots, thứ ba từ bên phải, được biểu thị trong Hình 2.11 bằng văn bản di chuột. Chọn nút này sẽ cho phép chúng tôi thay đổi lề của cốt truyện. Nút Chỉnh sửa Trục, thứ hai từ bên phải, sẽ cho phép chúng tôi chỉnh sửa tiêu đề và nhãn trục. Nút lưu, đầu tiên bên phải, sẽ cho phép bạn lưu đồ họa. Vui lòng thử nghiệm với các cài đặt này khi bạn nhận được đầu ra đồ họa từ mã trong văn bản này.

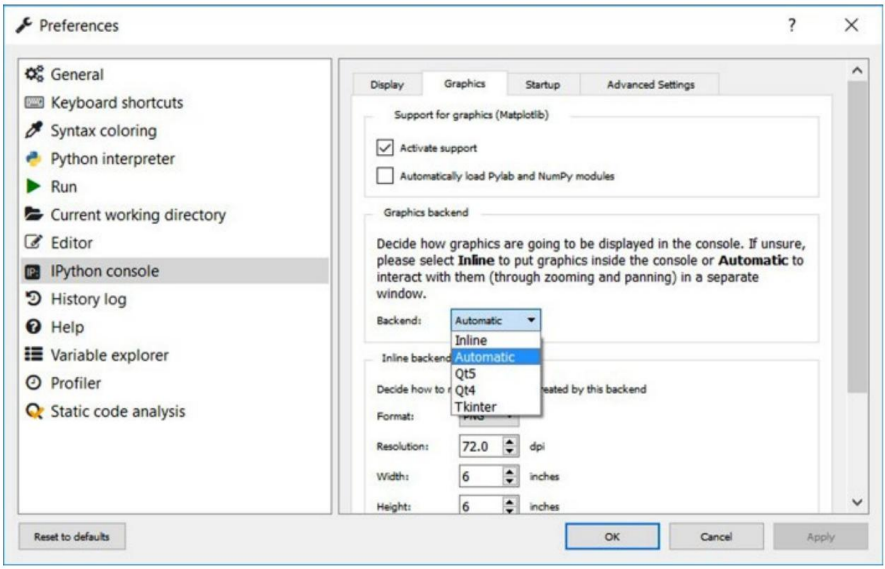
Bây giờ bạn đã học được các thao tác cốt lõi của lập trình bằng Python và đang trên đường trở thành một lập trình viên Python! Chúng tôi sẽ đề cập đến các lệnh, gói và đầu ra cụ thể hơn trong các chương sau, với các mẹo và giải thích xuyên suốt.

Machine Translated by Google


```
In [5]: bank_train['age'].plot(kind='hist')
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1aff31e1518>
```



Hình 2.9 Một ví dụ về biểu đồ được hiển thị trong phần đầu ra của Spyder.

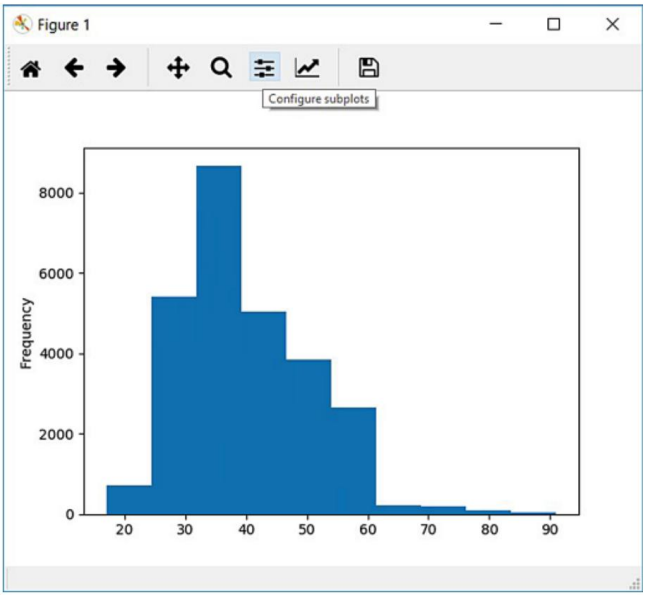


Hình 2.10 Cửa sổ Preferences trong Spyder khi bạn chuẩn bị thay đổi các tùy chọn đồ họa.

2.3 TẢI XUỐNG R VÀ RSTUDIO

Đã đến lúc chuyển đổi ngôn ngữ lập trình. Trong phần này, chúng tôi đề cập đến các kỹ năng cơ bản tương tự đối với ngôn ngữ lập trình thống kê R mà chúng tôi đã thực hiện trong phần trước đối với Python. Trong chương này và xuyên suốt cuốn sách này, bạn sẽ tìm thấy rất nhiều điểm tương đồng trong cách Python và R thực hiện mọi việc, chẳng hạn như tiết kiệm đầu ra và rất nhiều điểm khác biệt, chẳng hạn như nhập các gói.

Machine Translated by Google



Hình 2.11 Cửa sổ mới từ Spyder chứa đầu ra đồ họa của chúng ta, với nút Configure subplots được biểu thị.

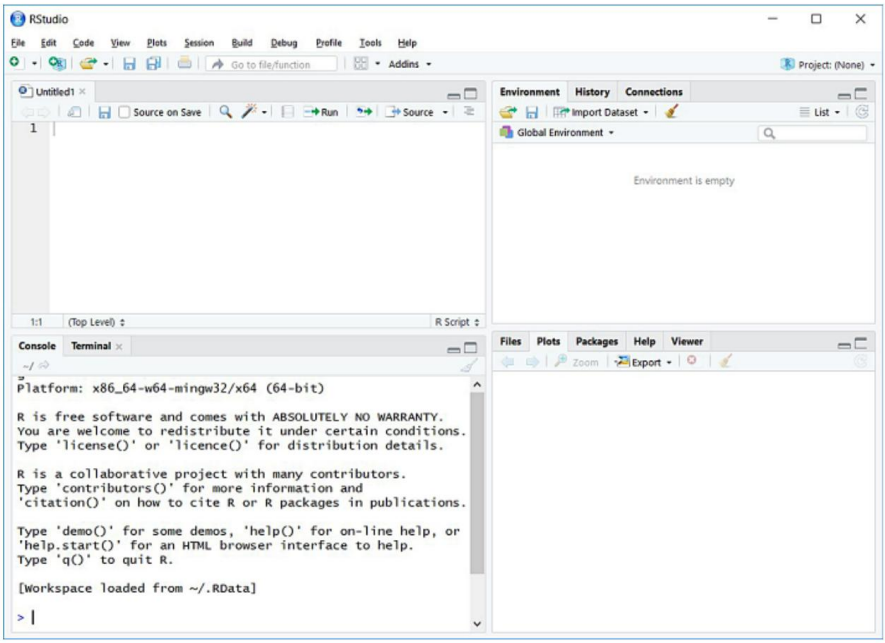
Để chạy mã R, chúng tôi cần tải xuống cả R và RStudio. Để tải xuống R, hãy truy cập trang cài đặt R,[2] chọn một máy nhân bản và làm theo hướng dẫn để tải xuống R cho hệ điều hành của bạn. Để tải xuống RStudio, hãy truy cập trang cài đặt RStudio[3] và chọn liên kết tải xuống cho hệ điều hành của bạn. Sau khi cài đặt, tìm RStudio và mở nó.

Hình 2.12 hiển thị cửa sổ RStudio khi bạn mở nó lần đầu tiên. Nếu bạn mở RStudio và chỉ có ba bảng điều khiển, hãy nhấp vào Tệp > Tệp mới > Tập lệnh R để hiển thị màn hình bốn bảng như trong Hình 2.12. Ô trên cùng bên trái là nơi bạn sẽ nhập mã R của mình. Hộp trên cùng bên phải có nút “Nhập tập dữ liệu”, chúng tôi sẽ sử dụng nút này để đọc dữ liệu vào R. Hộp này cũng có tab “Môi trường”, tab này sẽ hiển thị tất cả các tập dữ liệu và đối tượng bạn nhập hoặc tạo trong R. Hộp dưới cùng - hộp bên trái là nơi đầu ra dựa trên văn bản sẽ xuất hiện, cũng như bất kỳ phản hồi hoặc thông báo lỗi nào. Cuối cùng, hộp dưới cùng bên phải có nhiều tab. Chúng tôi sẽ dành phần lớn thời gian của mình trong tab “Plots”, nơi mọi đầu ra đồ họa sẽ được hiển thị. Hộp dưới cùng bên phải cũng có tab “Trợ giúp”, rất hữu ích để truy cập nhanh tài liệu trên các lệnh R.

[2] https://cran.r-project.org/mirrors.html

[3] http://rstudio.com/products/rstudio/download

Machine Translated by Google



Hình 2.12 Cửa sổ RStudio khi bạn mở chương trình lần đầu tiên.

2.4 CƠ BẢN VỀ MÃ HÓA TRONG R

Với R, cũng như với Python, bạn thực thi một lệnh tạo đầu ra. Phần lớn cấu trúc của mã R sẽ giống với mã Python, với một số điểm khác biệt quan trọng. Trong phần này, chúng ta thảo luận về năm loại hành động lập trình mà trước đây chúng ta đã đề cập bằng cách sử dụng Python, lần này là sử dụng R: Sử dụng nhận xét, Nhập gói, Thực thi lệnh, Lưu đầu ra và Lấy dữ liệu vào Python.

2.4.1 Sử dụng Nhận xét trong R

Việc sử dụng các nhận xét cũng quan trọng đối với mã hóa trong R cũng như đối với mã hóa trong Python. Nhận xét cho phép bạn mô tả chức năng của mã và các thông tin quan trọng khác.

Nhận xét là các dòng mã bắt đầu bằng dấu thăng, #, chẳng hạn như mã bên dưới.

```
# Đây là một nhận xét và sẽ không được biên dịch bởi R!
```

Hãy nhớ rằng mã R sẽ được in đậm trong suốt văn bản này. Đừng ngại đưa nhận xét vào mã của bạn, ngay cả khi không được gợi ý bởi các ví dụ và bài tập trong văn bản này. Sử dụng công cụ này để giữ cho mã của bạn rõ ràng và dễ hiểu đối với người khác nhất có thể!

Machine Translated by Google

2.4.2 Thực thi các lệnh trong R

Mã R mà bạn viết cần phải được thực thi trước khi nó thực hiện những gì nó phải làm.

Hầu hết thời gian, bạn sẽ muốn chạy một dòng mã. Có hai cách để thực hiện điều này. Trước tiên, bạn có thể nhấp vào nút Chạy bên trong bảng trên cùng bên trái trong RStudio, đây cũng chính là bảng mà bạn sẽ nhập mã R. Bạn cũng có thể sử dụng phím tắt, phím này sẽ hiển thị nếu bạn di chuột qua nút Chạy. Cả nút và văn bản di chuột (đối với hệ điều hành Windows) được hiển thị trong Hình 2.13.

2.4.3 Nhập gói trong R

Mặc dù "cơ sở" hoặc bản tải xuống ban đầu của R bao gồm nhiều lệnh hữu ích cho các nhà khoa học dữ liệu, nhưng nó không bao gồm mọi thứ chúng ta sẽ cần trong sách giáo khoa này. Do đó, trong suốt cuốn sách, chúng ta sẽ cần tải xuống và mở các lô lệnh R được thiết kế đặc biệt, được gọi là các gói. Có hai bước để cung cấp mã bổ sung này để sử dụng: (i) Tải xuống gói chứa mã và (ii) Mở gói.

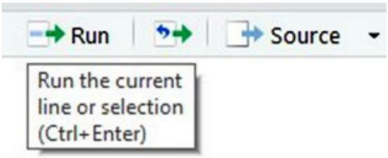
Hãy để chúng tôi chứng minh quá trình này cho một gói có tên là ggplot2. Gói này sẽ được sử dụng trong Chương 4, vì vậy bằng cách nhập mã này, chúng tôi sẽ chuẩn bị cho chương đó ngay bây giờ. Hai dòng mã để tải xuống và truy cập mã được đưa ra bên dưới.

```
install.packages("ggplot2")
thư viện (ggplot2)
```

Đầu tiên, chúng ta hãy xem dòng mã đầu tiên. Lệnh tải xuống một gói là cài đặt. gói(). Để tải xuống gói ggplot2 , hãy đặt tên của gói bên trong tệp cài đặt. lệnh packages() trong dấu ngoặc kép. Làm như vậy sẽ cho chúng ta dòng mã đầu tiên ở trên.

Nếu bạn đang tải xuống một gói lần đầu tiên, bạn sẽ phải chọn một máy nhân bản CRAN trước khi quá trình cài đặt có thể tiếp tục. Mặc dù bạn có thể chọn bất kỳ, nhưng bạn có thể muốn chọn một địa điểm gần với vị trí địa lý của mình.

Điều quan trọng cần lưu ý là bạn chỉ cần tải xuống một gói một lần. Sau khi gói được cài đặt xong, nó đã ở trên máy tính của bạn, sẵn sàng để mở và sử dụng



Hình 2.13 Nút và văn bản di chuột để chạy một lựa chọn hoặc dòng mã R trong RStudio.

Machine Translated by Google

trong bất kỳ mã nào bạn muốn. Tuy nhiên, trước khi bạn sử dụng các lệnh trong gói ggplot2 , bạn cần mở nó. Bạn sẽ cần làm điều này bất cứ khi nào bạn muốn sử dụng các chức năng trong gói.

Để làm như vậy, chúng tôi sử dụng dòng mã thứ hai ở trên. Lệnh để mở một gói là thư viện(). Để mở gói ggplot2 , hãy đặt tên của gói bên trong lệnh library() , lần này không có dấu ngoặc kép. Làm như vậy sẽ cho chúng ta dòng mã thứ hai ở trên.

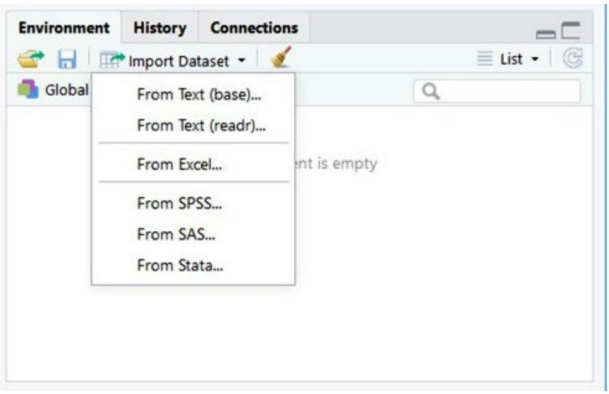
2.4.4 Lấy dữ liệu vào R

Có hai cách để đưa tập dữ liệu vào R: sử dụng nút "Nhập tập dữ liệu" trong tab Môi trường RStudio (chúng tôi đặc biệt khuyên dùng!) hoặc mã hóa đường dẫn tệp vào R.

Phương pháp đơn giản nhất để đưa một tập dữ liệu vào R là nút Nhập Tập dữ liệu, được tìm thấy ở đầu cửa sổ trên cùng bên phải trong RStudio. Cửa sổ trên cùng bên phải được hiển thị trong Hình 2.14, với nút Nhập tập dữ liệu được chọn. Việc chọn nút sẽ gửi trước cho bạn các tùy chọn khác nhau. Chúng tôi sẽ chọn "Từ văn bản (cơ sở)." làm tùy chọn của mình.

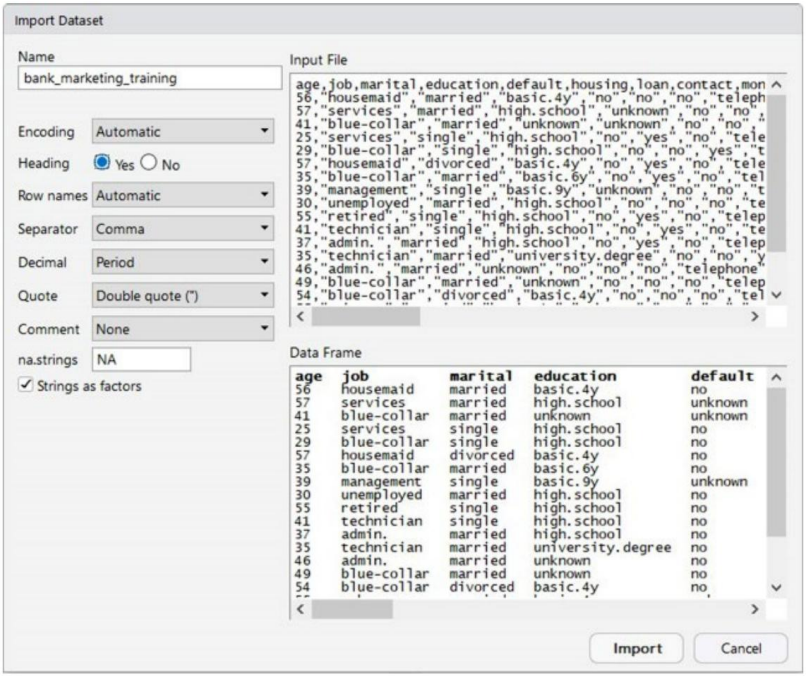
Sau khi chọn "Từ văn bản (cơ sở).", bạn sẽ thấy một cửa sổ duyệt tệp. Sử dụng cửa sổ để định vị tập dữ liệu của bạn. Điều hướng đến tập dữ liệu của bạn và chọn "Mở".

Sau khi bạn chọn tệp tập dữ liệu của mình và nhấp vào "Mở", một cửa sổ "Nhập tập dữ liệu" mới sẽ xuất hiện. Cửa sổ, như trong Hình 2.15, liệt kê ở phía bên trái nhiều tùy chọn để nhập tập dữ liệu. Việc lựa chọn "Heading" có tầm quan trọng đặc biệt. Tất cả các bộ dữ liệu chúng tôi sẽ sử dụng trong văn bản này đều có tiêu đề cột, vì vậy điều quan trọng là phải cho R biết rằng các tiêu đề này tồn tại. Đảm bảo nút "Có" cho "Tiêu đề" được chọn. Nếu bạn di chuyển giữa các tùy chọn "Có" và "Không", bạn sẽ thấy các thay đổi trong cửa sổ phía dưới bên phải, đây là bản xem trước của tập dữ liệu vì nó sẽ được nhập vào R. Đảm bảo tên cột ("tuổi, " "công việc," v.v.) được in đậm.



Hình 2.14 Sau khi nhấp vào nút "Nhập tập dữ liệu" trong RStudio, một menu thả xuống sẽ xuất hiện, từ đó chúng tôi chọn "Từ thử nghiệm (cơ sở)."

Machine Translated by Google



Hình 2.15 Cửa sổ “Nhập tập dữ liệu” trong R cho tập dữ liệu bank_marketing_training , với “Tiêu đề: Có” được chọn.

Phần in đậm cho biết rằng chúng sẽ được nhập dưới dạng tên biến chứ không phải dưới dạng bản ghi thực tế trong tập dữ liệu của chúng tôi. Trong cửa sổ này, bạn cũng có thể thay đổi dấu phân cách, mã giá trị bị thiếu và các tùy chọn khác để nhập tập dữ liệu. Cuối cùng, để nhập tập dữ liệu, hãy nhấp vào “Nhập”. Một tab mới sẽ mở ở cửa sổ trên cùng bên trái hiển thị tập dữ liệu. Quay lại tab chứa mã của bạn để tiếp tục làm việc.

Mặc dù bạn có thể thay đổi tên của tập dữ liệu khi nó sẽ được nhập, bằng cách sử dụng trường “Tên”, chúng tôi khuyên bạn nên để nó làm giá trị mặc định. Trong chính mã R, bạn có thể rút ngắn tên bằng cách lưu tập dữ liệu dưới dạng tên ngắn hơn. Hướng dẫn trong Chương 4 nêu rõ “Đọc tập dữ liệu bank_marketing_training dưới dạng bank_train.”
Sau khi nhập tập dữ liệu, bạn có thể lưu tập dữ liệu bằng cách sử dụng tên rút gọn bằng cách sử dụng mũi tên trỏ sang trái, như minh họa bên dưới.

```
bank_train <- bank_marketing_training
```

Lưu ý rằng các dấu trừ và nhỏ hơn hiển thị bên dưới tạo thành một mũi tên chỉ sang trái, <-. Mũi tên này trỏ từ đối tượng được đổi tên, ở bên phải, sang tên mới của nó, ở bên trái. Không thể có khoảng cách giữa các dấu nhỏ hơn và dấu trừ, mặc dù có thể có khoảng cách ở hai bên của mũi tên chỉ bên trái mà chúng tạo thành. Hình thức chung để đổi tên một tập dữ liệu là:

Machine Translated by Google


```
object_name <- object_to_be_saved
```

Nếu bạn đã rất quen thuộc với việc viết mã và truy cập các thuộc tính của tệp của mình, bạn có thể mở tập dữ liệu bằng cách mã hóa đường dẫn tệp vào R thông qua cấu trúc sau:

```
your_name_for_the_data <- read.csv(file = "đường dẫn đến tệp")
```

Sử dụng phương pháp này, chúng tôi sẽ sử dụng lệnh read.csv() để nhập dữ liệu. Tệp = đầu vào trong lệnh read.csv() cho biết đường dẫn đến tệp bên trong dấu ngoặc kép.

Dưới đây là một ví dụ về mã để mở tập dữ liệu bank_marketing_training .

```
bank_train <- read.csv(file = "C:/Users/Data Science/
Data/bank_marketing_training")
```

2.4.5 Tiết kiệm đầu ra trong R

Như trong Python, bạn sẽ thường muốn lưu đầu ra của mã dưới một tên cụ thể để sử dụng sau này trong chương trình. Để lưu đầu ra của bạn dưới một tên cụ thể, hãy sử dụng cùng cấu trúc mà chúng tôi đã sử dụng để đổi tên tập dữ liệu:

```
object_name <- object_to_be_saved
```

Ví dụ: để tạo một bảng dự phòng, bạn sẽ sử dụng mã được cung cấp dưới. Mã và đầu ra của nó được hiển thị trong Hình 2.16.

```
bảng(bank_train$response, bank_train$previous_outcome)
```

Nếu bạn muốn tạo tổng số hàng và cột hoặc tính toán tỷ lệ của bảng này, trước tiên bạn nên lưu bảng dưới tên riêng của mình. Để đơn giản, chúng ta hãy lưu bảng dưới dạng t1 cho "Bảng #1."

Mã được đưa ra dưới đây và được hiển thị trong Hình 2.17. Lưu ý cách bảng dự phòng không được hiển thị nếu chúng tôi lưu nó dưới tên t1. Để xem bảng, hãy tự chạy mã t1 . Kết quả chỉ chạy t1 được thể hiện trong Hình 2.18.

```
> table(bank_train$response, bank_train$previous_outcome)

      failure nonexistent success
  no     2390       21176     320
  yes     385        2034     569
```

Hình 2.16 Tạo bảng dự phòng trong R mà không lưu đầu ra.

Machine Translated by Google


```
> t1 <- table(bank_train$response, bank_train$previous_outcome)
>
```

Hình 2.17 Tạo bảng dự phòng trong R và lưu đầu ra với tên t1.

```
> t1

      failure nonexistent success
  no     2390       21176     320
  yes     385        2034     569
```

Hình 2.18 Đầu ra của bảng đã lưu t1 trong R.

2.4.6 Truy cập Bản ghi và Biến trong R

Như với Python, sẽ có lúc bạn muốn R cung cấp cho bạn một bản ghi hoặc biến cụ thể. Ví dụ: làm cách nào để chúng tôi truy cập bản ghi đầu tiên trong tập dữ liệu bank_train ? Hoặc biến tuổi trong tập dữ liệu đó?

R tham chiếu các bản ghi của nó bắt đầu từ một, vì vậy nếu chúng ta muốn xem bản ghi đầu tiên, chúng ta yêu cầu bản ghi 1. Ví dụ: để truy cập bản ghi đầu tiên trong tập dữ liệu bank_train , hãy sử dụng mã sau:

```
bank_train[1, ]
```

Lưu ý rằng không có lệnh riêng để sử dụng để tách riêng một bản ghi cụ thể. Thay vào đó, ký hiệu ngoặc được sử dụng. Cấu trúc của ký hiệu ngoặc là:

```
data_set_name[ hàng quan tâm                  ,  cột quan tâm ]
```

Lưu ý rằng không được có khoảng cách giữa tên tập dữ liệu và dấu ngoặc mở.

Chúng ta hãy xem xét một số ví dụ. Tập dữ liệu mà chúng ta sẽ làm việc có tên là bank_train. Vì chúng tôi quan tâm đến hàng đầu tiên, chúng tôi đặt 1 trong các hàng của khu vực quan tâm. Hiện tại, không có cột quan tâm, vì vậy chúng tôi để trống đầu vào đó.

```
ngân_hàng[ 1, ]
```

Nếu bạn muốn bản ghi thứ nhất, thứ ba và thứ tư, hãy sử dụng đoạn mã sau:

```
bank_train[c(1,3,4), ]
```

Chúng tôi chỉ định các hàng quan tâm bằng cách sử dụng đầu vào 1, 3, 4 . Đặt các số đó trong lệnh c() cho R biết rằng tất cả các số đó thuộc về nhau, vì vậy chúng đều được coi là các hàng quan tâm. Vì các cột quan tâm không được chỉ định nên tất cả các cột sẽ được trả về. Đầu ra (chỉ hiển thị bốn cột đầu tiên) được hiển thị trong Hình 2.19.

Machine Translated by Google


```
> bank_train[c(1,3,4), ]
  age         job marital   education
1  56   housemaid married    basic.4y
3  41 blue-collar married     unknown
4  25    services  single high.school
```

Hình 2.19 R cho chúng ta thấy nội dung của bản ghi thứ nhất, thứ ba và thứ tư trong ngân hàng_ tập dữ liệu đào tạo (bốn biến đầu tiên được hiển thị).

```
> bank_train[, c(1, 3)]
      age  marital
1      56  married
2      57  married
3      41  married
4      25   single
5      29   single
```

Hình 2.20 R cho chúng ta thấy nội dung của biến thứ nhất và biến thứ ba (năm bản ghi đầu tiên được hiển thị).

Còn các biến thì sao? Chúng ta có thể xác định các biến quan tâm theo cách tương tự như các bản ghi quan tâm: bằng cách sử dụng ký hiệu ngoặc và đặt số cột chúng ta muốn vào phần cột quan tâm . Ví dụ, biến age và marital là biến thứ nhất và thứ ba trong tập dữ liệu. Để truy cập các biến này, hãy đặt 1 và 3 vào vị trí thích hợp trong ngoặc đơn, như minh họa bên dưới.

```
bank_train[, c(1, 3)]
```

Kết quả được thể hiện trong Hình 2.20.

Tất nhiên, bạn có thể kết hợp các hàng quan tâm và cột quan tâm để chỉ định các hàng và cột cụ thể mà bạn quan tâm. Ví dụ: bạn có thể yêu cầu ba bản ghi đầu tiên của các biến số tuổi và hôn nhân . Chúng tôi để lại kết quả này như một bài tập.

Khi chúng tôi nhập tập dữ liệu, chúng được nhập dưới dạng khung dữ liệu. Các khung dữ liệu này có một thuộc tính rất hay: chúng ta có thể xác định các biến quan tâm bằng cách sử dụng ký hiệu đô la, $. Ví dụ: giả sử chúng tôi muốn biến tuổi từ tập dữ liệu bank_train . Chúng tôi viết tên của tập dữ liệu và tên của biến, được kết nối bằng ký hiệu đô la, như hình bên dưới.

```
bank_train$age
```

Xin chúc mừng, bây giờ bạn đã học được các hành động cốt lõi của lập trình trong R! Bạn đang trên con đường trở thành một lập trình viên R. Cũng giống như với Python, phần còn lại của cuốn sách này sẽ đề cập đến các lệnh, gói và đầu ra cụ thể hơn, đi kèm với các mẹo và giải thích về lập trình.

Bây giờ bạn đã có kiến thức cơ bản về hai ngôn ngữ lập trình của chúng tôi, bạn đã sẵn sàng làm Khoa học dữ liệu bằng Python và R!

Machine Translated by Google

NGƯỜI GIỚI THIỆU

K. Jarrod Millman và Michael Aivazis, Python dành cho các nhà khoa học và kỹ sư, Máy tính trong Khoa học & Kỹ thuật, 13, 9–12, 2011, doi:https://doi.org/10.1109/MCSE.2011.36.

Wes McKinney, Cấu trúc dữ liệu cho tính toán thống kê bằng Python. Trong Kỷ yếu của Python trong Hội nghị Khoa học lần thứ 9, trang 51–56, 2010.

Travis E. Oliphant, Hướng dẫn về NumPy, Trelgol Publishing, Spanish Fork, 2006.

Travis E. Oliphant, Python cho điện toán khoa học, Điện toán trong Khoa học & Kỹ thuật, 9, 10–20, 2007, doi:https://doi.org/10.1109/MCSE.2007.58.

R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Điện toán thống kê, Vienna, Áo, 2018.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Năm hành động của Python và mã R mà chúng ta thảo luận trong chương này là gì?
2. Bình luận dùng để làm gì? Đầu ra nào được tạo ra bởi một nhận xét? Cái gì đặc biệt nhân vật bắt đầu một bình luận?
3. Tại sao chúng tôi muốn nhập các gói?
4. Việc sử dụng mã "as" khi nhập các gói Python là gì?
5. Làm cách nào để chúng tôi lưu đầu ra do mã Python tạo ra?
6. Làm cách nào để lưu đầu ra do mã R tạo ra?
7. Tại sao chúng ta muốn lưu đầu ra?
8. Làm cách nào để đưa một tập dữ liệu vào Python?
9. Tại sao việc xác định tập dữ liệu của chúng ta có tiêu đề cột hay không lại quan trọng?
10. Hai cách chúng ta có thể đưa tập dữ liệu vào R là gì?

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu bank_marketing_training . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Tải xuống chương trình và mở trình biên dịch. Những gì được chứa ở dưới cùng bên phải cửa sổ? Bên trái (đối với Python) hoặc trên cùng bên trái (đối với R)?
12. Nhập nhận xét cho biết bạn đang làm bài tập Chương 2.
13. Xác định vị trí nút "Run" và lưu ý xem có phím tắt không.
14. Thực hiện nhận xét từ bài tập trước. Đầu ra là gì? giải thích của bạn trả lời.
15. Nhập các gói sau:

Machine Translated by Google

Một. Đối với Python, hãy nhập các gói pandas và numpy . Đổi tên gói pandas "pd" và đổi tên gói numpy thành "np."

b. Đối với R, hãy nhập gói ggplot2 . Hãy chắc chắn rằng bạn vừa cài đặt vừa mở gói.

16. Nhập tập dữ liệu bank_marketing_training và đặt tên là bank_train.

17. Tạo một bảng dự phòng gồm các biến response và previous_outcome từ tập dữ liệu bank_train . Không lưu đầu ra từ mã.

18. Chạy lại mã từ bài tập trước, lần này lưu kết quả dưới dạng crosstab_01 (đối với mã Python) hoặc t1 (đối với mã R).

19. Sau khi lưu đầu ra trong bài tập trước, hãy hiển thị đầu ra bằng tên của đầu ra đã lưu.

20. Lưu bảng dự phòng dưới một tên khác. Lần này, hãy sử dụng họ của bạn và số yêu thích như tên; ví dụ larose42.

21. Lưu chín bản ghi đầu tiên của tập dữ liệu bank_train làm khung dữ liệu của riêng chúng.

22. Lưu các bản ghi tuổi và hôn nhân của tập dữ liệu bank_train làm khung dữ liệu của riêng chúng.

23. Lưu ba bản ghi đầu tiên về các biến số tuổi và hôn nhân làm khung dữ liệu của riêng chúng.

PHÂN TÍCH THỰC TẾ

24. Nhập tập dữ liệu adult_ch3_training bằng cách sử dụng cài đặt "Heading: Yes". Đổi tên tập dữ liệu thành người lớn sau khi được nhập.

25. Viết nhận xét giải thích sự thay đổi tên tập dữ liệu.

26. Nhập các gói sau:

    Một. Đối với Python, hãy nhập lệnh DecisionTreeClassifier từ gói sklearn.tree .

    b. Đối với R, hãy nhập gói rpart . Hãy chắc chắn rằng bạn vừa cài đặt vừa mở gói.

27. Tạo một bảng dự phòng về tầng lớp lao động và giới tính. Lưu đầu ra dưới dạng table01.

28. Tạo một bảng dự phòng về giới tính và tình trạng hôn nhân. Lưu đầu ra dưới dạng bảng02.

29. Hiển thị giá trị giới tính và tầng lớp lao động của người trong bản ghi đầu tiên. Chúng thuộc ô nào của bảng01 ? Có bao nhiêu bản ghi khác trong tập dữ liệu có cùng giới tính và các giá trị của tầng lớp lao động ?

30. Hiển thị giá trị giới tính và tình trạng hôn nhân của những người trong bản ghi 6–10. Chúng thuộc những ô nào của bảng02 ? Có bao nhiêu bản ghi khác trong tập dữ liệu có cùng sự kết hợp của các giá trị tình trạng hôn nhân và giới tính ?

31. Tạo tập dữ liệu mới chỉ có các bản ghi có tình trạng hôn nhân là "Đã kết hôn-vợ/chồng" và đặt tên cho tập dữ liệu là người lớnĐã kết hôn.

32. Tạo lại bảng dự phòng về giới tính và tầng lớp lao động bằng cách sử dụng tập dữ liệu người lớnMarried . Bạn nhận thấy sự khác biệt nào giữa hai giới?

33. Tạo tập dữ liệu mới chỉ có các bản ghi có giá trị tuổi lớn hơn 40. Đặt tên cho bộ dữ liệu mới dành cho người lớnOver40.

34. Tạo lại bảng dự phòng về giới tính và tình trạng hôn nhân bằng cách sử dụng bộ dữ liệu dành cho người lớnOver40 . Bạn nhận thấy sự khác biệt nào?

Machine Translated by Google

chương 3

CHUẨN BỊ DỮ LIỆU

3.1 BỘ DỮ LIỆU MARKETING NGÂN HÀNG

Chúng tôi sẽ minh họa cách thực hiện hai giai đoạn đầu tiên của Phương pháp khoa học dữ liệu bằng cách sử dụng bộ dữ liệu bank_marketing_training và bank_marketing_test . Độc giả có thể tải các bộ dữ liệu này từ trang web của bộ sách: www.
dataminingconsultant.com. Các bộ dữ liệu này được điều chỉnh từ ngân hàng-bổ sung-
tập dữ liệu full.txt1 từ Kho lưu trữ máy học UCI.2 Chúng tôi chỉ sử dụng bốn yếu tố dự đoán (tuổi, trình độ học vấn, kết_quả trước và ngày_kể từ_trước), cùng với mục tiêu, phản hồi. Dữ liệu liên quan đến một chiến dịch tiếp thị trực tiếp dựa trên điện thoại được thực hiện bởi một ngân hàng ở Bồ Đào Nha. Ngân hàng quan tâm đến việc liệu những người liên hệ có đăng ký tài khoản tiền gửi có kỳ hạn với ngân hàng hay không. ngân hàng_
tập dữ liệu marketing_training chứa 26.874 bản ghi, trong khi bank_marketing_test
chứa 10.255 bản ghi.

3.2 GIAI ĐOẠN TÌM HIỂU VẤN ĐỀ

Chúng ta bắt đầu với Giai đoạn Tìm hiểu Vấn đề, để đảm bảo rằng chiếc thang mà chúng ta đang dày công leo lên không dựa nhầm vào bức tường.

3.2.1 Trình bày rõ ràng các mục tiêu của dự án

Mục tiêu của phân tích này như sau:

1. Tìm hiểu về khách hàng tiềm năng của chúng tôi. Đó là, tìm hiểu các đặc điểm của những người chọn ngân hàng với chúng tôi, cũng như những người không.

1Sérgio Moro, Paulo Cortez và Paulo Rita, Cách tiếp cận dựa trên dữ liệu để dự đoán sự thành công của Tiếp thị qua điện thoại của ngân hàng, Hệ thống hỗ trợ quyết định, Elsevier, 62,:22–31, tháng 6 năm 2014.

2Kho lưu trữ học máy của Đại học California tại Irvine, https://archive.ics.uci.edu/ml/
index.php.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

2. Phát triển một phương pháp hữu ích để xác định những người có khả năng phản hồi tích cực, để chúng ta có thể tiết kiệm thời gian và tiền bạc. Đó là, phát triển một hoặc nhiều mô hình sẽ xác định những người có khả năng phản hồi tích cực. Định lượng lợi nhuận dự kiến từ việc sử dụng các mô hình này.

3.2.2 Chuyển những Mục tiêu này thành Bài toán Khoa học Dữ liệu

Chúng ta sẽ sử dụng khoa học dữ liệu như thế nào để hoàn thành các mục tiêu của dự án?

1. Có nhiều cách để tìm hiểu về khách hàng tiềm năng của chúng ta.

   Một. Sử dụng Phân tích dữ liệu khám phá để thể hiện một số mối quan hệ đồ họa đơn giản giữa các biến. Ví dụ: sử dụng biểu đồ độ tuổi phủ thông tin về câu trả lời có/không để hình dung liệu độ tuổi có ảnh hưởng đến phản hồi của khách hàng hay không.

   b. Sử dụng Phân cụm để xác định xem có các nhóm tự nhiên trong khách hàng tiềm năng của chúng tôi hay không, ví dụ: trẻ hơn/có trình độ học vấn cao hơn so với lớn tuổi hơn/ ít được giáo dục. Sau đó, xem liệu các cụm này có khác nhau về phản ứng của chúng đối với hoạt động tiếp thị hay không.

   c. Sử dụng Quy tắc kết hợp để xem liệu có mối quan hệ hữu ích nào giữa các tập hợp con của bản ghi hay không. Ví dụ: giả sử quy tắc, "Nếu là điện thoại di động, thì phản hồi = có" được hỗ trợ tốt và có độ tin cậy cao. Điều này sẽ cho phép những người tiếp thị của chúng tôi phát triển một chiến dịch được nhắm mục tiêu đến người dùng điện thoại di động, không phụ thuộc vào kết quả của mô hình tổng thể của chúng tôi.

2. Chúng tôi có thể phát triển một bộ mô hình khoa học dữ liệu mạnh mẽ để xác định những người có khả năng phản hồi tích cực. Lưu ý rằng, vì câu trả lời (có/không) là phân loại, nên chúng ta có thể sử dụng các mô hình phân loại chứ không phải các mô hình ước lượng.

   Một. Phát triển mô hình phân loại tốt nhất có thể, sử dụng cách sau thuật toán:

   Tôi. Cây quyết định

   thứ hai. rừng ngẫu nhiên

   iii. Phân loại Naïve Bayes

   v.v. Mạng thần kinh

   v. Hồi quy logistic

   b. Đánh giá từng mô hình dựa trên các tiêu chí đánh giá mô hình được xác định trước, chẳng hạn như chi phí phân loại sai. Xây dựng một bảng các mô hình tốt nhất và chi phí của chúng.

   c. Tham khảo ý kiến của ban quản lý về mô hình hoặc các mô hình tốt nhất mà để chuyển sang giai đoạn triển khai.

Do đó, chúng tôi đã (i) trình bày rõ ràng các mục tiêu của mình và (ii) chuyển các mục tiêu này thành một tập hợp các nhiệm vụ khoa học dữ liệu sẽ được triển khai. Như vậy, Giai đoạn Một: Giai đoạn Tìm hiểu Vấn đề đã hoàn tất.

Machine Translated by Google

3.3 GIAI ĐOẠN CHUẨN BỊ DỮ LIỆU

Tiếp theo, chúng tôi chuyển sang Giai đoạn chuẩn bị dữ liệu, trong đó dữ liệu được làm sạch và chuẩn bị để phân tích. Hướng dẫn đầy đủ về chuẩn bị dữ liệu sẽ cần nhiều dung lượng hơn chúng ta có ở đây. (Người đọc được khuyến khích xem Khai thác dữ liệu và Phân tích dự đoán3 để biết thêm về cách chuẩn bị dữ liệu.) Mỗi bộ dữ liệu đều có các nhiệm vụ chuẩn bị dữ liệu cần thiết của riêng nó. Trong chương này, chúng ta sẽ tập trung vào các nhiệm vụ chuẩn bị dữ liệu sau:

- Thêm trường chỉ mục
- Thay đổi giá trị trường sai lệch
- Biểu thị lại dữ liệu phân loại dưới dạng dữ liệu số
- Chuẩn hóa các trường số
- Xác định các giá trị ngoại lệ.

3.4 THÊM MỘT TRƯỜNG CHỈ SỐ

Nhà khoa học dữ liệu có thể muốn tăng cường tập dữ liệu bằng các biến mới có thể nâng cao hiểu biết. Ví dụ: không phải tất cả các bộ dữ liệu, bao gồm cả bank_marketing bộ dữ liệu, được trang bị trường ID. Do đó, chúng ta có thể thêm một trường chỉ mục vào dữ liệu, trường này sẽ phục vụ hai mục đích: (i) nó hoạt động như một trường ID cho các tập dữ liệu không có trường như vậy và (ii) nó theo dõi thứ tự sắp xếp của các bản ghi trong cơ sở dữ liệu. Trong khoa học dữ liệu, chúng tôi thường phân vùng lại và sắp xếp lại dữ liệu; do đó, rất hữu ích khi có trường chỉ mục để khôi phục thứ tự sắp xếp ban đầu khi muốn. Sau đây là cách thêm trường chỉ mục bằng Python và R.

3.4.1 Cách thêm trường chỉ mục bằng Python

Đầu tiên, chúng ta cần mở gói cần thiết, sử dụng mã đã thảo luận trong chương trước.

```
nhập gấu trúc dưới dạng pd
```

Tiếp theo, nhập tập dữ liệu dưới tên bank_train bằng cách sử dụng hàm read_csv() lệnh và chỉ định vị trí của tệp.

```
bank_train = pd.read_csv("C:/.../bank_marketing_training")
```

Như đã thảo luận trong chương trước, vì lệnh read_csv nằm trong gói pandas nên chúng ta cần đặt tên của gói trước lệnh. Khi chúng tôi mở gói pandas dưới dạng pd, lệnh đầy đủ là pd.read_csv().

3Tác giả Daniel T. Larose và Chantal D. Larose, John Wiley and Sons, Inc., 2015.

Machine Translated by Google

Để tạo chỉ mục, trước tiên chúng ta cần tìm số lượng bản ghi và cột trong tập dữ liệu.

```
bank_train.shape
```

Sử dụng .shape sau tên của tập dữ liệu sẽ cho chúng ta biết số lượng hàng và cột trong tập dữ liệu. Số đầu tiên trong đầu ra là số bản ghi, 26.874. Thứ hai là số lượng biến.

Khi chúng tôi biết số lượng bản ghi, chúng tôi tạo một biến mới gán cho mỗi bản ghi một số nguyên duy nhất.

```
bank_train['index'] = pd.Series(range(0,26874))
```

Các lệnh lồng nhau Series() và range() tạo ra một chuỗi các số có giới hạn dưới bằng 0 và giới hạn trên là số bản ghi. Vì lệnh Series() được chứa trong gói pandas và chúng tôi đã đổi tên gói pandas là pd, nên chúng tôi bắt đầu lệnh Series() bằng pd và dấu chấm. Kết quả là mã pd.Series(). Lưu ý rằng giới hạn dưới của lệnh range() là 0 chứ không phải 1, vì Python bắt đầu đếm các vị trí từ 0. Chúng tôi lưu chuỗi số dưới dạng một biến mới trong tập dữ liệu, chỉ mục, bằng cách gán đầu ra của pd.Series(range()) cho biến chỉ mục của tập dữ liệu bank_train bằng cách sử dụng bank_train['index'].

Để xem tập dữ liệu với biến mới của nó, chúng ta có thể xem phần đầu của tập dữ liệu.

```
bank_train.head
```

Sử dụng .head sau tên của tập dữ liệu sẽ tạo đầu ra có chứa 30 bản ghi đầu tiên và cuối cùng cho mỗi biến trong tập dữ liệu.

3.4.2 Cách thêm trường chỉ mục bằng R

Nhập tập dữ liệu dưới tên bank_train, sử dụng nút Nhập tập dữ liệu trong RStudio. Để tạo một trường chỉ mục, trước tiên chúng ta cần biết có bao nhiêu bản ghi trong tập dữ liệu.

```
n <- dim(bank_train)[1]
```

Lệnh dim() cung cấp số lượng bản ghi và số lượng biến cho tập dữ liệu có tên được sử dụng làm giá trị đầu vào; trong trường hợp này, bank_train bộ dữ liệu. Việc thêm [1] vào cuối dim(bank_train) sẽ chỉ dẫn đến số đầu tiên, số lượng bản ghi, được cung cấp dưới dạng đầu ra. Chúng tôi lưu đầu ra dưới dạng chữ cái viết thường “n”, đây là ký hiệu thường được sử dụng cho kích thước mẫu. Nếu bạn tự thực hiện n , đầu ra sẽ là số 26.874, là số lượng bản ghi trong tập dữ liệu.

Machine Translated by Google

Khi chúng tôi có số lượng bản ghi, chúng tôi tạo một biến mới cung cấp mỗi bản ghi một số nguyên duy nhất xác định thứ tự của nó trong tập dữ liệu.

```
bank_train$Index <- c(1:n)
```

Hàm c() sẽ kết hợp các giá trị đầu vào của nó thành một đối tượng duy nhất. Đầu vào trong trường hợp của chúng tôi là 1:n, viết tắt của "bao gồm các số nguyên từ 1 đến n". Lệnh c(1:n) sẽ cung cấp cho chúng ta một dãy số từ 1 đến số bản ghi trong tập dữ liệu ngân hàng . Chúng tôi lưu chuỗi số này dưới dạng một biến mới trong tập dữ liệu, có tên là Chỉ mục, bằng cách lưu c(1:n) dưới dạng bank_train$Index.

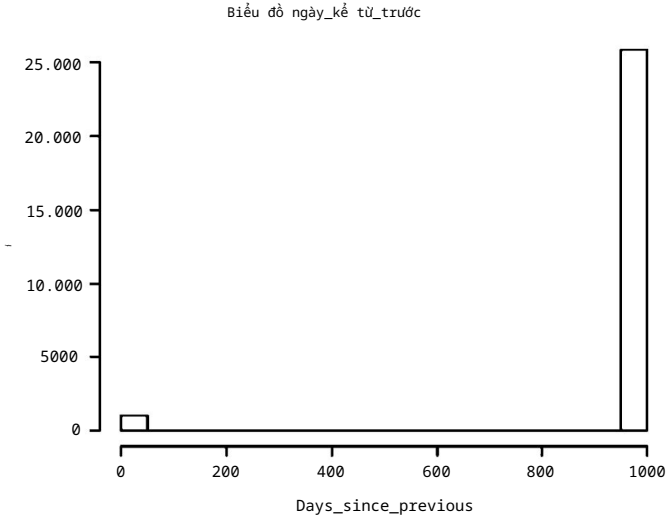
Để xem tập dữ liệu với biến chỉ mục mới của nó, hãy chạy lệnh head() với tập dữ liệu bank_train làm đầu vào duy nhất.

```
đầu (bank_train)
```

Đầu ra kết quả là sáu bản ghi đầu tiên trên tất cả các biến, bao gồm cả Biến chỉ mục.

3.5 THAY ĐỔI GIÁ TRỊ TRƯỜNG LỪA ĐẢO

Trường days_since_previous là tổng số ngày kể từ khi khách hàng được liên hệ lần cuối từ một chiến dịch trước đó. Trường này rõ ràng là số, vì vậy chúng ta có thể xem biểu đồ tần số4 của days_since_previous do R cung cấp trong Hình 3.1. Lưu ý rằng



Hình 3.1 Biểu đồ từ R của ngày_kể từ_ trước đó, với hầu hết các giá trị gần 1000.

4Xem Phụ lục, Tóm tắt và Trực quan hóa Dữ liệu.

Machine Translated by Google

hầu hết các giá trị dữ liệu đều gần 1000, với một số ít giá trị gần bằng không. Hóa ra quản trị viên cơ sở dữ liệu đã sử dụng mã 999 để đại diện cho những khách hàng chưa được liên hệ trước đó. Vì vậy, chúng ta cần thay đổi giá trị trường 999 thành thiếu, điều này được thực hiện như sau trong Python và R.

3.5.1 Cách thay đổi giá trị trường gây hiểu lầm bằng Python

Nếu bạn không mở gói gấu trúc hoặc đọc trong tập dữ liệu, như được mô tả trong phần Python trước, hãy làm như vậy ngay bây giờ. Chúng ta cũng cần nhập gói numpy cho phần này.

```
nhập numpy dưới dạng np
```

Chúng ta cần xác định tất cả các bản ghi có giá trị days_since_previous là 999 và thay thế chúng bằng mã Python cho các số bị thiếu, NaN. Sau khi thực hiện thay thế, chúng tôi sẽ lưu biến dưới tên biến days_since_previous , ghi đè lên các giá trị của biến trước đó một cách hiệu quả.

```
bank_train['days_since_previous'] =
       bank_train['days_since_previous'].replace({999: np.NaN})
```

Mã bank_train['days_since_previous'] truy cập biến days_
kể từ_trước. Lệnh replace({999: np.NaN }) tìm từng trường hợp của 999 trong biến days_since_previous và thay thế nó bằng giá trị NaN. Để lưu biến mới được chỉnh sửa dưới tên gốc của nó, chúng tôi đặt phía bên phải bằng với biến days_since_previous ban đầu ở bên trái bằng cách sử dụng lại tên biến bank_train['days_since_previous'].

Để tạo một biểu đồ của biến, hãy sử dụng lệnh hist() .

```
bank_train['days_since_previous'].plot(kind = 'hist',
         title = 'Histogram of Days Kể từ Trước')
```

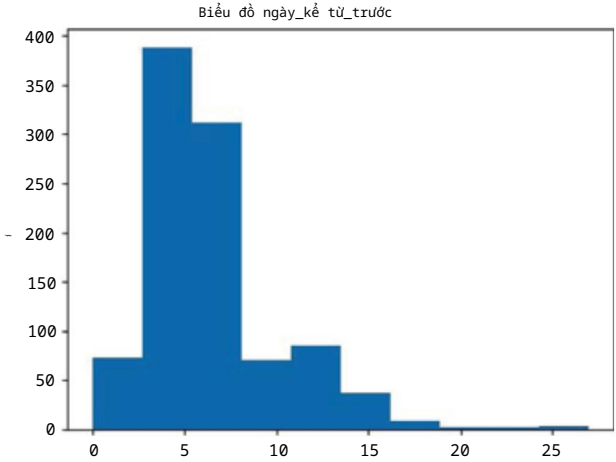
Sử dụng .plot() sau tên biến sẽ tạo thành biểu đồ của biến. Chúng tôi sử dụng kind = 'hist' để chỉ định rằng một biểu đồ sẽ được tạo. Đầu vào tiêu đề , chứa trong dấu ngoặc đơn, tạo tiêu đề của biểu đồ. Đầu ra được hiển thị trong Hình 3.2.
Trong Chương 4, chúng ta sẽ xem xét các biểu đồ phức tạp hơn.

3.5.2 Cách thay đổi giá trị trường gây hiểu lầm bằng R

Nếu bạn chưa đọc trong tập dữ liệu, như được mô tả trong phần R trước đó, hãy làm như vậy ngay bây giờ.

Chúng ta cần xác định từng phiên bản của 999 trong biến days_since_previous và thay thế nó bằng giá trị mã R cho giá trị bị thiếu, NA.

Machine Translated by Google



Hình 3.2 Biểu đồ của days_since_previous trong Python với các giá trị bị thiếu được mã hóa chính xác.

```
bank_train$days_since_previous <- ifelse(test = bank_
đào tạo$days_since_previous == 999,
       có = NA, không = bank_train$days_since_previous)
```

Lệnh ifelse() kiểm tra điều kiện được chỉ định trong test =, sau đó gán giá trị cho biến days_since_previous sau yes = nếu điều kiện kiểm tra là đúng và gán giá trị cho biến days_since_previous sau no = nếu điều kiện kiểm tra là sai.

Trong trường hợp của chúng tôi, mỗi bản ghi được kiểm tra xem liệu nó có chứa giá trị 999 trong biến days_since_previous hay không . Nếu có, giá trị NA được trả về. Nếu không, giá trị ban đầu trong biến days_since_previous được trả về. Để lưu chuỗi giá trị được trả về dưới dạng biến days_since_previous, hãy lưu đầu ra dưới dạng biến bằng cách sử dụng bank_train$days_since_previous.

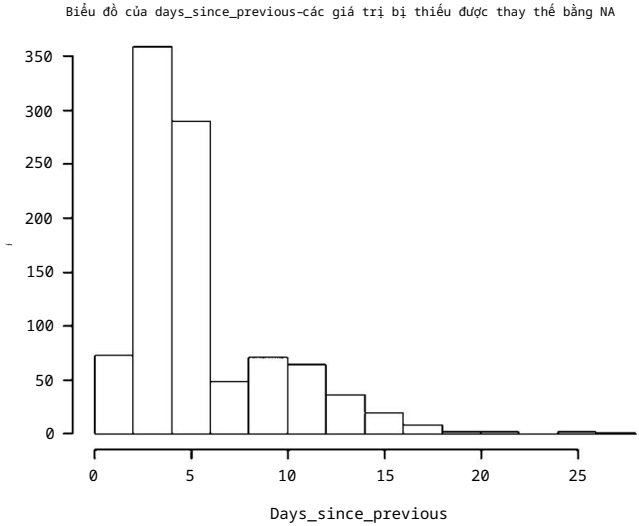
Để tạo một biểu đồ của biến, hãy sử dụng lệnh hist() .

```
lịch sử (bank_train$days_since_previous, xlab = "days_since_
trước",
          main = "Biểu đồ của days_since_previous - Thiếu
Các giá trị được thay thế bằng NA")
```

Lệnh hist() có một đầu vào bắt buộc, biến quan tâm. Chúng tôi sử dụng bank_train$days_since_previous làm biến quan tâm. Các giá trị đầu vào tùy chọn main và xlab chỉ định tiêu đề và nhãn trục X của biểu đồ tương ứng. Lưu ý rằng các nhãn phải được chứa trong dấu ngoặc kép.

Hình 3.3 hiển thị biểu đồ của days_since_previous bị thiếu các giá trị bị loại trừ.

Machine Translated by Google



Hình 3.3 Biểu đồ trong R của days_since_previous, loại trừ các giá trị bị thiếu.

3.6 THỂ HIỆN LẠI DỮ LIỆU PHÂN LOẠI NHƯ SỐ

Hình 3.4 cho thấy biểu đồ thanh5 của lĩnh vực giáo dục . Lưu ý rằng trường là phân loại, nghĩa là không có thứ tự của các giá trị trường. Nói cách khác, nếu chúng tôi để nguyên trường này, thì các thuật toán khoa học dữ liệu của chúng tôi sẽ không biết trường đại học đó_ bằng đại diện cho giáo dục nhiều hơn cơ bản.4yr. Để cung cấp thông tin này cho các thuật toán của chúng tôi, chúng tôi chuyển đổi các giá trị dữ liệu thành các giá trị số, trong đó rõ ràng là một giá trị lớn hơn giá trị khác. Người ta cần tiến hành cẩn thận khi làm điều này, để sự khác biệt tương đối giữa các loại khác nhau được bảo tồn.

Bảng 3.1 cho thấy cách chúng tôi dự định thực hiện chuyển đổi này. Giá trị 12 cho khóa học chuyên nghiệp được lấy từ ấn phẩm được trình bày ở phần chú thích, đại diện cho một lựa chọn thay thế cho khóa học thông thường ở trường trung học. Tất nhiên, các giá trị chưa biết cũng sẽ cần được biểu thị lại là bị thiếu.

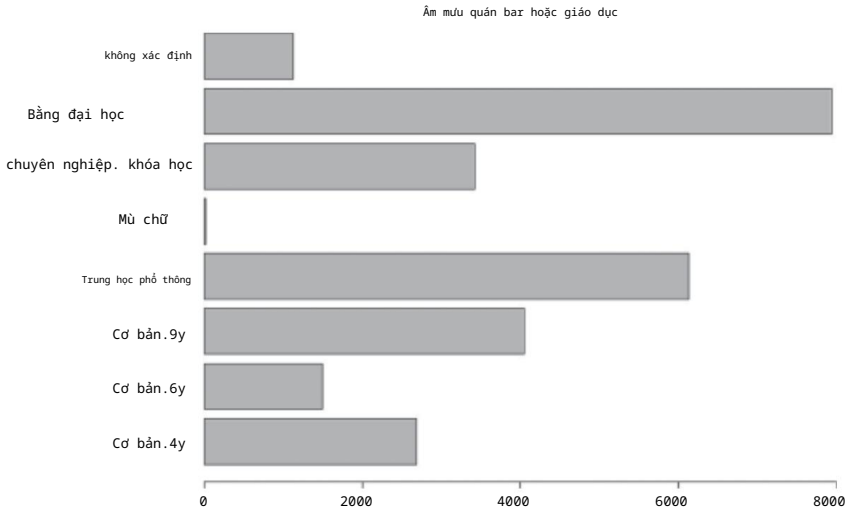
3.6.1 Cách thể hiện lại các giá trị trường phân loại bằng Python

Chúng tôi sẽ sao chép biến giáo dục và đặt tên là education_numeric, để chuẩn bị thay thế các giá trị phân loại của nó bằng các giá trị số.

```
bank_train['education_numeric'] = bank_
xe lửa['giáo dục']
```

5Xem Phụ lục, Tóm tắt và Trực quan hóa Dữ liệu.

Machine Translated by Google



Hình 3.4 Đồ thị thanh trong R của biến giáo dục .

BẢNG 3.1 Thể hiện lại các giá trị của giáo dục dưới dạng số

Giá trị phân loại	Giá trị số
mù chữ	0
cơ bản.4y	4
cơ bản.6y	6
cơ bản.9y	9
trung	12
học.trung cấp chuyên	12a
nghiệp.khóa đại học.bằng cấp	16
không xác định	Mất tích

a Giá trị này dựa trên In-Vet, Ngăn chặn tình trạng bỏ học ban đầu và thúc đẩy sự hòa nhập ban đầu, http://invet-project.eu/wp-content/uploads/2014/06/National-Report_Bồ Đào Nha_Final.pdf. Hai nhóm, khóa học chuyên nghiệp và trung học, sẽ được kết hợp với cùng một giá trị số, 12. Tỷ lệ người phản hồi tích cực giữa hai nhóm này là tương tự nhau (11,1 so với 10,7%), vì vậy chúng ta có thể chấp nhận việc chúng được kết hợp.

Vế bên phải của dấu bằng chỉ định biến giáo dục và dấu bằng chỉ định các giá trị đó cho vế trái. Hiện tại không có biến nào có tên là education_numeric, vì vậy, một biến sẽ được tạo và cung cấp các giá trị của biến education_numeric.

Chúng ta cần thiết lập một từ điển cụ thể để chuyển đổi các giá trị phân loại trong biến education_numeric thành các giá trị số. Từ điển được chứa trong dấu ngoặc nhọn, { }, để thiết lập từ điển của chúng tôi, như sau:

```
dict_edu = {"education_numeric": {"mù chữ": 0, "basic.4y": 4,
 "basic.6y": 6,
```

Machine Translated by Google


```
        "cơ bản.9y": 9, "trung học phổ thông":12, "chuyên nghiệp.
khóa học": 12, "đại học.bằng cấp": 16,
        "không xác định": np.NaN}}
```

Trong từ điển, chúng tôi sử dụng "education_numeric" để chỉ định biến mà chúng tôi muốn mã hóa lại, theo sau là dấu hai chấm và một bộ dấu ngoặc nhọn khác.
Trong bộ dấu ngoặc nhọn thứ hai này, chúng tôi chỉ định mã hóa lại theo thứ tự sau:

Giá trị ban đầu của biến : Giá trị mới của biến

Mỗi thông số kỹ thuật được phân tách bằng dấu phẩy. Lưu ý rằng chúng tôi sử dụng giá trị của Python cho các giá trị số bị thiếu, NaN, nếu cần.
Cuối cùng, chúng tôi yêu cầu Python sử dụng từ điển để thay thế các giá trị của biến.

```
bank_train.replace(dict_edu, inplace=True)
```

Lệnh replace() sẽ thay thế các giá trị theo quy tắc trong từ điển dict_edu.

3.6.2 Cách thể hiện lại các giá trị trường phân loại bằng R

Trước tiên, chúng ta cần cài đặt và tải gói plyr .

```
install.packages("plyr"); thư viện (plyr)
```

Chúng ta cần chỉ rõ giá trị nào của giáo dục đi với giá trị số nào,
theo các quy tắc được chỉ định trong Bảng 3.1.

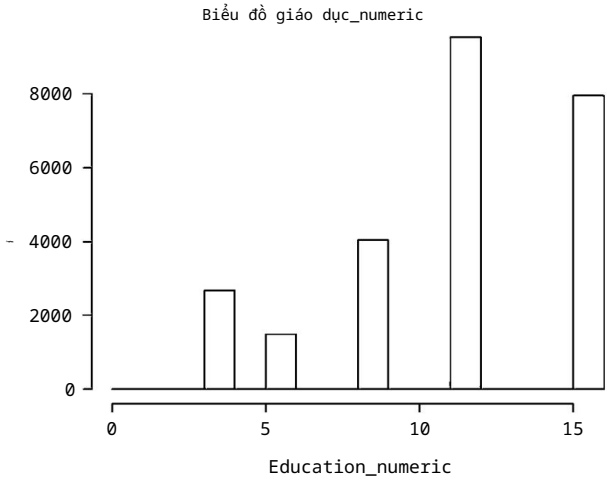
```
edu.num <- revalue(x = bank_train$education, replace = c("iliterate" = 0, "basic.4y"
= 4,
        "basic.6y" = 6, "basic.9y" = 9, "high.school" = 12,
 "chuyên nghiệp.khóa học" = 12,
        "đại học.bằng cấp" = 16, "không xác định" = NA))
```

Hàm revalue() thay thế các giá trị trong biến được cung cấp trong đầu vào x , theo các quy tắc được đưa ra trong đầu vào thay thế . Trong đầu vào thay thế = , chúng tôi sử dụng c() để xâu chuỗi từng phần mã hóa lại với nhau, sử dụng cấu trúc

Giá trị ban đầu của biến = Giá trị mới của biến

Mỗi thông số kỹ thuật được phân tách bằng dấu phẩy. Lưu ý rằng chúng tôi đang sử dụng giá trị của R đối với các giá trị bị thiếu, NA, nếu cần. Chúng tôi lưu đầu ra dưới dạng edu.num.
Hiện tại, các giá trị của edu.num không phải là số (ví dụ: bạn không thể tạo biểu đồ bằng cách sử dụng chúng), vì vậy chúng tôi cần chuyển đổi các mức của biến thành

Machine Translated by Google



Hình 3.5 Biểu đồ trong R của education_numeric.

kiểu số. Đối tượng edu.num là một thừa số và chúng ta chuyển đổi giá trị của nó thành số bằng mã sau:

```
bank_train$education_numeric <- as.numeric(levels (edu.num))[edu.num]
```

Lệnh levels() thu được các mức hệ số của biến edu.num , là các chuỗi. Lệnh as.numeric() chuyển đổi chúng thành số. Các giá trị mới này được áp dụng cho biến edu.num và kết quả được lưu dưới dạng biến mới của chúng tôi, education_numeric. Hình 3.5 cho thấy một biểu đồ của trường giáo dục được thể hiện lại education_numeric.

3.7 CHUẨN HÓA CÁC TRƯỜNG SỐ

Một số thuật toán hoạt động tốt hơn khi các trường số được chuẩn hóa sao cho giá trị trung bình của trường bằng 0 và độ lệch chuẩn của trường bằng 1,6 như sau:

$$z = \text{Giá trị chuẩn hóa} = \frac{x - \bar{x}}{s} = \frac{\text{giá trị dữ liệu có nghĩa là}}{\text{Độ lệch chuẩn}}$$

Giá trị z dương có thể được hiểu là biểu thị số độ lệch chuẩn trên giá trị trung bình của giá trị dữ liệu, trong khi giá trị z âm biểu thị số độ lệch chuẩn dưới giá trị trung bình. Một số nhà phân tích chuẩn hóa tất cả

6Để biết giá trị trung bình và độ lệch chuẩn, hãy xem Phụ lục, Tóm tắt và trực quan hóa dữ liệu. Để biết thêm về tiêu chuẩn hóa, hãy xem Khám phá số liệu thống kê, Phiên bản thứ ba, của Daniel T. Larose, WH Freeman, 2016.

Machine Translated by Google

các trường số như một vấn đề của khóa học. Tiếp theo, chúng tôi trình bày cách chuẩn hóa các trường số trong Python và R.

3.7.1 Cách chuẩn hóa trường số bằng Python

Nhập gói yêu cầu.

```
từ số liệu thống kê nhập khẩu scipy
```

Chúng ta sẽ chuẩn hóa biến age và lưu nó thành một biến mới, age_z.

```
bank_train['age_z'] = stats.zscore(bank_train['age'])
```

Hàm zscore tính toán giá trị z của biến đã cho, trong trường hợp này là tuổi, được viết bằng mã dưới dạng bank_train['age']. Vì lệnh zscore() là một phần của gói thống kê , chúng tôi viết lệnh dưới dạng stats.zscore(). Chúng tôi lưu nó dưới dạng một biến mới trong tập dữ liệu, age_z.

3.7.2 Cách chuẩn hóa trường số bằng R

Chúng tôi không phải cài đặt hoặc tải gói cho mã này. Lệnh chúng ta sẽ sử dụng, scale(), được bao gồm trong lần tải xuống đầu tiên của R.

```
bank_train$age_z <- quy mô(x = bank_train$age)
```

Hàm scale() có thể căn giữa một biến bằng cách trừ đi giá trị trung bình của biến đó, chia tỷ lệ biến đó bằng cách chia cho độ lệch chuẩn hoặc cả hai. Theo mặc định, nó thực hiện cả hai, khi cần thiết để tính điểm số z. Do đó, sử dụng cài đặt mặc định của hàm scale() trên biến bank_train$age sẽ trả về điểm z của biến. Chúng ta có thể lưu các giá trị z dưới dạng một biến mới trong tập dữ liệu bằng cách sử dụng bank_train$age_z.

3.8 XÁC ĐỊNH NGOẠI LỆ

Sau khi các trường số được chuẩn hóa, người ta có thể sử dụng các giá trị z để xác định các giá trị ngoại lệ, là các bản ghi có giá trị cực đoan dọc theo một hoặc nhiều thứ nguyên cụ thể. Ví dụ: xem xét trường number_of_contacts, đại diện gửi số lượng liên hệ khách hàng được thực hiện trong suốt chiến dịch tiếp thị. Số lượng người liên hệ trung bình trên mỗi khách hàng là 2,6, với độ lệch chuẩn là 2,7 (có thể làm tròn số). Vì vậy, chúng tôi có được trường chuẩn hóa như sau:

$$\text{số liên lạc z} \quad = \frac{\text{số lượng địa chỉ liên lạc} - 2.6}{2.7}$$

Machine Translated by Google

Một nguyên tắc chung là một giá trị dữ liệu là ngoại lệ nếu giá trị z của nó lớn hơn 3 hoặc nhỏ hơn 3. Chẳng hạn, một khách hàng đã được liên hệ 10 lần (có vẻ như rất nhiều) sẽ có giá trị tiêu chuẩn hóa,

số o _ _fdanh bạ _ z = 10 2 6 . / 2 .7 = 2 .7

Do đó, 10 liên hệ, trong khi rất nhiều, không được xác định là ngoại lệ khi sử dụng phương pháp này, vì 2,7<3.

Nhà khoa học dữ liệu nên tham khảo ý kiến của khách hàng về những gì họ muốn làm với bất kỳ ngoại lệ nào. Ngoại lệ không nên được tự động loại bỏ!
Chúng cũng không nên được tự động thay đổi. Các giá trị bất thường của chúng có thể làm sáng tỏ các khía cạnh quan trọng của dữ liệu nên được thảo luận với khách hàng hoặc với quản trị viên cơ sở dữ liệu.

3.8.1 Cách xác định ngoại lệ bằng Python

Đối với ví dụ này, chúng tôi sẽ tiếp tục sử dụng biến age_z mà chúng tôi đã tạo trong phần trước. Chúng ta sẽ tìm các giá trị ngoại lệ bằng cách sử dụng hàm query() , hàm này xác định các hàng đáp ứng một điều kiện cụ thể.

```
bank_train.query('age_z > 3 | age_z < -3')
```

Điều kiện mà chúng tôi muốn tất cả các bản ghi được trả về đáp ứng được đưa ra là 'age_z > 3 | age_z < -3'. Nói cách khác, điều kiện này yêu cầu mỗi bản ghi phải có age_z giá trị lớn hơn 3 hoặc giá trị age_z nhỏ hơn 3. "hoặc" được chỉ định bởi ký tự | giữa hai điều kiện.

Tất cả các bản ghi đáp ứng điều kiện đã chỉ định đều được trả về. Trong ví dụ của chúng tôi, có 228 bản ghi có giá trị age_z lớn hơn 3 hoặc nhỏ hơn 3. Chúng tôi có thể sử dụng các bản ghi này để tạo một tập dữ liệu mới, chỉ bao gồm các giá trị này.

```
bank_train_outliers = bank_train.query('age_z > 3 | age_z < -3')
```

Bằng cách đặt tên cho đầu ra của lệnh query() , chúng tôi tạo một tập dữ liệu mới chỉ bao gồm các ngoại lệ mà chúng tôi gọi là bank_train_outliers.

Hãy để chúng tôi sắp xếp tập dữ liệu bank_train_outliers theo biến age_z của nó .

```
bank_train_sort = bank_train.sort_values(['age_z'], tăng dần = Sai)
```

Lệnh sort_values() sẽ sắp xếp các bản ghi trong tập dữ liệu dựa trên một biến cụ thể. Sắp xếp có thể tăng dần hoặc giảm dần. Trong ví dụ này, chúng tôi muốn các giá trị age_z lớn nhất ở trên cùng, vì vậy chúng tôi sắp xếp theo thứ tự giảm dần theo

Machine Translated by Google

chỉ định tăng dần = Sai. Chúng ta có thể lưu tập dữ liệu đã sắp xếp này dưới tên riêng của nó, chẳng hạn như bank_train_sort.

Cuối cùng, giả sử chúng ta muốn báo cáo tuổi và tình trạng hôn nhân của 15 người có giá trị age_z lớn nhất . Điều kiện này chỉ định cả số hàng cần báo cáo (15) và số cột cần báo cáo (các biến có tên là tuổi và hôn nhân).

```
bank_train_sort[['tuổi', 'hôn nhân']].head(n=15)
```

Ký hiệu dấu ngoặc kép được cung cấp sau tên tập dữ liệu cho phép chúng tôi chỉ định cột nào sẽ bao gồm. Lệnh head () sẽ cung cấp các bản ghi trên cùng, dừng sau n bản ghi nếu n được cung cấp hoặc sau năm bản ghi nếu không có giá trị n nào được cung cấp. Trong trường hợp của chúng tôi, chúng tôi chỉ định n = 15. Kết quả là tuổi và tình trạng hôn nhân của những người có 15 giá trị age_z lớn nhất .

3.8.2 Cách xác định ngoại lệ bằng R

Đối với ví dụ này, chúng tôi sẽ tiếp tục sử dụng biến age_z mà chúng tôi đã tạo trong phần trước. Chúng ta có thể cô lập các bản ghi riêng lẻ bằng cách sử dụng ký hiệu ngoặc được nêu chi tiết trong chương trước. Cấu trúc của mã của chúng tôi sẽ bắt đầu như sau:

```
bank_train[ hàng quan tâm, ]
```

Lưu ý rằng phía bên phải của dấu phẩy được để trống. Vì chúng tôi chưa chỉ định bất kỳ cột quan tâm nào, tất cả các cột sẽ được trả về trong đầu ra của lệnh này.

Bây giờ chúng ta cần điền vào những hàng mà chúng ta quan tâm. Lệnh which() sẽ xác định các hồ sơ đáp ứng các điều kiện quy định.

```
bank_outliers <- bank_train[ which(bank_train$age_z < -3 | bank_train$age_z
 > 3), ]
```

Điều kiện được đưa ra làm đầu vào cho lệnh which() nói rằng chúng ta muốn tất cả các bản ghi có giá trị age_z nhỏ hơn 3 hoặc lớn hơn 3. Điều kiện which() lệnh trả về chỉ số hàng của tất cả các bản ghi như vậy. Ký hiệu ngoặc sẽ trả về một tập hợp con của tập dữ liệu bank_train chỉ chứa các bản ghi đó. Chúng tôi có thể lưu các bản ghi đó dưới dạng tập dữ liệu mới, bank_outliers.

Để sắp xếp một tập dữ liệu theo một biến, chúng ta sử dụng lệnh order() .

```
bank_train_sort <- bank_train[ đơn hàng(- bank_
xe lửa$age_z), ]
```

Lệnh order() lấy đầu vào là một biến cần sắp xếp. Nó trả về các chỉ số hàng của biến sau khi nó đã được sắp xếp. Theo mặc định, các giá trị được sắp xếp theo thứ tự tăng dần. Thay vào đó, để sắp xếp theo thứ tự giảm dần, chúng ta thêm dấu trừ trước biến, như được minh họa.

Machine Translated by Google

Đặt lệnh order() bên trong ký hiệu dấu ngoặc, cụ thể là vị trí của các hàng quan tâm, sẽ sắp xếp lại các bản ghi trong tập dữ liệu dựa trên thứ tự được trả về bởi lệnh order() . Chúng tôi lưu tập dữ liệu mới, được sắp xếp này dưới dạng bank_train_sort.

Để xem 10 bản ghi đầu tiên của tập dữ liệu mới, đó là 10 bản ghi có các giá trị age_z lớn nhất , hãy sử dụng ký hiệu ngoặc và chỉ định các hàng từ 1 đến 10.

```
bank_train_sort[ 1:10, ]
```

Một lần nữa, để trống các cột sẽ trả về tất cả các cột (biến) của tập dữ liệu.

Để chỉ trả về một số cột, chúng ta cần lưu ý xem biến nào ở cột nào. Chúng ta có thể làm điều này bằng cách sử dụng head().

```
đầu (bank_train_sort)
```

Biến age nằm trong cột 1 và biến marital nằm trong cột 3. Để trả về 10 bản ghi đầu tiên của cột 1 và 3, hãy chỉ định cả hàng và cột quan tâm trong ký hiệu ngoặc đơn.

```
bank_train_sort[1:10, c(1,3)]
```

Đầu ra chứa tuổi và tình trạng hôn nhân của 10 khách hàng có giá trị age_z lớn nhất .

Các chủ đề được đề cập trong chương này nhằm cung cấp hương vị của các loại thách thức đang chờ bạn trong Giai đoạn chuẩn bị dữ liệu. Trong các bài tập Thực hành Phân tích , chúng ta khám phá cách lấy các biến mới là hàm của các biến ban đầu, để trích xuất thêm thông tin từ tập dữ liệu. Để biết thêm về chuẩn bị dữ liệu, hãy xem Khai thác dữ liệu và Phân tích dự đoán, [7] cung cấp nhiều chủ đề chuẩn bị dữ liệu hơn, cùng với cách thực hiện chúng bằng R.

NGƯỜI GIỚI THIỆU

Gói scipy trong Python xử lý nhiều tác vụ toán học và tính toán khác nhau. Để biết thêm chi tiết, hãy xem trang web: E. Jones, E. Oliphant, và P. Peterson, et al., SciPy: Open Source Scientific Tools for Python, 2001-, www.scipy.org

Gói plyr trong R có mục tiêu tương tự là xử lý và sắp xếp dữ liệu. Để biết thêm chi tiết, hãy xem ấn bản gốc: Hadley Wickham, Chiến lược tách-áp dụng-kết hợp để phân tích dữ liệu, Tạp chí Phần mềm Thống kê, 40(1), 1-29, 2011.

7Khai thác dữ liệu và phân tích dự đoán, của Daniel T. Larose và Chantal D. Larose, Phiên bản thứ hai, John Wiley và Sons, Inc., 2015.

Machine Translated by Google

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Hai mục tiêu chính của phân tích bank_marketing là gì , như đã nêu trong Giai đoạn tìm hiểu vấn đề?
2. Ba cách chúng tôi dự định thực hiện mục tiêu tìm hiểu về khách hàng tiềm năng của mình là gì?
3. Giải thích cách chúng tôi dự định hoàn thành mục tiêu phát triển các mô hình có lợi để xác định những người có khả năng phản hồi tích cực.
4. Mô tả hai lý do tại sao nên thêm trường chỉ mục vào tập dữ liệu.
5. Giải thích tại sao trường days_since_previous về cơ bản là vô dụng cho đến khi chúng ta xử lý mã 999.
6. Tại sao việc thể hiện lại giáo dục dưới dạng trường số lại quan trọng ?
7. Giả sử một giá trị dữ liệu có giá trị z là 1. Chúng ta có thể diễn giải giá trị này như thế nào?
8. Quy tắc ngón tay cái sơ bộ để xác định các ngoại lệ sử dụng giá trị z là gì?
9. Có nên tự động loại bỏ hoặc thay đổi các ngoại lệ không? Tại sao hay tại sao không?
10. Chúng ta nên làm gì với những điểm khác biệt mà chúng ta đã xác định?

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu bank_marketing_training . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Lấy một trường chỉ mục và thêm nó vào tập dữ liệu.
12. Đối với trường days_since_previous , hãy thay đổi giá trị trường 999 thành mã thích hợp cho giá trị bị mất.
13. Đối với trường giáo dục , hãy biểu thị lại các giá trị của trường dưới dạng các giá trị số được hiển thị trong Bảng 3.1.
14. Chuẩn hóa tuổi đồng ruộng . In ra danh sách 10 bản ghi đầu tiên, bao gồm cả các biến tuổi và tuổi_z.
15. Lấy danh sách tất cả các bản ghi là ngoại lệ theo trường age_z. In ra danh sách 10 giá trị age_z lớn nhất .
16. Đối với trường công việc , gộp các công việc chiếm dưới 5% số bản ghi vào một trường gọi là khác.
17. Đổi tên bộ dự đoán mặc định thành credit_default.
18. Đối với biến tháng, hãy thay đổi giá trị trường thành 1–12, nhưng giữ nguyên biến là phân loại.
19. Thực hiện thao tác sau cho trường thời lượng .

    Một. Chuẩn hóa biến.

    b. Xác định có bao nhiêu ngoại lệ và xác định ngoại lệ cực đoan nhất.

Machine Translated by Google

20. Thực hiện thao tác sau cho trường chiến dịch .

    Một. Chuẩn hóa biến.

    b. Xác định có bao nhiêu ngoại lệ và xác định ngoại lệ cực đoan nhất.

PHÂN TÍCH THỰC TẾ

Đối với Bài tập 21–25, hãy làm việc với tập dữ liệu Nutrition_subset . Tập dữ liệu chứa trọng lượng tính bằng gam cùng với lượng chất béo bão hòa và lượng cholesterol của một tập hợp gồm 961 loại thực phẩm. Sử dụng Python hoặc R để giải quyết từng vấn đề.

21. Các phần tử trong tập dữ liệu là các mặt hàng thực phẩm với nhiều kích cỡ khác nhau, từ một thìa cà phê quế đến nguyên một chiếc bánh cà rốt.

    Một. Sắp xếp tập dữ liệu theo chất béo bão hòa (chất béo bão hòa) và tạo danh sách năm loại thực phẩm chứa nhiều chất béo bão hòa nhất.

    b. Nhận xét về giá trị của việc so sánh các mặt hàng thực phẩm có kích cỡ khác nhau.

22. Lấy một biến số mới, bão hòa_chất béo_per_gram, bằng cách chia lượng chất béo bão hòa cho trọng lượng tính bằng gam.

    Một. Sắp xếp bộ dữ liệu theo sat_fat_per_gram và tạo danh sách năm mặt hàng thực phẩm có hàm lượng chất béo bão hòa cao nhất trên mỗi gram.

    b. Thực phẩm nào có nhiều chất béo bão hòa nhất trên mỗi gam?

23. Lấy một biến mới, cholesterol_per_gram.

    Một. Sắp xếp tập dữ liệu theo cholesterol_per_gram và tạo danh sách năm mặt hàng thực phẩm có hàm lượng chất béo cholesterol cao nhất trên mỗi gam.

    b. Thực phẩm nào có nhiều chất béo cholesterol nhất trên mỗi gram?

24. Chuẩn hóa trường sat_fat_per_gram. Lập danh sách tất cả các mặt hàng thực phẩm ngoại lệ ở mức cao nhất của thang đo. Có bao nhiêu mặt hàng thực phẩm là ngoại lệ ở mức thấp nhất của quy mô?

25. Chuẩn hóa trường cholesterol_per_gram. Lập danh sách tất cả các mặt hàng thực phẩm ngoại lệ ở mức cao nhất của thang đo.

    Đối với Bài tập 26–30, hãy làm việc với tập dữ liệu adult_ch3_training . Câu trả lời là liệu thu nhập có vượt quá 50.000 đô la hay không.

26. Thêm trường chỉ mục bản ghi vào tập dữ liệu.

27. Xác định xem có tồn tại ngoại lệ nào đối với lĩnh vực giáo dục hay không .

28. Làm như sau cho trường tuổi .

    Một. Chuẩn hóa biến.

    b. Xác định có bao nhiêu ngoại lệ và xác định ngoại lệ cực đoan nhất.

29. Lấy một lá cờ cho lãi vốn, được gọi là cờ lãi vốn, bằng 0 cho lãi vốn bằng 0 và 1 nếu không.

30. Độ tuổi dị thường? Chỉ chọn các bản ghi có độ tuổi ít nhất là 80. Xây dựng biểu đồ độ tuổi. Giải thích những gì bạn thấy trong một câu và tại sao lại như vậy trong một câu khác.

Machine Translated by Google

chương 4

PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

4.1 EDA SO VỚI HT

Khách hàng hoặc nhà phân tích thường có giả thuyết tiên nghiệm rằng họ muốn kiểm tra dữ liệu. Một ví dụ về giả thuyết như vậy là: Người dùng điện thoại di động có tỷ lệ phản hồi tích cực cao hơn người dùng điện thoại cố định không? Kiểm tra giả thuyết kết quả (HT) có thể được thực hiện bằng phương pháp thống kê cổ điển hoặc sử dụng phương pháp xác thực chéo của khoa học dữ liệu (Chương 5).

Mặt khác, khách hàng hoặc nhà phân tích có thể không có bất kỳ khái niệm tiên nghiệm nổi bật nào về những gì dữ liệu có thể phát hiện ra. Trong những trường hợp như vậy, họ muốn sử dụng phân tích dữ liệu khám phá (EDA) hoặc phân tích dữ liệu đồ họa. EDA cho phép người dùng:

- Sử dụng đồ họa để khám phá mối quan hệ giữa các biến dự đoán và biến mục tiêu.
- Sử dụng đồ họa và bảng để rút ra các biến mới giúp tăng khả năng dự đoán giá trị.
- Sử dụng tính năng tạo thùng một cách hiệu quả để tăng giá trị dự đoán.

Trong chương này, chúng ta sẽ tiếp tục khám phá tập dữ liệu bank_marketing_training từ Chương 3. Chúng ta bắt đầu bằng cách sử dụng đồ họa để điều tra mối quan hệ đặt cược giữa phản hồi mục tiêu và một yếu tố dự đoán phân loại.

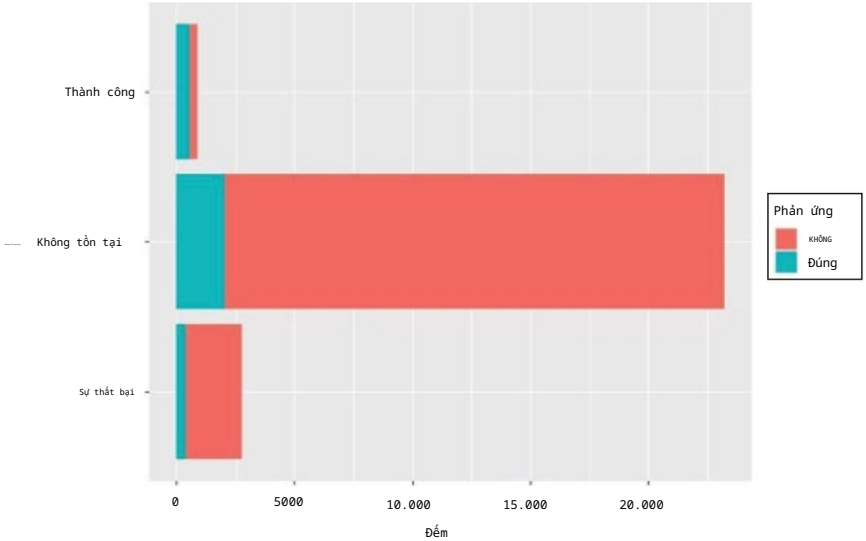
4.2 BIỂU ĐỒ THANH CÓ LỚP PHẢN ỨNG

Chúng ta có thể sử dụng biểu đồ thanh với lớp phủ phản hồi để khám phá mối quan hệ đặt cược giữa một yếu tố dự đoán phân loại và biến mục tiêu. Hình 4.1 hiển thị biểu đồ thanh của previous_outcome với lớp phủ của phản hồi mục tiêu. Previous_outcome đề cập đến kết quả của chiến dịch tiếp thị trước đó với cùng khách hàng này, với hầu hết khách hàng không có trải nghiệm như vậy trước đó.

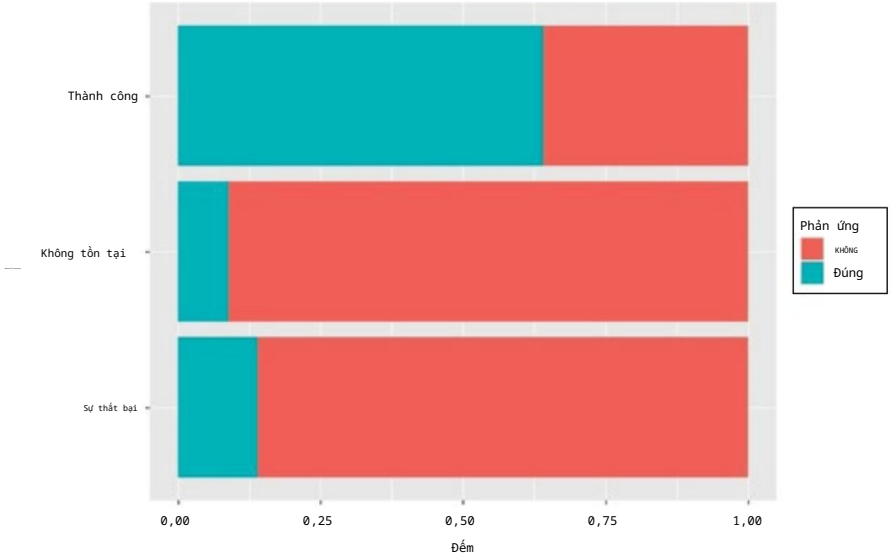
Rõ ràng, hầu hết khách hàng không có bất kỳ kinh nghiệm tiếp thị nào trước đó với công ty (không tồn tại giá trị thay đổi). Nói chung, biểu đồ thanh (không chuẩn hóa)

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google



Hình 4.1 Biểu đồ thanh từ R của previous_outcome với lớp phủ phản hồi .



Hình 4.2 Biểu đồ thanh được chuẩn hóa từ R của previous_outcome với lớp phủ phản hồi .

rất hữu ích để hiển thị phân phối các giá trị của biến phân loại.
Tuy nhiên, không rõ danh mục nào có tỷ lệ người trả lời cao hơn.
Không tồn tại có nhiều người phản hồi nhất nhưng nó cũng có nhiều người không phản hồi nhất.

Để làm rõ những tình huống như thế này, chúng ta có thể thu được biểu đồ thanh chuẩn hóa, biểu đồ này làm cân bằng độ dài của mỗi thanh, để chúng ta có thể so sánh tỷ lệ phản hồi dễ dàng hơn. Hình 4.2 biểu thị một biểu đồ thanh chuẩn hóa của previous_outcome,

Machine Translated by Google

với lớp phủ phản hồi . Từ Hình 4.2, rõ ràng là nhóm có tỷ lệ phản hồi tích cực cao nhất là thành công, những khách hàng đã phản hồi tích cực với chiến dịch trước đó của công ty. Điều thú vị là, những người phản hồi tiêu cực lần trước (thất bại) cũng có tỷ lệ thành công lần này cao hơn một chút so với những người không có liên hệ trước đó.

Bài tập này minh họa hai phương pháp hay nhất khi làm việc với biểu đồ thanh.

Thực tiễn tốt nhất: Biểu đồ thanh

- Khi biểu đồ thanh có lớp phủ không rõ ràng về tỷ lệ phản hồi, hãy bổ sung biểu đồ thanh đó bằng biểu đồ thanh chuẩn hóa.
- Tuy nhiên, không bao giờ cung cấp biểu đồ thanh được chuẩn hóa mà không có phiên bản không được chuẩn hóa của nó, vì phiên bản được chuẩn hóa không đưa ra dấu hiệu nào về sự phân phối ban đầu (có bao nhiêu bản ghi trong mỗi danh mục).

4.2.1 Cách tạo biểu đồ thanh có lớp phủ bằng Python

Tải gói yêu cầu và đọc trong bộ dữ liệu bank_marketing_training dưới dạng bank_train.

```
nhập gấu trúc dưới dạng pd
bank_train = pd.read_csv("C:/.../bank_marketing_
đào tạo")
```

Bước đầu tiên trong việc tạo biểu đồ thanh là tạo bảng dự phòng gồm các giá trị trong biến dự đoán và biến mục tiêu. Chúng tôi tạo bảng bằng cách sử dụng crosstab() yêu cầu.

```
crosstab_01 = pd.crosstab(bank_train['previous_
kết quả'], bank_train['phản hồi'])
```

Đoạn mã này sẽ được xem xét chi tiết sau. Hiện tại, chúng tôi lưu bảng dưới dạng crosstab_01.

Bây giờ, chúng ta có thể tạo biểu đồ thanh dựa trên bảng.

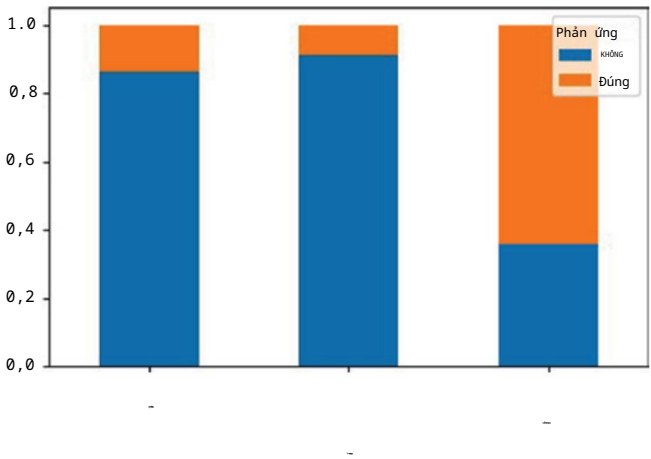
```
crosstab_01.plot(loại='thanh', xếp chồng lên nhau = Đúng)
```

Để tạo biểu đồ thanh, hãy thêm .plot() vào cuối đối tượng crosstab_01 . Lệnh plot() nhận các giá trị đầu vào tùy chọn khác nhau. Chúng tôi chỉ định loại đầu vào = 'bar' để vẽ biểu đồ thanh và xếp chồng lên nhau = True để chỉ định biểu đồ thanh xếp chồng lên nhau.

Để tạo một phiên bản chuẩn hóa, chúng ta cần thay đổi bảng sao cho các giá trị trong mỗi ô là tỷ lệ của các giá trị phản hồi "không" và "có" trong mỗi giá trị của bộ dự đoán trước_kết quả, như sau:

```
crosstab_norm = crosstab_01.div(crosstab_01.sum(1), axis = 0)
```

Machine Translated by Google



Hình 4.3 Biểu đồ thanh được chuẩn hóa từ Python của previous_outcome với lớp phủ phản hồi .

Lệnh div() sẽ chia các giá trị của bảng cho một đối tượng khác, trong mỗi trục được chỉ định. Trong ví dụ của chúng tôi, chúng tôi muốn chia các ô trong hàng 1 của bảng cho tổng các ô trong hàng 1, v.v. cho hàng 2 và hàng 3. Để thực hiện điều này, trước tiên chúng tôi đặt giá trị mà chúng tôi muốn chia bởi là crosstab_01.sum(1), là tổng của mỗi hàng trong bảng. Sau đó, chúng tôi đặt axis = 0 để xác định rằng chúng tôi muốn chia các hàng của bảng theo các giá trị này. Kết quả là một bảng có các ô là tỷ lệ dữ liệu trong hàng đó nằm trong cột đó. Chúng tôi lưu bảng kết quả dưới dạng crosstab_norm.

Sau khi bảng được lưu, hãy trực quan hóa bảng bằng cách sử dụng mã cho biểu đồ thanh xếp chồng lên nhau, như trên.

```
crosstab_norm.plot(kind='bar', stacked = True)
```

Đồ thị kết quả được hiển thị trong Hình 4.3.

4.2.2 Cách tạo biểu đồ thanh có lớp phủ bằng R

Đọc trong bộ dữ liệu bank_marketing_training dưới dạng bank_train. Chúng tôi sẽ sử dụng gói ggplot2 để tạo biểu đồ của mình. Bạn cần cài đặt gói một lần bằng cách sử dụng install.packages(), sau đó mở nó mỗi khi bạn viết mã mới bằng thư viện().

```
install.packages("ggplot2"); thư viện (ggplot2)
```

Mã ggplot sử dụng các lệnh khác nhau được xâu chuỗi với nhau bằng dấu cộng (+), như minh họa trong ví dụ sau. Lưu ý rằng các dấu cộng phải ngay lập tức

Machine Translated by Google

làm theo lệnh trước, không có ngắt dòng ở giữa, nhưng có thể theo sau bởi ngắt dòng.

Để tạo biểu đồ thanh của biến previous_outcome , chúng ta sử dụng phương thức ggplot() và các lệnh geom_bar() .

```
ggplot(bank_train, aes(previous_outcome)) + geom_bar() + coord_flip()
```

Lệnh ggplot() bắt đầu biểu đồ. Đầu vào bank_train chỉ định tập dữ liệu đang được sử dụng và biến quan tâm được liệt kê trong aes() ("aes" cho "aes thetics"). Đoạn mã thứ hai là geom_bar(), chỉ định rằng một biểu đồ thanh sẽ được tạo. Mã này biết nên sử dụng biến nào nhờ vào lệnh ggplot() trước đó . Đoạn mã thứ ba tùy chọn là coord_flip(), sẽ làm cho các thanh chạy theo chiều ngang.

Để tạo biểu đồ thanh có lớp phủ phản hồi, chúng tôi thêm đầu vào điền .

```
ggplot(bank_train, aes(previous_outcome)) + geom_
bar(aes(fill = response)) + coord_flip()
```

Lưu ý rằng thay đổi duy nhất là thêm aes(fill = response) bên trong geom_ quán ba(). Kết quả thu được đồ thị như Hình 4.1.

Để chuẩn hóa biểu đồ thanh, hãy thêm position = "fill" bên trong geom_bar(), như hình bên dưới.

```
ggplot(bank_train, aes(previous_outcome)) + geom_
thanh (aes (điền = phản hồi),
          vị trí = "điền") + coord_flip()
```

Vị trí đầu vào = "fill" được thêm vào bên trong lệnh geom_bar() , nhưng bên ngoài aes(). Kết quả thu được đồ thị như hình 4.2.

4.3 BẢNG DỰ PHÒNG

Để giúp định lượng mối quan hệ giữa một yếu tố dự đoán phân loại và mục tiêu, chúng ta có thể xây dựng một bảng dự phòng, là một bảng chéo của hai biến và chứa một ô cho mọi kết hợp giá trị biến (nghĩa là cho mọi tình huống dự phòng).

Hình 4.4 chứa một bảng dự phòng của previous_outcome với phản hồi. Lưu ý rằng thực tế thông thường là có biến mục tiêu đại diện cho các hàng, với bộ dự đoán đại diện cho các cột. Đối với EDA, việc bao gồm cột phần trăm tuổi cũng rất hữu ích. Hình 4.5 chứa bảng có cột tỷ lệ phần trăm. Hầu hết khách hàng không có chiến dịch tiếp thị trước đó (không tồn tại), vì vậy hãy lưu ý rằng 21.176 trong số này trả lời không trong khi 2034 trả lời có. Nhìn chung, lưu ý rằng tỷ lệ phản hồi có chỉ là 13,9% cho trường hợp thất bại và chỉ 8,8% cho trường hợp không tồn tại, nhưng tỷ lệ rất cao là 64% khi chiến dịch tiếp thị trước đó của khách hàng thành công.

Thực hành tốt nhất cho các bảng dự phòng làm theo.

Machine Translated by Google

Thực tiễn tốt nhất: Bảng dự phòng

- Để biến trả lời đại diện cho các hàng.
- Sau đó, lấy phần trăm cột để so sánh trực tiếp tỷ lệ phản hồi cho từng loại của yếu tố dự báo.

	failure	nonexistent	success	total
no	2390	21176	320	23886
yes	385	2034	569	2988
total	2775	23210	889	26874

Hình 4.4 Bảng dự phòng từ R của previous_outcome có phản hồi.

	failure	nonexistent	success
no	86.1	91.2	36.0
yes	13.9	8.8	64.0

Hình 4.5 Bảng dự phòng từ R của previous_outcome với bảng phản hồi có cột phần trăm thay vì số đếm.

4.3.1 Cách xây dựng bảng dự phòng bằng Python

Chúng tôi phải tạo một bảng dự phòng để tạo biểu đồ thanh, nhưng không nghiên cứu chi tiết về mã. Hãy để chúng tôi kiểm tra mã cho bảng kỹ hơn.

```
crosstab_01 = pd.crosstab(bank_train['previous_
kết quả'], bank_train['kết quả'])
```

Lưu ý thứ tự của các biến. Bảng được tạo bởi mã này sẽ có các hàng là previous_outcome . Bảng này đã tạo biểu đồ thanh mà chúng ta cần trong phần trước, nhưng để tuân theo phương pháp hay nhất và biến mục tiêu đại diện cho các hàng, chúng ta cần thay đổi mã thành:

```
crosstab_02 = pd.crosstab(bank_train['response'], bank_
tàu['previous_outcome'])
```

Đảm bảo lưu bảng. Chúng tôi lưu đầu ra dưới dạng crosstab_02. Kết quả sẽ là một bảng tương đương với Hình 4.4.

Để tính tỷ lệ cột của bảng, chúng ta cần chia từng cột cho tổng cột. Chúng tôi sử dụng các lệnh sum() và div() theo cách tương tự như phần trước. Tuy nhiên, lần này chúng tôi đang lấy tỷ lệ phần trăm theo cột chứ không phải theo hàng.

```
vòng(crosstab_02.div(crosstab_02.sum(0), axis = 1)*100, 1)
```

Machine Translated by Google

Lưu ý rằng chúng tôi nhân bảng kết quả với 100 để có được tỷ lệ phần trăm thay vì tỷ lệ. Ngoài lệnh sum() và div() , chúng ta đặt mã cho bảng bên trong lệnh round() . Lệnh round() sẽ làm tròn các số trong bảng tới số chữ số có nghĩa đã chỉ định; ở đây, chúng tôi chỉ định một chữ số. Kết quả là một bảng tương đương với Hình 4.5.

4.3.2 Cách xây dựng các bảng dự phòng bằng R

Lệnh tạo bảng là table(), với các biến quan tâm bên trong dấu ngoặc đơn.

```
t.v1 <- table(bank_train$response, bank_train$previous_
kết quả)
```

Biến đầu tiên, bank_train$response, tạo nên các hàng, trong khi biến thứ hai, bank_train$previous_outcome, tạo nên các cột. Chúng tôi sẽ lưu bảng của chúng tôi dưới dạng t.v1, vì vậy chúng tôi có thể chỉnh sửa nó.

Để thêm tổng số hàng và cột vào bảng, hãy sử dụng lệnh addmargins() .

```
t.v2 <- addmargins(A = t.v1, FUN = danh sách (tổng = tổng), yên lặng = TRUE)
```

Đầu vào A = t.v1 chỉ định bảng sẽ được chỉnh sửa, trong trường hợp của chúng tôi là t.v1. Đầu vào FUN = list(total = sum) chỉ định danh sách các chức năng sẽ được thực hiện để tạo hàng và cột biên. Trong trường hợp của chúng tôi, chúng tôi muốn tạo một hàng và cột có tên là tổng chứa tổng của các hàng và cột. Chúng tôi lưu bảng đã chỉnh sửa dưới dạng t.v2. Để xem bảng đã hoàn thành, hãy tự chạy t.v2 . Kết quả được thể hiện trong Hình 4.4.

Bây giờ chúng tôi muốn chỉnh sửa bảng t.v1 để nó cung cấp cho chúng tôi tỷ lệ phần trăm cột.

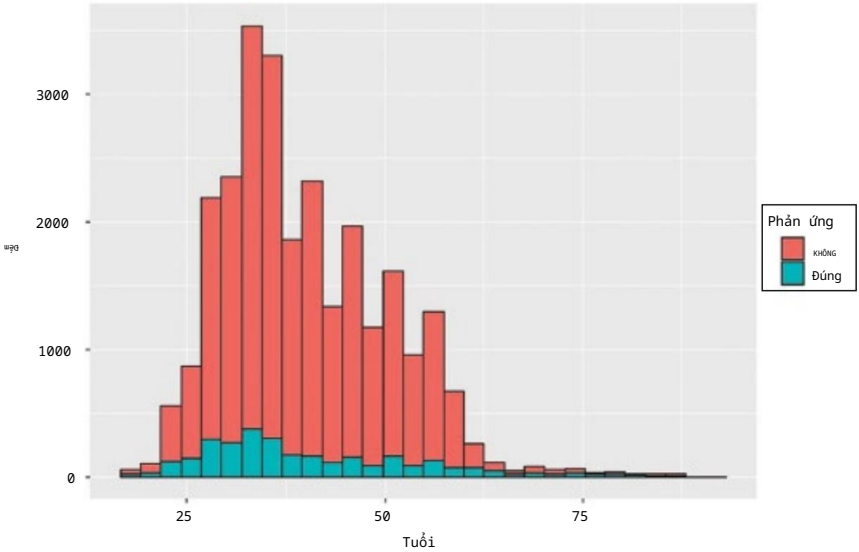
```
round(prop.table(t.v1, margin = 2)*100, 1)
```

Để tính tỷ lệ các mục nhập trong các ô của bảng, hãy sử dụng prop. bàn(). Đầu vào t.v1 cho prop.table() biết bảng nào cần tính tỷ lệ.
Đầu vào margin = 2 yêu cầu R tính phần trăm cột. Nhân kết quả với 100 bằng cách sử dụng *100 sẽ cho chúng ta tỷ lệ phần trăm thay vì tỷ lệ. Cuối cùng, đặt lệnh prop.table() bên trong lệnh round() sẽ làm tròn các mục đến một số điểm thập phân nhất định; trong trường hợp của chúng tôi, một (1) dấu thập phân. Kết quả được thể hiện trong Hình 4.5.

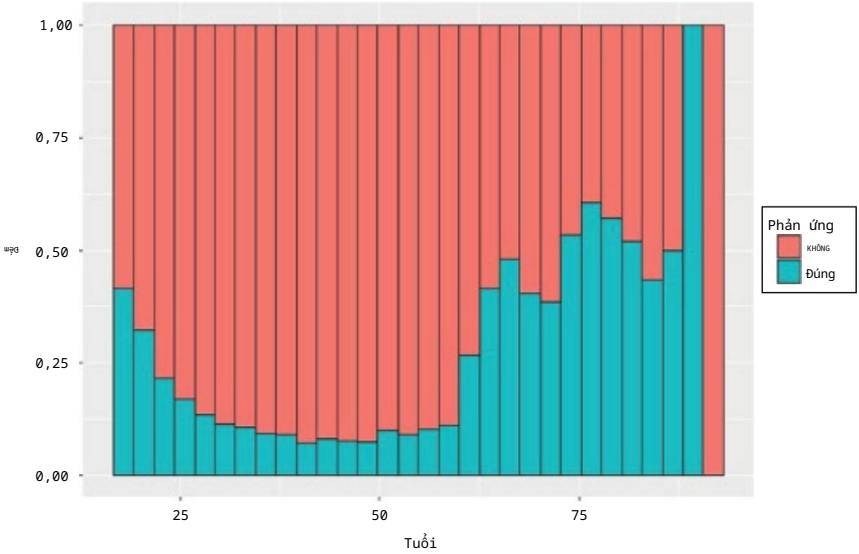
4.4 BIỂU ĐỒ VỚI LỚP PHẢN ỨNG

Biểu đồ là biểu diễn đồ họa của phân phối tần số cho một biến số. Hình 4.6 cho thấy biểu đồ của biến tuổi với lớp phủ phản hồi. Hầu hết khách hàng nằm trong độ tuổi từ giữa 20 đến khoảng 60 tuổi. Vì thế

Machine Translated by Google



Hình 4.6 Biểu đồ từ R của độ tuổi với lớp phủ phản hồi .



Hình 4.7 Biểu đồ chuẩn hóa từ R tuổi với lớp phủ phản hồi .

biểu đồ (không chuẩn hóa) rất hữu ích để xem phân phối các giá trị của một biến số.

Tuy nhiên, một lần nữa, hơi khó để xác định bất kỳ mô hình nào trong tỷ lệ phản hồi . Để làm rõ hơn các tỷ lệ phản hồi này, chúng tôi chuyển sang biểu đồ tần suất chuẩn với lớp phủ phản hồi, được hiển thị trong Hình 4.7. Đột nhiên mô hình phản ứng trở nên rõ ràng. Phản hồi của khách hàng bắt đầu cao cho

Machine Translated by Google

những người 20 tuổi, giảm dần, đi ngang ở mức thấp đối với những người 30-60 tuổi và tăng mạnh trở lại đối với những người trên 60 tuổi. Vì vậy, biểu đồ chuẩn hóa cho phép chúng ta phân biệt rõ hơn các kiểu phản ứng này, nhưng tiếc là biểu đồ chuẩn hóa không cho biết chúng tôi về sự phân bố ban đầu của độ tuổi trong dân số khách hàng của chúng tôi.

Điều này dẫn chúng ta đến hai phương pháp hay nhất cho biểu đồ.

Thực tiễn tốt nhất: Biểu đồ

- Sử dụng biểu đồ không chuẩn hóa để có được phân phối dữ liệu ban đầu các giá trị.
- Nếu cần, hãy sử dụng biểu đồ đã chuẩn hóa để giúp phân biệt rõ hơn phản hồi hoa văn.

4.4.1 Cách tạo biểu đồ có lớp phủ bằng Python

Tải các gói cần thiết.

```
nhập numpy dưới dạng np
nhập matplotlib.pyplot dưới dạng plt
```

Tách biến bạn muốn vẽ biểu đồ theo lớp phủ bạn muốn sử dụng. Vì chúng tôi đang tạo biểu đồ tuổi bằng cách sử dụng lớp phủ phản hồi, nên chúng tôi tách biến tuổi, bank_train['age'], bằng hai giá trị của phản hồi biến. Lưu từng phần dưới dạng biến riêng của nó.

```
bt_age_y = bank_train[bank_train.response == "có"]
 ['tuổi']
bt_age_n = bank_train[bank_train.response == "không"]
 ['tuổi']
```

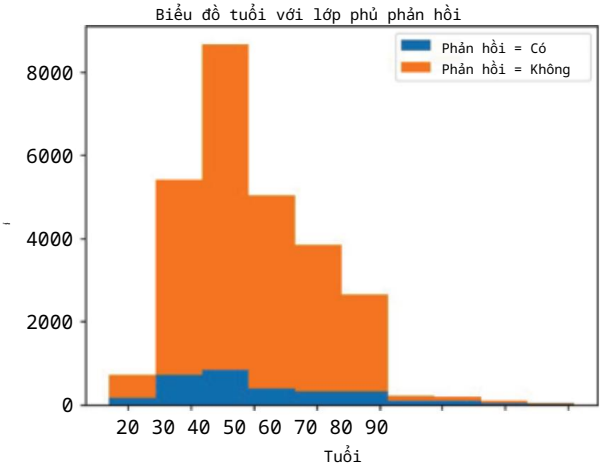
Kết quả là hai biến, bt_age_y và bt_age_n, có giá trị tuổi chỉ từ những bản ghi có phản hồi = “có” và phản hồi = “không,”
tương ứng.

Khi các biến được tạo, hãy tạo biểu đồ xếp chồng của hai biến.

```
plt.hist([bt_age_y, bt_age_n], bins = 10, stacked = True)
plt.legend(['Phản hồi = Có', 'Phản hồi = Không'])
plt.title('Histogram of Age with Response Overlay')
plt.xlabel('Tuổi'); plt.ylabel('Tần số'); plt.show()
```

Đối với lệnh hist() , đầu vào Stacked = True sẽ xếp chồng hai biến, trong khi bins = 10 chỉ định số lượng ngăn trong biểu đồ tần suất. Các lệnh legend(), title(), xlabel() và ylabel() chỉ định các giá trị của chú giải, tiêu đề, nhãn trục x và nhãn trục y. Cuối cùng, show() hiển thị hình. Kết quả được thể hiện trong Hình 4.8.

Machine Translated by Google



Hình 4.8 Biểu đồ từ tuổi Python với lớp phủ phản hồi .

Bây giờ chúng ta sẽ tạo một biểu đồ chuẩn hóa. Đầu tiên, tạo một biểu đồ xếp chồng lên nhau, nhưng lần này hãy lưu thông tin do biểu đồ tạo ra.

```
(n, thùng, bản vá) = plt.hist([bt_age_y, bt_age_n], thùng = 10, xếp chồng = True)
```

Phía bên trái của mã lưu một số mẫu thông tin từ biểu đồ. Cụ thể, n là chiều cao của các thanh biểu đồ và các thùng là ranh giới của mỗi ngăn trong biểu đồ. Lưu ý rằng, vì hai biến đang được vẽ trong biểu đồ tần số nên n có hai dãy số. Chuỗi đầu tiên dành cho biến thứ nhất và chuỗi thứ hai dành cho biến thứ hai. Số đầu tiên trong mỗi chuỗi là chiều cao của thanh đầu tiên cho mỗi biến.

Để tạo biểu đồ được chuẩn hóa của chúng tôi, chúng tôi cần biết tỷ lệ của mỗi thùng mà mỗi biến thể hiện. Để thực hiện điều này, chúng ta cần đưa thông tin chứa trong n vào một ma trận và thu được tỷ lệ cột.

Để bắt đầu xây dựng ma trận, hãy kết hợp độ cao của hai biến ' thanh thành một mảng bằng lệnh column_stack() .

```
n_table = np.column_stack((n[0], n[1]))
```

Kết quả, n_table, là một ma trận hai cột trong đó các mục của mỗi cột chứa chiều cao của mỗi thanh.

Để tính tỷ lệ của thanh được chiếm bởi mỗi biến, chúng ta cần chia mỗi hàng cho tổng của hàng đó.

```
n_norm = n_table / n_table.sum(axis=1)[:, Không]
```

Bây giờ, mỗi hàng trong n_norm tính tổng bằng một và các cột trong mỗi hàng đưa ra tỷ lệ của biến đó tạo nên hàng.

Machine Translated by Google

Trong bước chuẩn bị cuối cùng của chúng tôi, chúng tôi tạo một mảng có các hàng là các phần cắt chính xác của mỗi thùng.

```
ourbins = np.column_stack((bins[0:10], bins[1:11]))
```

Mỗi hàng trong thùng rác của chúng tôi cung cấp giới hạn trên và dưới của mỗi thùng.

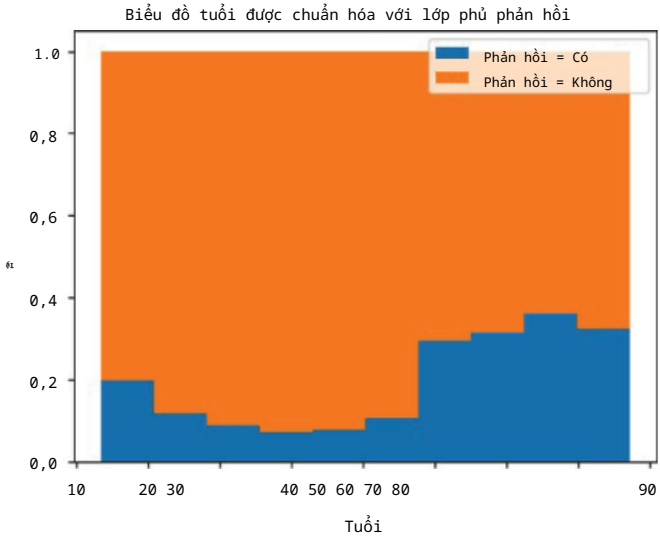
Bây giờ, chúng tôi đã sẵn sàng để tạo biểu đồ chuẩn hóa của mình.

```
p1 = plt.bar(x = ourbins[:,0], height = n_norm[:,0], width = ourbins[:, 1] -
ourbins[:, 0])
p2 = plt.bar(x = ourbins[:,0], height = n_norm[:,1], width = ourbins[:, 1] -
ourbins[:, 0],
         đáy = n_norm[:,0])
plt.legend(['Phản hồi = Có', 'Phản hồi = Không'])
plt.title('Biểu đồ tuổi được chuẩn hóa với lớp phủ phản hồi')

plt.xlabel('Tuổi'); plt.ylabel('Tỷ lệ'); plt.show()
```

Trong các lệnh bar() , đầu vào x chỉ định giới hạn trên và dưới của các ngăn, đầu vào chiều cao sử dụng các giá trị đếm được chuẩn hóa mà chúng ta đã tạo trước đó để chỉ định chiều cao của từng phần trong số hai phần của mỗi thanh và đầu vào chiều rộng sử dụng lại chiều rộng thanh từ biểu đồ thanh ban đầu. Đáy = n_norm[:,0]

đầu vào trong lệnh bar() thứ hai chỉ định phần thứ hai trong số hai phần thanh bắt đầu trên phần đầu tiên. Các lệnh còn lại là các tùy chọn tùy chỉnh giống như chúng tôi đã sử dụng trong biểu đồ thanh xếp chồng lên nhau trước đó. Kết quả được thể hiện trong Hình 4.9.



Hình 4.9 Biểu đồ được chuẩn hóa từ Python of Age với lớp phủ phản hồi .

Machine Translated by Google

4.4.2 Cách xây dựng biểu đồ với lớp phủ bằng R

Chúng ta bắt đầu với lệnh ggplot() , với tập dữ liệu bank_train được chỉ định và biến age được chỉ định trong lệnh aes() . Để tạo biểu đồ, chúng tôi thêm lệnh geom_histogram() .

```
ggplot(bank_train, aes(tuổi)) + geom_
biểu đồ (màu = "đen")
```

Đầu vào color = “black” tùy chọn tạo ra các đường màu đen xung quanh mỗi thanh của biểu đồ. Kết quả là một biểu đồ không có lớp phủ.

Để thêm lớp phủ vào biểu đồ sử dụng biến mục tiêu, đầu vào aes(fill = response) được thêm vào geom_histogram().

```
ggplot(bank_train, aes(age)) + geom_histogram(aes(fill = response), color="black")
```

Biểu đồ kết quả với lớp phủ được hiển thị trong Hình 4.6.

Để chuẩn hóa biểu đồ, đầu vào position = “fill” được thêm vào geom_histogram().

```
ggplot(bank_train, aes(tuổi)) +
geom_histogram(aes(fill = response), color="black", position = "fill")
```

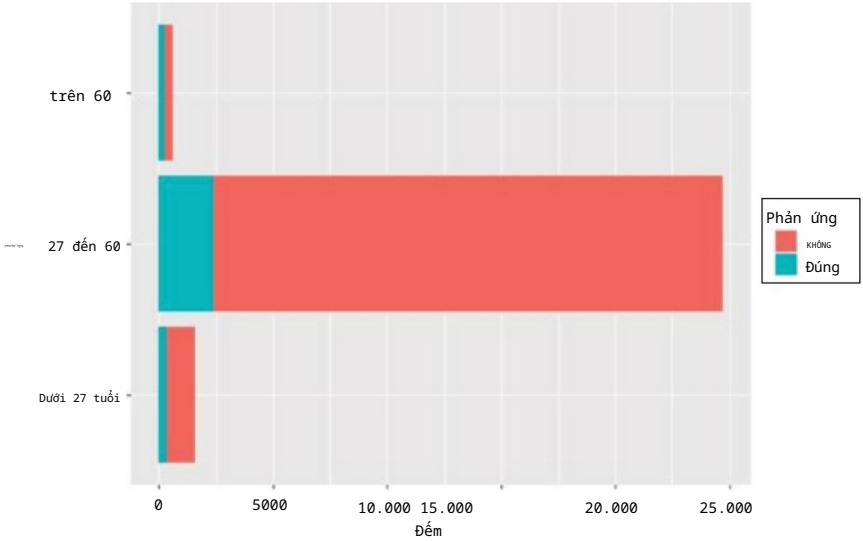
Kết quả biểu đồ chuẩn hóa với lớp phủ được hiển thị trong Hình 4.7.

4.5 GỬI DỰA TRÊN GIÁ TRỊ DỰ ĐOÁN

Một số thuật toán hoạt động tốt hơn với các biến phân loại thay vì biến số, do đó, có thể hữu ích cho nhà phân tích khi sử dụng tính năng tạo thùng để lấy các biến phân loại mới dựa trên cách các bộ giá trị khác nhau của công cụ dự đoán số hoạt động đối với phản hồi. Ví dụ, lấy hình 4.7. Để tối ưu hóa tín hiệu của chúng tôi từ dữ liệu, chúng tôi tự hỏi: Làm cách nào chúng tôi có thể phân loại các giá trị số của tuổi sao cho các danh mục có tỷ lệ phản hồi rất khác nhau? Rõ ràng, một danh mục sẽ là những khách hàng từ 60 tuổi trở lên, những người có tỷ lệ phản hồi cao. Điều này trái ngược với nhóm trung bình (khoảng giữa những năm 20 đến 60) có xác suất phản hồi thấp. Cuối cùng là nhóm trẻ nhất (đến giữa độ tuổi 20) cũng có tỷ lệ phản hồi cao. Do đó, chúng ta có thể định nghĩa biến mới của mình như sau (ngưỡng 27 hơi tùy ý; 25 hoặc 26 cũng sẽ hoạt động):

$$\text{tuổi bị hạn chế} = \left\{ \begin{array}{ll} 1: \text{Dưới} & 27 \\ 2 & 27 \text{ đến } 60 \\ 3 & 60 \text{ trở lên} \end{array} \right\}$$

Machine Translated by Google



Hình 4.10 Biểu đồ thanh từ R của age_binned với lớp phủ phản hồi .

Hình 4.10 hiển thị biểu đồ thanh có lớp phủ cho age_binned. Hình 4.11 cho thấy phiên bản chuẩn hóa của biểu đồ thanh. Sau đó, Hình 4.12 hiển thị bảng dự phòng của age_binned với phản hồi và Hình 4.13 đưa ra phần trăm cột của bảng dự phòng. Rõ ràng, cả nhóm lớn tuổi và trẻ tuổi đều có tỷ lệ phản hồi cao hơn nhiều so với nhóm trung bình. Thật không may, hơn 90% khách hàng của chúng tôi thuộc nhóm trung lưu này.

Đây là một phương pháp hay nhất quan trọng cần nhớ về việc tạo thùng.

Thực hành tốt nhất: Binning

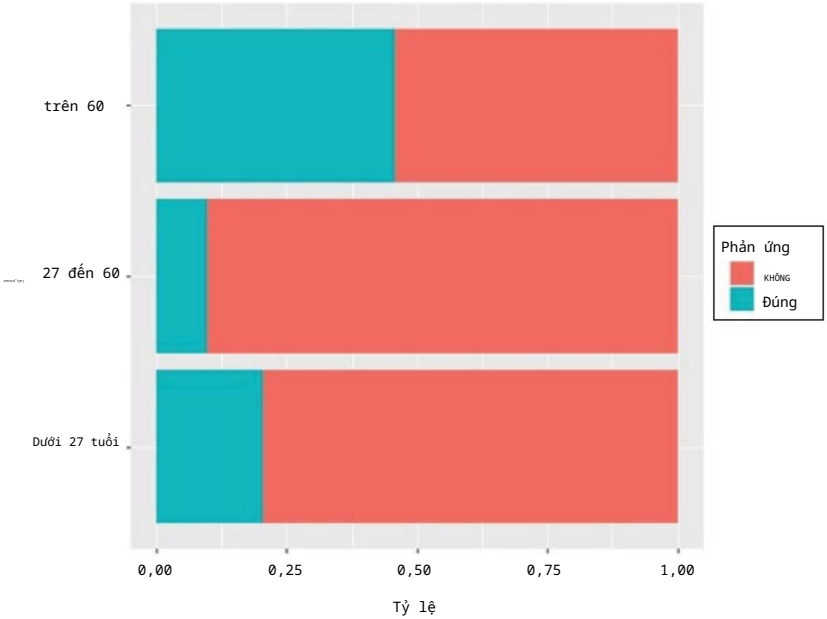
- Nhiều gói phần mềm cung cấp các phương pháp tạo thùng "tự động", chẳng hạn như equal-tạo thùng theo chiều rộng danh mục hoặc tạo thùng theo số bản ghi bằng nhau trên mỗi danh mục. Mặc dù những điều này có thể có công dụng của chúng, nhưng nếu bạn quan tâm đến việc nâng cao khả năng dự đoán của phân tích của mình, bạn nên luôn cố gắng sử dụng tính năng tạo thùng dựa trên giá trị dự đoán mà chúng tôi đã trình bày ở đây.

4.5.1 Cách thực hiện Binning dựa trên giá trị dự đoán bằng Python

Tải gói cần thiết.

```
nhập gấu trúc dưới dạng pd
```

Machine Translated by Google



Hình 4.11 Biểu đồ thanh được chuẩn hóa từ R của age_binned với lớp phủ phản hồi .

	Under 27	27 to 60	Over 60	total
no	1255	22315	316	23886
yes	322	2399	267	2988
total	1577	24714	583	26874

Hình 4.12 Bảng dự phòng từ R của age_binned và phản hồi.

	Under 27	27 to 60	Over 60
no	79.6	90.3	54.2
yes	20.4	9.7	45.8

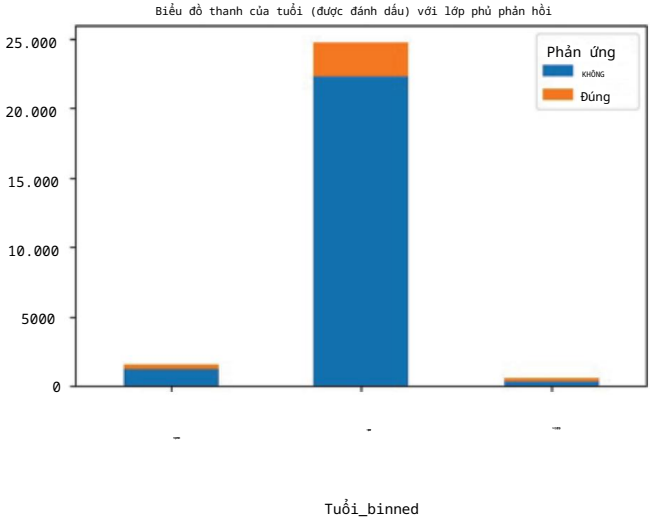
Hình 4.13 Bảng dự phòng từ R của age_binned và phản hồi với cột tỷ lệ phần trăm.

Bin các giá trị bằng cách sử dụng cut() từ gói gấu trúc .

```
bank_train['age_binned'] = pd.cut(x = bank_train['age'], bins = [0, 27, 60.01,
100],
        nhãn=["Dưới 27", "27 đến 60", "Trên 60"], phải =
SAI)
```

Đầu vào x = chỉ định biến mà bạn muốn chia thành các danh mục. Các thùng = đầu vào chỉ định các cạnh của mỗi thùng. Các nhãn = đầu vào chỉ định nhãn bin. Đầu vào đúng = Sai chỉ định rằng chúng tôi muốn các thùng của mình loại trừ bên phải-điểm cắt tay. Ví dụ: ngăn đầu tiên sẽ bao gồm tất cả các độ tuổi từ 0 đến (nhưng

Machine Translated by Google



Hình 4.14 Biểu đồ thanh từ Python của age_binned với lớp phủ phản hồi.

ngoại trừ) 27 tuổi. Chúng tôi lưu biến phân loại mới trong tập dữ liệu dưới tên age_binned bằng cách sử dụng bank_train['age_binned'].

Lưu ý rằng, mặc dù chúng tôi có hầu hết các điểm cắt trực quan cho các ngăn, chẳng hạn như 0 cho giới hạn dưới của ngăn đầu tiên và 100 cho giới hạn trên của ngăn cuối cùng, chúng tôi cũng có điểm cắt là 60,01. Việc chỉ định điểm cắt là 60,01 (hoặc bất kỳ số nào trong khoảng nhưng không bao gồm 60 và 61), kết hợp với giá trị nhập phải = Sai, sẽ đảm bảo rằng danh mục giữa của chúng tôi bao gồm tất cả các độ tuổi từ 27 đến 60. Cụ thể, chúng tôi yêu cầu Python tạo một thùng từ 27 trở lên nhưng không bao gồm 60.01. Vì độ tuổi là số nguyên trong tập dữ liệu này nên điều này thực sự làm cho thùng bao gồm cả độ tuổi từ 27–60.

Để vẽ biểu đồ tạo thùng với lớp phủ phản hồi, hãy tạo bảng dự phòng cần thiết và vẽ biểu đồ đó, sử dụng mã tương tự như mã chúng tôi đã sử dụng trước đó. Đồ thị kết quả được hiển thị trong Hình 4.14.

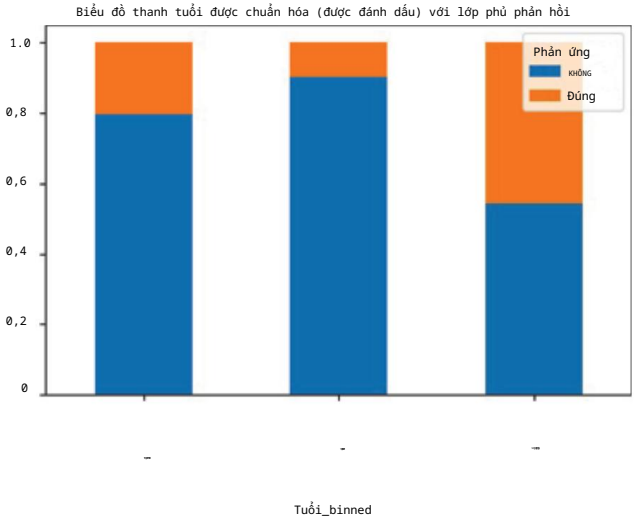
```
crosstab_02 = pd.crosstab(bank_train['age_binned'], bank_train['response'])

crosstab_02.plot(kind='bar', stacked = True,
          title = 'Biểu đồ Thanh Tuổi (Binned) với Lớp phủ Phản hồi')
```

Để tạo biểu đồ thanh được chuẩn hóa, hãy làm theo các hướng dẫn tương tự như đã thảo luận trước đây. Đồ thị kết quả được hiển thị trong Hình 4.15.

Để có được bảng dự phòng của biến phản hồi và biến phân loại mới của chúng ta, hãy sử dụng lệnh crosstab() . Hãy nhớ sử dụng phản hồi biến mục tiêu
như các hàng của bảng. Để có được bảng tỷ lệ cột, hãy sử dụng các lệnh div() và sum() trên bảng kết quả. Chúng tôi đã thảo luận mã này trước đó khi tạo biểu đồ bằng Python.

Machine Translated by Google



Hình 4.15 Biểu đồ thanh được chuẩn hóa từ Python của age_binned với lớp phủ phản ứng.

4.5.2 Cách thực hiện Binning dựa trên giá trị dự đoán bằng R

Để tạo biến phân loại, chúng ta sẽ sử dụng lệnh cut() trên biến tuổi.

```
bank_train$age_binned <- cut(x = bank_train$age, break = c(0, 27, 60,01, 100),

        phải = FALSE, nhãn = c("Dưới 27", "27 đến 60",
        "Trên 60"))
```

Đầu vào x chỉ định biến được đánh dấu. Đầu vào ngắt chỉ định các điểm cắt của mỗi ngăn. Đầu vào bên phải = FALSE cho biết rằng điểm cắt bên phải của mỗi ngăn phải được loại trừ khỏi danh mục. Nhập nhãn tùy chọn thay thế các nhãn mặc định của từng ngăn với các nhãn do người dùng chỉ định, nhãn này sẽ áp dụng cho từng ngăn theo thứ tự (ví dụ: “Dưới 27 tuổi” cho ngăn 0–27). Chúng tôi sử dụng giá trị bin 60,01, thay vì 60 và đầu vào đúng = FALSE vì những lý do tương tự được thảo luận trong phần Python ở trên. Chúng tôi lưu kết quả trong tập dữ liệu bank_train là age_binned.

Khi chúng tôi có biến phân loại của mình, chúng tôi có thể vẽ biểu đồ đó với lớp phủ phản hồi bằng cách sử dụng các lệnh ggplot đã đề cập trước đó.

```
ggplot(bank_train, aes(age_binned)) + geom_bar(aes(fill = response)) + coord_flip()
```

Biểu đồ thanh kết quả có lớp phủ được hiển thị trong Hình 4.10. Để tạo biểu đồ thanh chuẩn hóa cho các biến này, hãy làm theo các nguyên tắc tương tự như đã thảo luận cho

Machine Translated by Google

biểu đồ thanh chuẩn hóa trước đó. Biểu đồ thanh chuẩn hóa kết quả được hiển thị trong Hình 4.11.

Chúng ta có thể xây dựng một bảng dự phòng của biến phân loại và biến phản hồi bằng cách sử dụng lệnh table() và một bảng tỷ lệ cột bằng cách sử dụng lệnh prop.table() . Các chi tiết của cả hai lệnh đã được thảo luận trước đây. Mã được đưa ra dưới đây.

```
t2 <- table(bank_train$response, bank_train$age_bin); t2
vòng(prop.table(t2, lề = 2)*100, 1)
```

Lưu ý cách sử dụng dấu chấm phẩy trong dòng đầu tiên cho phép cả việc xây dựng và lưu bảng cũng như in bảng diễn ra trên một dòng. Các bảng kết quả được thể hiện trong Hình 4.12 và 4.13.

Các phương pháp khác trong EDA có thể được tìm thấy trong Khai thác dữ liệu và Phân tích dự đoán[1].

NGƯỜI GIỚI THIỆU

Gói matplotlib của Python có nhiều tùy chọn đồ họa. Để bắt đầu tìm hiểu thêm, hãy xem ấn phẩm: John D. Hunter, Matplotlib: môi trường đồ họa 2D, Điện toán trong Khoa học & Kỹ thuật, 9, 90–95, 2007, doi:https://doi.org/10.1109/MCSE.2007.55

Chúng tôi vừa lướt qua bề mặt của gói ggplot2 ! Để biết thêm thông tin, hãy xem ấn phẩm sau: H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer-Verlag, New York, 2009.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Khi nào các nhà phân tích nên sử dụng phân tích dữ liệu khám phá (EDA) thay vì giả thuyết thử nghiệm?
2. Một số ví dụ về những gì EDA cho phép người dùng làm là gì?
3. Chúng ta sử dụng biểu đồ nào để khám phá mối quan hệ giữa một yếu tố dự đoán phân loại và biến mục tiêu?
4. Biểu đồ thanh (không chuẩn hóa) hữu ích để làm gì?
5. Nêu một ưu điểm và một nhược điểm của việc sử dụng biểu đồ thanh chuẩn hóa.
6. Nêu hai phương pháp hay nhất khi làm việc với biểu đồ thanh cho EDA?
7. Bảng dự phòng giúp chúng ta làm gì?
8. Giải thích hai phương pháp hay nhất khi làm việc với các bảng dự phòng trong EDA?

1Bởi Daniel T. Larose và Chantal D. Larose, John Wiley and Sons, Inc., 2015.

Machine Translated by Google

9. Biểu đồ là gì?

10. Mô tả một ưu điểm và một nhược điểm của việc sử dụng biểu đồ chuẩn hóa.

11. Các phương pháp hay nhất để làm việc với biểu đồ trong EDA là gì?

12. Tại sao việc phân tích một biến số có thể hữu ích cho nhà phân tích?

13. Tại sao chúng ta sử dụng phương pháp tạo hộp được trình bày trong chương này thay vì tạo hộp tự động phương pháp?

14. Ngoại suy từ câu trả lời của bạn cho câu hỏi trước và giải thích tại sao các nhà khoa học dữ liệu nên sử dụng các phương pháp phân tích dữ liệu tự động rất cẩn thận.

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu bank_marketing_training . Sử dụng Python hoặc R để giải quyết từng vấn đề.

15. Tạo biểu đồ thanh của biến previous_outcome , với lớp phủ phản hồi .

16. Tạo biểu đồ thanh chuẩn hóa của biến previous_outcome với lớp phủ phản hồi . Mô tả mối quan hệ giữa previous_outcome và phản hồi.

17. Tạo một bảng dự phòng của trước_kết quả và phản hồi. So sánh bảng dự phòng với biểu đồ thanh không chuẩn hóa và biểu đồ thanh chuẩn hóa.

18. Tạo biểu đồ tuổi với lớp phủ phản hồi .

19. Tạo biểu đồ tuổi đã chuẩn hóa với lớp phủ phản hồi . Mô tả mối quan hệ giữa tuổi tác và phản ứng.

20. Bin biến tuổi bằng cách sử dụng các bin được chỉ định trong chương này và tạo biểu đồ thanh của biến tuổi đã được binned với lớp phủ phản hồi .

PHÂN TÍCH THỰC TẾ

Đối với Bài tập 21–30, hãy tiếp tục làm việc với tập dữ liệu bank_marketing_training . Sử dụng Python hoặc R để giải quyết từng vấn đề.

21. Tạo các đồ thị sau. Sức mạnh của mỗi biểu đồ là gì? Yếu đuối?

    Một. Biểu đồ thanh của hôn nhân.

    b. Biểu đồ thanh của hôn nhân, với lớp phủ phản hồi.

    c. Biểu đồ thanh chuẩn hóa của hôn nhân, với lớp phủ phản hồi.

22. Sử dụng đồ thị từ Bài tập 21c, mô tả mối quan hệ giữa hôn nhân và phản ứng.

23. Thực hiện các thao tác sau với các biến marital và response.

    Một. Xây dựng một bảng dự phòng, cẩn thận để có các biến chính xác đại diện cho các hàng và cột. Báo cáo số lượng và phần trăm cột.

    b. Mô tả những gì bảng dự phòng đang nói với bạn.

Machine Translated by Google

24. Lặp lại bài tập trước, lần này báo cáo phần trăm hàng. Giải thích sự khác biệt giữa cách giải thích của bảng này và bảng dự phòng trước đó.

25. Tạo các đồ thị sau. Sức mạnh của mỗi biểu đồ là gì? Yếu đuối?

    Một. Biểu đồ thời lượng.

    b. Biểu đồ thời lượng, với lớp phủ phản hồi.

    c. Biểu đồ thời lượng được chuẩn hóa, với lớp phủ phản hồi.

26. Sử dụng đồ thị từ Bài tập 25c, mô tả mối quan hệ giữa thời lượng và phản ứng.

27. Kiểm tra các biểu đồ thời lượng không chuẩn hóa và chuẩn hóa , với lớp phủ phản hồi. Xác định (các) điểm giới hạn trong khoảng thời gian, phân tách các giá trị phản hồi thấp từ những giá trị cao. Xác định một biến phân loại mới, duration_binned, sử dụng các điểm giới hạn mà bạn đã xác định.

28. Cung cấp những điều sau đây. Mô tả từng kết quả.

    Một. Bảng dự phòng của duration_binned có phản hồi, có số lượng và cột tỷ lệ phần trăm.

    b. Biểu đồ thanh không chuẩn hóa của duration_binned, với lớp phủ phản hồi .

    c. Biểu đồ thanh chuẩn hóa của duration_binned, với lớp phủ phản hồi .

29. Lập bảng dự phòng công việc có phản hồi, có số lượng và cột tỷ lệ phần trăm.

30. Tham khảo bài tập trước, hãy làm như sau:

    Một. Kết hợp các danh mục công việc theo tỷ lệ phần trăm phản hồi sau : 0<10, 10<25, 25<33. Đặt tên cho biến mới là job2.

    b. Cung cấp một bảng dự phòng của job2 với phản hồi, với số lượng và cột phần trăm lứa tuổi. Mô tả những gì bạn nhìn thấy.

    c. Cung cấp biểu đồ đã chuẩn hóa của công việc 2 kèm theo phản hồi. Mô tả mối quan hệ.

Đối với Bài tập 31–36, hãy làm việc với bộ dữ liệu ngũ cốc . Sử dụng Python hoặc R để giải quyết từng vấn đề.

31. Tạo biểu đồ cột của biến Manuf với Type overlay.

32. Tạo biểu đồ thanh được chuẩn hóa của biến Manuf với Type overlay.

33. Tạo bảng dự phòng Manuf và Type.

34. Tạo biểu đồ Calo với lớp phủ Manuf .

35. Tạo biểu đồ Calo chuẩn hóa với lớp phủ Manuf .

36. Bin biến Calo sử dụng bin cho 0–90, 90–110, và hơn 110 calo. Tạo một biểu đồ thanh của biến số lượng calo được kiểm soát với lớp phủ Manuf .

Đối với Bài tập 37–60, hãy sử dụng bộ dữ liệu adult_ch3_training .

Đối với Bài tập 37–40, chúng tôi trình bày một ví dụ về lý do tại sao không nên xóa các giá trị ngoại lệ ở giai đoạn EDA, vì nó dẫn đến việc thay đổi đặc tính của tập dữ liệu.

Machine Translated by Google

37. Xem xét lỗ vốn. Xác định các ngoại lệ trong tổn thất vốn bằng phương pháp điểm Z. Có bao nhiêu ngoại lệ?

38. Xây dựng biểu đồ thanh Thu nhập cho các bản ghi ngoại lệ này.

39. Xây dựng biểu đồ thanh Thu nhập cho toàn bộ tập dữ liệu adult_ch3_training , không có bỏ qua các ngoại lệ.

40. So sánh biểu đồ thanh của bạn từ hai bài tập trước. Mô tả sự khác biệt giữa hai biểu đồ thanh. Mô tả sự thay đổi đối với đặc tính của tập dữ liệu sẽ xảy ra nếu chúng ta xóa các bản ghi ngoại lệ này. Nêu kết luận của bạn về việc loại bỏ những kẻ nói dối ở giai đoạn EDA.

41. Lấy cờ cho lỗ vốn, được gọi là cờ lỗ vốn, bằng 0 khi lỗ vốn bằng 0, và 1 nếu không. Cung cấp một biểu đồ thanh của cờ vốn-lỗ-cờ.

42. Lặp lại bài tập trước đối với cờ vốn-tăng vốn.

43. Xây dựng bảng dự phòng cờ vốn-lỗ- thu nhập. Bao gồm số lượng và phần trăm cột. Mô tả rõ ràng ảnh hưởng của bất kỳ khoản lỗ vốn nào đối với Thu nhập.

44. Xây dựng bảng dự phòng cờ vốn-tăng-lợi so với Thu nhập. Bao gồm số lượng và phần trăm cột. Mô tả rõ ràng tác động của việc có bất kỳ khoản lãi vốn nào đối với Thu nhập.

45. Để chuẩn bị cho công việc tiếp theo, hãy đổi tên lớp công nhân thành workclass-old, marital-status thành marital-status-old, và ococation thành occupation-old.

46. Xây dựng một bảng dự phòng với thu nhập cho các hàng và workclass-old cho các cột, yêu cầu số lượng và tỷ lệ phần trăm của cột.

47. Tham khảo bảng dự phòng từ bài tập trước, hãy làm như sau:

    Một. Cung cấp một lý do hợp lý trong một câu về lý do tại sao chúng ta nên kết hợp không bao giờ làm việc và không trả lương thành không trả tiền.

    b. Cung cấp một lý do hợp lý trong một câu về lý do tại sao chúng tôi có thể kết hợp chính quyền địa phương và chính phủ tiểu bang thành bang-địa phương-gov.

48. Thực hiện các thay đổi được đề cập trong bài tập trước, cùng với việc thay đổi “?” đến "không biết." Gọi biến mới là workclass. Lập bảng dự phòng với thu nhập cho các hàng và lớp làm việc cho các cột, với số lượng và tỷ lệ phần trăm của cột. Mô tả bảng của bạn bằng cách sử dụng một vài câu.

49. Lập một bảng dự phòng với thu nhập cho các hàng và tình trạng hôn nhân cho các cột, yêu cầu số đếm và tỷ lệ phần trăm của cột.

50. Tham khảo bảng dự phòng từ câu trước, cung cấp hai câu lý do tại sao chúng ta nên kết hợp Vợ chồng-AF-vợ chồng và Vợ chồng-công dân vào danh mục mới Đã kết hôn và kết hợp các trạng thái khác vào danh mục mới Khác.

51. Thực hiện các thay đổi được đề cập trong bài tập trước. Xây dựng một bảng dự phòng với thu nhập cho các hàng và tình trạng hôn nhân cho các cột, với số lượng và tỷ lệ phần trăm của cột. Mô tả bảng của bạn bằng cách sử dụng một vài câu.

52. Lập một bảng dự phòng với thu nhập cho các hàng và nghề nghiệp cũ cho các cột, yêu cầu số lượng và tỷ lệ phần trăm của cột.

Machine Translated by Google

53. Cung cấp một câu cho mỗi lý do tại sao chúng ta nên thực hiện các kết hợp sau:

    Một. Exec-managerial và Prof-specialist vào danh mục mới Exec/prof.

    b. Các nghề có tỷ lệ phần trăm thu nhập >50 nghìn vào danh mục mới Trung cấp.

    c. Các nghề còn lại vào danh mục mới Cấp thấp.

    d. Gấp danh mục không xác định vào danh mục Cấp thấp .

54. Thực hiện các thay đổi được đề cập trong bài tập trước. Gọi biến occupa tion mới. Xây dựng một bảng dự phòng với thu nhập cho các hàng và nghề nghiệp cho các cột, với số lượng và tỷ lệ phần trăm của cột. Mô tả bảng của bạn bằng cách sử dụng một vài câu.

55. Thực hiện các thao tác sau đối với biến giáo dục .

    Một. Cung cấp biểu đồ giáo dục không chuẩn hóa và chuẩn hóa , với thu nhập lớp phủ.

    b. Cung cấp một câu mô tả về mối quan hệ và một câu khác về kỳ vọng của bạn về tính hữu ích của giáo dục trong việc dự đoán thu nhập.

56. Thực hiện các thao tác sau đối với biến tuổi .

    Một. Cung cấp biểu đồ tuổi không chuẩn hóa và chuẩn hóa , với lớp phủ thu nhập .

    b. Cung cấp mô tả một câu về mối quan hệ.

    c. Cung cấp một lý do hợp lý khác cho cách sắp xếp sau: tuổi <30, tuổi 30-60, tuổi >60.

57. Thực hiện tính năng tạo thùng được đề cập trong bài tập trước, bằng cách lấy một biến mới, age_binned.

    Một. Cung cấp biểu đồ thanh được chuẩn hóa về age_binned với lớp phủ thu nhập .

    b. Giải thích biểu đồ thanh bằng cách sử dụng một câu.

58. Đưa ra phân tích sau về yếu tố dự đoán giới tính .

    Một. Biểu đồ thanh không chuẩn hóa về giới tính, với lớp phủ thu nhập.

    b. Biểu đồ thanh giới tính được chuẩn hóa với lớp phủ thu nhập.

    c. Giải thích biểu đồ thanh được chuẩn hóa bằng một câu.

59. Dựng như sau:

    Một. Biểu đồ nghề nghiệp dạng thanh không chuẩn hóa với lớp phủ giới tính .

    b. Biểu đồ thanh nghề nghiệp được chuẩn hóa với lớp phủ giới tính . Mô tả mối quan hệ.

60. Xây dựng một bảng dự phòng với giới tính cho các hàng và nghề nghiệp cho các cột, với số lượng và tỷ lệ phần trăm của cột. So sánh bảng dự phòng với biểu đồ thanh chuẩn hóa.

Machine Translated by Google

chương 5

CHUẨN BỊ LÀM MẪU DỮ LIỆU

5.1 CÂU CHUYỆN CHO ĐẾN NAY

Để tóm tắt lại tiến trình của chúng tôi cho đến nay, chúng tôi đang thực hiện theo Phương pháp khoa học dữ liệu.

1. Trong Chương 3, chúng ta đã thảo luận về tầm quan trọng của việc Hiểu vấn đề Giai đoạn.
2. Cũng trong Chương 3, chúng tôi đã giải quyết một số vấn đề liên quan đến Chuẩn bị Dữ liệu Giai đoạn.
3. Trong Chương 4, chúng ta đã đề cập đến một số chủ đề quan trọng trong Giai đoạn Phân tích Dữ liệu Khám phá.
4. Bây giờ, ở đây trong Chương 5, chúng ta đã sẵn sàng giải quyết Giai đoạn Thiết lập.

Giai đoạn thiết lập bao gồm một số nhiệm vụ rất quan trọng phải được hoàn thành trước khi chúng tôi có thể bắt đầu mô hình hóa dữ liệu của mình. Bao gồm các:

- Phân vùng dữ liệu
- Xác thực phân vùng dữ liệu
- Cân bằng dữ liệu
- Thiết lập hiệu suất mô hình cơ sở

Chúng ta lần lượt đề cập đến từng chủ đề này trong chương này.

5.2 PHÂN VÙNG DỮ LIỆU

Phương pháp khoa học dữ liệu không sử dụng mô hình suy luận thống kê trong đó quá trình khái quát hóa được thực hiện từ một mẫu thành một tổng thể. Có hai lý do cho việc này.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

1. Áp dụng suy luận thống kê cho các cỡ mẫu khổng lồ gặp phải trong khoa học dữ liệu có xu hướng dẫn đến ý nghĩa thống kê, ngay cả khi kết quả không có ý nghĩa thực tế.

2. Trong mô hình thống kê, nhà thống kê có một giả thuyết tiên nghiệm trong đầu, trong khi Phương pháp khoa học dữ liệu không yêu cầu giả thuyết tiên nghiệm như vậy, thay vào đó tự do tìm kiếm dữ liệu để tìm kết quả khả thi.

Do thiếu các giả thuyết tiên nghiệm, các nhà khoa học dữ liệu cần cẩn thận với việc nạo vét dữ liệu, theo đó các kết quả giả mạo ảo được phát hiện, chỉ do sự thay đổi ngẫu nhiên chứ không phải do tác động thực. Khoa học dữ liệu tránh việc nạo vét dữ liệu thông qua quá trình xác thực chéo, một kỹ thuật để đảm bảo rằng kết quả có thể khái quát hóa đối với một tập dữ liệu độc lập, vô hình. Các phương pháp phổ biến nhất là xác thực chéo gấp đôi và xác thực chéo k-fold. Trong xác thực chéo kép, dữ liệu được phân vùng, sử dụng phép gán ngẫu nhiên, thành tập dữ liệu huấn luyện và tập dữ liệu thử nghiệm (còn gọi là tập dữ liệu holdout).

Các bản ghi tập huấn luyện đã hoàn tất, nhưng các bản ghi tập kiểm tra nên bỏ qua biến mục tiêu (tạm thời). Vì vậy, các mô hình khoa học dữ liệu tìm hiểu về các mẫu và xu hướng trong dữ liệu bằng cách sử dụng tập dữ liệu huấn luyện. Các mô hình này được áp dụng cho tập hợp thử nghiệm, trong đó chúng đưa ra dự đoán cho các giá trị tạm thời chưa biết của biến mục tiêu. Những dự đoán này sau đó được đánh giá dựa trên các giá trị đích thực (hiện đã được khôi phục), sử dụng các biện pháp đánh giá như tỷ lệ lỗi tổng thể hoặc giá trị trung bình-lỗi bình phương. Theo cách này, xác thực chéo bảo vệ chống lại các kết quả giả mạo, vì rất khó có thể tìm thấy cùng một biến thể ngẫu nhiên trong cả tập dữ liệu huấn luyện và kiểm tra.

Kích thước của các phân vùng khác nhau tùy thuộc vào kích thước và độ phức tạp của tập dữ liệu. Đối với các tập dữ liệu rất phức tạp, chẳng hạn như mô hình mạng thần kinh cần tìm hiểu về nhiều mối quan hệ phi tuyến tính trong dữ liệu, thì nên sử dụng nhiều bản ghi đào tạo hơn, chẳng hạn như 75–90% dữ liệu gốc. Ngoài ra, nếu tập dữ liệu rất lớn, sẽ thuận tiện khi có nhiều bản ghi hơn trong tập huấn luyện. Mặt khác, đối với các tập dữ liệu nhỏ hơn hoặc ít phức tạp hơn, người ta nên giữ lại đủ hồ sơ để đánh giá chính xác, sao cho các tập huấn luyện chỉ chứa 50–67% dữ liệu gốc.

5.2.1 Cách phân vùng dữ liệu trong Python

Nhập các gói sau:

```
nhập gấu trúc dưới dạng pd
từ sklearn.model_selection nhập train_test_split
nhập ngẫu nhiên
```

Đọc trong tập dữ liệu bank_additional và đặt tên là bank.

```
ngân hàng = pd.read_csv("C:/.../bank-additional.csv")
```

Để phân vùng tập dữ liệu, chúng ta sẽ sử dụng lệnh train_test_split().

Machine Translated by Google


```
bank_train, bank_test = train_test_split(ngân hàng, test_size = 0,25,
random_state = 7)
```

Lệnh tạo hai bộ dữ liệu, bank_train và bank_test. Mặc dù tên của các tập dữ liệu là tùy ý, nhưng tập dữ liệu thử nghiệm luôn là tập thứ hai được tạo.
Đầu vào đầu tiên, ngân hàng, chỉ định rằng chúng tôi đang phân vùng tập dữ liệu ngân hàng và đầu vào test_size = 0,25 cho biết rằng 25% tập dữ liệu ngân hàng phải nằm trong tập dữ liệu thử nghiệm, trong khi 75% còn lại phải nằm trong dữ liệu huấn luyện bộ. trạng thái ngẫu nhiên đầu vào đặt hạt giống cho trình tạo số ngẫu nhiên sẽ phân chia ngẫu nhiên dữ liệu thành tập dữ liệu huấn luyện và kiểm tra. Bản thân giá trị đầu vào là tùy ý. Điều quan trọng là chỉ định hạt giống và sử dụng cùng một số đó khi bạn muốn sao chép kết quả của mình. Đặt hạt giống ngẫu nhiên sẽ đảm bảo bạn sẽ nhận được câu trả lời giống như trước đây.

Để xác nhận rằng tập dữ liệu đã được phân vùng chính xác, bạn có thể so sánh hình dạng của tập dữ liệu gốc, huấn luyện và thử nghiệm bằng cách sử dụng tính năng hình dạng.

```
ngân hàng.shape
bank_train.shape
bank_test.shape
```

Các số đầu tiên được cung cấp từ bank_train.shape và bank_test.shape đầu ra phải tổng bằng số đầu tiên của đầu ra bank.shape . Ngoài ra, số đầu tiên từ đầu ra bank_test.shape phải bằng khoảng 25% số đầu tiên từ đầu ra bank.shape .

5.2.2 Cách phân vùng dữ liệu trong R

Đọc trong ngân hàng- dữ liệu bổ sung được đặt làm ngân hàng. Tiếp theo, chúng ta cần đặt "hạt giống" cho trình tạo số ngẫu nhiên mà chúng ta sẽ sử dụng sau này trong phần này.

```
set.seed(7)
```

Số đầu vào cho lệnh set.seed() là tùy ý. Tuy nhiên, nếu bạn muốn chạy lại mã và nhận được các kết quả ngẫu nhiên giống nhau, thì phần khởi đầu (bất kể đó là gì) phải khớp với phần khởi tạo được sử dụng trong lần chạy đầu tiên. Trong trường hợp của chúng tôi, hạt giống là bảy, 7.
Để chuẩn bị phân vùng dữ liệu, chúng tôi xác định có bao nhiêu bản ghi trong tập dữ liệu

```
n <- dim(ngân hàng)[1]
```

Việc sử dụng lệnh dim() và thông số kỹ thuật [1] bổ sung của nó giống như trong các chương trước.
Khi chúng tôi có số lượng bản ghi, n, chúng tôi có thể xác định bản ghi nào sẽ nằm trong tập dữ liệu huấn luyện thông qua trình tạo số ngẫu nhiên.

```
train_ind <- runif(n) < 0,75
```

Machine Translated by Google

Lệnh runif() rút ngẫu nhiên các số từ 0 đến 1, mỗi số có xác suất bằng nhau. Đầu vào n sẽ tạo n số như vậy, một số cho mỗi bản ghi trong tập dữ liệu. Điều kiện < 0,75 sẽ xem xét từng số được tạo bởi runif() và xem liệu số đó cao hơn hay thấp hơn 0,75. Nếu số dưới 0,75, giá trị TRUE được trả về, trong khi nếu số trên 0,75, giá trị FALSE được trả về. Bạn có thể tự chạy train_ind để xem loạt TRUE

và các giá trị FALSE . Mặc dù các số được tạo bởi runif() mỗi lần sẽ khác nhau, nhưng trung bình khoảng 75% sẽ là TRUE.

Bây giờ chúng ta có một loạt giá trị TRUE và FALSE , chúng ta sẽ sử dụng chúng để tạo tập dữ liệu huấn luyện và kiểm tra. Chúng tôi sử dụng ký hiệu ngoặc để chia tập dữ liệu ngân hàng thành hai nửa bằng cách chỉ định các hàng quan tâm cho từng phân vùng. Hãy nhớ ký hiệu dấu ngoặc chỉ định các hàng quan tâm trước dấu phẩy và các cột quan tâm sau dấu phẩy.

```
bank_train <- ngân hàng[ train_ind, ]
bank_test <- ngân hàng[ !train_ind, ]
```

Bằng cách chạy bank[train_ind,] chúng tôi chỉ tập hợp con những bản ghi của ngân hàng có giá trị train_ind bằng TRUE. Vì khoảng 75% giá trị trong train_ind bằng TRUE nên khoảng 75% bản ghi từ ngân hàng sẽ nằm trong tập dữ liệu bank_train . Đối với bank_test, chúng tôi muốn các bản ghi ngân hàng có giá trị train_ind bằng FALSE. Bằng cách sử dụng ngân hàng [!train_ind,] chúng tôi chỉ tập hợp con những bản ghi của ngân hàng có train_ind giá trị không bằng TRUE (trong đó "không" được biểu thị bằng dấu chấm than, !).

Bây giờ chúng tôi có tập dữ liệu thử nghiệm và đào tạo của mình.

5.3 XÁC NHẬN PHẦN MỀM CỦA BẠN

Do tính hợp pháp của toàn bộ Phương pháp khoa học dữ liệu phụ thuộc vào tính hợp lệ của phân vùng, nên điều quan trọng là phải kiểm tra xem tập dữ liệu huấn luyện và tập dữ liệu kiểm tra không khác nhau về mặt hệ thống. Chúng ta có thể làm điều này bằng cách kiểm tra, trên cơ sở từng biến, xem tập huấn luyện và tập kiểm tra có khác nhau hay không. Vì có thể có nhiều biến trong tập dữ liệu nên chúng tôi hạn chế kiểm tra tại chỗ một tập hợp nhỏ các biến được chọn ngẫu nhiên. Tùy thuộc vào các loại biến có liên quan, các thử nghiệm thống kê khác nhau được yêu cầu.

- Đối với một biến số, hãy sử dụng kiểm định t hai mẫu cho sự khác biệt về giá trị trung bình.
- Đối với một biến phân loại có hai lớp, hãy sử dụng kiểm định Z hai mẫu cho sự khác biệt về tỷ lệ.
- Đối với một biến phân loại có nhiều hơn hai lớp, hãy sử dụng phép thử cho sự đồng nhất về tỷ lệ.

Để biết chi tiết về cách thực hiện các thử nghiệm này, vui lòng xem phần trước của chúng tôi.1

1Khai thác dữ liệu và phân tích dự đoán, Phiên bản thứ hai, của Daniel T. Larose và Chantal D. Larose, John Wiley và Sons, Inc., 2015.

Machine Translated by Google

5.4 CÂN BẰNG TẬP DỮ LIỆU ĐÀO TẠO

Trong một số mô hình phân loại, một trong các lớp biến mục tiêu có tần suất tương đối thấp hơn nhiều so với các lớp khác. Trong những trường hợp như vậy, việc cân bằng tập dữ liệu huấn luyện có thể được khuyến nghị. Mục đích của việc cân bằng là cung cấp cho thuật toán phân loại nhiều lựa chọn bản ghi cho mỗi danh mục. Bằng cách này, các thuật toán thuật toán có cơ hội tìm hiểu về tất cả các loại bản ghi, không chỉ những bản ghi có tần suất cao. Chẳng hạn, giả sử 1000 trong số 100.000 giao dịch thẻ tín dụng là gian lận. Một mô hình phân loại có thể đạt được độ chính xác 99% chỉ bằng cách dự đoán "không gian lận" cho mọi giao dịch. Rõ ràng, mô hình này là vô dụng.

Thay vào đó, nhà phân tích nên cân bằng tập huấn luyện để tăng tỷ lệ giao dịch gian lận. Sự cân bằng này đạt được thông qua việc lấy mẫu lại một số bản ghi gian lận (hiếm).

Lấy mẫu lại là quá trình lấy mẫu ngẫu nhiên và thay thế từ một tập dữ liệu. Ví dụ: hiện tại hồ sơ gian lận của chúng tôi chiếm 1% trong tập huấn luyện của chúng tôi. Giả sử chúng ta muốn tăng tỷ lệ này lên 25%. Sau đó, chúng tôi sẽ thêm 32.000 hồ sơ gian lận được lấy mẫu lại, do đó chúng tôi có tổng cộng 33.000 hồ sơ gian lận. Tổng số bản ghi trong tập huấn luyện khi đó sẽ là 100.000+ 33 000

32.000 = 132.000. Chúng tôi đã mong $\frac{'}{132\ 000} = 0\ .25\ \ 25$ % hồ sơ gian lận, như muốn.

Vì vậy, làm cách nào chúng tôi đưa ra con số kỳ diệu là 32.000 bản ghi được lấy mẫu lại? Chúng tôi đã sử dụng phương trình sau:

$$1000\ x + \ = 0\ 2.5\left(100\ 000\ \ + x\right)$$

và giải quyết cho x, số lượng bản ghi bổ sung cần thiết để lấy mẫu lại. Nói chung, phương trình này là

$$\text{hồ sơ} + x\text{p hiếm}\left(x \qquad + \ \right)$$

trong đó, giải quyết cho x cho chúng ta:

$$x = \frac{\text{hồ}\left(\text{sơ p hiếm}\right) -}{1\ \overline{\text{giờ}}\ \text{chiều}}$$

trong đó x là số bản ghi được lấy mẫu lại bắt buộc, p là tỷ lệ mong muốn của các giá trị hiếm trong tập dữ liệu cân bằng, bản ghi là số bản ghi trong tập dữ liệu không cân bằng và hiếm biểu thị số lượng giá trị mục tiêu hiếm hiện tại.

Tập dữ liệu thử nghiệm không bao giờ được cân bằng! Hãy nhớ rằng tập dữ liệu thử nghiệm đại diện cho dữ liệu nắm giữ mà các mô hình chưa từng thấy. Các tập dữ liệu trong thế giới thực chắc chắn sẽ không được cân bằng lại một cách thuận tiện để thuận tiện cho các mô hình phân loại của chúng tôi, vì vậy tập kiểm tra của chúng tôi cũng không nên được cân bằng lại. Hơn nữa, tất cả đánh giá mô hình sẽ được áp dụng cho tập dữ liệu thử nghiệm, nghĩa là các mô hình sẽ được đánh giá trong các điều kiện dữ liệu giống như thế giới thực.

Machine Translated by Google

5.4.1 Cách cân bằng tập dữ liệu huấn luyện trong Python

Trước tiên, chúng tôi xác định có bao nhiêu bản ghi trong bank_train có giá trị ít phổ biến hơn, "có" cho phản hồi, bằng cách sử dụng lệnh value_counts() .

```
bank_train['response'].value_counts()
```

Số lượng câu trả lời "có" sẽ thay đổi tùy thuộc vào phân vùng. Đối với phân vùng sử dụng hạt giống ngẫu nhiên được chỉ định trong mã Python trước đó, có 3089 bản ghi trong tập dữ liệu huấn luyện, với 338 bản ghi có giá trị phản hồi "có" .
Do đó, khoảng 12% tập dữ liệu huấn luyện có giá trị phản hồi "có" .

Giả sử, chúng tôi muốn tăng tỷ lệ phần trăm câu trả lời "có" lên 30%. kể từ khi chúng tôi có p = 0,3, bản ghi = 3089 và hiếm = 338, chúng tôi thu được

$$x = \frac{0.3\,(3089 - 338)}{0.7} = 841$$

Nghĩa là, chúng tôi cần lấy mẫu lại 841 bản ghi có câu trả lời là "có" và thêm chúng vào tập dữ liệu đào tạo của chúng tôi.

Để bắt đầu lấy mẫu lại, chúng tôi cô lập các bản ghi mà chúng tôi muốn lấy mẫu lại.

```
to_resample = bank_train.loc[bank_train['response'] == "yes"]
```

Lệnh loc tập hợp con dữ liệu bank_train dựa trên điều kiện bank_train['response'] == "yes" và lưu tập dữ liệu kết quả dưới tên to_resample.

Tiếp theo, chúng tôi cần lấy mẫu từ hồ sơ quan tâm của chúng tôi

```
our_resample = to_resample.sample(n = 841, thay thế = True)
```

Lệnh sample() lấy ngẫu nhiên các bản ghi từ to_resample, chứa các bản ghi mà chúng ta muốn lấy mẫu lại. Đầu vào n = 841 chỉ định số lượng bản ghi sẽ vẽ, trong khi đầu vào replace = True chỉ định lấy mẫu có thay thế.
Đầu ra là một tập dữ liệu được tạo thành từ 841 bản ghi được lấy lại mẫu ngẫu nhiên này mà chúng tôi lưu dưới tên our_resample.

Cuối cùng, chúng tôi thêm các bản ghi được lấy mẫu lại vào tập dữ liệu đào tạo ban đầu của mình.

```
bank_train_rebal = pd.concat([bank_train, our_resample])
```

Lệnh concat() gắn hai tập dữ liệu bằng cách đặt các hàng chồng lên nhau. Kết quả là một tập dữ liệu duy nhất được tạo thành từ các bản ghi trong cả bank_train và our_resample, được lưu dưới dạng tập dữ liệu riêng dưới tên bank_train_rebal.

Machine Translated by Google


```
In [44]: bank_train_rebal['response'].value_counts()
Out[44]:
no     2751
yes    1179
Name: response, dtype: int64
```

Hình 5.1 Bảng giá trị phản hồi trong Python sau khi cân bằng lại.

Để kiểm tra xem đã thu được phần trăm mong muốn của câu trả lời "có" hay chưa, hãy kiểm tra bảng biến số câu trả lời .

```
bank_train_rebal['response'].value_counts()
```

Bảng kết quả được thể hiện trong Hình 5.1. Hiện có 1179 hồ sơ trong số 3930 với phản hồi "có", chiếm khoảng 30%.

5.4.2 Cách cân bằng tập dữ liệu huấn luyện trong R

Trước tiên, chúng ta hãy tìm hiểu xem có bao nhiêu bản ghi trong tập dữ liệu bank_train có giá trị phản hồi là "có".

```
bảng(bank_train$response)
```

Số lượng câu trả lời "có" sẽ khác nhau đối với mỗi phân vùng. Đối với phân vùng dựa trên hạt giống ngẫu nhiên được sử dụng trong mã R trước đó, có 3103 bản ghi trong tập dữ liệu huấn luyện và 336 trong số chúng có giá trị phản hồi "có". Điều này mang lại cho chúng tôi khoảng 11% tập dữ liệu huấn luyện có phản hồi "có".

Hãy để chúng tôi lấy mẫu lại để tăng tỷ lệ phần trăm câu trả lời "có" lên 30%. Vì chúng tôi có p = 0,3, hồ sơ = 3103 và hiếm = 336, chúng tôi thu được

$$x = \frac{0.3\,(3103 - 336)}{0.7} = 850$$

Nghĩa là, chúng tôi cần lấy mẫu lại 850 bản ghi có câu trả lời là "có" và thêm chúng vào tập dữ liệu đào tạo của chúng tôi.

Đầu tiên, chúng tôi xác định các chỉ số bản ghi mà chúng tôi muốn lấy mẫu lại bằng cách sử dụng which () yêu cầu.

```
to.resample <- which(bank_train$response == "yes")
```

Lệnh which() trả về số hàng tương ứng với các bản ghi đáp ứng điều kiện đã chỉ định. Trong trường hợp của chúng tôi, chúng tôi muốn số hàng của những bản ghi có giá trị của y là "có", vì vậy điều kiện của chúng tôi là bank_train$response ==

Machine Translated by Google

"Đúng". Chúng tôi lưu các số bản ghi dưới dạng to.resample, vì đây là những giá trị mà chúng tôi muốn lấy mẫu lại.

Tiếp theo, chúng tôi lấy mẫu ngẫu nhiên từ các giá trị trong to.resample.

```
our.resample <- sample(x = to.resample, size = 850, replace = TRUE)
```

Đầu vào x = to.resample chỉ định rằng chúng tôi muốn lấy mẫu từ chuỗi chỉ số bản ghi mà chúng tôi đã tạo trước đó. Đầu vào size = 850 chỉ định số lượng số sẽ được lấy mẫu lại. Đầu vào thay thế = TRUE báo cho thuật toán lấy mẫu bằng thay thế. Đầu ra là một chuỗi gồm 850 chỉ mục bản ghi, được lấy mẫu từ vectơ to.resample của chúng tôi.

Bây giờ chúng tôi muốn lấy các bản ghi có số bản ghi là những bản ghi trong tệp của chúng tôi. lấy mẫu lại.

```
our.resample <- bank_train[our.resample, ]
```

Việc sử dụng ký hiệu ngoặc thực hiện công việc cho chúng tôi. Số lượng bản ghi trong tập dữ liệu mới our.resample là 850, đây là số lượng bản ghi mà chúng tôi muốn lấy mẫu lại.

Cuối cùng, chúng ta cần thêm các bản ghi đã được lấy mẫu lại vào tập dữ liệu đào tạo ban đầu của mình.

```
train_bank_rebal <- rbind(bank_train, our.resample)
```

Lệnh rbind() là viết tắt của “row bind”, nghĩa là nối hai tập dữ liệu bằng cách đặt các hàng chồng lên nhau. Bằng cách sử dụng rbind() trên dữ liệu đào tạo ban đầu bank_train và 850 bản ghi được lấy mẫu lại trong our.resample, chúng tôi đã tạo tập dữ liệu cân bằng lại mà chúng tôi đặt tên là train_bank_rebal.

Để xác nhận rằng việc lấy mẫu lại đã cung cấp số lượng bản ghi hiếm như mong muốn, hãy xem bảng giá trị phản hồi trong tập dữ liệu đã cân bằng lại của chúng tôi.

```
t.v1 <- table(train_bank_rebal$response)
t.v2 <- rbind(t.v1, round(prop.table(t.v1), 4))
colnames(t.v2) <- c("Phản hồi = Không", "Phản hồi = Có");
rownames(t.v2) <- c("Count", "Proportion")
t.v2
```

Kết quả được thể hiện trong hình 5.2. Phần trăm câu trả lời “có” xấp xỉ 30%, đây là phần trăm các bản ghi hiếm mà chúng tôi muốn.

	Phản hồi = Không	có phản hồi = Có
Đếm	2767.0	1186.0
Tỷ lệ	0,7	0,3

Hình 5.2 Bảng giá trị đáp ứng R sau khi cân bằng lại.

Machine Translated by Google

5.5 THIẾT LẬP HIỆU SUẤT MÔ HÌNH CƠ SỞ

Trước khi đánh giá hiệu suất của mô hình, trước tiên, các nhà khoa học dữ liệu nên hiệu chỉnh kết quả dựa trên một số hiệu suất của mô hình cơ sở. Ví dụ: trong trường hợp gian lận ở trên, giả sử chúng ta đã phát triển một mô hình phát hiện gian lận phức tạp báo cáo độ chính xác 98%. Nghe có vẻ ấn tượng, cho đến khi chúng ta nhớ rằng một mô hình "tất cả đều tiêu cực" chỉ phân loại tất cả các bản ghi là không gian lận sẽ có tỷ lệ chính xác là 99%.
Nếu không so sánh với đường cơ sở, khách hàng của chúng tôi không thể xác định liệu kết quả của chúng tôi có tốt hay không.

Chúng tôi cung cấp hai mô hình cơ sở sau đây cho trường hợp phân loại nhị phân.

Mô hình cơ sở để phân loại nhị phân

Đặt một trong các lớp đích nhị phân biểu thị dương và lớp còn lại biểu thị âm.

Đặt p đại diện cho tỷ lệ các bản ghi tích cực trong dữ liệu.

- Tất cả Mô hình Tích cực. Chỉ định tất cả các dự đoán là tích cực.
  - Độ chính xác của mô hình này sẽ là p.
- Tất cả các mô hình tiêu cực. Chỉ định tất cả các dự đoán là tiêu cực.
  - Độ chính xác của mô hình này sẽ là 1–p.

Ví dụ: trong trường hợp gian lận của chúng tôi, p = 0,01 hoặc 1% hồ sơ gian lận, trong đó chúng tôi để hồ sơ gian lận đại diện cho giá trị dương. Sau đó, Mô hình Tất cả Tích cực sẽ có độ chính xác 0,01 và Mô hình Tất cả Tiêu cực sẽ có độ chính xác 0,99. Bất kỳ mô hình nào chúng tôi phát triển sẽ cần phải vượt qua độ chính xác 99% này để được coi là hữu ích.

Chúng ta có thể mở rộng điều này cho trường hợp k-nary, k ≥ 3, như sau:

Mô hình cơ sở cho phân loại k-nary

Giả sử có k lớp cho biến trả lời, C1 , C2 , ., Ck .
Đặt pi đại diện cho tỷ lệ các bản ghi lớp Ci trong dữ liệu, i = 1, ., k.

- Hạng Mục Mô Hình Lớn Nhất. Chỉ định tất cả các dự đoán là thuộc về lớn nhất loại.
  - Độ chính xác của mô hình này là pmax, số pi lớn nhất .

Ví dụ: nếu tập huấn luyện của bạn có 30% Đảng viên Đảng Dân chủ, 30% Đảng viên Đảng Cộng hòa và 40% Đảng viên Độc lập, thì mô hình cơ sở sẽ chỉ định tất cả các bản ghi cho Đảng viên Độc lập và có độ chính xác là 40%.

2Thật vậy, nếu chúng ta đang sử dụng độ chính xác làm phương pháp chọn mô hình tốt nhất. Điều này thường không phải là trường hợp. Xem Chương 7.

Machine Translated by Google

Vì vậy, loại mô hình cơ sở nào chúng ta nên sử dụng cho các bài toán ước lượng? Đối với hồi quy, người ta có thể chỉ cần so sánh các ước tính của chúng tôi với mô hình yy , trong đó ước tính cho mỗi phản hồi của bản ghi chỉ đơn giản là phản hồi trung bình. Nhưng đây thực sự là ngưỡng quá thấp, vì hầu hết mọi mô hình hồi quy chắc chắn sẽ đánh bại ngưỡng đó. Lưu ý rằng mô hình yy cơ sở hoàn toàn bỏ qua lượng thông tin phong phú nằm trong các yếu tố dự đoán. Thay vào đó, chúng ta có thể hỏi một câu hỏi chẳng hạn như, "Một chuyên gia về chủ đề sẽ coi một lỗi dự đoán tầm thường là gì?" Ví dụ: nếu chúng tôi là một công ty cho vay cố gắng ước tính số tiền khách hàng thế chấp của chúng tôi có thể chi trả, chúng tôi có thể nói rằng chúng tôi có thể sống với một mô hình thường giảm 50.000 đô la, do đó sẽ có một lỗi dự đoán tầm thường. Sau đó, khi chúng tôi xây dựng các mô hình hồi quy của mình, chúng tôi cần các giá trị của s, sai số chuẩn của ước tính, thấp hơn 50.000 đô la.3

Tất nhiên, điểm chuẩn tối ưu để hiệu chỉnh bất kỳ mô hình nào là so sánh nó với hiệu suất của mô hình tiêu chuẩn vàng hiện tại, như được cung cấp bởi tài liệu (hoặc các mô hình kinh doanh độc quyền). Ví dụ: giả sử công ty cho vay của chúng tôi đã thực hiện nghiên cứu cách đây vài năm, trong đó họ đạt được sai số chuẩn là 25.000 đô la. Sau đó, s= $25.000 này là điểm chuẩn cơ bản của chúng tôi để so sánh các mô hình mà chúng tôi xây dựng.

Tóm lại, trong chương này, chúng ta đã học về Giai đoạn Thiết lập của Phương pháp khoa học dữ liệu, bao gồm các bước sau:

- Phân vùng dữ liệu
- Xác thực phân vùng dữ liệu
- Cân bằng dữ liệu
- Thiết lập hiệu suất mô hình cơ sở

Lần tới, chúng ta có thể bắt đầu Giai đoạn Lập mô hình khi chuyển sang Chương 6.

NGƯỜI GIỚI THIỆU

Đối với xuất bản ở trung tâm của gói ngẫu nhiên : M. Matsumoto và T. Nishimura, Mersenne twister: bộ tạo số giả ngẫu nhiên thống nhất được phân phối đều 623 chiều, Giao dịch ACM trên Mô hình hóa và Mô phỏng Máy tính, 8(1), 3–30, Tháng 1 năm 1998.

Chúng tôi sẽ dành nhiều thời gian hơn với gói sklearn ! Để làm quen với nó, hãy xem những điều sau: F. Pedregosa, et al., Scikit-learning: machine learning in Python, Journal of Machine Learning Research, 12, 2825–2830 , 2011.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Bốn nhiệm vụ nào nên được thực hiện trong Giai đoạn Thiết lập?
2. Nêu hai lý do tại sao Phương pháp khoa học dữ liệu không tuân theo thông thường mô hình suy luận thống kê.

3Xem Chương 11.

Machine Translated by Google

3. Mô tả nạo vét dữ liệu là gì và tại sao các nhà khoa học dữ liệu cần tránh nó.

4. Làm thế nào để các nhà khoa học dữ liệu tránh việc nạo vét dữ liệu?

5. Mô tả sự khác biệt giữa tập dữ liệu huấn luyện và tập dữ liệu kiểm tra.

6. Khi xác thực phân vùng, nhà khoa học dữ liệu có cần kiểm tra mọi trường không?

7. Khi xác thực một phân vùng, phép kiểm tra thống kê nào được sử dụng cho một biến số?

8. Cân bằng là gì? Tại sao nó được sử dụng?

9. Mô tả ý nghĩa của việc lấy mẫu lại.

10. Khi nào nên cân bằng bộ dữ liệu thử nghiệm?

11. Tại sao thiết lập hiệu suất mô hình cơ bản lại quan trọng?

12. Mô tả hai mô hình cơ sở để phân loại nhị phân.

13. Đúng hay sai: không có mô hình cơ sở cho phân loại k-nary.

14. Điểm chuẩn tối ưu để hiệu chỉnh hiệu suất mô hình của bạn là gì?

LÀM VIỆC VỚI DỮ LIỆU

Đối với Bài tập 15–20, hãy làm việc với tập dữ liệu bank_additional . Sử dụng Python hoặc R để giải quyết từng vấn đề.

15. Phân vùng tập dữ liệu, sao cho 75% bản ghi được đưa vào tập dữ liệu huấn luyện và 25% được đưa vào tập dữ liệu thử nghiệm. Sử dụng biểu đồ thanh để xác nhận tỷ lệ của bạn.

16. Xác định tổng số bản ghi trong tập dữ liệu huấn luyện và có bao nhiêu bản ghi trong tập dữ liệu huấn luyện tập dữ liệu huấn luyện có "có" cho một giá trị biến phản hồi .

17. Sử dụng câu trả lời của bạn từ bài tập trước để tính xem bạn cần lấy mẫu lại bao nhiêu bản ghi phản hồi "có" để có 20% tập dữ liệu được cân bằng lại có giá trị phản hồi "có".

18. Thực hiện tái cân bằng được mô tả trong bài tập trước và xác nhận rằng 20% bản ghi trong tập dữ liệu đã cân bằng lại có giá trị phản hồi "có".

19. Chúng ta có nên cân bằng tập dữ liệu thử nghiệm mà bạn đã tạo ở trên không? Giải thích lý do tại sao và tại sao không.

20. Chúng tôi sử dụng mô hình cơ sở nào để so sánh hiệu suất mô hình phân loại của chúng tôi? Giá trị nào mà mô hình cơ sở này gán tất cả các dự đoán? Độ chính xác của mô hình cơ sở này là gì?

PHÂN TÍCH THỰC TẾ

Đối với Bài tập 21–28, hãy làm việc với bộ dữ liệu dành cho người lớn .

21. Phân vùng tập dữ liệu sao cho 50% bản ghi được đưa vào tập dữ liệu huấn luyện và 50% được đưa vào tập dữ liệu kiểm tra. Sử dụng biểu đồ thanh để xác nhận tỷ lệ của bạn.

22. Xác định tổng số bản ghi trong tập dữ liệu huấn luyện và có bao nhiêu bản ghi trong tập dữ liệu huấn luyện tập dữ liệu huấn luyện có giá trị thu nhập >50 K.

Machine Translated by Google

23. Sử dụng câu trả lời của bạn từ bài tập trước để tính xem bạn cần lấy mẫu lại bao nhiêu bản ghi có thu nhập >50K để có 35% tập dữ liệu được cân bằng lại có thu nhập >50K.

24. Thực hiện tái cân bằng được mô tả trong bài tập trước và xác nhận rằng 35% bản ghi trong tập dữ liệu đã cân bằng lại có thu nhập >50K.

25. Chúng tôi sử dụng mô hình cơ sở nào để so sánh hiệu suất mô hình phân loại của chúng tôi? Giá trị nào mà mô hình cơ sở này gán tất cả các dự đoán? Độ chính xác của mô hình cơ sở này là gì?

26. Xác thực phân vùng của bạn bằng cách thực hiện kiểm tra Z hai mẫu để tìm sự khác biệt về phương tiện cho tuổi trung bình trong tập huấn luyện so với tuổi trung bình trong tập kiểm tra.4

27. Xác thực phân vùng của bạn bằng cách thực hiện kiểm tra Z hai mẫu để biết sự khác biệt về tỷ lệ đối với tỷ lệ >50K bản ghi trong tập huấn luyện so với tỷ lệ >50K bản ghi trong tập kiểm tra.

Đối với Bài tập 28–34, hãy làm việc với bộ dữ liệu rời bỏ .

28. Phân vùng tập dữ liệu sao cho 67% bản ghi được đưa vào tập dữ liệu huấn luyện và 33% được đưa vào tập dữ liệu kiểm tra. Sử dụng biểu đồ thanh để xác nhận tỷ lệ của bạn.

29. Xác định tổng số bản ghi trong tập dữ liệu huấn luyện và có bao nhiêu bản ghi trong tập dữ liệu huấn luyện tập dữ liệu huấn luyện có giá trị rời là true.

30. Sử dụng các câu trả lời của bạn từ bài tập trước để tính xem bạn cần lấy mẫu lại bao nhiêu bản ghi số lần rời bỏ thực tế để có 20% tập dữ liệu được cân bằng lại có giá trị số lần rời bỏ thực sự.

31. Thực hiện tái cân bằng được mô tả trong bài tập trước và xác nhận rằng 20% các bản ghi trong tập dữ liệu được cân bằng lại có giá trị rời thực sự.

32. Chúng tôi sử dụng mô hình cơ sở nào để so sánh hiệu suất mô hình phân loại của chúng tôi? Giá trị nào mà mô hình cơ sở này gán tất cả các dự đoán? Độ chính xác của mô hình cơ sở này là gì?

33. Xác thực phân vùng của bạn bằng cách kiểm tra sự khác biệt về số phút trung bình trong ngày đối với quá trình đào tạo tập so với tập kiểm tra.

34. Xác thực phân vùng của bạn bằng cách kiểm tra sự khác biệt về tỷ lệ bản ghi rời bỏ thực sự của tập huấn luyện so với tập kiểm tra.

Đối với Bài tập 35, hãy làm việc với bộ dữ liệu ngũ cốc . Ở đây, chúng tôi đang cố gắng ước tính một mục tiêu số, xếp hạng (xếp hạng dinh dưỡng), dựa trên một tập hợp các yếu tố dự đoán.

35. Chúng tôi sử dụng mô hình đường cơ sở nào để so sánh hiệu quả của mô hình ước lượng? Giá trị nào mà mô hình cơ sở này gán tất cả các dự đoán?

4Chi tiết về cách thực hiện kiểm tra giả thuyết để xác thực phân vùng của bạn có thể được tìm thấy trong Data Mining and Predictive Analytics, Second Edition, của Daniel T. Larose và Chantal D. Larose, John Wiley và Sons, Inc., 2015.

Machine Translated by Google

chương 6

CÂY QUYẾT ĐỊNH

6.1 GIỚI THIỆU VỀ CÂY QUYẾT ĐỊNH

Cho đến nay, chúng ta đã làm quen với bốn giai đoạn đầu tiên của Khoa học dữ liệu phương pháp luận:

1. Giai đoạn hiểu dữ liệu
2. Giai đoạn chuẩn bị dữ liệu
3. Giai đoạn phân tích dữ liệu khám phá
4. Giai đoạn thiết lập.

Cuối cùng, chúng tôi đã sẵn sàng để bắt đầu lập mô hình dữ liệu của mình, trong Giai đoạn lập mô hình. Khoa học dữ liệu cung cấp nhiều phương pháp và thuật toán để mô hình hóa các tập dữ liệu lớn. Chúng ta bắt đầu ở đây với một trong những phương pháp đơn giản nhất: cây quyết định. Trong chương này, chúng ta sẽ làm việc với bộ dữ liệu adult_ch6_training và adult_ch6_test . Chúng được điều chỉnh từ bộ dữ liệu Người lớn từ kho lưu trữ UCI.1 Để đơn giản, chỉ có hai yếu tố dự đoán và mục tiêu được giữ lại, như sau:

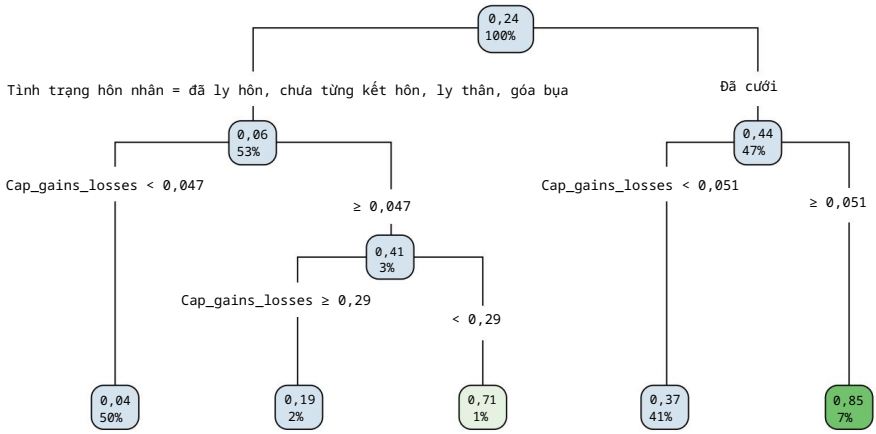
- Tình trạng hôn nhân, một yếu tố dự đoán phân loại với các lớp đã kết hôn, ly hôn, chưa bao giờ-đã kết hôn, ly thân và góa bụa.
- Cap_gains_losses, một công cụ dự báo bằng số, bằng với lãi vốn +|vốn thiệt hại|.
- Thu nhập, một biến mục tiêu phân loại có hai loại, >50k và ≤50k, đại diện cho những cá nhân có thu nhập lớn hơn $50.000 mỗi năm và những người có thu nhập thấp hơn hoặc bằng $50.000 mỗi năm.

Cây quyết định bao gồm một tập hợp các nút quyết định, được kết nối bởi các nhánh, kéo dài xuống từ nút gốc cho đến khi kết thúc ở các nút lá. Bắt đầu từ nút gốc, theo quy ước được đặt ở đầu cây quyết định

1C.L. Blake và CJ Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998 Bộ dữ liệu dành cho người lớn do Ron Kohavi tặng.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google



Hình 6.1 Cây quyết định từ R để phân loại kết quả phản hồi cho người lớn_ch6_ dữ liệu huấn luyện .

sơ đồ, các biến được kiểm tra tại các nút quyết định, với mỗi kết quả có thể dẫn đến một nhánh. Sau đó, mỗi nhánh dẫn đến một nút quyết định khác hoặc một nút lá kết thúc. Hình 6.1 cung cấp một ví dụ về cây quyết định đơn giản, sử dụng thuật toán phân loại và cây hồi quy (GIỎI)2 , áp dụng cho 18.761 bản ghi trong tập dữ liệu adult_ch6_training .

Người ta có thể tưởng tượng tất cả các bản ghi dữ liệu đi vào cây quyết định thông qua nút gốc ở trên cùng của cây và đi xuống qua cây dựa trên các quyết định tại mỗi nút quyết định liên quan đến các giá trị biến. 100% trong nút gốc minh chứng cho điều này. Nút gốc cũng cho chúng ta biết rằng 24% (0,24) bản ghi trong tập dữ liệu adult_ch6_training có thu nhập cao (>50K). Do đó, mỗi nút cho chúng ta biết tỷ lệ các bản ghi có thu nhập cao trong nút, cùng với tỷ lệ phần trăm các bản ghi đạt đến nút đó. Tại sự phân chia nút gốc, GIỎ HÀNG xác định cách phân chia nhị phân hiệu quả nhất có thể bằng cách tách các bản ghi thành hai nhóm, tùy thuộc vào giá trị của chúng đối với tình trạng hôn nhân có thể thay đổi, một nhóm có tình trạng hôn nhân đã kết hôn và nhóm còn lại bao gồm tất cả những người khác. tình trạng hôn nhân: đã ly hôn, chưa từng kết hôn, ly thân và góa bụa. Lưu ý rằng nhóm đã kết hôn có 44% thu nhập cao, trong khi nhóm còn lại chỉ có 6% thu nhập cao. Nhóm đã kết hôn có tỷ lệ hồ sơ thu nhập cao cao gấp bảy lần. Sự khác biệt nổi bật này là lý do tại sao sự phân tách này được chọn làm nút gốc được phân tách bằng thuật toán GIỎ HÀNG. Cũng lưu ý rằng nút gốc phân chia dữ liệu gần một nửa, với 47% đã kết hôn và 53% chưa kết hôn.

Tại nút đã kết hôn , GIỎ HÀNG sau đó thực hiện phân tách thứ hai, dựa trên biến Cap_Gains_Losses. Nếu lãi và lỗ vốn (bình thường hóa tối thiểu-tối đa) vượt quá 0,051, thì 85% hồ sơ có thu nhập cao. Tuy nhiên, nhóm hồ sơ đại diện cho những người đã kết hôn có lãi và lỗ vốn cao chỉ chiếm 7%.

2Leo Breiman, Jerome Friedman, Richard Olshen, và Charles Stone, Cây phân loại và hồi quy, Chapman & Hall/ CRC Press, Boca Raton, FL, 1984.

Machine Translated by Google

của tập dữ liệu huấn luyện, như được hiển thị trong nút lá ở phía dưới bên phải. Mặt khác, những người đã kết hôn nhưng không có lãi và lỗ vốn cao chiếm 41% bộ dữ liệu, nhưng chỉ chứa 37% có thu nhập cao. Hai nút ở phía dưới bên phải là các nút lá vì không có sự phân tách nào được thực hiện trên chúng.

Quay lại các bản ghi từ nút tình trạng hôn nhân không kết hôn , chúng tôi thấy rằng sau đó việc phân chia được thực hiện, cũng dựa trên lãi và lỗ vốn được chuẩn hóa. Các cá nhân chưa kết hôn và không có lãi và lỗ vốn cao chiếm 50% toàn bộ tập dữ liệu đào tạo. Trong số này chỉ có 4% có thu nhập cao. Những người chưa kết hôn có lãi và lỗ vốn cao chỉ chiếm 3% trong bộ dữ liệu, nhưng có tỷ lệ thu nhập cao cao hơn nhiều: 41%. Sự phân chia cuối cùng được thực hiện trên 3% hồ sơ này, điều chỉnh chính xác hơn mức lãi và lỗ vốn, với nhóm cao hơn có thu nhập cao là 71% và nhóm thấp hơn có thu nhập cao là 19%. Cây quyết định ngừng phát triển khi không thể thực hiện phân tách thêm nữa.

Vậy cây quyết định hoạt động như thế nào? Cây quyết định tìm cách tạo ra một tập hợp các nút lá càng "thuần túy" càng tốt, nghĩa là mỗi bản ghi trong một nút lá cụ thể có cùng phân loại. Theo cách này, cây quyết định có thể cung cấp các bài tập phân loại với độ tin cậy cao nhất hiện có.
Tuy nhiên, làm thế nào để đo lường tính đồng nhất, hoặc ngược lại, làm thế nào để đo lường tính không đồng nhất? Chúng ta sẽ xem xét hai trong số nhiều phương pháp để đo độ tinh khiết của nút lá, dẫn đến hai thuật toán hàng đầu để xây dựng cây quyết định:

- Thuật toán GIỎ HÀNG
- Thuật toán C5.0

6.2 CÂY PHÂN LOẠI VÀ HỒI QUY

Phương thức GIỎ HÀNG3 tạo ra các cây quyết định hoàn toàn là nhị phân, chứa chính xác hai nhánh cho mỗi nút quyết định. GIỎ HÀNG phân vùng đệ quy các bản ghi trong tập dữ liệu huấn luyện thành các tập hợp con các bản ghi có giá trị tương tự cho thuộc tính đích. Thuật toán GIỎ HÀNG phát triển cây bằng cách tiến hành cho từng nút quyết định, tìm kiếm toàn diện tất cả các biến có sẵn và tất cả các giá trị phân tách có thể, chọn cách phân tách tối ưu theo Chỉ số Gini (từ Kennedy et al.4) .

Đặt Φ(s|t) là thước đo "độ tốt" của một ứng cử viên được tách s tại nút t, trong đó

$$\Phi(s|t) = 2 P_L P_R \sum_{j=1}^{\#\text{ các lớp học}} \left| P(j|t_L) - P(j|t_R) \right|$$

3Leo Breiman, Jerome Friedman, Richard Olshen, và Charles Stone, Cây phân loại và hồi quy, Chapman & Hall/CRC Press, Boca Raton, FL, 1984.

4Ruby L. Kennedy, Yuchun Lee, Benjamin Van Roy, Christopher D. Reed, và Richard P. Lippman, Giải quyết các vấn đề khai thác dữ liệu thông qua nhận dạng mẫu, Pearson Education, Upper Saddle River, NJ, 1995.

Machine Translated by Google

và ở đâu

$$t_L = \text{nút con trái của nút } t$$

$$t_r = \text{nút con phải của nút } t$$

$$P_L = \frac{\text{số bản ghi tại } t_L}{\text{số bản ghi trong tập huấn luyện}}$$

$$P_r = \frac{\text{số bản ghi tại } t_r}{\text{số bản ghi trong tập huấn luyện}}$$

$$P_j(t_L) = \frac{\text{số hồ sơ lớp } j \text{ tại } t_L}{\text{số bản ghi tại } t}$$

$$P_j(t_r) = \frac{\text{Số lượng hồ sơ lớp } j \text{ tại } t_r}{\text{số bản ghi tại } t}$$

Sau đó, sự phân chia tối ưu là bất kỳ sự phân chia nào tối đa hóa độ đo này Φ(s|t) trên tất cả các lần phân tách có thể có tại nút t. Vì vậy, GIỎI đã xác định sự phân chia nút gốc trong Hình 6.1 là tối đa hóa Φ(s|t) trên tất cả các lần phân tách nút gốc ứng cử viên.

6.2.1 Cách xây dựng cây quyết định GIỎ HÀNG bằng Python

Tải các gói cần thiết và đọc trong tập dữ liệu đào tạo là adult_tr.

```
nhập gấu trúc dưới dạng
pd nhập numpy dưới dạng
np nhập statsmodels.tools.tools dưới dạng stattools từ
sklearn.tree nhập DecisionTreeClassifier, export_ graphviz adult_tr =

pd.read_csv("C:/.../adult_ch6_training")
```

Để đơn giản, chúng ta lưu biến Thu nhập là y.

```
y = adult_tr[['Thu nhập']]
```

Chúng tôi có một biến phân loại, Tình trạng hôn nhân, trong số các yếu tố dự đoán của chúng tôi. Mô hình GIỎ HÀNG được triển khai trong gói sklearn cần các biến phân loại được chuyển đổi thành dạng biến giả. Vì vậy, chúng ta sẽ tạo một loạt các biến giả cho Tình trạng hôn nhân bằng cách sử dụng lệnh categorical() .

```
mar_np = np.array(adult_tr['Marital status']) (mar_cat,
mar_cat_dict) = stattools.categorical(mar_np, drop=True, dictnames =
True)
```

Chúng ta biến biến Tình trạng hôn nhân thành một mảng bằng cách sử dụng mảng(), sau đó sử dụng lệnh categorical() từ gói stattools để tạo một ma trận các biến giả cho mỗi giá trị của Tình trạng hôn nhân. Chúng tôi lưu ma trận và từ điển riêng biệt bằng cách sử dụng (mar_cat, mar_cat_dict).

Machine Translated by Google

Ma trận mar_cat chứa năm cột, một cột cho mỗi danh mục trong biến Tình trạng hôn nhân ban đầu . Mỗi hàng đại diện cho một bản ghi trong tập dữ liệu adult_tr . Mỗi hàng sẽ có 1 trong cột khớp với giá trị mà bản ghi có trong biến Tình trạng hôn nhân ban đầu . Bạn có thể biết cột nào đại diện cho danh mục nào bằng cách kiểm tra mar_cat_dict. Trong trường hợp của chúng tôi, hàng đầu tiên của mar_cat có 1 trong cột thứ ba. Bằng cách kiểm tra mar_cat_dict, chúng tôi biết cột thứ ba đại diện gửi danh mục "Chưa kết hôn". Chắc chắn rồi, bản ghi đầu tiên của adult_tr có

"Chưa kết hôn" làm giá trị biến Tình trạng hôn nhân .

Bây giờ, chúng ta cần thêm các biến giả mới được tạo trở lại vào các biến X.

```
mar_cat_pd = pd.DataFrame(mar_cat)
X = pd.concat((adult_tr[['Cap_Gains_Losses']], mar_cat_
pd), trục = 1)
```

Trước tiên, chúng tôi tạo ma trận mar_cat thành khung dữ liệu bằng cách sử dụng DataFrame() yêu cầu. Sau đó, chúng tôi sử dụng lệnh concat() để đính kèm biến dự báo Cap_Gains_Losses vào khung dữ liệu của các biến giả biểu thị tình trạng hôn nhân. Chúng tôi lưu kết quả dưới dạng X.

Trước khi chúng tôi chạy thuật toán GIỎ HÀNG, hãy lưu ý rằng các cột của X không bao gồm các giá trị khác nhau của biến Tình trạng hôn nhân . Chạy mar_cat_dict để thấy rằng cột đầu tiên dành cho giá trị "Đã ly hôn", cột thứ hai dành cho "Đã kết hôn", v.v. Vì cột đầu tiên của X là Cap_Gains_Losses nên chúng tôi có thể chỉ định tên của từng cột của X.

```
X_names = ["Cap_Gains_Losses", "Đã ly hôn", "Đã kết hôn",
 "Không bao giờ kết hôn",
        "Biệt ly", "Góa chồng"]
```

Nó sẽ giúp chúng ta khi hình dung mô hình GIỎ HÀNG để biết các cấp độ của y .

```
y_names = ["<=50K", ">50K"]
```

Bây giờ, chúng ta đã sẵn sàng để chạy thuật toán GIỎ HÀNG!

```
cart01 = DecisionTreeClassifier(criterion = "gini", max_
leaf_nodes=5).fit(X,y)
```

Để chạy thuật toán GIỎ HÀNG, chúng ta sử dụng lệnh DecisionTreeClassifier() . Lệnh DecisionTreeClassifier() thiết lập các tham số khác nhau cho cây quyết định. Ví dụ: đầu vào criteria = "gini" chỉ định rằng chúng ta đang sử dụng mô hình GIỎ HÀNG sử dụng tiêu chí Gini và đầu vào max_leaf_nodes cắt cây GIỎ HÀNG để có nhiều nhất số lượng nút lá đã chỉ định. Đối với ví dụ này, chúng tôi đã giới hạn cây của mình ở năm nút lá.

Lệnh fit() yêu cầu Python điều chỉnh cây quyết định đã được chỉ định trước đó cho dữ liệu. Các biến dự đoán được đưa ra đầu tiên, tiếp theo là biến mục tiêu. Do đó, hai đầu vào cho fit() là các đối tượng X và y mà chúng ta đã tạo. Chúng tôi lưu cây quyết định dưới dạng cart01.

Machine Translated by Google

Cuối cùng, để có được cấu trúc cây, chúng ta sử dụng lệnh export_graphviz() .

```
export_graphviz(cart01, out_file = "C:/.../cart01.dot", feature_names=X_names,
class_names=y_names)
```

Đầu vào đầu tiên là chính cây quyết định mà chúng tôi đã lưu dưới dạng giỏ hàng01. ra_ _ tệp đầu vào sẽ lưu cấu trúc cây vào vị trí đã chỉ định và đặt tên cho tệp là cart01.dot. Chạy nội dung của tệp thông qua gói graphviz để hiển thị mô hình GIỎ HÀNG. Chỉ định feature_names = X_names và class_names = y_
tên thêm tên biến dự đoán và giá trị biến mục tiêu vào giỏ hàng01.
dot, tăng đáng kể khả năng đọc của nó.

Để có được các phân loại của biến Thu nhập cho mọi biến trong tập dữ liệu huấn luyện, hãy sử dụng lệnh predict() .

```
preIncomeCART = cart01.predict(X)
```

Sử dụng lệnh predict() trên cart01 nói rằng chúng tôi muốn sử dụng mô hình GIỎ HÀNG của mình để thực hiện phân loại. Bao gồm các biến dự đoán X làm đầu vào xác định rằng chúng tôi muốn dự đoán cụ thể cho các bản ghi đó. Kết quả là phân loại, theo mô hình GIỎ HÀNG của chúng tôi, cho mọi bản ghi trong tập dữ liệu huấn luyện. Chúng tôi lưu các dự đoán dưới dạng preIncomeCART.

6.2.2 Cách xây dựng cây quyết định GIỎ HÀNG bằng R

Nhập tập dữ liệu huấn luyện và đặt tên là adult_tr. Sau khi tập dữ liệu được tải vào R, hãy đổi tên “Marital status” thành “maritalStatus” để xóa khoảng trống. Thay đổi này sẽ giúp ích cho code của chúng ta sau này.

```
colnames(adult_tr)[1] <- "maritalStatus"
```

Lệnh colnames() liệt kê tên của từng biến trong tập dữ liệu adult_tr . Tự chạy nó sẽ xuất tên các cột theo thứ tự. Lưu ý rằng cột đầu tiên chứa biến Tình trạng hôn nhân . Sử dụng [1] ở cuối colnames(adult_tr) sẽ cô lập tên biến cho Tình trạng hôn nhân. Sau đó, chúng tôi đổi tên biến bằng cách tạo chuỗi “maritalStatus” và lưu nó dưới dạng biến

tên.

Sau đó, chúng tôi thay đổi cả hai biến phân loại thành các yếu tố.

```
adult_tr$Income <-yếu tố(adult_tr$Income)
adult_tr$maritalStatus <-yếu tố(adult_tr$maritalStatus)
```

Để chạy và trực quan hóa mô hình GIỎ HÀNG, chúng ta cần cài đặt và mở gói cần thiết, rpart và rpart.plot.

```
install.packages(c("rpart", "rpart.plot"))
thư viện(rpart); thư viện (rpart.plot)
```

Machine Translated by Google

Cuối cùng, chúng ta hãy chạy lệnh rpart() để xây dựng mô hình GIỎ HÀNG.

```
giỏ hàng01 <- rpart(công thức = Thu nhập ~ Tình trạng hôn nhân + Giới hạn_
Lãi_Lỗ,
         data=adult_tr, method="class")
```

Đầu vào công thức có cấu trúc Mục tiêu ~ Dự đoán, trong đó tên của các yếu tố dự đoán được phân tách bằng dấu cộng. Đầu vào dữ liệu chỉ định tập dữ liệu mà các biến đến từ. Đầu vào method = "class" chỉ định rằng chúng ta muốn sử dụng mô hình phân loại (GIỎ HÀNG). Cuối cùng, chúng tôi lưu mô hình kết quả là cart01.

Khi mô hình GIỎ HÀNG được xây dựng, chúng ta có thể vẽ mô hình GIỎ HÀNG với giá trị mặc định tùy chọn hiển thị bằng lệnh rpart.plot .

```
rpart.plot(cart01)
```

Lệnh rpart.plot() nhận đầu vào được yêu cầu duy nhất là tên mà bạn đã lưu mô hình GIỎ HÀNG. Vì chúng tôi đã gọi mô hình của chúng tôi là giỏ hàng01, đó là đầu vào của chúng tôi.

Lưu ý rằng cốt truyện được tạo bằng cách sử dụng cài đặt mặc định của rpart.plot() không thay thế cate Hình 6.1. Có những cài đặt nào khác?

```
?rpart.plot
```

Chạy ?rpart.plot để xem xét các tùy chọn hiển thị khác nhau, nằm bên dưới đối số loại và đối số phụ . Hãy thử sử dụng type = 4 để gắn nhãn cho mỗi nhánh với giá trị cụ thể của nó, thay vì có/không ở đầu phần tách; và extra = 2 để thêm tỷ lệ phân loại chính xác cho mỗi nút.

```
rpart.plot(cart01, type = 4, extra = 2)
```

Kết quả được thể hiện trong hình 6.1.

Để có được các phân loại cho từng bản ghi trong tập dữ liệu bằng mô hình GIỎ HÀNG, trước tiên bạn cần tạo một khung dữ liệu bao gồm các biến dự đoán của các bản ghi mà bạn muốn phân loại.

```
X = data.frame(maritalStatus = adult_tr$maritalStatus,
Cap_Gains_Losses =
        adult_tr$Cap_Gains_Losses)
```

Khung dữ liệu X chứa hai biến dự đoán đã được sử dụng để xây dựng mô hình GIỎ HÀNG. Điều cực kỳ quan trọng là tên biến trong khung dữ liệu mới này phải khớp chính xác với tên biến được sử dụng để xây dựng mô hình GIỎ HÀNG.

Khi bạn có các biến dự đoán mà bạn muốn phân loại, hãy sử dụng hàm dự đoán () yêu cầu.

```
predIncomeCART = dự đoán (đối tượng = giỏ hàng01, dữ liệu mới = X, loại = "lớp")
```

Machine Translated by Google

Đối tượng đầu vào = cart01 cho biết rằng việc phân loại được thực hiện bằng cách sử dụng mô hình GIỎ HÀNG mà bạn đã lưu dưới dạng giỏ hàng01. newdata = X gửi dữ liệu trong khung dữ liệu X xuống mô hình GIỎ HÀNG để đến nút lá và phân loại. Đầu vào type = "class" chỉ định rằng chúng ta muốn chính phân loại làm đầu ra. Chúng tôi lưu các dự đoán dưới dạng preIncomeCART.

6.3 THUẬT TOÁN C5.0 XÂY DỰNG CÂY QUYẾT ĐỊNH

Thuật toán C5.0 là phần mở rộng của thuật toán C4.5 của riêng J. Ross Quinlan để tạo cây quyết định.5 Không giống như GIỎ HÀNG, thuật toán C5.0 không bị hạn chế đối với các phân tách nhị phân. Thuật toán 5.0 sử dụng khái niệm tăng thông tin hoặc giảm entropy để chọn phân chia tối ưu. Giả sử rằng chúng ta có một biến X có k các giá trị có thể có xác suất p1 , p2 , ., pk . Số bit nhỏ nhất, trung bình trên mỗi ký hiệu, cần thiết để truyền một luồng ký hiệu đại diện cho các giá trị của X được quan sát được gọi là entropy của X, được định nghĩa là

$$HX(\quad) = -\sum_{j} p_j \log_2(p_j)$$

C5.0 sử dụng entropy như sau. Giả sử rằng chúng ta có một ứng cử viên phân chia S, mà Giá trị trung bình phân vùng tập dữ liệu huấn luyện T thành nhiều tập con, T1 , T2 , ., Tk . yêu cầu thông tin sau đó có thể được tính là tổng trọng số của các entropy cho các tập hợp con riêng lẻ, như sau:

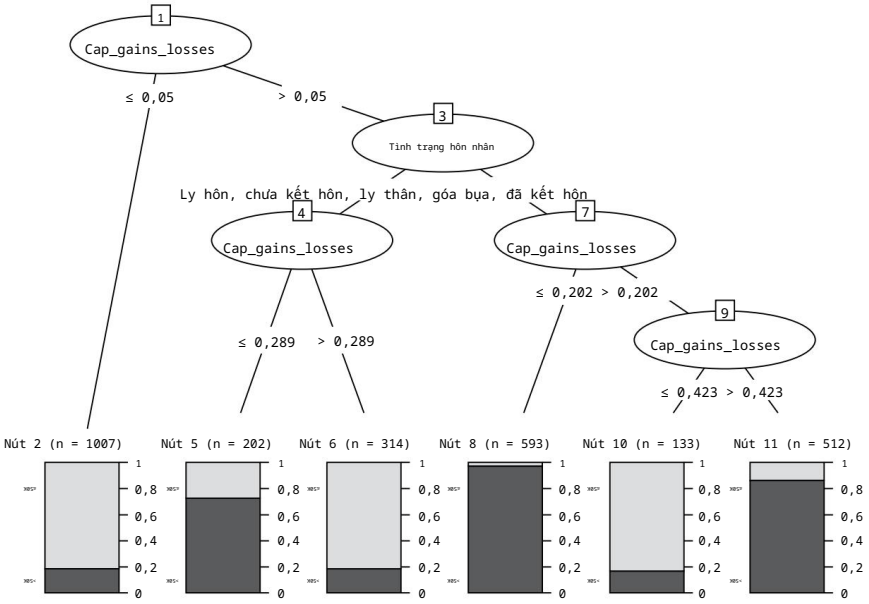
$$HT_S(\quad) = -\sum_{i=1}^{k} p_i \, HT_i(\quad)$$

trong đó Pi đại diện cho tỷ lệ các bản ghi trong tập hợp con i. Sau đó, chúng tôi có thể xác định mức tăng thông tin của mình là mức tăng (S) = H(T) – HS (T), nghĩa là mức tăng thông tin được tạo ra bằng cách phân vùng dữ liệu đào tạo T theo phân chia ứng viên S này. Tại mỗi nút quyết định, C5.0 chọn phần tách tối ưu là phần tách có mức tăng thông tin lớn nhất, gain(S).

Hình 6.2 cho thấy đầu ra của cây quyết định C5.0 từ R cho adult_ch6_ tập dữ liệu huấn luyện . Việc phân tách nút gốc (Nút 1) phụ thuộc vào việc có hay không Cap_Gains_ Lỗ (CGL) vượt quá 0,05. Nếu không, thì nhánh ngay lập tức kết thúc ở nút lá (Nút 2) với phần lớn các bản ghi có thu nhập thấp. Lưu ý rằng Nút 2 chứa 17.007 bản ghi, chiếm 90,7% trên tập dữ liệu. Phần còn lại của cây quyết định, Nút 3–11, chỉ hoạt động cùng nhau với 9,3% tập dữ liệu.

[5] J. Ross Quinlan, C4.5: Các chương trình dành cho Máy học, Morgan Kaufmann, San Francisco, CA, 1992.

Machine Translated by Google



Hình 6.2 Đầu ra của cây quyết định C5.0 từ R.

Nếu CGL vượt quá 0,05, thì nút quyết định tiếp theo (Nút 3) là về Tình trạng hôn nhân. Nếu đã kết hôn, thì cây sẽ phân nhánh tới Nút 7. Nếu một số tình trạng hôn nhân khác, cây sẽ phân nhánh tới Nút 4, sau đó kiểm tra lại CGL, lần này xem nó có vượt quá 0,289 hay không. Ngược lại, nhóm có CGL>0,289 có tỷ lệ thu nhập cao ở Nút 6 thấp hơn so với nhóm có CGL thấp hơn ở Nút 5. Nút 7 nhận các cá nhân đã kết hôn có CGL>0,05. Sau đó, nó phân tách xem CGL>0,202. Nếu không, thì nút lá (Node 8) hầu hết đều có thu nhập cao. Nếu CGL>0,202, một lần phân tách cuối cùng trong Nút 9 sẽ kiểm tra xem CGL>0,423 hay không. Nếu vậy, hồ sơ 512 chủ yếu là thu nhập cao, nếu không thì không.

Có một số khác biệt thú vị giữa cây quyết định được xây dựng bởi CART và C5.0. Ngược lại với cây C5.0 chuyển hơn 90% bản ghi thành một nút duy nhất, nút gốc của cây GIỎI được phân chia (Hình 6.1) theo tình trạng hôn nhân dẫn đến sự phân chia khá cân bằng, có thể là do bản chất của Chỉ số Gini , chứa hệ số PL PR , ưu tiên các nhánh cân bằng.

6.3.1 Cách xây dựng cây quyết định C5.0 bằng Python

Các gói Python không triển khai trực tiếp C5.0. Thay vào đó, chúng ta sẽ lại sử dụng gói sklearn , lần này thay đổi tiêu chí phân tách từ Gini thành entropy.

Trước khi chạy mã trong phần này, hãy chạy mã trong phần Python trước đó, tối đa nhưng không bao gồm đoạn bắt đầu "Bây giờ chúng tôi đã sẵn sàng để chạy thuật toán GIỎ HÀNG!" Mã có trong đó sẽ thiết lập các biến và tên biến của chúng ta để chuẩn bị cho việc chạy cây quyết định này.

Machine Translated by Google

Để có được cây quyết định sử dụng entropy làm tiêu chí phân tách, chúng ta lại sử dụng lệnh DecisionTreeClassifier () .

```
c50_01 = DecisionTreeClassifier(criterion="entropy",
max_leaf_nodes=5).fit(X,y)
```

Tiêu chí đầu vào = "entropy" sử dụng thông tin thu được để xác định phân chia ứng viên tốt nhất.

Để xuất phần tóm tắt của cây C5.0, hãy chạy lệnh export_graphviz() . Mã này được đưa ra bên dưới, với lời giải thích trong phần Python trước đó.

```
export_graphviz(c50_01, out_file = "C:/.../c50_01.dot", feature_names=X_names,

        class_names=y_names)
```

Để có được các phân loại của từng bản ghi trong tập dữ liệu huấn luyện, hãy chạy lệnh predict() trên tên của đầu ra đã lưu, c50_01, sử dụng biến dự đoán X làm đầu vào.

```
c50_01.predict(X)
```

Kết quả là phân loại cho từng bản ghi trong tập dữ liệu huấn luyện.

6.3.2 Cách xây dựng cây quyết định C5.0 bằng R

Nếu bạn không thay đổi tên biến Trạng thái Võ thuật hoặc chuyển đổi các biến phân loại thành các yếu tố, hãy làm như vậy ngay bây giờ bằng cách làm theo các bước có liên quan trong phần R trước đó. Sau đó, cài đặt và tải gói cần thiết để chạy C5.0.

```
install.packages("C50"); thư viện(C50)
```

Chạy thuật toán, sử dụng lệnh C5.0() .

```
C5 <- C5.0(công thức = Thu nhập ~ Tình trạng hôn nhân + Cap_Gains_
Tổn thất, dữ liệu = adult_tr,
          điều khiển = C5.0Control(minCases=75))
```

Giá trị đầu vào cốt lõi là đầu vào công thức , giống hệt với đối tác GIỎ HÀNG của nó. Như trước đây, biến mục tiêu nằm ở bên trái của dấu ngã và các biến tiền tố ở bên phải, được phân tách bằng dấu cộng. Đầu vào data = adult_tr xác định rằng chúng tôi đang vẽ các biến của mình từ tập dữ liệu adult_tr . Đầu vào control = C5.0Control(minCases = 75)) yêu cầu các nút lá trong cây quyết định phải có ít nhất 75 bản ghi. Chúng tôi lưu cây kết quả là C5.

Chúng ta có thể hình dung cây bằng cách sử dụng lệnh plot() .

```
cốt truyện (C5)
```

Machine Translated by Google

Đầu ra từ ô (C5) được hiển thị trong Hình 6.2.

Để có được phân loại cho từng bản ghi trong tập dữ liệu, hãy tạo dữ liệu X frame như trong phần R trước đó và chạy lệnh predict() .

dự đoán (đối tượng = C5, dữ liệu mới = X)

Đặt đối tượng = C5 để sử dụng cây C5.0 mà bạn đã xây dựng trong phần này. Giữ lại đầu vào newdata = X , như trong phần R trước đó. Kết quả là phân loại biến Thu nhập cho mỗi bản ghi trong tập dữ liệu huấn luyện.

6.4 RỪNG NGẪU NHIÊN

GIỎ HÀNG và C5.0 đều tạo ra một cây quyết định duy nhất dựa trên tất cả các bản ghi và các biến được chỉ định trong tập dữ liệu huấn luyện. Tuy nhiên, có một phương pháp sử dụng nhiều cây, trong đó đầu ra của mỗi cây được xem xét khi xác định phân loại cuối cùng của mỗi bản ghi.

Rừng ngẫu nhiên6 xây dựng một loạt các cây quyết định và kết hợp các phân loại tỷ lệ phân tán cây của từng bản ghi thành một phân loại cuối cùng. Rừng ngẫu nhiên là một ví dụ về phương pháp tập hợp. Các phương pháp tập hợp là một loại kỹ thuật lập mô hình có tính đến đầu ra của nhiều mô hình để đi đến một câu trả lời duy nhất. Các phương pháp tập hợp khác nhau sẽ xem xét đầu ra của các mô hình theo những cách khác nhau. Để biết thêm về các phương pháp tập hợp, vui lòng xem văn bản trước đó của chúng tôi.7

Thuật toán rừng ngẫu nhiên bắt đầu xây dựng từng cây quyết định bằng cách lấy một mẫu ngẫu nhiên, có thay thế, từ tập dữ liệu huấn luyện ban đầu. Theo cách này, mỗi cây sẽ có một bộ dữ liệu khác nhau để xây dựng. Đối với mỗi nút của cây quyết định, một tập hợp con của các biến dự đoán được chọn để xem xét. Theo cách này, có thể là biến số đưa ra sự phân chia "tốt nhất" (ví dụ, theo tiêu chí Gini) sẽ không có sẵn để xem xét.

Cây quyết định được hoàn thành theo cách này, không có hạn chế về độ lớn của cây.

Khi các cây khác nhau được tạo, chúng được sử dụng để phân loại các bản ghi trong tập dữ liệu huấn luyện ban đầu. Mỗi bản ghi trong tập dữ liệu được phân loại theo từng cây. Vì các phân loại này rất khó có thể thống nhất cho tất cả các bản ghi, nên mỗi phân loại được coi là một "phiếu bầu" cho giá trị biến mục tiêu cụ thể đó. Giá trị có số phiếu bầu lớn nhất được coi là phân loại cuối cùng của bản ghi.

6Leo Breiman, Rừng ngẫu nhiên, Học máy, 45, 5–32, 2001.

7Khai thác dữ liệu và phân tích dự đoán, Phiên bản thứ hai, của Daniel T. Larose và Chantal D. Larose, John Wiley và Sons, Inc., 2015.

Machine Translated by Google

6.4.1 Cách xây dựng rừng ngẫu nhiên trong Python

Như trước đây, hãy chạy mã trong phần GIỎ HÀNG Python, tối đa nhưng không bao gồm mã trong đoạn bắt đầu "Bây giờ chúng tôi đã sẵn sàng để chạy thuật toán GIỎ HÀNG!" Bạn sẽ cần chạy mã để thiết lập biến dự đoán và biến mục tiêu cho mã trong phần này.

Tiếp theo, tải các thư viện cần thiết.

```
từ sklearn.ensemble nhập RandomForestClassifier
nhập numpy dưới dạng np
```

Lệnh rừng ngẫu nhiên trong Python yêu cầu một biến phản hồi cho mờ dưới dạng mảng một chiều, vì vậy chúng tôi sử dụng lệnh ravel() của numpy để tạo định dạng đó.

```
rfy = np.ravel(y)
```

Chúng tôi sử dụng lệnh RandomForestClassifier() để tạo rừng ngẫu nhiên.

```
rf01 = RandomForestClassifier(n_estimators = 100,
criteria="gini").fit(X,rfy)
```

Như trước đây, lệnh RandomForestClassifier() thiết lập các tham số của thuật toán. Đầu vào n_estimators chỉ định số lượng cây sẽ được tạo, trong khi tiêu chí = “gini” chỉ định Chỉ số Gini được sử dụng để xác định các phần tách tốt nhất. Lệnh fit() sử dụng biến dự đoán X và biến đích y để xây dựng cây quyết định thực tế. Lưu kết quả dưới dạng rf01.

Để xem các phân loại được thực hiện trên tập dữ liệu huấn luyện bằng thuật toán ước tính ngẫu nhiên, hãy sử dụng lệnh predict() .

```
rf01.predict(X)
```

Kết quả là một loạt các phân loại, một phân loại cho mỗi bản ghi trong tập dữ liệu.

6.4.2 Cách xây dựng rừng ngẫu nhiên trong R

Nếu bạn không thay đổi tên biến Trạng thái Võ thuật hoặc chuyển đổi các biến phân loại thành các yếu tố, hãy làm như vậy ngay bây giờ bằng cách làm theo các bước có liên quan trong phần R trước đó. Tải dữ liệu.

Cài đặt và mở gói RandomForest .

```
install.packages("Rừng ngẫu nhiên"); thư viện (rừng ngẫu nhiên)
```

Bây giờ, chúng tôi chạy thuật toán rừng ngẫu nhiên, sử dụng đầu vào công thức giống như trước đây.

```
rf01 <- RandomForest(công thức = Thu nhập ~ tình trạng hôn nhân +
Cap_Gains_Losses,
```

Machine Translated by Google


```
data=adult_tr, ntree=100, type="classification")
```

Lệnh để xây dựng các khu rừng ngẫu nhiên là RandomForest() và đầu vào đầu tiên, công thức, giống với công thức GIỎ HÀNG và C5.0 trong các phần R trước.
Dữ liệu đầu vào chỉ định nguồn gốc của các biến trong công thức . cây thông
đầu vào cho thuật toán biết cần tạo bao nhiêu cây. Đối với tập dữ liệu tương đối nhỏ của chúng tôi, chúng tôi sử dụng 100 cây. Đầu vào cuối cùng, type = "classification," xác định rằng chúng ta đang phân loại dữ liệu của mình. Chúng tôi lưu đầu ra dưới dạng rf01.

Để xem các phân loại được thực hiện bởi thuật toán, hãy xem dự đoán
các giá trị được lưu dưới rf01.

```
rf01$dự đoán
```

Kết quả là một phân loại cho từng bản ghi trong tập dữ liệu.

NGƯỜI GIỚI THIỆU

Để kiểm tra các chi tiết của gói C50 , hãy xem tài liệu: Max Kuhn và Ross Quinlan, C50: Cây quyết định C5.0 và Mô hình dựa trên quy tắc. Gói R phiên bản 0.1.2, 2018. https://CRAN.R-project.org/package=C50.

Nếu bạn muốn khám phá toàn bộ gói RandomForest, hãy bắt đầu với ấn phẩm cốt lõi: A. Liaw và M. Wiener, Phân loại và hồi quy theo RandomForest, R News, 2(3), 18–22, 2002.

Chúng tôi đã trình bày một số tùy chọn mà gói rpart.plot phải đại diện cho các mô hình GIỎ HÀNG. Bạn có thể tìm tài liệu đầy đủ cho gói tại đây: Stephen Milborrow, rpart.plot: Plot "rpart" Models: An Enhanced Version of "plot.rpart." Gói R phiên bản 2.2.0, 2018. https://CRAN.R-project.org/package=rpart.plot.

Xem quy trình trình bày chi tiết gói mô hình thống kê cho Python: Skipper Seabold và Josef Perktold, "Statsmodels: Mô hình kinh tế lượng và thống kê với trăn." Trong Kỷ yếu Hội nghị Khoa học Python lần thứ 9, 2010.

Gói dành cho các mô hình GIỎ HÀNG trong R: Terry Therneau và Beth Atkinson, rpart: Recursive Partitioning and Regression Trees. Phiên bản gói R 4.1-13, 2018. https://CRAN.R-project.org/package=rpart.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Cây quyết định là gì?
2. Điểm khác biệt giữa nút quyết định và nút lá là gì?
3. Trong cây quyết định, đâu là cách phân chia mạnh nhất có thể được thực hiện?
4. Khi nào thì cây quyết định ngừng phát triển?
5. Cây quyết định hoạt động như thế nào?

Machine Translated by Google

6. GIỎ HÀNG có phải là một thuật toán tốt để sử dụng nếu chúng ta quan tâm đến phân loại ba dự đoán?
7. GIỎ HÀNG sử dụng tiêu chí nào để đánh giá cách phân chia nào là tối ưu?
8. Thuật toán C5.0 sử dụng khái niệm nào để chọn cách chia tối ưu?
9. Rừng ngẫu nhiên là gì?
10. Rừng ngẫu nhiên hoạt động như thế nào?
11. Có phải tất cả các biến dự đoán đều là ứng cử viên để phân chia "tốt nhất" cho mỗi nút trong một cây được xây dựng bởi các khu rừng ngẫu nhiên không?
12. Các bộ dữ liệu được sử dụng để xây dựng từng cây trong các khu rừng ngẫu nhiên có giống nhau không?
13. Thuật toán rừng ngẫu nhiên phân loại cuối cùng cho tập dữ liệu huấn luyện như thế nào?

LÀM VIỆC VỚI DỮ LIỆU

Đối với Bài tập 14–20, hãy làm việc với bộ dữ liệu adult_ch6_training và adult_ch6_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

14. Tạo mô hình GIỎ HÀNG bằng cách sử dụng tập dữ liệu đào tạo dự đoán thu nhập dựa trên tình trạng hôn nhân và các khoản lãi và lỗ vốn. Trực quan hóa cây quyết định (nghĩa là cung cấp đầu ra của cây quyết định). Mô tả một số lần phân tách đầu tiên trong cây quyết định.
15. Phát triển mô hình GIỎ HÀNG bằng cách sử dụng bộ dữ liệu thử nghiệm sử dụng cùng các biến mục tiêu và biến dự đoán. Trực quan hóa cây quyết định. So sánh các cây quyết định Kết quả dữ liệu thử nghiệm có khớp với kết quả dữ liệu huấn luyện không?
16. Sử dụng tập dữ liệu huấn luyện để xây dựng mô hình C5.0 nhằm dự đoán thu nhập dựa trên tình trạng hôn nhân và lãi/lỗ vốn. Chỉ định tối thiểu 75 trường hợp cho mỗi nút đầu cuối. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.
17. Mô hình C5.0 của bạn so với mô hình GIỎ HÀNG như thế nào? Nêu sự giống nhau và sự khác biệt.
18. Xây dựng mô hình C5.0 bằng cách sử dụng tập dữ liệu thử nghiệm sử dụng cùng một biến mục tiêu, các biến tiền tố và tiêu chí trường hợp tối thiểu. Trực quan hóa cây quyết định. Kết quả dữ liệu thử nghiệm có khớp với kết quả dữ liệu huấn luyện không?
19. Sử dụng rừng ngẫu nhiên trên tập dữ liệu huấn luyện để dự đoán thu nhập bằng cách sử dụng tình trạng hôn nhân và lãi và lỗ vốn.
20. Sử dụng các khu rừng ngẫu nhiên bằng cách sử dụng bộ dữ liệu thử nghiệm sử dụng cùng các biến dự đoán và mục tiêu. Kết quả dữ liệu thử nghiệm có khớp với kết quả dữ liệu huấn luyện không?

PHÂN TÍCH THỰC TẾ

Đối với Bài tập 21–27, hãy làm việc với bộ dữ liệu loan_training và loan_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

21. Tạo mô hình GIỎ HÀNG bằng cách sử dụng tập dữ liệu đào tạo dự đoán Phê duyệt bằng cách sử dụng Tỷ lệ Nợ trên Thu nhập, Điểm FICO và Số tiền Yêu cầu. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

Machine Translated by Google

22. Phát triển mô hình GIỎ HÀNG bằng cách sử dụng bộ dữ liệu thử nghiệm sử dụng cùng các biến mục tiêu và biến dự đoán. Trực quan hóa cây quyết định. Điều tra các phần tách trong cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

23. Xây dựng mô hình C5.0 bằng cách sử dụng tập dữ liệu đào tạo dự đoán Phê duyệt bằng cách sử dụng Tỷ lệ Nợ trên Thu nhập, Điểm FICO và Số tiền Yêu cầu. Chỉ định tối thiểu 1000 trường hợp cho mỗi nút đầu cuối. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

24. Làm thế nào để mô hình C5.0 của bạn so với mô hình GIỎ HÀNG của bạn đối với dữ liệu cho vay_đào tạo ? Mô tả những điểm tương đồng và khác biệt.

25. Tạo mô hình C5.0 bằng cách sử dụng tập dữ liệu thử nghiệm sử dụng cùng biến mục tiêu, biến tiền tố và tiêu chí trường hợp tối thiểu. Trực quan hóa cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

26. Sử dụng rừng ngẫu nhiên trên tập dữ liệu huấn luyện để thu được giá trị dự đoán của Phê duyệt bằng cách sử dụng cùng các biến dự đoán như trong mô hình GIỎ HÀNG và C5.0.

27. Sử dụng rừng ngẫu nhiên trên tập dữ liệu thử nghiệm để thu được giá trị dự đoán của Phê duyệt trong tập dữ liệu thử nghiệm. Xây dựng bảng so sánh các dự đoán từ tập dữ liệu huấn luyện và kiểm tra. Làm thế nào để họ so sánh?

Đối với Bài tập 28–34, hãy làm việc với bộ dữ liệu bank_marketing_training và bank_marketing_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

28. Tạo mô hình GIỎ HÀNG bằng cách sử dụng tập dữ liệu huấn luyện dự đoán Phản hồi bằng bất kỳ yếu tố dự báo nào bạn cho là phù hợp. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

29. Phát triển mô hình GIỎ HÀNG bằng cách sử dụng tập dữ liệu thử nghiệm và cùng các biến mục tiêu và biến dự đoán. Trực quan hóa cây quyết định. Điều tra các phần tách trong cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

30. Xây dựng mô hình C5.0 sử dụng tập dữ liệu huấn luyện dự đoán Phản hồi và cùng mục tiêu và các biến dự báo. Chỉ định tối thiểu 1000 trường hợp cho mỗi nút đầu cuối. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

31. Mô hình C5.0 của bạn so với mô hình GIỎ HÀNG của bạn cho ngân hàng_marketing_ như thế nào dữ liệu đào tạo ? Mô tả những điểm tương đồng và khác biệt.

32. Tạo mô hình C5.0 bằng cách sử dụng tập dữ liệu thử nghiệm sử dụng cùng biến mục tiêu, biến dự đoán và tiêu chí trường hợp tối thiểu. Trực quan hóa cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

33. Sử dụng rừng ngẫu nhiên trên tập dữ liệu huấn luyện để thu được giá trị dự đoán của Phản hồi bằng cách sử dụng cùng các biến dự đoán như trong mô hình GIỎ HÀNG và C5.0.

34. Sử dụng rừng ngẫu nhiên trên tập dữ liệu thử nghiệm để thu được giá trị dự đoán của Phản hồi trong tập dữ liệu thử nghiệm. Xây dựng bảng so sánh các dự đoán từ tập dữ liệu huấn luyện và kiểm tra. Làm thế nào để họ so sánh?

Đối với Bài tập 35–41, hãy làm việc với tập dữ liệu huấn luyện và kiểm tra mà bạn thu được bằng cách phân vùng tập dữ liệu Churn trong Bài tập Chương 5. Sử dụng Python hoặc R để giải quyết từng vấn đề.

35. Tạo mô hình GIỎ HÀNG bằng cách sử dụng tập dữ liệu đào tạo dự đoán Churn bằng bất kỳ yếu tố dự báo nào bạn cho là phù hợp. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

Machine Translated by Google

36. Xây dựng mô hình GIỎ HÀNG sử dụng tập dữ liệu thử nghiệm và cùng mục tiêu và các biến dự đoán. Trực quan hóa cây quyết định. Điều tra các phần tách trong cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

37. Xây dựng mô hình C5.0 sử dụng tập dữ liệu huấn luyện dự đoán Churn và cùng mục tiêu và các biến dự đoán. Chỉ định tối thiểu 1000 trường hợp cho mỗi nút đầu cuối. Trực quan hóa cây quyết định. Mô tả một số lần phân tách đầu tiên trong cây quyết định.

38. Mô hình C5.0 của bạn so với mô hình GIỎ HÀNG của bạn như thế nào đối với dữ liệu churn_training ? Mô tả những điểm tương đồng và khác biệt.

39. Tạo một mô hình C5.0 bằng cách sử dụng tập dữ liệu thử nghiệm sử dụng cùng một biến mục tiêu, các biến định trước và tiêu chí trường hợp tối thiểu. Trực quan hóa cây quyết định. Cây được tạo bằng dữ liệu thử nghiệm có khớp với cây được tạo bằng dữ liệu huấn luyện không?

40. Sử dụng rừng ngẫu nhiên trên tập dữ liệu huấn luyện để thu được giá trị dự đoán của Churn bằng cách sử dụng cùng các biến dự đoán như trong mô hình GIỎ HÀNG và C5.0.

41. Sử dụng rừng ngẫu nhiên trên tập dữ liệu thử nghiệm để thu được giá trị Churn dự đoán trong tập dữ liệu thử nghiệm. Xây dựng bảng so sánh các dự đoán từ tập dữ liệu huấn luyện và kiểm tra. Làm thế nào để họ so sánh?

Machine Translated by Google

chương 7

ĐÁNH GIÁ MÔ HÌNH

7.1 GIỚI THIỆU ĐÁNH GIÁ MÔ HÌNH

Cho đến nay trong Khoa học dữ liệu sử dụng Python và R, chúng ta đã đề cập đến năm giai đoạn đầu tiên của Phương pháp khoa học dữ liệu:

1. Giai đoạn hiểu dữ liệu
2. Giai đoạn chuẩn bị dữ liệu
3. Giai đoạn phân tích dữ liệu khám phá
4. Giai đoạn thiết lập
5. Giai đoạn lập mô hình (ít nhất là một chút)

Tuy nhiên, cho đến nay chúng tôi vẫn chưa kiểm tra xem các mô hình của chúng tôi có tốt không. Đó là, chúng tôi chưa đánh giá tính hữu ích của chúng trong việc đưa ra dự đoán. Lưu ý sự khác biệt giữa đánh giá và xác nhận. Xác thực mô hình chỉ đơn giản là đảm bảo rằng kết quả mô hình của chúng tôi nhất quán giữa tập dữ liệu huấn luyện và kiểm tra. Tuy nhiên, việc xác thực mô hình không cho chúng tôi biết các mô hình của chúng tôi chính xác đến mức nào hoặc tỷ lệ lỗi của chúng là bao nhiêu. Đối với các biện pháp như thế này, chúng ta cần chuyển sang đánh giá mô hình. Vì các mô hình duy nhất mà chúng ta đã học cho đến nay là các cây quyết định để phân loại, nên chúng ta sẽ hạn chế thảo luận về các biện pháp đánh giá cho các mô hình phân loại.

7.2 BIỆN PHÁP ĐÁNH GIÁ PHÂN LOẠI

Chúng tôi sẽ phát triển các biện pháp đánh giá phân loại cho trường hợp chúng tôi có biến mục tiêu nhị phân. Để áp dụng các thước đo mà chúng ta sẽ tìm hiểu trong chương này, chúng ta cần biểu thị (tùy ý, nếu muốn) một trong hai kết quả mục tiêu là tích cực và một là tiêu cực. Ví dụ: giả sử chúng ta đang cố gắng dự đoán thu nhập,

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

BẢNG 7.1 Dạng tổng quát của bảng dự phòng cho phân loại nhị phân

		Danh mục dự đoán		
		0	1	Tổng cộng
danh mục thực tế	0 Âm tính thực sự: dự đoán 0 Trên thực tế 0		Dương tính giả: dự đoán 1 Trên thực tế 0	Tổng số thực sự tiêu cực
	1 Âm tính giả: dự đoán 0 Trên thực tế 1		Tích cực thực sự: dự đoán 1 Trên thực tế 1	Tổng số thực sự tích cực
	Tổng cộng Dự đoán tiêu cực		Tổng cộng dự đoán tích cực	Tổng cộng

một biến nhị phân với các giá trị thu nhập cao và thu nhập thấp. Chúng ta có thể biểu thị thu nhập cao là dương và thu nhập thấp là âm.1

Bây giờ, các biện pháp đánh giá mô hình phân loại mà chúng ta sẽ tìm hiểu trong chương này là các chức năng của các mục trong bảng dự phòng2 được tạo bởi mô hình phân loại, dạng tổng quát của nó được thể hiện trong Bảng 7.1. Lưu ý rằng, theo quy ước, các giá trị thực tế được biểu thị bằng các hàng, trong khi các giá trị dự đoán được biểu thị bằng các cột. Ô phía trên bên trái trong Bảng 7.1 biểu thị số lượng bản ghi trong đó mô hình dự đoán một phản hồi tiêu cực và giá trị phản hồi thực tế thực sự là âm, làm cho dự đoán này trở thành một giá trị âm thực sự. Ô bên dưới biểu thị số lượng bản ghi trong đó mô hình dự đoán phản hồi tiêu cực nhưng giá trị phản hồi thực tế lại dương, khiến dự đoán này trở thành âm tính giả. Các ô khác được xác định tương tự.

Đặt TN, FN, FP và TP đại diện cho số lượng âm tính thật, âm tính giả, lần lượt là dương tính giả và dương tính thật trong bảng dự phòng của chúng tôi. Ngoài ra, hãy để

TAN Tổng A TN FP thực sự âm = +

TAP Tổng số FN TP thực sự tích = cực +

TPN Tổng số P được xác định lại TN = FN +

TPN Tổng số P được xác định lại Tích cực = FP TP +

Hơn nữa, đặt GT = TN+FN+FP+TP đại diện cho tổng số lớn trong bốn tế bào. Sau đó, chúng ta có thể diễn đạt lại Bảng 7.1 thành Bảng 7.2.

Sau đó, sử dụng ký hiệu trong Bảng 7.2, chúng tôi xác định tập hợp các biện pháp đánh giá phân loại của chúng tôi.

$$\text{Sự chính xác} = \frac{TN + TP.}{TN + FN + FP + TP} + = \frac{TN + TP.}{GT}$$

1Không có ý nghĩa tích cực hay tiêu cực nào đối với các nhãn này. Chúng đơn giản cho phép chúng ta áp dụng các biện pháp này cho bất kỳ bài toán phân loại nhị phân nào.

2Còn được gọi là ma trận nhầm lẫn hoặc ma trận lỗi.

Machine Translated by Google

BẢNG 7.2 Dạng tổng quát của bảng dự phòng, được thể hiện lại

		Danh mục dự đoán		
		0	1	Tổng cộng
danh mục thực tế	0	TN	FP TÂN	
	1	FN	TP.TẤP _	
	Tổng cộng	TPN	CTPT GT	

$$\text{Tỷ lệ lỗi} = 1 - \text{Sự chính xác} = \frac{FN + FP}{TN + FN + FP + TP} = \frac{FN + FP}{GT}$$

Độ chính xác đại diện cho thước đo tổng thể về tỷ lệ phân loại chính xác do mô hình tạo ra, trong khi tỷ lệ lỗi đo lường tỷ lệ phân loại không chính xác, trên tất cả các ô trong bảng dự phòng. Tuy nhiên, các biện pháp này không phân biệt giữa các loại lỗi khác nhau hoặc các loại quyết định đúng khác nhau. Chúng tôi bắt đầu làm như vậy bằng cách sử dụng độ nhạy và độ đặc hiệu, như sau.

7.3 ĐỘ NHẠY VÀ ĐỘ ĐẶC BIỆT

$$\text{Nhạy cảm} = \frac{\text{Số dương tính thật}}{\text{Tổng số thực sự tích cực e}} = \frac{TP.}{\text{VÔ NHE}} = \frac{TP.}{TP + FN}$$

$$\text{độ đặc hiệu} = \frac{\text{Số lượng âm tính thực sự}}{\text{Tổng thực sự âm e}} = \frac{TN}{\text{TÂN}} = \frac{TN}{FP + TN}$$

Độ nhạy đo khả năng mô hình phân loại bản ghi một cách tích cực, trong khi độ đặc hiệu đo khả năng phân loại bản ghi một cách tiêu cực. Độ nhạy đo lường tỷ lệ tất cả các bản ghi tích cực được mô hình của bạn ghi lại, trong khi độ đặc hiệu đo lường tỷ lệ tất cả các bản ghi tiêu cực được mô hình của bạn ghi lại. Tất nhiên, một mô hình phân loại hoàn hảo sẽ có độ nhạy = 1,0 = 100%. Tuy nhiên, một mô hình đơn giản chỉ phân loại tất cả các khách hàng là tích cực cũng sẽ có độ nhạy = 1,0. Rõ ràng, chỉ xác định các phản ứng tích cực thôi là chưa đủ. Một mô hình phân loại cũng cần phải cụ thể, nghĩa là nó phải xác định được một tỷ lệ cao khách hàng có thái độ tiêu cực. Tất nhiên, một mô hình phân loại hoàn hảo sẽ có độ đặc hiệu = 1.0. Tuy nhiên, một mô hình phân loại tất cả khách hàng là tiêu cực cũng vậy. Một mô hình phân loại tốt nên có mức độ chấp nhận được về cả độ nhạy và độ đặc hiệu, nhưng những gì cấu thành có thể chấp nhận được rất khác nhau giữa các lĩnh vực.

7.4 ĐỘ CHÍNH XÁC, GHI NHỚ VÀ Fβ SCORES

Trong số các bản ghi được mô hình của chúng tôi phân loại là tích cực, tỷ lệ nào là tích cực thực sự? Số liệu giải quyết câu hỏi này được gọi là độ chính xác và được định nghĩa như sau:

Machine Translated by Google

$$\text{Độ chính xác} = \frac{\text{TP.}}{\text{TPP}}$$

Trong lĩnh vực truy xuất thông tin (ví dụ: công cụ tìm kiếm), số liệu chính xác trả lời câu hỏi, "Tỷ lệ các mục được chọn có liên quan là bao nhiêu?" Số liệu này thường được kết hợp với thu hồi, đây chỉ là một tên gọi khác của độ nhạy.

$$\text{Nhớ lại} = \frac{\text{TN}}{\text{TÂN}}$$

Cụ thể Sẽ rất hữu ích nếu kết hợp độ chính xác và thu hồi thành một thước đo duy nhất. Để làm như vậy, chúng ta có thể sử dụng điểm Fβ , được định nghĩa như sau:

$$F_{\beta} = \left(1+\beta^{2}\right)\cdot\frac{\text{thu hồi chính xác}}{\left(\beta^{2}\cdot\text{thu hồi chính xác}\right)+}$$

đối với β>0.

- Khi β = 1, giá trị này được gọi là giá trị trung bình điều hòa của độ chính xác và độ thu hồi, trong đó do đó có trọng số như nhau trong chỉ số F1 .
- Khi β>1, trọng số Fβ thu hồi cao hơn độ chính xác. •
- Khi β<1, trọng số Fβ thu hồi thấp hơn độ chính xác. •
- Do đó, F2 sẽ thu hồi trọng số cao gấp đôi độ chính xác, trong khi F0.5 sẽ trọng lượng nhớ lại một nửa so với độ chính xác.

7.5 PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH

Phương pháp chung để đánh giá mô hình được đưa ra như sau. Điều này áp dụng cho bất kỳ mô hình phân loại hoặc ước tính nào.

Phương pháp đánh giá mô hình

1. Phát triển mô hình sử dụng tập dữ liệu huấn luyện.
2. Đánh giá mô hình bằng tập dữ liệu thử nghiệm. Tức là lấy mô hình được phát triển bằng cách sử dụng tập dữ liệu huấn luyện và áp dụng nó vào tập dữ liệu thử nghiệm. Nói cách khác, chuyển tập kiểm tra thông qua mô hình được tạo bởi tập huấn luyện.

7.6 ÁP DỤNG ĐÁNH GIÁ MÔ HÌNH

Chúng tôi sẽ làm việc với tập dữ liệu clothing_data_drive_training và clothing_data_ Driven_test . Nhiệm vụ là dự đoán liệu khách hàng có phản hồi chiến dịch tiếp thị qua điện thoại/thư hay không, dựa trên ba yếu tố dự đoán liên tục:

- Số ngày kể từ khi mua hàng • Số lượt truy cập mua hàng • Doanh số trên mỗi lượt truy cập

Machine Translated by Google

Biến mục tiêu là một cờ, Phản hồi, được mã hóa 1 cho phản hồi tích cực và 0 cho phản hồi tiêu cực.

Chúng tôi phát triển mô hình C5.0 để phân loại phản hồi bằng cách sử dụng clothing_data_tập dữ liệu drive_training . Gọi đây là Model 1. Chúng tôi sẽ sử dụng clothing_data_driver_tập dữ liệu thử nghiệm để đánh giá Mô hình 1. Việc này được thực hiện như sau:

Phương pháp đánh giá mô hình 1

1. Nhận các giá trị dự đoán cho Phản hồi, được tạo bởi Mô hình 1 trong quá trình đào tạo bộ dữ liệu.
2. Áp dụng Mô hình 1 cho tập dữ liệu thử nghiệm và so sánh các giá trị Phản hồi được dự đoán từ Bước 1 với các giá trị Phản hồi thực tế từ tập dữ liệu thử nghiệm.

Khi chúng tôi so sánh các giá trị Phản hồi được dự đoán từ Mô hình 1 với các giá trị Phản hồi thực tế từ tập dữ liệu thử_nghiệm_dữ_liệu_quần_áo , chúng tôi thu được bảng dự phòng trong Bảng 7.3.

Lưu ý rằng TAN = 9.614, TAP = 1.940 và GT = 11.554 sẽ đúng trên bất kỳ mô hình nào vì đây là các giá trị thực, không phải dự đoán. Các số còn lại khác nhau giữa các mô hình khác nhau, tùy thuộc vào dự đoán. Kết quả của tám biện pháp đánh giá của chúng tôi được thể hiện trong Bảng 7.4.

Mô hình 1 có độ chính xác là 0,8410 hoặc 84,10%. Cái này có tốt không? Chà, hãy nhớ lại từ Chương 5 rằng chúng ta phải luôn hiệu chỉnh kết quả của mình so với hiệu suất cơ bản. Trong trường hợp này, có TAN = 9614 bản ghi phủ định, trong tổng số GT = 11.554 bản ghi, do đó, Mô hình Tất cả Phủ định cơ sở, chỉ định tất cả các dự đoán là phủ định, sẽ có độ chính xác là

$$\text{Độ chính xác}_{\text{Tất cả Mô hình tiêu cực}} = \frac{9614}{11{,}554} = 0{,}8321$$

Vì vậy, thực sự Mô hình 1 hầu như không quản lý để vượt qua độ chính xác của đường cơ sở Tất cả Mô hình Tiêu cực. (Chắc chắn, chúng ta có thể làm tốt hơn? Xem bên dưới.)

Độ đặc hiệu của Mô hình 1 là 0,9541 là tuyệt vời, có nghĩa là mô hình đã quản lý để phân loại chính xác

$$\frac{\text{TN}}{\text{TÂN}} = \frac{9173}{9614} = 95{,}41\ \%$$

BẢNG 7.3 Bảng dự phòng để đánh giá Mô hình 1

		Danh mục dự đoán		
		0	1	Tổng cộng
danh mục thực tế	0	TN = 9173	FP = 441	TÂN = 9614
	1	FN = 1396	TP = 544	TAP = 1940
	Tổng cộng	TPN = 10,569	CTPT = 985	GT = 11,554

Machine Translated by Google

BẢNG 7.4 Các biện pháp đánh giá của mô hình R C5.0

Biện pháp đánh giá	Công thức	Giá trị
Sự chính xác	$\frac{TN + TP.}{GT} = \frac{9173 + 544}{11\ 554,}$	0,8410
tỷ lệ lỗi	1 Độ chính	0,1590
Nhạy cảm	$\frac{TP.}{VÔ\ NHE} = \frac{xác\ 544}{1940}$	0,2804
độ đặc hiệu	$\frac{TN}{TÂN} = \frac{9173}{9614}$	0,9541
Độ chính xác	$\frac{TP.}{TPP} = \frac{544}{985}$	0,5523
F1	$2 \cdot \frac{thu\ hồi\ chính\ xác}{thu\ hồi\ chính\ xác}$	0,3720
F2	$5 \cdot \frac{thu\ hồi\ chính\ xác}{(4 \cdot thu\ hồi\ chính\ xác)}$	0,3110
F0.5	$1.25 \cdot \frac{thu\ hồi\ chính\ xác}{(Thu\ hồi\ độ\ chính\ xác\ 0\ 25)}$	0,4626

của các bản ghi tiêu cực thực tế là tiêu cực. Nghĩa là, mô hình đang làm rất tốt trong việc xác định chính xác những khách hàng không hưởng ứng chiến dịch tiếp thị.
Tuy nhiên, độ nhạy 0,2804 của mô hình khá kém, nghĩa là chỉ

$$\frac{TP.}{VÔ\ NHE} = \frac{544}{1940} = 28\ 04\ .\ \%$$

của các hồ sơ tích cực thực tế đã được mô hình phân loại là tích cực. Nói cách khác, mô hình không làm tốt công việc xác định chính xác những khách hàng sẽ phản hồi tích cực với chiến dịch.

Độ chính xác của mô hình không tốt hơn nhiều so với tung đồng xu:

$$\frac{TP.}{TPP} = \frac{544}{985} = 55\ 23\ .\ \%$$

có nghĩa là, trong số những khách hàng được mô hình phân loại là tích cực, 55,23% thực sự sẽ phản hồi tích cực với hoạt động tiếp thị. Lưu ý rằng ba điểm Fβ nằm trong khoảng giữa các giá trị về độ chính xác và khả năng thu hồi, nếu chúng phải như vậy. F1 , trung bình điều hòa của độ chính xác và thu hồi, có giá trị là 0,372. Giá trị của F2 , có trọng số thu hồi cao hơn độ chính xác, gần với khả năng thu hồi hơn, trong khi giá trị của F0.5 gần với độ chính xác hơn. Lưu ý rằng chúng tôi không cung cấp các diễn giải hình thu nhỏ của các giá trị này như chúng tôi làm cho các giá trị khác. Thay vào đó, các số liệu này được sử dụng để lựa chọn mô hình, so sánh Fβ giá trị trực tiếp để chọn mô hình tốt nhất.

Machine Translated by Google

7.6.1 Cách thực hiện đánh giá mô hình bằng R

Để xác nhận mô hình, chúng ta cần

1. Phát triển Mô hình 1 bằng cách sử dụng dữ liệu huấn luyện và
2. Sau đó chạy dữ liệu thử nghiệm thông qua Mô hình 1.

Đọc trong tập dữ liệu clothing_data_driven_training là clothing_train và dữ liệu clothing_data_drive_test được đặt là kiểm tra quần áo. Tải gói yêu cầu, nếu cần, sử dụng thư viện (C50). Chạy tập dữ liệu huấn luyện thông qua C5.0 để lấy Mô hình 1, sử dụng mã được cung cấp tại đây.3 Lưu kết quả dưới dạng C5.

```
C5 <- C5.0(Phản hồi ~ Days.since.Purchase + Number.of.
Mua.Lượt truy cập + Bán hàng.per.Lượt truy cập, dữ liệu = clothing_train)
```

Tiếp theo, tập hợp con các biến dự đoán từ tập dữ liệu thử nghiệm vào khung dữ liệu của chính chúng.

```
test.X <- subset(x = clothing_test, select = c("Days.
kể từ.Mua hàng",
          "Number.of.Purchase.Lượt truy cập", "Doanh số.per.Lượt truy cập"))
```

Lệnh subset() nhận đầu vào x = clothing_test để chỉ định đối tượng nào cần tập hợp con. Đối với trường hợp của chúng ta, chúng ta đang đặt con từ tập dữ liệu clothing_test . lựa chọn đầu vào chỉ định tên của các biến để tập hợp con từ dữ liệu. Chúng tôi đang đặt con tất cả ba biến dự đoán, vì vậy chúng tôi đặt tên của các biến đó trong dấu ngoặc kép và liệt kê chúng bên trong lệnh c() . Chúng tôi lưu tập dữ liệu kết quả dưới dạng test.X.

Bây giờ, chúng tôi đã sẵn sàng để chạy dữ liệu thử nghiệm thông qua mô hình dữ liệu huấn luyện (Mô hình 1).

```
ypred <-predict(object = C5, newdata = test.X)
```

Lệnh dự đoán () yêu cầu đầu vào là mô hình bạn đang sử dụng để đưa ra dự đoán, đối tượng được gắn nhãn và dữ liệu bạn đang sử dụng để đưa ra dự đoán, dữ liệu mới được gắn nhãn. Trong trường hợp của chúng tôi, chúng tôi đang sử dụng C5 từ tập dữ liệu huấn luyện làm mô hình của mình và các biến dự đoán từ tập dữ liệu thử nghiệm.

Đầu ra của lệnh predict() khi sử dụng mô hình C5.0 làm đối tượng là danh sách các nhãn biến mục tiêu được dự đoán, một nhãn cho mỗi bản ghi trong tập dữ liệu thử nghiệm. Chúng tôi lưu chuỗi dự đoán này dưới tên ypred. Vì biến phản hồi có thể có trong mô hình C5.0 có các giá trị 0 và 1, nên nội dung của ypred sẽ là các giá trị 0 hoặc 1 cho mỗi bản ghi trong tập dữ liệu clothing_test .

Bây giờ chúng ta đã có các dự đoán, chúng ta có thể so sánh chúng với các giá trị thu nhập thực tế trong tập dữ liệu thử nghiệm. Chúng tôi làm điều này bằng cách sử dụng một bảng.

```
t1 <- table(clothing_test$Response, ypred)
row.names(t1) <- c("Thực tế: 0", "Thực tế: 1")
```

3Để biết giải thích về mã, hãy xem phần R trên C5.0 trong Chương 6.

Machine Translated by Google


```
colnames(t1) <- c("Dự đoán: 0", "Dự đoán: 1")
t1 <- addmargins(A = t1, FUN = danh sách (Tổng = tổng), yên tĩnh =
ĐÚNG VẬY); t1
```

Chúng tôi tự xây dựng bảng bằng cách sử dụng lệnh table() , theo sau là biến mục tiêu từ bộ dữ liệu thử nghiệm clothing_test$Response cho các hàng và các giá trị biến mục tiêu được xác định trước ypred cho các cột. Để rõ ràng, chúng tôi thêm tên hàng và cột vào bảng để phân biệt giữa giá trị thực và giá trị dự đoán.
Chúng tôi sử dụng row.names() để đặt tên cho các hàng "Thực tế: 0" và "Thực tế: 1" và colnames() để đặt tên cho các cột "Dự đoán: 0" và "Dự đoán: 1" để làm cho bảng kết quả dễ đọc nhất có thể. Lệnh addmargins() , như chúng ta đã thấy trước đây, thêm một hàng và cột Tổng cộng . Bảng kết quả tương đương với bảng 7.3.

7.7 KẾ TOÁN CHI PHÍ LỖI KHÔNG CÔNG BẰNG

Bây giờ, giả định nội tại trong Mô hình 1 là hai loại lỗi phân loại, dương tính giả và âm tính giả, đều tốn kém như nhau. Tuy nhiên, với tư cách là một nhà bán lẻ quần áo, với hàng triệu đô la có khả năng bị đe dọa, chúng ta cần tự hỏi mình, giả định này có hợp lệ không? Hãy để chúng tôi điều tra, bằng cách thừa nhận một số chi phí liên quan đến các trường hợp dự phòng khác nhau. Những điều này được thể hiện trong Bảng 7.5.

Cơ sở lý luận cho các chi phí này như sau:

- Tiêu cực thực sự. Điều này thể hiện một người không phản hồi được phân loại chính xác là người không phản hồi. Nhà bán lẻ sẽ không gặp khó khăn khi liên hệ với khách hàng này, người sẽ không trả lời nếu được liên hệ. Không có tiền bị mất hoặc đạt được.

  Chi phíTN = $0.
- Dương Tính Giả. Điều này thể hiện một người không phản hồi được phân loại không chính xác là người phản hồi. Đây không phải là lỗi quá nghiêm trọng đối với nhà bán lẻ, vì chi phí liên hệ với mỗi khách hàng, qua điện thoại và gửi thư, chẳng hạn là 10 đô la.

  Chi phíFP = $10.
- Âm tính giả. Điều này cho thấy một người phản hồi tích cực bị phân loại không chính xác là người không phản hồi. Buồn, nhưng không có chi phí phát sinh trực tiếp. Chi phíFN = $0.
- Tích cực thực sự. Điều này thể hiện một người phản hồi tích cực được phân loại chính xác là người phản hồi tích cực. Nhà bán lẻ sẽ liên hệ với khách hàng này và khách hàng sẽ đến và chi tiêu trung bình khoảng 40 đô la. Vì lợi nhuận bằng chi phí âm, chúng ta có CostTP = $40.

BẢNG 7.5 Ma trận chi phí cho nhà bán lẻ

		Danh mục dự đoán	
		0	1
danh mục thực tế	0	Chi phíTN = $0	CostFP = $10
	1	CostFN = $0	CostTP = $40

Machine Translated by Google

BẢNG 7.6 Ma trận chi phí đã điều chỉnh đối với bên cho vay

		Danh mục dự đoán	
		0	1
danh mục thực tế	0	Chi phí TN = 0	Chi phíFP = 1
	1	Chi phíFN = 4	TP chi phí = 0

Chúng ta có thể thêm 40 đô la vào hàng dưới cùng, sau đó chia tất cả các ô cho 10 đô la, sao cho ma trận chi phí đã điều chỉnh của chúng ta trở thành Bảng 7.6.

Nói cách khác, chi phí âm tính giả của nhà bán lẻ cao gấp bốn lần chi phí dương tính giả. Vì vậy, rõ ràng, giả định của Mô hình 1 rằng hai lỗi gây thiệt hại như nhau là không hợp lệ. Thay vào đó, chúng ta nên chuyển sang một mô hình mới, Mô hình 2, mô hình này kết hợp các mức chi phí lỗi không đồng đều trong Bảng 7.6.

7.7.1 Tính chi phí lỗi không đồng đều bằng R

Khi bạn tạo ma trận chi phí như trong Bảng 7.6, bạn có thể thêm ma trận vào mô hình C5.0 của mình. Đầu tiên, tự tạo ma trận.

```
cost.C5 <- matrix(c(0,4,1,0), byrow = TRUE, ncol=2)
dimnames(cost.C5) <- list(c("0", "1"), c("0", "1"))
```

Điều quan trọng cần lưu ý là C5.0 xác định ma trận chi phí của nó bằng cách sử dụng các hàng làm giá trị dự đoán và các cột làm giá trị thực (chi tiết về điều này có trong trang trợ giúp của C5.0, có thể tìm thấy bằng cách chạy ? C5 .0). Điều này ngược lại với cách xác định ma trận chi phí trong Bảng 7.6. Khi chúng ta tạo ma trận chi phí cho mô hình C5.0, nó sẽ có số "1" trong đó số "4" nằm trong Bảng 7.6 và số "4" trong số "1". Lưu ma trận chi phí dưới dạng cost.C5. Chỉ định dimnames() của ma trận chi phí sẽ đảm bảo rằng thuật toán C5.0 xác định chính xác các chi phí khác nhau.

Bây giờ chúng ta chạy lại mô hình C5.0, lần này thêm ma trận sai số không bằng nhau chi phí.

```
C5.chi phí <- C5.0(Phản hồi ~ Days.since.Purchase +
Number.of.Purchase.Visits + Sales.per.Visit, dữ liệu = clothing_train,
chi phí = cost.C5)
```

Mã ở trên khác với mã R trước đó cho C5.0 theo hai cách. Đầu tiên, chúng tôi lưu mô hình dưới một tên khác, vì vậy chúng tôi không làm mất thông tin từ mô hình trước đó của chúng tôi. Thứ hai, chúng tôi thêm chi phí đầu vào = cost.C5 vào C5.0()
lệnh thêm ma trận chi phí vào mô hình.

4Để biết chi tiết về lý do đằng sau những hành động này, vui lòng xem Chương 16 về Khai thác dữ liệu và phân tích dự đoán.

Machine Translated by Google

7.8 SO SÁNH CÁC MÔ HÌNH CÓ VÀ KHÔNG CÓ CHI PHÍ LỖI KHÔNG CÔNG BẰNG

Khi chúng tôi phát triển Mô hình 2 bằng R (xem phần trước để biết chi tiết về mã) và đánh giá nó, chúng tôi thu được bảng dự phòng như trong Bảng 7.7.

Vì chi phí lỗi cho Mô hình 2 là không bằng nhau nên các biện pháp đánh giá được giới thiệu trước đó hiện được thay thế bằng một biện pháp đánh giá mới, chi phí mô hình trên mỗi bản ghi (hoặc nghịch đảo cộng của nó, lợi nhuận mô hình tổng thể trên mỗi bản ghi).

Mô hình Chi phí trên mỗi Bản ghi và Mô hình Lợi nhuận trên mỗi Bản ghi

Khi chi phí lỗi không bằng nhau, một trong những biện pháp đánh giá quan trọng nhất cho mục đích lựa chọn mô hình là chi phí mô hình trên mỗi bản ghi. Đầu tiên, tính Chi phí mô hình tổng thể, như sau:

$$\text{Chi phí mô hình tổng thể} = \text{Chi phí}_{TN} \cdot TN + \text{Chi phí}_{FP} \cdot FP + \text{Chi phí}_{FN} \cdot FN + \text{Chi phí}_{TP} \cdot TP.$$

Thông thường, chi phí lỗi đối với phủ định thực và phủ định sai bằng 0, vì không có hành động nào được thực hiện. Trong trường hợp này, chi phí mô hình tổng thể đơn giản hóa thành:

$$\text{Chi phí mô hình tổng thể} = \text{Chi phí}_{FP} \cdot FP + \text{Chi phí}_{TP} \cdot TP.$$

Sau đó,

$$\text{Chi phí mô hình cho mỗi bản ghi} = \frac{\text{Chi phí mô hình tổng thể}}{GT}$$

Cuối cùng,

$$\text{Mô hình P lợi nhuận trên mỗi bản ghi} = -\text{Mô hình Chi phí trên mỗi bản ghi}$$

Đối với lựa chọn mô hình, hãy chọn mô hình giúp giảm thiểu chi phí mô hình trên mỗi bản ghi hoặc ngược lại, tối đa hóa lợi nhuận của mô hình. Điều quan trọng là phải báo cáo ghi lại chi phí hoặc lợi nhuận, bởi vì tổng chi phí thay đổi theo kích thước của tập dữ liệu.

Sử dụng các chi phí được mô tả trong Bảng 7.5, chúng tôi tính toán tổng chi phí mô hình đối với Mô hình 1 và Mô hình 2 như sau:

$$\begin{aligned}
&\text{Chi phí mô hình tổng thể}_{\text{Người mẫu 1}} \\
&= \text{Chi phí}_{FP} \cdot FP + \text{Chi phí}_{TP} \cdot TP \\
&= \$441 \cdot 10\,544 + \cdot -\$40 \\
&= -\$17\,350
\end{aligned}$$

BẢNG 7.7 Bảng dự phòng để đánh giá Mô hình 2, kết hợp bất đối xứng chi phí lỗi

		Danh mục dự đoán		
		0	1	Tổng cộng
danh mục thực tế	0	TN = 7163	FP = 2451	TÂN = 9614
	1	PN = 618	TP = 1322	TAP = 1940
	Tổng cộng	TPN = 7781	TPP = 3773	GT = 11,554

Machine Translated by Google

$$\begin{aligned}
&\text{Chi phí mô hình tổng thể}_{\text{Người mẫu 2}} \\
&= \text{Chi phí FP}_{FP} + \text{Chi phí TP}_{TP} \\
&= \$2451 \cdot 10\ 1322 + \$40 \cdot \\
&= \$28\ 370
\end{aligned}$$

Sau đó, lợi nhuận mô hình trên mỗi khách hàng cho mỗi mô hình được đưa ra là:

$$\text{Lợi nhuận trên mỗi khách hàng}_{\text{người mẫu 1}} = \frac{\text{Chi phí mô hình tổng thể}_{\text{Người mẫu 1}}}{GT} = \frac{\$17\ 350}{11\ 554} = \$1.5016$$

$$\text{Lợi nhuận trên mỗi khách hàng}_{\text{người mẫu 2}} = \frac{\text{Chi phí mô hình tổng thể}_{\text{Người mẫu 2}}}{GT} = \frac{\$28\ 370}{11\ 554} = \$2.4554$$

Nói cách khác, chúng tôi đã tăng lợi nhuận của nhà bán lẻ lên gần 64%, đơn giản chỉ từ chiếm các chi phí lỗi không đồng đều!

7.9 CHI PHÍ LỖI DO DỮ LIỆU

Trong thời đại dữ liệu lớn này, các doanh nghiệp nên tận dụng thông tin trong cơ sở dữ liệu hiện có của họ để giúp khám phá các mô hình dự đoán tối ưu. Nói cách khác, như một giải pháp thay thế cho việc ấn định chi phí lỗi vì "các giá trị chi phí này có vẻ phù hợp với chuyên gia tư vấn của chúng tôi" hoặc "đó là cách chúng tôi luôn mô hình hóa chúng", thay vào đó, chúng tôi khuyên bạn nên lắng nghe dữ liệu và học hỏi từ chính dữ liệu đó chi phí lỗi nên là bao nhiêu. Hãy để chúng tôi minh họa sức mạnh của chi phí lỗi dựa trên dữ liệu bằng cách tiếp tục ví dụ của chúng tôi.

Hãy nhớ lại rằng chi phí khác không duy nhất của chúng tôi là CostFP = $10 và CostTP = $40. Tuy nhiên, may mắn thay, chúng tôi có quyền truy cập vào dữ liệu sẽ cho chúng tôi ý tưởng tốt hơn về CostTP, cụ thể là Công cụ dự đoán Doanh số trên mỗi lượt truy cập. Công cụ dự đoán này cung cấp số tiền trung bình chi cho mỗi lượt truy cập của mỗi khách hàng. Vì vậy, nếu chúng tôi tính toán Doanh số bán hàng trung bình trên mỗi lượt truy cập của tất cả khách hàng, thì chúng tôi có thể sử dụng giá trị này làm ước tính tốt hơn về số tiền trung bình mà khách hàng sẽ đến và chi tiêu. Ước tính trước đây của chúng tôi là 40 đô la. Nhưng Doanh số trung bình trên mỗi lượt truy cập cho tất cả các bản ghi trong tập dữ liệu đào tạo là $113,58. Do đó, chi phí dựa trên dữ liệu hoặc dựa trên dữ liệu của chúng tôi đối với các kết quả tích cực thực sự được cập nhật thành CostTP = $113,58. Thật không may, không có dữ liệu tương tự để chúng tôi ước tính tốt hơn CostFP = $10, do đó vẫn ở mức $10.

Điều này cho chúng ta ma trận chi phí sửa đổi trong Bảng 7.8.

BẢNG 7.8 Ma trận chi phí dựa trên dữ liệu cho bài toán cửa hàng quần áo

		Danh mục dự đoán	
		0	1
danh mục thực tế	0	Chi phíTN = $0	Chi phíFP = $10
	1	Chi phíFN = $0	CostTP = $113,58

Machine Translated by Google

BẢNG 7.9 Ma trận chi phí dựa trên dữ liệu được đơn giản hóa cho bài toán cửa hàng quần áo.

		Danh mục dự đoán	
		0	1
danh mục thực tế	0	0	Chi phíFP = 1
	1	Chi phíFN = 11.358	0

BẢNG 7.10 Bảng dự phòng cho Mô hình 3 với chi phí lỗi dựa trên dữ liệu

		Danh mục dự đoán		
		0	1	Tổng cộng
danh mục thực tế	0	TN = 4237	FP = 5377	TÂN = 9614
	1	FN = 201	TP = 1739	TAP = 1940
	Tổng cộng	TPN = 4438	TPP = 7116	GT = 11,554

Trừ CostTP = $113,58 từ hàng dưới cùng rồi chia cho mỗi ô bằng 10 đô la cho chúng ta ma trận chi phí dựa trên dữ liệu được đơn giản hóa trong Bảng 7.9.

Do đó, âm tính giả của chúng tôi đắt gấp 11,358 lần so với dương tính giả của chúng tôi.

Một mô hình C5.0 khác được phát triển bằng cách sử dụng tập huấn luyện, được gọi là Mô hình 3, sau đó được đánh giá bằng cách sử dụng tập dữ liệu thử nghiệm. Bảng dự phòng kết quả được thể hiện trong Bảng 7.10.

Chúng tôi tính toán chi phí mô hình tổng thể cho Mô hình 3 như sau:

$$\begin{aligned}\text{Chi phí mô hình tổng thể}_{\text{Người mẫu}3} &= \text{Chi phí FP}_{FP} \cdot \text{Chi phí TP}_{TP} \\ &= \$5377 \cdot 10 + \$1739 \cdot -\$113.58 \\ &= -\$143,745.62\end{aligned}$$

Sau đó, lợi nhuận mô hình trên mỗi khách hàng cho Mô hình 3 được đưa ra là:

$$\text{Lợi nhuận trên mỗi khách hàng}_{\text{Người mẫu}3} = \frac{-\text{Chi phí mô hình tổng thể}_{\text{Người mẫu}3}}{GT} = \frac{\$143,745.62}{11,554} = \$12.4412$$

Đây là một sự gia tăng lớn về lợi nhuận so với các mô hình trước đó. Tuy nhiên, phần lớn mức tăng này là do mức tăng CostTP dựa trên dữ liệu lên $113,58. Để công bằng, chúng ta nên so sánh cả ba mô hình theo chi phí mô hình mới. Điều này được thể hiện cho tất cả các biện pháp đánh giá của chúng tôi trong Bảng 7.11.

Bảng 7.11 chứa các biện pháp đánh giá cho Mô hình 1–3, với kết quả của mô hình hoạt động tốt nhất được in đậm. Lưu ý rằng Mô hình 1, mô hình có lợi nhuận thấp nhất (không có chi phí lỗi), là mô hình chính xác nhất, ở mức 84,10%, trong khi Mô hình 3, mô hình có lợi nhuận cao nhất của chúng tôi (có chi phí lỗi do dữ liệu), là mô hình kém chính xác nhất, ở mức chỉ 51,72%. Do đó, độ chính xác không phải là số liệu thích hợp để so sánh các mô hình có chi phí lỗi không bằng nhau.

Machine Translated by Google

BẢNG 7.11 Số liệu đánh giá mô hình cho tất cả các mô hình

	Mẫu C5.0		
Sự đánh giá Đo lường	Mẫu 1: Không chi phí lỗi	Mô hình 2: Chi phí lỗi gấp 4 lần	Mô hình 3: Lỗi chi phí 11,358x
Sự chính xác	0,8410	0,7344	0,5172
tỷ lệ lỗi	0,1590	0,2656	0,4828
Nhạy cảm	0,2804	0,6814	0,8964
độ đặc hiệu	0,9541	0,7451	0,4407
Độ chính xác	0,5523	0,3504	0,2444
F1	0,3720	0,4628	0,3841
F2	0,3110	0,5731	0,5845
F0.5	0,4626	0,3881	0,2860
Tổng chi phí mô hình	$57.377,52	$125.642,76	$143.745,62
Lợi nhuận trên mỗi khách hàng	$4,97	$10,87	$12,44

Bởi vì mỗi kết quả tích cực thực sự mang lại cho chúng tôi 113,58 đô la, nên các mô hình có xu hướng đưa ra nhiều dự đoán tích cực hơn đã hoạt động tốt hơn. Do đó, độ nhạy (thu hồi), tỷ lệ của tất cả những người phản hồi tích cực mà mô hình nắm bắt được, hóa ra lại quan trọng hơn tính đặc hiệu (nắm bắt những người không phản hồi). Do đó, mô hình hoạt động tốt nhất của chúng tôi có độ nhạy cao nhất, trong khi mô hình hoạt động kém nhất của chúng tôi có độ nhạy thấp nhất. Mối quan hệ ngược lại được tổ chức cho tính đặc hiệu.

Độ chính xác dường như cũng không quan trọng lắm, vì độ chính xác chắc chắn của Mô hình 1 một phần là do nó có ít dự đoán tích cực. F2 , ưu tiên thu hồi (độ nhạy) hơn độ chính xác, cao nhất với Mô hình 3, trong khi F0.5, ưu tiên độ chính xác hơn thu hồi, cao nhất khi mất đi Mô hình 1.

Cuối cùng, chúng ta cũng nên đảm bảo rằng các mô hình của chúng ta hoạt động tốt hơn các mô hình cơ bản: Mô hình Hoàn toàn Tích cực và Mô hình Hoàn toàn Tiêu cực. Vì không có khách hàng nào được liên hệ trong Mô hình Tất cả Phủ định nên không có lợi nhuận. Mặt khác, Mô hình Tất cả Tích cực đã hoạt động khá tốt, với Lợi nhuận Mô hình Tổng thể là 124.205,20 đô la và Lợi nhuận trên mỗi khách hàng là 10,75 đô la, gần như đánh bại Mô hình 2 ở trên.

Tóm lại, các nhà khoa học dữ liệu phải luôn đánh giá các mô hình của họ. Trong chương này, chúng ta đã học một loạt các thước đo để đánh giá các mô hình phân loại. Chúng tôi phát hiện ra rằng chi phí lỗi không phải lúc nào cũng bằng nhau và khi không bằng nhau, chi phí mô hình cho mỗi bản ghi có thể là thước đo tốt nhất. Cuối cùng, chúng tôi đã minh họa cách chi phí lỗi dựa trên dữ liệu có thể nâng cao hơn nữa lợi nhuận của các mô hình phân loại của chúng tôi.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Giải thích sự khác biệt giữa đánh giá mô hình và xác nhận mô hình.
2. Bảng dự phòng bao gồm những gì?

Machine Translated by Google

3. Chứng minh rằng GT = TPN+TPP và TAP = GT TAN.

4. Chứng tỏ rằng Tỷ lệ lỗi = 1 Độ chính xác.

5. Đo độ nhạy và độ đặc hiệu để làm gì?

6. Giải thích Phương pháp Đánh giá Mô hình.

7. Tại sao chúng tôi chọn mô hình Tất cả phủ định làm cơ sở để hiệu chỉnh độ chính xác của Mô hình 1, thay vì mô hình Tất cả tích cực?

8. Khi chi phí lỗi không bằng nhau, thước đo đánh giá quan trọng nhất đối với mục đích của việc lựa chọn mô hình?

9. Giải thích tại sao một nhà phân tích ngây thơ lại thích Mô hình 1 hơn Mô hình 2 một cách sai lầm.

10. Đối với các mô hình Tất cả Tích cực và Tất cả Tiêu cực, hãy tính toán các chỉ số đánh giá từ Bảng 7.11.

LÀM VIỆC VỚI DỮ LIỆU

Đối với Bài tập 11–22, hãy làm việc với clothing_data_driven_training và clothing_data_ bộ dữ liệu drive_test . Sử dụng R để giải quyết từng vấn đề.

11. Sử dụng tập dữ liệu đào tạo, tạo mô hình C5.0 (Mô hình 1) để dự đoán Phản hồi của khách hàng bằng Số ngày kể từ khi mua, Số lượt truy cập mua hàng và Doanh số trên mỗi lượt truy cập. Có được các câu trả lời dự đoán.

12. Đánh giá Mô hình 1 bằng bộ dữ liệu thử nghiệm. Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán của Phản hồi.

13. Đối với Mô hình 1, tóm tắt lại Bảng 7.4 từ văn bản, tính toán tất cả các biện pháp đánh giá mô hình được trình bày trong bảng. Gọi bảng này là Bảng đánh giá mô hình. Để lại không gian cho Mô hình 2 và 3.

14. Giải thích rõ ràng và đầy đủ từng biện pháp đánh giá Mô hình 1 từ Bảng Đánh giá Mô hình.

15. Tạo một ma trận chi phí, được gọi là ma trận chi phí 4x , cho biết kết quả dương tính giả tệ gấp bốn lần kết quả âm tính giả.

16. Sử dụng tập dữ liệu đào tạo, xây dựng mô hình C5.0 (Mô hình 2) để dự đoán Phản hồi của khách hàng bằng cách sử dụng Số ngày kể từ khi mua, Số lượt truy cập mua hàng và Doanh số mỗi lượt truy cập, sử dụng ma trận chi phí 4x .

17. Đánh giá dự đoán của bạn từ Mô hình 2 bằng cách sử dụng các giá trị phản hồi thực tế từ tập dữ liệu thử nghiệm. Thêm Chi phí Mô hình Tổng thể và Lợi nhuận trên mỗi Khách hàng vào Bảng Đánh giá Mô hình. Tính toán tất cả các biện pháp từ Bảng đánh giá mô hình.

18. So sánh các biện pháp đánh giá từ Mô hình 1 và Mô hình 2 bằng cách sử dụng ma trận chi phí 4x . Thảo luận về điểm mạnh và điểm yếu của từng mô hình.

19. Xây dựng ma trận chi phí dựa trên dữ liệu được đơn giản hóa như sau:

    Một. Lấy giá trị trung bình của biến Doanh số trên mỗi lượt truy cập từ tập dữ liệu đào tạo và đặt giá trị âm của giá trị đó thành "chi phí" của giá trị dương thực sự. Đặt chi phí dương tính giả bằng $10.

Machine Translated by Google

b. Xây dựng ma trận chi phí phù hợp và đơn giản hóa nó để thu được dữ liệu đơn giản hóa-
ma trận chi phí định hướng.

20. Sử dụng tập huấn luyện, xây dựng mô hình C5.0 (Mô hình 3) để dự đoán Phản hồi của khách hàng
sử dụng Số ngày kể từ khi mua hàng, Số lượt truy cập mua hàng và Doanh số trên mỗi lượt truy cập, sử dụng dữ liệu-
ma trận chi phí định hướng.

21. Điền vào Bảng đánh giá mô hình các biện pháp đánh giá cho Mô hình 3, sử dụng
ma trận chi phí dựa trên dữ liệu.

22. So sánh Mô hình 1, Mô hình 2 và Mô hình 3 bằng Bảng Đánh giá Mô hình.

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu adult_ch6_training và adult_ch6_test .
Sử dụng R để giải quyết từng vấn đề.

23. Sử dụng tập dữ liệu huấn luyện, tạo mô hình C5.0 (Mô hình 1) để dự đoán Thu nhập của khách hàng
sử dụng Tình trạng hôn nhân và lãi và lỗ vốn. Có được các câu trả lời dự đoán.

24. Đánh giá Mô hình 1 bằng bộ dữ liệu thử nghiệm. Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán
của Thu nhập.

25. Đối với Mô hình 1, tóm tắt lại Bảng 7.4 từ văn bản, tính toán tất cả các biện pháp đánh giá mô hình được trình bày
trong bảng. Gọi bảng này là Bảng đánh giá mô hình. Để lại không gian cho Mô hình 2.

26. Giải thích rõ ràng và đầy đủ từng biện pháp đánh giá Mô hình 1 từ Bảng Đánh giá Mô hình.

27. Tạo một ma trận chi phí, được gọi là ma trận chi phí 3x , cho biết kết quả dương tính giả tệ gấp bốn lần kết quả âm
tính giả.

28. Sử dụng tập dữ liệu huấn luyện, xây dựng mô hình C5.0 (Mô hình 2) để dự đoán Thu nhập của khách hàng
sử dụng Tình trạng hôn nhân và lãi và lỗ vốn, sử dụng ma trận chi phí 3x .

29. Đánh giá dự đoán của bạn từ Mô hình 2 bằng cách sử dụng các giá trị phản hồi thực tế từ tập dữ liệu thử nghiệm. Thêm
Chi phí Mô hình Tổng thể và Lợi nhuận trên mỗi Khách hàng vào Bảng Đánh giá Mô hình. Tính toán tất cả các biện
pháp từ Bảng đánh giá mô hình.

30. So sánh các biện pháp đánh giá từ Mô hình 1 và Mô hình 2 bằng cách sử dụng ma trận chi phí 3x .
Thảo luận về điểm mạnh và điểm yếu của từng mô hình.

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu Loans_Training và Loans_Test . Sử dụng R để giải quyết từng vấn đề.

31. Sử dụng tập dữ liệu đào tạo, tạo mô hình C5.0 (Mô hình 1) để dự đoán Phê duyệt của người đăng ký khoản vay bằng cách
sử dụng Tỷ lệ Nợ trên Thu nhập, Điểm FICO và Số tiền Yêu cầu. Có được các câu trả lời dự đoán.

32. Đánh giá Mô hình 1 bằng bộ dữ liệu thử nghiệm. Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán
của Phê duyệt.

33. Đối với Mô hình 1, tóm tắt lại Bảng 7.4 từ văn bản, tính toán tất cả các biện pháp đánh giá mô hình được trình bày
trong bảng. Gọi bảng này là Bảng đánh giá mô hình. Để lại không gian cho Mô hình 2.

Machine Translated by Google

34. Giải thích rõ ràng và đầy đủ từng biện pháp đánh giá Mô hình 1 từ Bảng Đánh giá Mô hình.

35. Làm như sau để xây dựng ma trận chi phí dựa trên dữ liệu được đơn giản hóa.

Một. Tính giá trị trung bình của Lãi suất trên mỗi người xin vay từ tập dữ liệu huấn luyện. Bộ giá trị âm của giá trị đó là "chi phí" của giá trị dương thực sự.

b. Tính toán Số tiền yêu cầu trung bình cho mỗi người đăng ký khoản vay từ tập dữ liệu đào tạo. Bộ giá trị này là chi phí của một dương tính giả.

c. Có được ma trận chi phí dựa trên dữ liệu được đơn giản hóa.

36. Sử dụng tập huấn luyện, xây dựng mô hình C5.0 (Mô hình 2) để dự đoán Phê duyệt của người xin vay bằng cách sử dụng Tỷ lệ nợ trên thu nhập, Điểm FICO và Số tiền yêu cầu, sử dụng ma trận chi phí dựa trên dữ liệu được đơn giản hóa.

37. Điền vào Bảng đánh giá mô hình các biện pháp đánh giá cho Mô hình 2, sử dụng ma trận chi phí dựa trên dữ liệu.

38. Giải thích rõ ràng và đầy đủ từng biện pháp đánh giá Mô hình 2 từ Bảng đánh giá Mô hình.

39. So sánh Mô hình 1 và Mô hình 2 bằng Bảng Đánh giá Mô hình. Thảo luận về điểm mạnh và điểm yếu của từng mô hình.

40. Chúng tôi đã kiếm được bao nhiêu tiền cho ngân hàng của mình bằng cách sử dụng chi phí lỗi dựa trên dữ liệu để đánh giá mô hình của chúng tôi?

Machine Translated by Google

chương số 8

PHÂN LOẠI NAÏVE BAYES

8.1 GIỚI THIỆU VỀ VỊNH NAÏVE

Tất nhiên, mô hình phân loại không bị giới hạn đối với cây quyết định. Có nhiều phương pháp phân loại khác, bao gồm phân loại Naïve Bayes. Các phương pháp phân loại Naïve Bayes dựa trên Định lý Bayes, được phát triển bởi Reverend Thomas Bayes.1 Định lý Bayes cập nhật kiến thức của chúng ta về các tham số dữ liệu bằng cách kết hợp kiến thức trước đây của chúng ta (được gọi là phân phối trước) với thông tin mới thu được từ dữ liệu được quan sát, dẫn đến tham số được cập nhật kiến thức (được gọi là phân phối sau).

8.2 ĐỊNH LÝ BAYES

Hãy xem xét một tập dữ liệu được tạo thành từ hai biến dự đoán X = X1 , X2 và một biến phản hồi Y, trong đó biến phản hồi nhận một trong ba giá trị lớp có thể có: y1 , y2 và y3 Mục tiêu của chúng ta là xác định giá trị nào trong số y1 , y2 và y3 có nhiều khả năng nhất đối với sự kết hợp cụ thể của các giá trị biến dự báo. Chúng ta hãy gọi sự kết hợp có nhiều khả năng xảy ra nhất này là X* = {X1 = x1 , X2 = x2 }.

Chúng ta có thể sử dụng Định lý Bayes để xác định lớp nào có khả năng nhất cho một sự kết hợp cụ thể của các giá trị biến dự báo bằng cách:

1. tính xác suất sau cho từng y1 , y2 và y3 , kết hợp các yếu tố dự đoán x1 và x2 và 2. chọn giá trị của y có xác suất sau cao nhất.

Đặt y* là một trong ba giá trị thế của Y. Định lý Bayes cho chúng ta biết:

$$p\left(Y = y^{*} \mid X = x^{*}\right) = \frac{p\left(X = x^{*} \mid Y = y^{*}\right) p\left(Y = y^{*}\right)}{p\left(X = x^{*}\right)} \tag{8.1}$$

1Thomas Bayes, Tiểu luận hướng tới giải quyết vấn đề trong Học thuyết về những thay đổi, Giao dịch triết học của Hiệp hội Hoàng gia Luân Đôn, 1793.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose. © 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

Bây giờ, p(Y = y*) đại diện cho kiến thức mà chúng ta có về khả năng xảy ra của giá trị lớp y* , thậm chí trước khi chúng ta bắt đầu. Vì thông tin đó được biết trước khi phân tích hậu môn của chúng tôi, chúng tôi gọi p(Y = y*) là xác suất trước. Thông tin trước này được kết hợp với p(X* Y = y*), đại diện cho cách dữ liệu hoạt động, khi phản hồi bằng y* . Mẫu số, p(X*), là cách dữ liệu hoạt động mà không cần tham chiếu đến các giá trị của lớp phản hồi, còn được gọi là xác suất cận biên của dữ liệu.

Kết quả của công thức, p(Y = y* X*), biểu thị thông tin hoặc ý tưởng mà chúng ta có về khả năng xảy ra giá trị lớp y* của chúng ta nếu chúng ta quan sát các giá trị biến dự đoán cụ thể X* . Vì thông tin được cập nhật từ P(Y = y*) sau khi kiểm tra dữ liệu, nên chúng tôi gọi p(Y = y* X*) là xác suất hậu nghiệm.

Nếu bạn không có kiến thức trước về các thông số thì sao? Trong trường hợp này, bạn có thể sử dụng phần trước không mang tính thông tin, cho biết rằng mọi giá trị của lớp đều có khả năng xảy ra như nhau. Bằng cách sử dụng tiên nghiệm không cung cấp thông tin, xác suất sau của bạn chỉ dựa trên dữ liệu.

8.3 TỐI ĐA MỘT GIẢ THUYẾT SAU

Làm thế nào chúng ta có thể sử dụng xác suất của Định lý Bayes để phân loại một bản ghi? Trong ví dụ của chúng tôi ở trên, chúng tôi có ba giá trị có thể có khác nhau của y*. Đối với một giá trị cố định , chúng tôi gọi của X* , hãy xác định xác suất Định lý Bayes cho từng giá trị trong số ba giá trị có thể có của Y:

$$p(Y=y_1 \mid X^*) = \frac{p(X^* \mid Y=y_1)\, p(Y=y_1)}{p(X^*)}$$

$$p(Y=y_2 \mid X^*) = \frac{p(X^* \mid Y=y_2)\, p(Y=y_2)}{p(X^*)}$$

$$p(Y=y_3 \mid X^*) = \frac{p(X^* \mid Y=y_3)\, p(Y=y_3)}{p(X^*)}$$

Giả thuyết hậu nghiệm tối đa cho chúng ta phân loại bản ghi X* là giá trị của Y có xác suất hậu nghiệm cao nhất. Nói cách khác, hãy chọn giá trị lớp của Y tương ứng với giá trị lớn nhất trong ba xác suất sau mà chúng tôi đã tính toán.

8.4 LOẠI ĐỘC LẬP CÓ ĐIỀU KIỆN

Nếu chúng ta có nhiều hơn một biến dự đoán, thì giả định về tính độc lập có điều kiện của lớp cho phép chúng ta viết p(X* Y = y*) dưới dạng tích của các sự kiện độc lập. Ví dụ: giả sử chúng ta có hai biến dự đoán X* = {X1 = x1 , X2 = x2 }. Để viết p(X* Y = y*), chúng ta sẽ viết p(X1 = x1 Y = y*)×p(X2 = x2 Y = y*). Chúng ta sẽ thấy minh họa về điều này trong Phần 8.5.

Machine Translated by Google

8.5 ÁP DỤNG PHÂN LOẠI NAÏVE BAYES

Chúng tôi sẽ sử dụng bộ dữ liệu wine_ flag_training và wine_flag_test để chứng minh cách chúng tôi sử dụng Naïve Bayes để phân loại một biến phản hồi. Giả sử chúng ta muốn dự đoán rượu vang có màu đỏ hay trắng dựa trên việc rượu có hàm lượng cồn và đường cao hay thấp. Giá trị hàm lượng cồn và đường được coi là thấp nếu chúng ở dưới mức trung bình của biến đó và cao nếu chúng ở trên mức trung bình.

Đầu tiên, chúng tôi xây dựng hai bảng dự phòng, một cho Type và Alcohol_flag và một cho Type và Sugar_flag. Nhớ lại rằng các giá trị lớp của biến mục tiêu tạo thành các hàng và các giá trị lớp của các biến dự đoán tạo thành các cột.
Bảng dự phòng cho Type và Alcohol_flag được hiển thị trong Hình 8.1, trong khi bảng dự phòng cho Type và Sugar_flag được hiển thị trong Hình 8.2.

Chúng ta có thể sử dụng Hình 8.1 và 8.2 để tính các giá trị cần thiết để thực hiện phân loại Naïve Bayes. Hãy để chúng tôi bắt đầu bằng cách kiểm tra biến phản hồi, Type. Loại có hai cấp độ: Đỏ và Trắng. Sử dụng một trong hai bảng dự phòng, chúng ta có thể tính toán xác suất trước của từng cấp độ Loại :

- p(Loại = Đỏ) = 500/1000 = 0,5
- p(Loại = Trắng) = 500/1000 = 0,5

Hai xác suất Loại tạo thành hai giá trị có thể có của p(Y), phân phối trước của Loại. Ví dụ: bây giờ chúng tôi biết rằng bất kỳ loại rượu nào được chọn ngẫu nhiên từ tập dữ liệu này đều có 50% khả năng là rượu vang đỏ.

Chúng ta sử dụng Hình 8.1 để tính xác suất cận biên của biến Alcohol_flag của biến dự báo. Alcohol_flag có hai cấp độ: Cao và Thấp. Hai giá trị này sẽ tạo thành phân phối của biến dự đoán đầu tiên, p(X1):

- p (Rượu • p _ cờ Cao = 486 1000) = 0 486 ⁄ = ·
  Rượu (_ cờ Thấp 514 1000) = 0 514 ⁄ = ·

Ví dụ, bây giờ chúng ta biết rằng một loại rượu được chọn ngẫu nhiên từ bộ dữ liệu này có xác suất chứa nồng độ cồn cao là 48,6%. Lưu ý rằng phân phối cận biên này không tính đến các giá trị phản hồi Y.

	Alcohol = High	Alcohol = Low	Total
Type = Red	218	282	500
Type = White	268	232	500
Total	486	514	1000

Hình 8.1 Bảng dự phòng từ R cho Type và Alcohol_flag.

	Sugar = High	Sugar = Low	Total
Type = Red	116	384	500
Type = White	300	200	500
Total	416	584	1000

Hình 8.2 Bảng dự phòng từ R cho Type và Sugar_flag.

Machine Translated by Google

Chúng tôi sử dụng Hình 8.2 để tính xác suất cận biên của biến dự báo Sugar_flag. Sugar_flag cũng có hai mức: Cao và Thấp. Hai giá trị này sẽ tạo thành phân phối của biến dự đoán thứ hai, p(X2):

- p Đường _ cờ Cao = 416 1000 0 416 / p Đường _ cờ Thấp 584 1000 0 584 =) = / = .

Ví dụ, bây giờ chúng ta biết rằng một loại rượu được chọn ngẫu nhiên từ bộ dữ liệu này có 41,6% xác suất có hàm lượng đường cao.

Còn về p(X* Y), xác suất có điều kiện của từng biến dự đoán với biến mục tiêu thì sao? Đối với mỗi biến dự đoán, chúng ta có bốn xác suất khác nhau, một xác suất cho mỗi cặp trong số bốn cặp mức biến dự báo và mục tiêu.

Từ Hình 8.1 và 8.2, chúng ta có thể tính bốn xác suất có điều kiện cho Alcohol_flag và Type, như sau:

- p Rượu • p _ / cờ Loại cao | Màu đỏ = 218 500 0 436 = .
Rượu • p Rượu _ 282 500 0 564 | / cờ Loại thấp Đỏ = .
- p Rượu _ 268 500 0 536 | / cờ Loại cao Trắng = .
(_ 232 500 0 464 | / cờ Loại thấp Trắng = .

Bây giờ chúng ta đã biết, ví dụ, nếu một loại rượu vang đỏ, nó có 56,4% khả năng có nồng độ cồn thấp, so với 43,6% khả năng có nồng độ cồn cao. Để hình dung sự khác biệt này, chúng ta có thể sử dụng biểu đồ thanh chuẩn hóa, như ở phía bên trái của Hình 8.1.

Bốn xác suất có điều kiện cho Sugar_flag và Type được đưa ra bên dưới.

- p Đường _ | p Đường Loại _ Cao Đỏ 116 500 0 232 324 500 0 768 • p
Đường _ cờ Cao Loại trắng 300 500 0 600 200 500 0 400 Đường _ = .
(= | =) = = . 0 6
(= | =) = = . 0 4

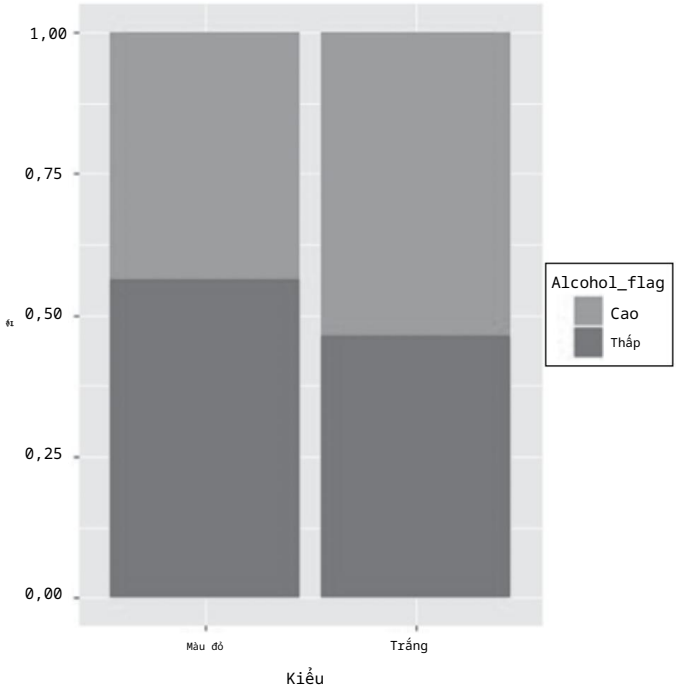
Bây giờ chúng ta đã biết, ví dụ, nếu một loại rượu vang đỏ, nó có 76,8% khả năng có hàm lượng đường thấp, so với 23,2% khả năng có hàm lượng đường cao. Để hình dung sự khác biệt này, chúng ta có thể sử dụng biểu đồ thanh chuẩn hóa, như ở phía bên tay phải của Hình 8.3.

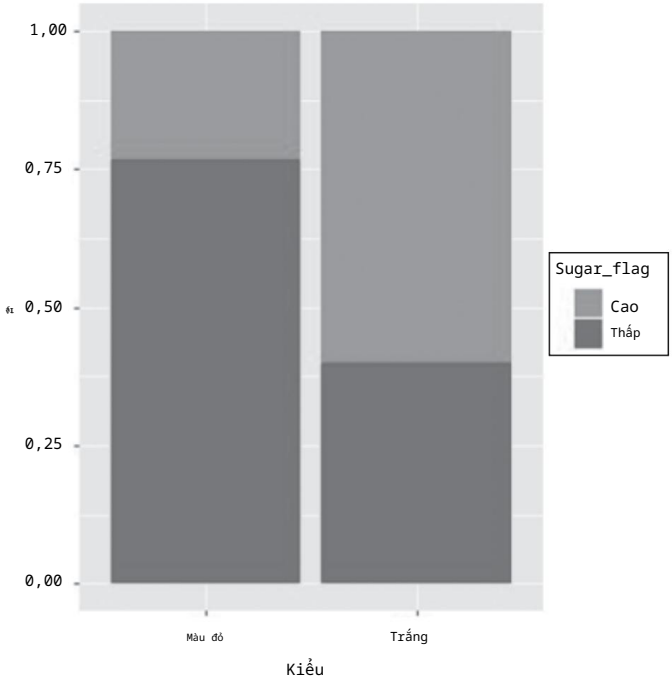
Bây giờ chúng ta có tất cả các giá trị của p(Y), p(X*) và p(X* Y), chúng ta sẵn sàng tính toán xác suất sau của từng cấp Loại, p(Y = y* X *). Nói cách khác, cuối cùng chúng ta cũng có thể giải quyết câu hỏi hiện tại: Naïve Bayes sẽ phân loại rượu như thế nào, dựa trên hàm lượng cồn và đường? Để tìm ra điều đó, chúng tôi sử dụng giả thuyết hậu nghiệm tối đa . Chúng tôi kiểm tra xác suất sau của từng Loại có thể đưa ra một giá trị cụ thể của Alcohol_flag và Sugar_flag và chọn Loại có xác suất xảy ra sau cao nhất.

Đầu tiên, chúng ta hãy xem xét một loại rượu có hàm lượng cồn thấp và ít đường. Sử dụng ký hiệu xác suất của chúng tôi, chúng tôi muốn biết từng điều sau đây:

- p (y = X1 | *) = (p Rượu đỏ cờ Đường thấp , _ cờ Thấp) (=
- p (y = X2 | *) = p White Alcohol flag Low Sugar , _ flag Low) =

Machine Translated by Google





Hình 8.3 Biểu đồ thanh được chuẩn hóa của Loại có lớp phủ Alcohol_flag (ở trên cùng) và Sugar_flag (ở dưới cùng).

Machine Translated by Google

Để tìm p(Red | Alcohol_flag = Low, Sugar_flag = Low), chúng ta thiết lập Bayes Định lý:

$$p\left(Y = y_1 \mid X^*\right) = \frac{p\left(X^* \mid Y = y_1\right)\left(Y = y_1\right)}{p\left(X^*\right)}$$

$$= \frac{p\left(\text{Cồn thấp, Đường thấp, Đỏ} \mid \text{Đỏ}\right) \times \left(\text{Đỏ}\right)}{p\left(\text{Rượu Alcohol_flag = Thấp, Đường Sugar_flag = Thấp}\right)}$$

$$= \frac{p\left(\text{Rượu Alcohol_flag = Thấp} \mid \text{Đỏ}\right) p\left(\text{Đường Sugar_flag = Thấp} \mid \text{Đỏ}\right) \times \left(\text{Đỏ}\right)}{p\left(\text{Rượu Alcohol_flag = Thấp}\right) p\left(\text{Đường Sugar_flag = Thấp}\right)}$$

Lưu ý rằng chúng tôi đang sử dụng giả định độc lập có điều kiện để đi đến bước cuối cùng này.

Chúng tôi đã tính toán mọi xác suất mà chúng tôi cần để giải phương trình này, vì vậy chúng tôi cắm từng giá trị vào và xử lý các con số:

$$\frac{0.5640 \times 0.7680 \times 0.5}{0.5140 \times 0.5840} = 0.7215$$

Xác suất để rượu có màu đỏ do hàm lượng cồn và đường thấp là khoảng 72,15%.

Tiếp theo, chúng ta cần tìm p(White| Alcohol_flag = Low, Sugar_flag = Low):

$$p\left(Y = y_2 \mid X^*\right) = \frac{p\left(X^* \mid Y = y_2\right)\left(Y = y_2\right)}{p\left(X^*\right)}$$

$$= \frac{p\left(\text{Rượu Alcohol_flag = Ít Đường Sugar_flag = Thấp} \mid \text{Trắng}\right) \times \left(\text{Trắng}\right)}{p\left(\text{Rượu Alcohol_flag = Thấp, Đường Sugar_flag = Thấp}\right)}$$

$$= \frac{p\left(\text{Rượu Alcohol_flag = Thấp} \mid \text{Trắng}\right) p\left(\text{Đường Sugar_flag = Thấp} \mid \text{Trắng}\right) \times \left(\text{Trắng}\right)}{p\left(\text{Rượu Alcohol_flag = Thấp}\right) p\left(\text{Đường Sugar_flag = Thấp}\right)}$$

Chúng tôi đã tính toán mọi xác suất mà chúng tôi cần để giải phương trình này, vì vậy chúng tôi cắm từng giá trị vào và xử lý các con số:

$$\frac{0.4640 \times 0.4040 \times 0.5}{0.5140 \times 0.5840} = 0.3092$$

Xác suất để rượu vang trắng có hàm lượng cồn và đường thấp là khoảng 30,92%. Vì xác suất sau của rượu có độ cồn thấp, ít đường có màu đỏ cao hơn xác suất sau của cùng một loại rượu có màu trắng, nên thuật toán Naïve Bayes sẽ phân loại rượu là màu đỏ.

Machine Translated by Google

Chúng ta hãy so sánh xác suất trước của rượu vang có màu đỏ với xác suất sau của rượu vang có màu đỏ do hàm lượng cồn và đường thấp. Xác suất để một loại rượu được chọn ngẫu nhiên có màu đỏ là p(Type = Red) = 50%. Tuy nhiên, sau khi chúng tôi tính đến việc loại rượu này có hàm lượng cồn và đường thấp, xác suất nó là rượu vang đỏ đã tăng lên 72,15%! Tại sao? Bởi vì dữ liệu cho chúng tôi biết rằng rượu vang đỏ có nhiều khả năng có nồng độ cồn thấp hơn so với rượu vang trắng (56,4% so với 46,4%) và chúng cũng có nhiều khả năng có hàm lượng đường thấp hơn so với rượu vang trắng. (76,8% so với 40%). Naïve Bayes đã tính đến thông tin này và sử dụng nó để nói rằng rượu vang ít đường và cồn có nhiều khả năng có màu đỏ hơn màu trắng.

Còn rượu có độ cồn cao và hàm lượng đường cao thì sao? Chúng tôi sử dụng các công thức tương tự như trên, thay đổi các giá trị X* để phản ánh các giá trị mới của chúng tôi. Chúng tôi muốn so sánh hai xác suất sau:

• p = (1) yX* = p Rượu | Đỏ (_ cờ Đường cao, _ cờ Cao) =

Y(• p = ()2yX* | p = Rượu trắng _ cờ Đường cao, _ cờ Cao) =

Chúng ta hãy bắt đầu với p(Đỏ | Alcohol_ cờ = Cao, Đường_ cờ = Cao).

$$p\left(Y\ y=X_1 \mid {}^{*}\right) = \frac{\mid \left(p\ X^{*}Y\ y p = Y_1 y\right)\ \left(= {}_1\right)}{p\left(X\ {}^{*}\right)}$$

$$= \frac{(\mid p\ \text{Rượu cờ Cao Cờ đường Cao Đỏ}\ p = \text{Đỏ} \qquad)\times \left(\quad \right)}{p\left(\text{Rượu f} \quad \text{_ tụt hậu Cao, _ Đường cao cờ} \qquad \right)}$$

$$= \frac{p\left(\text{Rượu _| _| cờ = Loại Cao Đỏ = p Đường cờ Loại Cao Đỏ}\ p\ \text{Đỏ} \qquad = \qquad \right)\times \left(\quad \right)}{p\left(\text{Rượu} \quad \text{_ cờ Cao p Đường cờ Cao _} \qquad = \qquad \right)}$$

Cắm các xác suất cụ thể vào công thức trên, chúng ta có được xác suất sau:

$$\frac{0.436 \times 0.232 \times 0.5}{0.486 \times 0.416} = 0.2502$$

Xác suất để rượu có màu đỏ do hàm lượng cồn và đường cao là khoảng 25,02%.

Tiếp theo, p(Trắng | Alcohol_flag = Cao, Sugar_flag = Cao).

$$p\left(Y\ y=X_2 \mid {}^{*}\right) = \frac{\mid \left(p\ X^{*}Y\ y p = Y_2 y\right)\ \left(= {}_2\right)}{p\left(X\ {}^{*}\right)}$$

$$= \frac{(p\ \text{Rượu cờ Cao Đường cờ, Cao Trắng}\ p = \text{Trắng} \quad \mid \qquad)\times \left(\quad \right)}{p\left(\text{rượu} \quad \text{ol} _ \text{cờ Cao Đường, _ cờ Cao} \qquad = \qquad \right)}$$

$$= \frac{p\left(\text{Rượu} \quad \text{_ cờ cao T| ype Trắng p Đường cờ Loại cao Trắng}\ p = \text{Trắng} \quad \mid \qquad = \qquad \right)\times \left(\quad \right)}{p\left(\text{Rượu} \quad \text{_ cờ Cao p Đường cờ Cao _} \qquad = \qquad \right)}$$

Machine Translated by Google

Cắm các xác suất cụ thể vào công thức trên, chúng ta có được xác suất sau:

$$\frac{0.536 \times 0.6 \times 0.5}{0.486 \times 0.416} = 0.7953$$

Xác suất để rượu vang trắng có hàm lượng cồn và đường cao là khoảng 79,53%. Vì xác suất sau của rượu có độ cồn cao, rượu có hàm lượng đường cao có màu trắng cao hơn xác suất sau của cùng một loại rượu có màu đỏ, nên thuật toán Naïve Bayes sẽ phân loại rượu là màu trắng.

Có hai sự kết hợp nữa của các biến dự đoán mà chúng tôi chưa đề cập đến: lượng cồn thấp và lượng đường cao, và lượng đường cao và lượng cồn thấp. Phân loại Naïve Bayes cho rượu có độ cồn thấp, đường cao có màu trắng và phân loại Naïve Bayes cho rượu có độ cồn cao, đường thấp có màu đỏ. Chi tiết về các tính toán xác suất sau cho các loại rượu này được để lại dưới dạng bài tập. Tổng hợp tất cả lại với nhau, mô hình phân loại Naïve Bayes của chúng tôi có thể được tóm tắt trong Bảng 8.1.

Chúng ta có thể đánh giá mô hình Naïve Bayes mà chúng ta đã khám phá trong ví dụ trên bằng cách sử dụng tập dữ liệu thử nghiệm. Đối với ví dụ này, chúng tôi đánh giá mô hình được xây dựng bằng wine_ tập dữ liệu flag_training sử dụng dữ liệu wine_flag_test . Bảng dự phòng cho các giá trị thực tế và dự đoán của Loại được hiển thị trong Hình 8.4.

Từ hình 8.4, ta thấy độ chính xác của mô hình là (464+1082)/2345 = 0,6593. Sử dụng mô hình Naïve Bayes của mình, chúng tôi dự đoán chính xác loại rượu trong 65,93% thời gian. Mô hình của chúng tôi phân loại đúng rượu vang đỏ 464/585 = 0,7932 hay 79,32% thời gian và phân loại đúng rượu vang trắng 1082/1760 = 0,6148 hay 61,48% thời gian. Vì một nửa rượu vang của chúng tôi có màu đỏ và một nửa có màu trắng, nên độ chính xác cơ bản của chúng tôi là 50% và do đó, mô hình Naïve Bayes của chúng tôi sẽ thực hiện mô hình cơ sở.

BẢNG 8.1 Tóm tắt mô hình Naïve Bayes để dự đoán loại rượu dựa trên hàm lượng cồn và đường

Nếu rượu có.		• sau đó chúng tôi phân loại
. nồng độ cồn là.	· và hàm lượng đường.	loại rượu đó như một.
. Cao.	. Cao.	rượu trắng
. Cao.	. Thấp.	rượu vang đỏ
. Thấp.	. Cao.	rượu trắng
. Thấp.	. Thấp.	rượu vang đỏ

	Predicted: Red	Predicted: White	Total
Actual: Red	464	121	585
Actual: White	678	1082	1760
Total	1142	1203	2345

Hình 8.4 Bảng dự phòng từ R của Loại rượu thực tế so với Loại rượu được dự đoán cho mô hình Naïve Bayes được đánh giá trên tập dữ liệu thử nghiệm.

Machine Translated by Google

8.5.1 Naïve Bayes trong Python

Bắt đầu bằng cách nhập tất cả các thư viện cần thiết.

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
từ sklearn.naive_bayes nhập MultinomialNB
nhập statsmodels.tools.tools dưới dạng stattools
```

Tải tập dữ liệu huấn luyện và kiểm tra, đồng thời gọi chúng lần lượt là wines_tr và wines_test .

```
wine_tr = pd.read_csv("C:/.../wine_flag_training.csv")
wine_test = pd.read_csv("C:/.../wine_flag_test.csv")
```

Đầu tiên, chúng tôi xem xét dữ liệu bằng bảng dự phòng. Các bảng này sẽ cho phép chúng tôi thu được các xác suất cận biên và có điều kiện mà chúng tôi cần để thực hiện các phép tính Naïve Bayes bằng tay, nếu chúng tôi chọn.

```
t1 = pd.crosstab(wine_tr['Type'], wine_tr['Alcohol_flag'])
t1['Total'] = t1.sum(axis=1)
t1.loc['Tổng'] = t1.sum()
t1
```

Các bảng được thể hiện trong Hình 8.5. Từ những điều này, chúng ta có thể thu được xác suất biên cho Loại Alcohol_flag và Sugar_flag, và xác suất có điều kiện là Alcohol_flag cho Loại và Sugar_flag cho Loại.

Chúng tôi cũng có thể tạo biểu đồ thanh để trực quan hóa xác suất từ các bảng của mình. Để làm như vậy, chúng ta cần điều chỉnh mã cho bảng dự phòng, như đã giải thích trong Chương 4.

```
t1_plot = pd.crosstab(wine_tr['Alcohol_flag'], wine_tr['Type'])
t1_plot.plot(loại='thanh', xếp chồng lên nhau = Đúng)
```

Bây giờ chúng ta chuyển sang thuật toán Naïve Bayes. Như trước đây, sklearn không tự động xử lý các biến phân loại. Điều này có nghĩa là chúng ta cần chuyển đổi Alcohol_flag và Sugar_flag thành các phiên bản biến giả của chính chúng trước khi có thể chạy thuật toán. Chúng tôi làm theo cách tiếp cận tương tự như trong Chương 6.

```
X_Alcohol_ind = np.array(wine_tr['Alcohol_flag'])
(X_Alcohol_ind , X_Alcohol_ind_dict) = stattools. phân loại
(X_Alcohol_ind,
```

Alcohol_flag	High	Low	Total
Type			
Red	218	282	500
White	268	232	500
Total	486	514	1000

Sugar_flag	High	Low	Total
Type			
Red	116	384	500
White	300	200	500
Total	416	584	1000

Hình 8.5 Các bảng dự phòng từ Python cho Type và Alcohol_flag (trái) và Type và Sugar_flag (phải).

Machine Translated by Google


```
      drop=True, dictnames=True)
X_Alcohol_ind = pd.DataFrame(X_Alcohol_ind)
X_Sugar_ind = np.array(wine_tr['Sugar_flag'])
 (X_Sugar_ind , X_Sugar_ind_dict) = stattools.
phân loại (X_Sugar_ind,
      drop=True, dictnames=True)
X_Sugar_ind = pd.DataFrame(X_Sugar_ind)
X = pd.concat((X_Alcohol_ind, X_Sugar_ind), trục = 1)
```

Trong X, ma trận các biến giả dự đoán của chúng tôi, chúng tôi có bốn cột. Hai cột đầu tiên tương ứng với Alcohol_flag, trong đó cột đầu tiên có giá trị 1 nếu nồng độ cồn cao và giá trị 0 nếu ngược lại. Tương tự, cột thứ hai có 1 nếu nồng độ cồn thấp và 0 nếu ngược lại. Cột thứ ba và thứ tư tương ứng với hàm lượng đường cao và hàm lượng đường thấp.

Chúng tôi cũng lưu biến mục tiêu là Y, để rõ ràng.

```
Y = wine_tr['Loại']
```

Cuối cùng, chúng ta chuyển sang thuật toán Naïve Bayes.

```
nb_01 = Đa thứcNB().fit(X, Y)
```

Cũng như các thuật toán trước, có hai bước: Chỉ định các tham số của thuật toán và khớp thuật toán dành riêng cho tham số với dữ liệu. Đối với thuật toán NultinonomialNB() trong trường hợp này, không có tham số bổ sung nào để đặt. Khi chúng tôi khớp () mô hình với các biến X và Y của mình , chúng tôi lưu đầu ra dưới dạng nb_01.

Để kiểm tra công cụ ước tính Naïve Bayes trên dữ liệu thử nghiệm, trước tiên chúng ta cần thiết lập biến X trong tập dữ liệu thử nghiệm làm biến giả. Chúng tôi làm theo các bước tương tự như đối với dữ liệu đào tạo, được đưa ra dưới đây.

```
X_Alcohol_ind_test = np.array(wine_test['Alcohol_flag'])
(X_Alcohol_ind_test, X_Alcohol_ind_dict_test) =
stattools.categorical(X_Alcohol_ind_test,
       drop=True, dictnames=True)
X_Alcohol_ind_test = pd.DataFrame(X_Alcohol_ind_test)
X_Sugar_ind_test = np.array(wine_test['Sugar_flag'])
(X_Sugar_ind_test, X_Sugar_ind_dict_test) = stattools. phân
loại (X_Sugar_ind_test,
       drop=True, dictnames=True)
X_Sugar_ind_test = pd.DataFrame(X_Sugar_ind_test)
X_test = pd.concat((X_Alcohol_ind_test, X_Sugar_ind_
kiểm tra), trục = 1)
```

Khi chúng tôi đã thiết lập các biến dự đoán cho tập dữ liệu thử nghiệm, chúng tôi có thể tạo đánh giá dự đoán.

```
Y_predicted = nb_01.predict(X_test)
```

Machine Translated by Google

Predicted	Red	White	Total
Actual			
Red	464	121	585
White	678	1082	1760
Total	1142	1203	2345

Hình 8.6 Bảng dự phòng từ Python về các loại rượu thực tế và dự đoán từ Python.

Sử dụng lệnh predict() trên đối tượng Naïve Bayes nb_01 của chúng ta sẽ tạo ra một mảng nhãn, Đỏ hoặc Trắng, cho mỗi bản ghi trong tập dữ liệu thử nghiệm.

Cuối cùng, chúng tôi muốn có một bảng dự phòng về các loại rượu thực tế và dự đoán. Chúng tôi sử dụng lệnh crosstab() một lần nữa.

```
ypred = pd.crosstab(wine_test['Type'], Y_predicted, rownames = ['Actual'],

          colnames = ['Dự đoán'])
ypred['Total'] = ypred.sum(axis=1); ypred.loc['Total'] = ypred.sum(); ypred
```

Các loại rượu thực nằm trong biến wine_test['Type'] tạo thành các hàng và các loại rượu dự đoán Y_predicted tạo thành các cột. Các giá trị đầu vào tùy chọn rownames và colnames đánh dấu phần đầu của các hàng và cột để tăng khả năng đọc của bảng. Tổng số hàng và cột được thêm vào theo các bước tương tự như đã được sử dụng với bảng dự phòng đã bắt đầu phần này. Kết quả được thể hiện trong Hình 8.6.

8.5.2 Naïve Bayes trong R

Nhập tập dữ liệu wine_flag_training và wine_flag_test vào R. Đặt tên cho chúng lần lượt là wine_tr và wine_test .

Trước tiên, chúng tôi tạo các bảng cho phép chúng tôi tính toán xác suất cần thiết bằng tay, nếu chúng tôi chọn. Bảng đầu tiên sẽ là bảng dự phòng của Type và Alcohol_flag.

```
ta <- table(wine_tr$Type, wine_tr$Alcohol_flag)
colnames(ta) <- c("Rượu = Cao", "Rượu = Thấp")
rownames(ta) <- c("Type = Red", "Type = White")
addmargins(A = ta, VUI = danh sách (Tổng = tổng), yên lặng = TRUE)
```

Kết quả của lệnh addmargins() được hiển thị trong bảng dự phòng trong Hình 8.1.

Bảng thứ hai sẽ là bảng dự phòng của Type và Sugar_flag.

```
ts <- table(wine_tr$Type, wine_tr$Sugar_flag)
colnames(ts) <- c("Đường = Cao", "Đường = Thấp")
rownames(ts) <- c("Type = Red", "Type = White")
addmargins(A = ts, FUN = danh sách (Tổng = tổng), yên lặng = TRUE)
```

Machine Translated by Google

Kết quả của lệnh addmargins() được hiển thị trong bảng dự phòng trong Hình 8.2.

Chúng ta cũng có thể tạo các biểu đồ thanh cạnh nhau như trong Hình 8.3. Mã lõi là mã ggplot() , đã được trình bày trước đây trong văn bản này. Tuy nhiên, để đặt các biểu đồ cạnh nhau, chúng tôi sẽ điều chỉnh mã lõi.

Trước tiên, hãy cài đặt gói cho phép chúng ta đặt các biểu đồ cạnh nhau: gói gridExtra .

```
install.packages("gridExtra"); thư viện (gridExtra)
```

Sau đó, chạy mã ggplot() như được hiển thị.

```
plot1 <- ggplot(wine_tr, aes(Type)) + geom_bar( aes(fill =
Alcohol_flag), vị trí = "điền") +
          ylab("Tỷ lệ")
plot2 <- ggplot(wine_tr, aes(Type)) + geom_bar( aes(fill =
Sugar_flag), vị trí = "điền") +
          ylab("Tỷ lệ")
grid.arrange(plot1, plot2, nrow = 1)
```

Bản thân mã ggplot() sẽ quen thuộc trong Chương 4. Lưu ý rằng chúng ta lưu mỗi biểu đồ dưới tên riêng của nó; ô 1 cho lớp phủ Alcohol_flag và ô 2 cho lớp phủ Sugar_flag . Sau khi lưu từng biểu đồ, chúng ta chạy lệnh grid.arrange() với ba giá trị đầu vào: plot1, plot2 và nrow = 1 để xác định rằng chúng ta muốn các biểu đồ cạnh nhau trên một hàng. Kết quả được thể hiện trong các đồ thị cạnh nhau trong Hình 8.3.

Bây giờ chúng ta đã có các bảng dự phòng và đồ thị, chúng ta chuyển sang thuật toán Naïve Bayes. Gói e1071 chứa thuật toán phân loại Naïve Bayes. Cài đặt và mở gói.

```
install.packages("e1071"); thư viện(e1071)
```

Chạy công cụ ước tính Naïve Bayes.

```
nb01 <- ngây thơBayes(công thức = Type ~ Alcohol_flag +
Sugar_flag, dữ liệu = wine_tr)
```

Lệnh ngây thơBayes() sẽ xây dựng mô hình. Đầu vào công thức lấy biến mục tiêu Type ở bên trái dấu ngã và hai biến dự đoán Alcohol_flag và Sugar_flag ở bên phải dấu ngã, được phân tách bằng dấu cộng. Đầu vào dữ liệu chỉ định tập dữ liệu mà các biến này đến từ đó. Chúng tôi lưu mô hình dưới dạng nb01.

Để quan sát xác suất trước và xác suất có điều kiện được sử dụng trong mô hình Naïve Bayes, hãy tự chạy tên của mô hình.

```
nb01
```

Machine Translated by Google


```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  Red White
  0.5   0.5

Conditional probabilities:
       Alcohol_flag
Y        High   Low
  Red   0.436 0.564
  White 0.536 0.464

       Sugar_flag
Y        High   Low
  Red   0.232 0.768
  White 0.600 0.400
```

Hình 8.7 Đầu ra từ mô hình Naïve Bayes trong R.

Đầu ra được hiển thị trong Hình 8.7.

Hai mục quan tâm chính trong đầu ra là xác suất A-priori và xác suất có điều kiện. Xác suất A-priori là các giá trị của p(Y) và xác suất có điều kiện là các giá trị kết quả của p(Y X).

Để dự đoán loại rượu cho từng loại rượu trong tập dữ liệu thử nghiệm của mình, chúng tôi sử dụng lệnh dự đoán ()

```
ypred <-predict(object = nb01, newdata = wine_test)
```

Đối tượng = nb01 chỉ định rằng chúng tôi đang sử dụng mô hình Naïve Bayes của mình và newdata = wine_test cho biết tập dữ liệu thử nghiệm sẽ sử dụng. Thuật toán phân loại từng bản ghi trong tập dữ liệu thử nghiệm thành rượu vang trắng hoặc đỏ và lưu chuỗi phân loại là ypred.

Cuối cùng, chúng tôi tạo bảng dự phòng của các loại rượu thực tế so với dự đoán.

```
t.preds <- table(wine_test$Type, ypred)
rownames(t.preds) <- c("Actual: Red", "Actual: White")
colnames(t.preds) <- c("Dự đoán: Đỏ", "Dự đoán: Trắng")
addmargins(A = t.preds, FUN = danh sách (Tổng cộng = tổng), yên tĩnh = TRUE)
```

Kết quả của lệnh addmargins() được thể hiện trong Hình 8.4.

NGƯỜI GIỚI THIỆU

Chúng tôi đã sử dụng một gói mới, gridExtra, để định dạng nhiều biểu đồ ggplot . Nếu bạn muốn biết thêm chi tiết về hoạt động bên trong của gói mới, hãy xem tại đây: Baptiste Auguie, gridExtra: Miscellaneous Functions for “Grid” Graphics. Phiên bản gói R 2.3, 2017. https://CRAN.R-project.org/package=gridExtra.

Machine Translated by Google

Gói e1071 chứa nhiều hơn so với Naïve Bayes! Xem trích dẫn để biết mô tả đầy đủ: David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, và Friedrich Leisch, e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. Phiên bản gói R 1.6-8., 2017. http://CRAN.R-project. org/gói=e1071.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Định lý Bayes cập nhật kiến thức trước đây của chúng ta về định lý Bayes với thông tin gì? thông số dữ liệu?
2. Xác suất trước thể hiện điều gì?
3. Công thức nào thể hiện cách dữ liệu hoạt động trong các giá trị lớp của biến mục tiêu?
4. Công thức nào thể hiện cách dữ liệu hoạt động mà không cần tham chiếu đến các giá trị của lớp?
5. Công thức từ bài tập trước được gọi là gì?
6. Xác suất sau đại diện cho điều gì?
7. Chúng ta sử dụng xác suất biết trước để làm gì nếu chúng ta không có kiến thức trước về thông số?
8. Giả thuyết hậu nghiệm cực đại giúp chúng ta phân loại bản ghi như thế nào?
9. Giả định độc lập có điều kiện của lớp là gì?
10. Nếu chúng ta có nhiều hơn một biến dự đoán, làm thế nào để viết p(X* Y = y*) nếu chúng ta có hai biến dự đoán X* = {X1 = x1 , X2 = x2 }?

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu wine_flag_training và wine_flag_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Tạo hai bảng dự phòng, một với Type và Alcohol_flag và một với Type và Sugar_flag.
12. Sử dụng các bảng trong bài tập trước để tính toán:

    Một. Xác suất trước của Loại = Đỏ và Loại = Trắng.

    b. Xác suất của nồng độ cồn cao và thấp.

    c. Xác suất hàm lượng đường cao và thấp.

    d. Các xác suất có điều kiện p(Cờ_rượu = Cao Loại = Đỏ) và p(Cờ_rượu = Thấp Loại = Đỏ).
    đ. Các xác suất có điều

    kiện p(Cờ_rượu = Cao Loại = Trắng) và p(Cờ_rượu = Thấp Loại = Trắng).

Machine Translated by Google

f. Xác suất có điều kiện p(Sugar_flag = Cao | Loại = Đỏ) và p(Sugar_flag = Thấp | Loại = Đỏ).

g. Các xác suất có điều kiện p(Sugar_flag = Cao | Loại = Trắng) và p(Sugar_cờ = Thấp | Loại = Trắng).

13. Sử dụng xác suất trong bài tập trước để thảo luận

Một. Khả năng một loại rượu được chọn ngẫu nhiên có màu đỏ là bao nhiêu.

b. Khả năng một loại rượu được chọn ngẫu nhiên có nồng độ cồn cao là bao nhiêu.

c. Khả năng một loại rượu được chọn ngẫu nhiên có hàm lượng đường thấp là bao nhiêu.

14. Sử dụng các xác suất có điều kiện được tìm thấy trước đó để thảo luận

Một. Một loại rượu vang trắng điển hình có thể có hàm lượng cồn và đường như thế nào.

b. Một loại rượu vang đỏ điển hình có thể có hàm lượng cồn và đường như thế nào.

15. Tạo các biểu đồ thanh cạnh nhau cho Loại, một biểu đồ có lớp phủ Alcohol_flag và biểu đồ kia có lớp phủ Sugar_flag. So sánh các biểu đồ với các xác suất có điều kiện mà bạn đã tính toán.

16. Tính xác suất sau của Loại = Màu đỏ đối với rượu có nồng độ cồn thấp và hàm lượng đường cao. Tính xác suất sau của Loại = Trắng cho cùng một loại rượu.

17. Sử dụng câu trả lời của bạn cho bài tập trước để xác định loại rượu nào, đỏ hay trắng, có nhiều khả năng là rượu có hàm lượng đường cao và ít cồn. Bộ phân loại Naïve Bayes sẽ phân loại loại rượu này là gì?

18. Tính xác suất sau của Loại = Màu đỏ đối với rượu có hàm lượng cồn cao và hàm lượng đường thấp. Tính xác suất sau của Loại = Trắng cho cùng một loại rượu.

19. Sử dụng câu trả lời của bạn cho bài tập trước để xác định loại nào, đỏ hay trắng, có nhiều khả năng là rượu có hàm lượng cồn cao và ít đường. Bộ phân loại Naïve Bayes sẽ phân loại loại rượu này là gì?

20. Chạy bộ phân loại Naïve Bayes để phân loại rượu vang trắng hoặc đỏ dựa trên độ cồn và hàm lượng đường.

21. Đánh giá mô hình Naïve Bayes trên tập dữ liệu wines_test . Hiển thị kết quả trong một bảng dự phòng. Chỉnh sửa tên hàng và cột của bảng để làm cho bảng dễ đọc hơn. Bao gồm tổng số hàng và cột.

22. Theo bảng của bạn trong bài tập trước, hãy tìm các giá trị sau cho Naïve Mô hình Bayes:

Một. Sự chính xác

b. tỷ lệ lỗi

23. Theo bảng dự phòng của bạn, hãy tìm các giá trị sau cho mô hình Naïve Bayes:

Một. Tần suất nó phân loại chính xác rượu vang đỏ.

b. Tần suất nó phân loại chính xác rượu vang trắng.

Machine Translated by Google

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với framingham_nb_training và framingham_nb_ tập dữ liệu thử nghiệm . Sử dụng Python hoặc R để giải quyết từng vấn đề.

24. Chuyển đổi tất cả các biến (Death, Sex và Educ) thành thừa số.

25. Tạo hai bảng dự phòng, một với Death và Sex và một với Death và Educ.

26. Sử dụng các bảng trong bài tập trước để tính toán:

    Một. Xác suất một người được chọn ngẫu nhiên còn sống hoặc đã chết.

    b. Xác suất một người được chọn ngẫu nhiên là nam.

    c. Xác suất một người được chọn ngẫu nhiên có giá trị Educ là 3.

    d. Xác suất để một người chết là nam có trình độ học vấn 1 và một người sống là nam có trình độ học vấn 1.

    đ. Xác suất để người sống là nữ có học vấn cấp 2 và người chết là nữ có học vấn cấp 2.

27. Tạo các biểu đồ thanh cạnh nhau cho Death, một biểu đồ có lớp phủ Giới tính và biểu đồ kia có lớp phủ Giáo dục.

28. Sử dụng các đồ thị từ bài tập trước để trả lời các câu hỏi sau:

    Một. Nếu chúng ta biết một người đã chết, họ có nhiều khả năng là nam hay nữ?

    b. Nếu chúng ta biết một người còn sống, nhiều khả năng họ là nam hay nữ?

    c. Nếu chúng ta biết một người đã chết, họ có khả năng học ở trình độ nào nhất?

    d. Nếu chúng ta biết một người còn sống, trình độ học vấn của họ có khả năng nhất là bao nhiêu?

    đ. Trình độ học vấn nào phổ biến hơn đối với những người đã chết? Đối với người sống?

29. Tính xác suất tử vong sau = 0 (người đang sống) đối với nam giới có trình độ học vấn 1. Tính xác suất tử vong sau = 1 (người đã chết) đối với nam giới có trình độ học vấn 1.

30. Tính xác suất tử vong sau = 0 (người đang sống) đối với nữ có trình độ học vấn 2. Tính xác suất tử vong sau = 1 (người đã chết) đối với nữ có trình độ học vấn 2.

31. Chạy trình phân loại Naïve Bayes để phân loại người còn sống hay đã chết dựa trên giới tính và giáo dục.

32. Đánh giá mô hình Naïve Bayes trên tập dữ liệu framingham_nb_test . Hiển thị kết quả trong một bảng dự phòng. Chỉnh sửa tên hàng và cột của bảng để làm cho bảng dễ đọc hơn. Bao gồm tổng số hàng và cột.

33. Theo bảng của bạn trong bài tập trước, hãy tìm các giá trị sau cho Naïve Mô hình Bayes:

    Một. Sự chính xác

    b. tỷ lệ lỗi

34. Theo bảng dự phòng của bạn, hãy tìm các giá trị sau cho mô hình Naïve Bayes:

    Một. Tần suất nó phân loại chính xác người chết.

    b. Tần suất nó phân loại chính xác những người sống.

Machine Translated by Google

chương 9

MẠNG NƠ NE

9.1 GIỚI THIỆU VỀ MẠNG NƠ NE

Mạng lưới thần kinh đại diện cho một nỗ lực ở mức rất cơ bản để bắt chước kiểu học phi tuyến tính xảy ra trong mạng lưới tế bào thần kinh được tìm thấy trong tự nhiên, chẳng hạn như não người. Như được minh họa trong Hình 9.1, một nơ-ron từ não người sử dụng đuôi gai để thu thập thông tin đầu vào từ các nơ-ron khác và kết hợp thông tin đầu vào, tạo ra một phản ứng phi tuyến tính ("kích hoạt") khi đạt đến một ngưỡng nhất định, phản ứng này sẽ gửi đến các nơ-ron khác bằng cách sử dụng sợi trục. . Hình 9.1 cũng cho thấy một mô hình nơ-ron nhân tạo được sử dụng trong hầu hết các mạng nơ-ron. Các đầu vào (xi) được thu thập từ các nơ-ron ngược dòng (hoặc tập dữ liệu) và được kết hợp thông qua một hàm kết hợp, chẳng hạn như tổng (Σ), sau đó được nhập vào một hàm kích hoạt (thường là phi tuyến tính) để tạo ra phản hồi đầu ra (y) , mà sau đó có thể được truyền xuống các tế bào thần kinh khác.

Lợi ích chính của mạng thần kinh là chúng khá mạnh đối với dữ liệu nhiễu, phức tạp hoặc phi tuyến tính do tính chất phi tuyến tính của hàm kích hoạt.
Mặt khác, nhược điểm chính của mạng lưới thần kinh là chúng tương đối khó hiểu đối với cách hiểu của con người, trái ngược với các cây quyết định chẳng hạn.

9.2 CẤU TRÚC MẠNG NƠ-RON

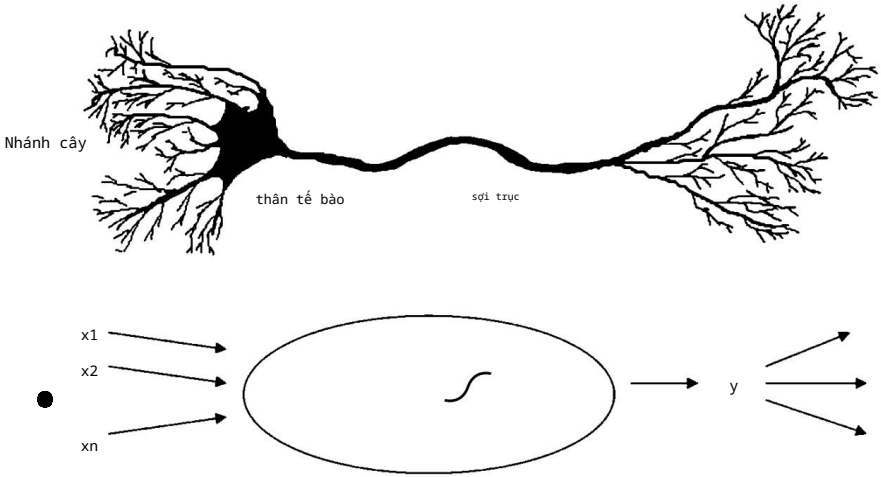
Chúng ta hãy xem xét mạng lưới thần kinh đơn giản được hiển thị trong Hình 9.2.

Một mạng lưới thần kinh bao gồm một lớp, chuyển tiếp, được kết nối hoàn toàn mạng nơ-ron nhân tạo hoặc các nút.

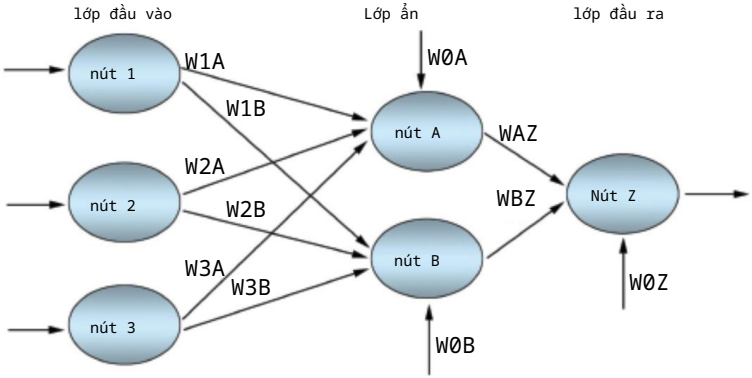
- Bản chất chuyển tiếp của mạng hạn chế mạng theo một hướng duy nhất của dòng chảy và không cho phép chạy vòng hoặc quay vòng.
- Hầu hết các mạng bao gồm ba lớp: lớp đầu vào, lớp ẩn và lớp lớp đầu ra.
  - Có thể có nhiều hơn một lớp ẩn, mặc dù hầu hết các mạng chỉ chứa một lớp, đủ cho hầu hết các mục đích.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google



Hình 9.1 Mô hình nơ ron thực và nơ ron nhân tạo.



Hình 9.2 Ví dụ đơn giản về mạng nơ-ron.

- Mạng thần kinh được kết nối hoàn toàn, nghĩa là mọi nút trong một lớp nhất định được kết nối với mọi nút trong các lớp liền kề, mặc dù không kết nối với các nút khác trong cùng một lớp.
  - Mỗi kết nối giữa các nút có một trọng số (ví dụ: W1A) được liên kết với nó.
  - Khi khởi tạo, các trọng số này được gán ngẫu nhiên cho các giá trị giữa 0 và 1.

Số lượng nút đầu vào phụ thuộc vào số lượng và loại thuộc tính trong tập dữ liệu. Lớp đầu vào chấp nhận đầu vào từ tập dữ liệu, chẳng hạn như giá trị thuộc tính và chỉ cần chuyển các giá trị này cho lớp ẩn mà không cần xử lý thêm.

Số lượng các lớp ẩn và số lượng nút trong mỗi lớp ẩn đều có thể được cấu hình bởi nhà phân tích. Một người nên có bao nhiêu nút trong

Machine Translated by Google

lớp ẩn? Vì có nhiều nút hơn trong lớp ẩn làm tăng sức mạnh và tính linh hoạt của mạng để xác định các mẫu phức tạp, người ta có thể muốn có một số lượng lớn nút trong lớp ẩn. Tuy nhiên, một lớp ẩn quá lớn dẫn đến trang bị quá mức, ghi nhớ tập huấn luyện với chi phí là khả năng khái quát hóa tập hợp lệ. Nếu xảy ra tình trạng quá khớp, người ta có thể xem xét giảm số lượng nút trong lớp ẩn. Ngược lại, nếu độ chính xác đào tạo thấp đến mức không thể chấp nhận được, người ta có thể xem xét việc tăng số lượng nút trong lớp ẩn.

9.3 TRỌNG LƯỢNG KẾT NỐI VÀ CHỨC NĂNG TỔNG HỢP

Các nút trong lớp ẩn và lớp đầu ra thu thập các đầu vào từ lớp trước đó và kết hợp chúng bằng chức năng kết hợp. Hàm kết hợp này (thường là tổng, Σ) tạo ra sự kết hợp tuyến tính của đầu vào nút và trọng số kết nối thành một giá trị vô hướng duy nhất, mà chúng ta sẽ gọi là net. Do đó, đối với một nút nhất định j,

$$\text{lưới } W_j x = \sum_{i=0}^{I} x_{ij} W_{ij} = x_{0j} W_{0j} + x_{1j} W_{1j} + \text{trọng } x_{ij}$$

trong đó xij đại diện cho đầu vào thứ i của nút j, Wij đại diện cho trọng số được liên kết với đầu vào thứ i của nút j và có I+1 đầu vào cho nút j. Lưu ý rằng x1 , x2 , ., xI đại diện gửi lại đầu vào từ các nút ngược dòng, trong khi x0 đại diện cho đầu vào không đổi , tương tự như hệ số không đổi trong các mô hình hồi quy, theo quy ước sẽ nhận giá trị duy nhất x0j = 1. Do đó, mỗi lớp ẩn hoặc đầu ra nút lớp j chứa đầu vào "thêm" bằng trọng số cụ thể W0j x0j = W0j , chẳng hạn như W0B cho nút B.

Chúng tôi minh họa cấu trúc của các nút lớp ẩn và các nút lớp đầu ra bằng cách sử dụng dữ liệu mẫu đồ chơi được cung cấp trong Bảng 9.1.

Ví dụ, đối với nút A trong lớp ẩn, chúng ta có

$$\text{lưới } W_{\text{một}} x = \sum_{i=0}^{I} W_{iA} x_{iA} = W_{0A}(1) + \text{Rộng } x_{1A} + \text{Rộng } x_{2A} + \text{Rộng } x_{3A}$$

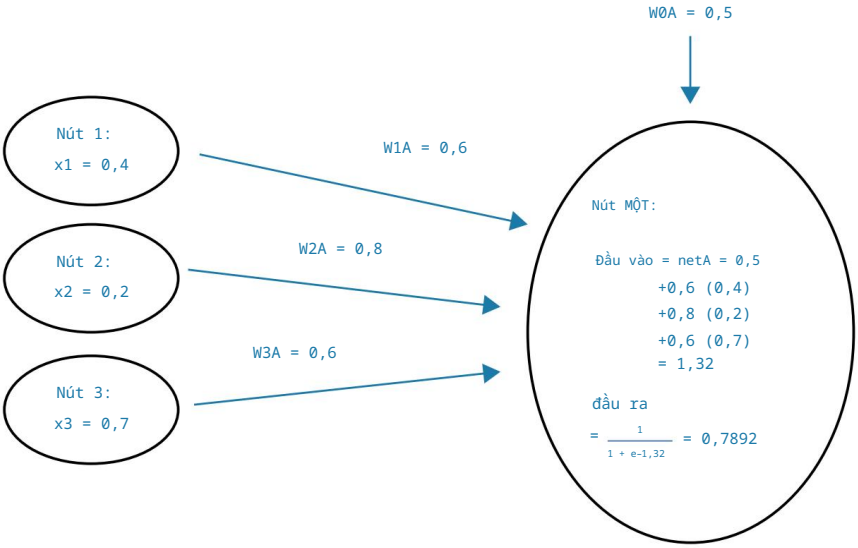
$$= 0.5(1) + 0.4(0.4) + 0.2(0.2) + 0.6(0.7) = 1.32$$

Vì vậy, trong Hình 9.3, chúng ta thấy rằng nút A đã kết hợp các đầu vào của nó thành một đầu vào ròng của 1,32. Trong nút A, hàm kết hợp này netA = 1,32 sau đó được sử dụng làm đầu vào cho hàm kích hoạt. Trong tế bào thần kinh sinh học, tín hiệu được gửi giữa

BẢNG 9.1 Dữ liệu đầu vào và giá trị ban đầu cho trọng số mạng thần kinh

x0 = 1,0	W0A = 0,5	W0B = 0,7	W0Z = 0,5
x1 = 0,4	W1A = 0,6	W1B = 0,9	WA = 0,9
x2 = 0,2	W2A = 0,8	W2B = 0,8	WBZ = 0,9
x3 = 0,7	W3A = 0,6	W3B = 0,4	

Machine Translated by Google



Hình 9.3 Chi tiết về mạng thần kinh, hiển thị đầu vào cho nút A, chức năng kết hợp và đầu ra từ nút A.

tế bào thần kinh khi sự kết hợp của các đầu vào cho một tế bào thần kinh cụ thể vượt qua một ngưỡng nhất định và tế bào thần kinh "cháy". Đây là hành vi phi tuyến tính, vì phản ứng kích hoạt không nhất thiết liên quan tuyến tính với mức tăng kích thích đầu vào.

Hàm kích hoạt phổ biến nhất là hàm sigmoid:

$$y = \frac{1}{1 + e^{-x}}$$

trong đó e là cơ số của logarit tự nhiên, bằng khoảng 2,718281828. Do đó, trong nút A, việc kích hoạt sẽ lấy netA = 1,32 làm đầu vào cho hàm kích hoạt sigmoid và tạo ra giá trị đầu ra là

$$\text{lưỡi } y\left(f_{\text{MỘT}}\right) = \frac{1}{1 + e^{-1.32}} = 0.7892$$

Công việc của nút A đã hoàn thành (tại thời điểm này) và giá trị đầu ra này sau đó sẽ được truyền dọc theo kết nối đến nút đầu ra Z, nơi nó sẽ hình thành (thông qua một tổ hợp tuyến tính khác) một thành phần của netZ .

Nó còn lại như một bài tập để tính netB = 1,5 và $f\left(\text{net}_b\right) = \frac{1}{1 + e^{-1.5}} = 0.8176$.

Sau đó, nút Z kết hợp các đầu ra này từ các nút A và B, thông qua netZ , một tổng trọng số, sử dụng các trọng số được liên kết với các kết nối giữa các nút này. Lưu ý rằng đầu vào xi đến nút Z không phải là giá trị thuộc tính dữ liệu mà là đầu ra từ các hàm sigmoid từ các nút ngược dòng:

Machine Translated by Google

$$\text{rộng}_Z = \sum \text{rộng}_{iZ}\, x_{iZ} = \text{rộng}_{0Z}(1) + \text{Rộng}_{AZ}\, x_{AZ} + \text{Rộng}_{BZ}\, x_{BZ}$$

$$= 0.5 + 0.09(0.7892) + 0.09(0.7108) = 1.9461$$

Cuối cùng, netZ được nhập vào chức năng kích hoạt sigmoid trong nút Z, kết quả là

$$f(\text{mạng}_Z) = \frac{1}{1 + e^{-1.9461}} = 0.8750$$

Giá trị 0,8750 này là đầu ra từ mạng thần kinh cho lần truyền đầu tiên này qua mạng và biểu thị giá trị được dự đoán cho biến mục tiêu cho lần quan sát đầu tiên.

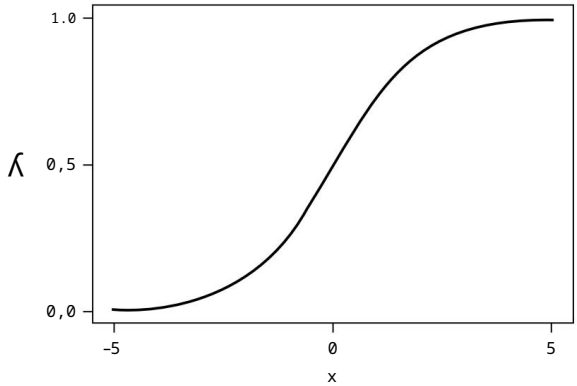
9.4 CHỨC NĂNG KÍCH HOẠT SIGMOID

Một chức năng kích hoạt phổ biến là chức năng sigmoid

$$y = f(x) = \frac{1}{1 + e^{-x}}$$

Tại sao sử dụng chức năng sigmoid? Bởi vì nó kết hợp hành vi gần như tuyến tính , hành vi đường cong và hành vi gần như không đổi , tùy thuộc vào giá trị của đầu vào. Hình 9.4 cho thấy đồ thị của hàm sigmoid cho 5<x<5. Thông qua phần lớn tâm miền của đầu vào x (ví dụ –1<x<1), hành vi của f(x) gần như tuyến tính. Khi đầu vào di chuyển ra khỏi tâm, f(x) trở thành đường cong.
Vào thời điểm đầu vào đạt đến các giá trị cực trị, f(x) gần như không đổi.

Các gia số vừa phải trong giá trị của x tạo ra các gia số khác nhau trong giá trị của f(x), tùy thuộc vào vị trí của x. Gần trung tâm, mức tăng vừa phải trong



Hình 9.4 Đồ thị của hàm sigmoid y = f(x) = 1/(1+e–x).

Machine Translated by Google

giá trị của x tạo ra sự gia tăng vừa phải trong giá trị của f(x); tuy nhiên, ở gần các điểm cực trị, giá trị của x tăng vừa phải tạo ra giá trị của f(x) tăng rất nhỏ.

9.5 TUYỆT ĐỐI

Làm thế nào để mạng lưới thần kinh học? Khi mỗi quan sát từ tập huấn luyện được xử lý thông qua mạng, một giá trị đầu ra được tạo ra từ nút đầu ra.
Giá trị đầu ra này sau đó được so sánh với giá trị thực tế của biến mục tiêu cho quan sát tập huấn luyện này và lỗi (thực tế - đầu ra) được tính toán. Lỗi dự đoán này tương tự như lỗi dự đoán trong các mô hình hồi quy tuyến tính. Để đo mức độ phù hợp của các dự đoán đầu ra với các giá trị mục tiêu thực tế, hầu hết các mô hình mạng thần kinh sử dụng tổng các lỗi bình phương (SSE):

$$\text{SSE} = \sum_{\text{ghi lại các nút đầu ra}} \sum \left(\text{sản lượng thực tế}\right)^2$$

trong đó các lỗi dự đoán bình phương được tính tổng trên tất cả các nút đầu ra và trên tất cả các bản ghi trong tập huấn luyện. Do đó, vấn đề là xây dựng một tập hợp các trọng số mô hình sẽ giảm thiểu SSE. Theo cách này, các trọng số tương tự như các tham số của mô hình hồi quy. Các giá trị "đúng" cho các trọng số sẽ giảm thiểu SSE là không xác định và nhiệm vụ của chúng tôi là ước tính chúng, dựa trên dữ liệu.

Tuy nhiên, do tính chất phi tuyến tính của các hàm sigmoid trong mạng, không tồn tại giải pháp dạng đóng nào để giảm thiểu SSE như tồn tại đối với hồi quy bình phương nhỏ nhất. Do đó, các phương pháp tối ưu hóa, cụ thể là các phương pháp giảm dần độ dốc, được sử dụng.

Thuật toán lan truyền ngược thực hiện như sau:

1. Nó nhận lỗi dự đoán (thực tế – đầu ra) cho một bản ghi cụ thể và mỗi xử lý lỗi thông qua mạng.
2. Gán trách nhiệm phân vùng về lỗi cho các kết nối khác nhau.
3. Sau đó, trọng số trên các kết nối này được điều chỉnh để giảm sai số, sử dụng giảm độ dốc.[1]

9.6 ỨNG DỤNG CỦA MÔ HÌNH MẠNG NƠ-RON

Tiếp theo, chúng ta chuyển sang một ví dụ về mô hình mạng thần kinh sử dụng một tập hợp con của dữ liệu Nghiên cứu Tim mạch Framingham.[2] Tập dữ liệu, Framingham_training, chứa thông tin về ba biến cho 7953 bệnh nhân. Giới tính là một yếu tố dự đoán nhị phân với

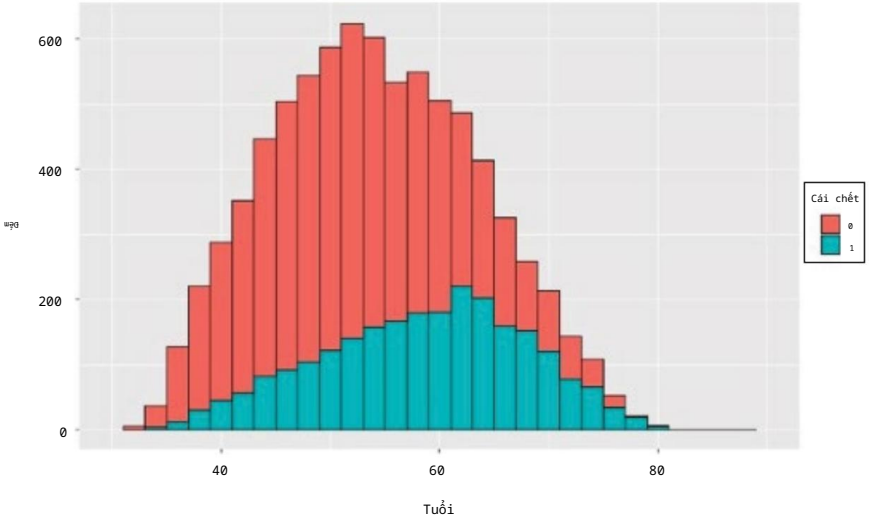
[1]Để biết thêm thông tin về các phương pháp giảm dần độ dốc và chi tiết về lan truyền ngược, hãy xem Khai thác dữ liệu và phân tích dự đoán, Chương 12.

[2]www.framinghamheartstudy.org.

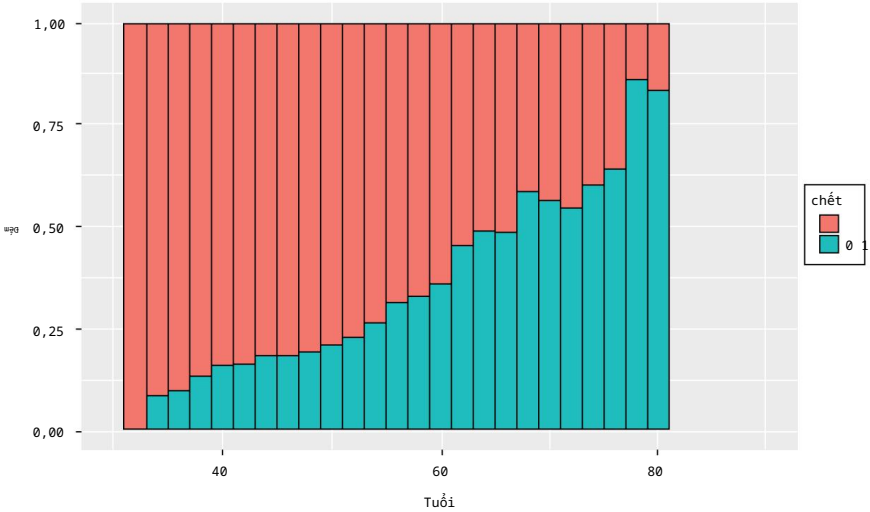
Machine Translated by Google

1 = Nam và 2 = Nữ. Tuổi tác là một yếu tố dự báo liên tục. Biến mục tiêu là Cái chết, với các giá trị 0 = sống sót và 1 = chết.

Manh mối về mối quan hệ giữa các yếu tố dự đoán và mục tiêu có được thông qua phân tích dữ liệu khám phá, cụ thể là qua Hình 9.5 và 9.6, và Bảng 9.2 và 9.3. Các biểu đồ trong Hình 9.5 và 9.6 cho thấy, khi Tuổi tăng lên, tỷ lệ Tử vong tăng lên. Bảng 9.2 và 9.3 cho thấy tỷ lệ nam chết nhiều hơn nữ. Do đó, những mối quan hệ tương hỗ này sẽ được phản ánh trong kết quả mô hình mạng thần kinh của chúng tôi.



Hình 9.5 Biểu đồ từ R of Age, với lớp phủ Death.



Hình 9.6 Biểu đồ được chuẩn hóa từ R of Age, với lớp phủ Death.

Machine Translated by Google

BẢNG 9.2 Bảng dự phòng Giới tính và Tử vong

		giới tính		
		Nam giới	Nữ Tổng số	
Cái chết	0	2113	3422 5535	
	1	1324	1094 2418	
	Tổng cộng	3437	4516 7953	

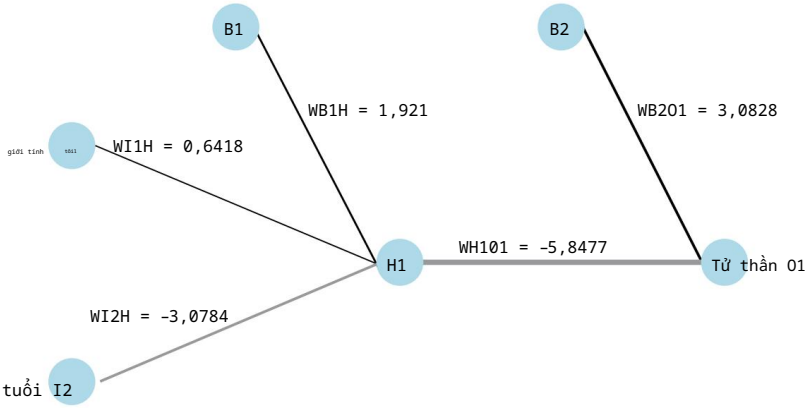
BẢNG 9.3 Bảng dự phòng Giới tính và Cái chết với cột tỷ lệ phần trăm

		giới tính	
		Nam (%)	Nữ (%)
Cái chết	0	61,5	75,8
	1	38,5	24.2

Để rõ ràng, chúng tôi chỉ chọn một lớp ẩn duy nhất, chỉ chứa một nơ-ron duy nhất. Hình 9.7 cho thấy kết quả mô hình mạng thần kinh, được xây dựng bằng R, được tạo bởi tập dữ liệu Framingham_training .

9.7 DIỄN GIẢI TRỌNG SỐ TRONG MÔ HÌNH MẠNG Nơ-ron

Các trọng số trong mô hình mạng thần kinh đại diện cho những gì mô hình đang cố nói với bạn, dựa trên dữ liệu. Các trọng số này tương tự như các hệ số dự báo trong một mô hình hồi quy. Chúng ta hãy thu thập những thông tin có thể từ các trọng số trong Hình 9.7.



Hình 9.7 Mô hình mạng lưới thần kinh từ dữ liệu Framingham Heart Study.

Machine Translated by Google

Đầu tiên, chúng ta hãy bỏ qua các trọng số sai lệch (số hạng không đổi) B1 và B2, vì chúng không ảnh hưởng đến mối quan hệ giữa các yếu tố dự đoán và phản hồi. Tiếp theo, hãy nhớ lại phân tích dữ liệu khám phá (EDA) của chúng tôi, trong đó chúng tôi nhận thấy rằng tuổi càng cao và là nam giới đều có liên quan đến xác suất tử vong cao hơn trong Nghiên cứu Tim mạch Framingham. Ngoài ra, Giới tính là một yếu tố dự đoán nhị phân với 1 = Nam và 2 = Nữ, do đó, giá trị của Giới tính tăng lên sẽ liên quan đến việc giảm xác suất tử vong. Chúng ta hãy xem làm thế nào và liệu các kết quả EDA này có được phản ánh trong các trọng số của mạng thần kinh hay không.

Bây giờ, trọng số giữa nút lớp ẩn H1 và nút đầu ra O1 có giá trị âm, WH1O1 = 5,8477. Do đó, khi nút lớp ẩn H1 được kích thích đến giá trị cao, nó thực sự có tác dụng bảo vệ khỏi cái chết, làm giảm xác suất của nó, vì trọng số âm này.

Tiếp theo, trọng số từ yếu tố dự đoán Giới tính đến nút lớp ẩn nhận giá trị dương, WI1H = 0,6418. Điều này có nghĩa là các giá trị Giới tính (nữ) lớn hơn sẽ có xu hướng kích thích tế bào thần kinh lớp ẩn H1 lên giá trị cao hơn. Như chúng ta vừa biết, giá trị H1 cao như vậy sẽ làm giảm xác suất tử vong. Do đó, các trọng số cho chúng ta biết rằng việc là phụ nữ bảo vệ khỏi xác suất tử vong, giống như chúng ta đã thấy trong EDA.

Cuối cùng, trọng số từ bộ dự đoán Tuổi đến nút lớp ẩn có giá trị âm, WI2H = 3,0784. Điều này có nghĩa là giá trị Tuổi cao hơn sẽ dẫn đến giá trị thấp hơn cho nơ-ron lớp ẩn H1. Các giá trị thấp từ H1 đến nút đầu ra O1 dẫn đến khả năng tử vong cao vì trọng số của nó cũng âm. Do đó, các trọng số cho chúng ta biết rằng sự gia tăng tuổi tác có liên quan đến sự gia tăng xác suất tử vong, chính xác như chúng ta đã thấy trong EDA.

9.8 CÁCH SỬ DỤNG MẠNG Nơ ron TRONG R

Trước tiên, hãy đọc tập dữ liệu Framingham_training là fram_train và chuyển đổi các biến nhị phân và biến thứ tự Death và Sex thành thừa số.

```
fram_train$Death <- as.factor(fram_train$Death)
fram_train$Sex <- as.factor(fram_train$Sex)
```

Thực hiện tiêu chuẩn hóa tối thiểu tối đa trên biến Tuổi

```
fram_train$Age.mm <- (fram_train$Age - min(fram_train$Age)) /
 (tối đa(fram_train$Age) - tối thiểu(fram_train$Age))
```

Cài đặt gói nnet và NeuralNetTools và mở cả hai gói.

```
install.packages("nnet"); install.packages("NeuralNetTools")
thư viện(nnet); thư viện (NeuralNetTools)
```

Chạy thuật toán mạng thần kinh.

```
nnet01 <- nnet(Cái chết ~ Giới tính + Tuổi.mm, dữ liệu = fram_train, kích thước = 1)
```

Lưu ý cấu trúc công thức, với biến mục tiêu Death ở phía bên trái của dấu ngã và hai biến dự đoán Giới tính và Tuổi tác ở phía bên phải. Các

Machine Translated by Google

dữ liệu = đầu vào fram_train chỉ định ba biến đến từ đâu. Lệnh size = 1 nói rằng có một đơn vị trong lớp ẩn.

Lưu ý rằng đầu ra mạng thần kinh được lưu dưới dạng nnet01. Tiết kiệm đầu ra là cần thiết để có được biểu đồ và trọng số của mạng lưới thần kinh.

Tiếp theo, vẽ sơ đồ mạng lưới thần kinh.

cốt truyện (nnet01)

Đầu ra từ lệnh plotnet() với đầu vào nnet01 sẽ khớp với Hình 9.7.

Cuối cùng, có được trọng lượng.

nnet01$wts

NGƯỜI GIỚI THIỆU

Gói NeuralNetTools đã trực quan hóa mô hình mạng thần kinh của chúng tôi. Để xem các tùy chọn và các tính năng khác, hãy xem tài liệu về gói:M. Beck, _NeuralNetTools: Công cụ phân tích và trực quan hóa cho mạng nơ-ron_. Phiên bản gói R 1.5.1, 2018. http://CRAN.R-project.org/package=NeuralNetTools.

Gói nnet đã được sử dụng để tạo mạng lưới thần kinh. Để biết thêm thông tin, hãy xem ấn phẩm xuất bản: WN Venables và BD Ripley, Modern Applied Statistics with S, Fourth Edition, Springer, New York, 2002.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Phân loại mạng nơ-ron thể hiện nỗ lực bắt chước cái gì?
2. Sử dụng Hình 9.1, hãy giải thích cách một mô hình nơ-ron nhân tạo bắt chước hoạt động của tế bào thần kinh thực tế bào thần kinh.
3. Lợi ích chính của mạng lưới thần kinh đối với mô hình hóa là gì? Điều gì mang lại cho mạng lưới thần kinh sức mạnh này?
4. Mô tả nhược điểm chính của mô hình mạng thần kinh.
5. Giải thích ý nghĩa của chúng tôi khi chúng tôi nói rằng một mạng lưới thần kinh được kết nối hoàn toàn.
6. Mô tả những lợi ích và hạn chế của việc sử dụng nhiều hơn hoặc ít nút hơn trong lớp ẩn.
7. Tham khảo ví dụ trong văn bản, tính netB = 1,5 và f net $(\quad _b) = \frac{1}{1 + e^{-1.5}} = 0.8176$.
8. Giải thích cách hàm sigmoid kết hợp hành vi gần như tuyến tính, hành vi đường cong, và hành vi gần như không đổi.
9. Mô tả quá trình lan truyền ngược.
10. Vấn đề cơ bản đối với mạng nơ-ron là xây dựng một bộ trọng số sẽ giảm thiểu cái gì?

Machine Translated by Google

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu Framingham_training và Framingham_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Chuyển đổi các biến số nhị phân và thứ tự Death, Sex và Educ thành thừa số.

12. Chạy thuật toán mạng thần kinh để dự đoán Cái chết bằng Sex và Educ.

13. Vẽ mạng lưới thần kinh.

14. Lấy trọng số cho mạng nơ-ron. Xác định phần nào của mạng có cân nặng.

15. Đánh giá mô hình mạng thần kinh của bạn bằng bộ dữ liệu Framingham_test . Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán của Cái chết.

16. Chúng tôi so sánh mô hình mạng thần kinh của bạn với mô hình cơ sở nào? đã làm nó ra thực hiện đường cơ sở theo độ chính xác?

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với tập dữ liệu adult_ch6_training và adult_ch6_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

17. Chuẩn bị bộ dữ liệu cho mô hình mạng nơ-ron bằng cách thực hiện như sau:

    Một. Tạo một biến nhị phân bằng một nếu Cap_Gains_Losses lớn hơn 0 và bằng 0 nếu ngược lại. Gọi nó là CapGainsLossesPositive.

    b. Chuyển đổi Marital.status, Income và CapGainsLossesPositive thành thừa số.

18. Sử dụng tập dữ liệu huấn luyện, tạo mô hình mạng thần kinh để dự đoán Thu nhập của khách hàng sử dụng Marital.status và CapGainsLossesPositive. Gọi đây là NNM1 (Đối với Mô hình mạng thần kinh 1). Có được các câu trả lời dự đoán.

19. Vẽ sơ đồ mạng nơ ron NNM1.

20. Đánh giá NNM1 bằng bộ dữ liệu thử nghiệm. Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán của Thu nhập.

21. Chúng tôi so sánh NNM1 với mô hình cơ sở nào? NNM1 có vượt trội so với cơ sở không dòng theo độ chính xác?

22. Thu thập kết quả (bảng dự phòng) từ mô hình trước đó của bạn về adult_ch6_ tập dữ liệu training và adult_ch6_test trong bài tập Chương 6 và Chương 8. Từ Chương 6, gọi mẫu CART là CARTM1 và gọi mẫu C5.0 là C5M1. Từ Chương 8, gọi mô hình Naïve Bayes là NBM1.

23. So sánh kết quả NNM1 với ba mô hình từ bài tập trước, theo các tiêu chí sau. Thảo luận chi tiết mô hình nào hoạt động tốt nhất và kém nhất theo từng tiêu chí.

    Một. Sự chính xác

    b. Nhạy cảm

    c. độ đặc hiệu

Machine Translated by Google

Đối với các bài tập sau, hãy làm việc với bank_marketing_training và bank_
bộ dữ liệu marketing_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

24. Chuẩn bị bộ dữ liệu cho mô hình mạng thần kinh, bao gồm chuẩn hóa các biến.

25. Sử dụng tập dữ liệu huấn luyện, tạo mô hình mạng thần kinh để dự đoán Phản hồi của khách hàng sử dụng bất kỳ yếu tố dự đoán nào mà bạn cho là phù hợp. Có được các câu trả lời dự đoán.

26. Vẽ sơ đồ mạng nơ-ron.

27. Đánh giá mô hình mạng nơ-ron bằng tập dữ liệu thử nghiệm. Xây dựng một bảng dự phòng để so sánh các giá trị thực tế và dự đoán của Phản hồi.

28. Chúng tôi so sánh mô hình mạng thần kinh của bạn với mô hình cơ sở nào? đã làm nó ra thực hiện đường cơ sở theo độ chính xác?

29. Sử dụng cùng một công cụ dự đoán mà bạn đã sử dụng cho mô hình mạng thần kinh của mình, hãy xây dựng các mô hình để xác định trước dict Response bằng các thuật toán sau:

    Một. XE ĐẨY

    b. C5.0

    c. Naïve Bayes

30. So sánh kết quả của mô hình mạng thần kinh của bạn với ba mô hình từ bài tập trước, theo các tiêu chí sau. Thảo luận chi tiết mô hình nào hoạt động tốt nhất và kém nhất theo từng tiêu chí.

    Một. Sự chính xác

    b. Nhạy cảm

    c. độ đặc hiệu

Machine Translated by Google

chương 10

CỤM

10.1 GÓP CỤM LÀ GÌ?

Phân cụm đề cập đến việc nhóm các bản ghi, quan sát hoặc trường hợp thành các lớp của các đối tượng tương tự. Một cụm là một tập hợp các bản ghi tương tự nhau và không giống với các bản ghi trong các cụm khác. Phân cụm khác với phân loại ở chỗ không có biến mục tiêu cho phân cụm. Nhiệm vụ phân cụm không cố gắng phân loại, ước tính hoặc dự đoán giá trị của một biến mục tiêu. Thay vào đó, các thuật toán phân cụm tìm cách phân đoạn toàn bộ tập dữ liệu thành các nhóm con hoặc cụm tương đối đồng nhất, trong đó độ giống nhau của các bản ghi trong cụm được tối đa hóa và độ tương tự với các bản ghi bên ngoài cụm này được giảm thiểu.

Ví dụ: các phân khúc Nielsen PRIZM, do Claritas, Inc. phát triển, đại diện cho hồ sơ nhân khẩu học của từng khu vực địa lý ở Hoa Kỳ, về các kiểu lối sống riêng biệt, như được xác định bằng mã vùng. Ví dụ: các cụm được xác định cho mã zip 90210, Beverly Hills, California, là:

- Cụm 01: Địa chất vỏ trên
- Cụm 03: Movers and Shakers
- Cụm 04: Young Digerati
- Cụm 07: Tiền và Não
- Cụm 16: Bohemian Mix

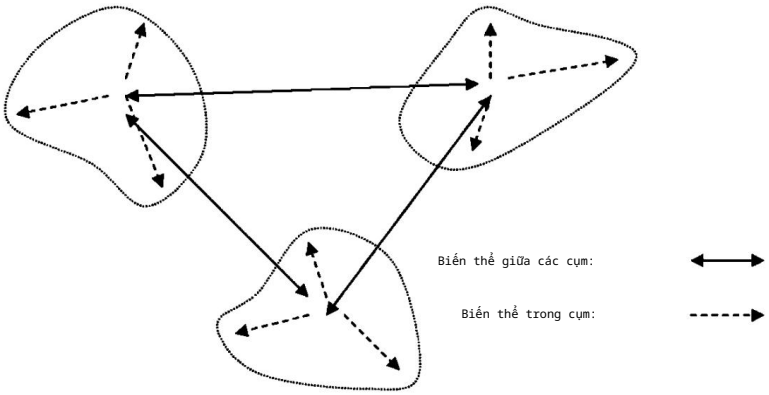
Mô tả cho Cụm 01: Upper Crust là "Địa chỉ độc quyền nhất của quốc gia, Upper Crust là lối sống giàu có nhất ở Mỹ, Nơi trú ẩn cho các cặp vợ chồng trong độ tuổi từ 45 đến 64. Không có phân khúc nào có mức độ tập trung cư dân có thu nhập trên $100,000 một năm và có bằng sau đại học. Và không nơi nào có mức sống sang trọng hơn."

Ví dụ về các nhiệm vụ phân cụm trong kinh doanh và nghiên cứu bao gồm:

- Nhắm mục tiêu tiếp thị một sản phẩm thích hợp cho một doanh nghiệp vốn nhỏ không có ngân sách tiếp thị lớn.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google



Hình 10.1 Các cụm nên có biến thể nhỏ bên trong cụm so với giữa-
biến thể cụm.

- Đối với mục đích kiểm toán kế toán, để phân chia hành vi tài chính thành lành tính và các danh mục đáng ngờ.
- Đối với cụm biểu hiện gen, trong đó số lượng rất lớn gen có thể biểu hiện hành vi tương tự.

Phân cụm thường được thực hiện như một bước sơ bộ trong quy trình khai thác dữ liệu, với các cụm kết quả được sử dụng làm đầu vào tiếp theo cho một kỹ thuật khác xuôi dòng, chẳng hạn như mạng thần kinh. Do kích thước khổng lồ của nhiều hiện tại-cơ sở dữ liệu ngày, trước tiên thường hữu ích khi áp dụng phân tích phân cụm để giảm không gian tìm kiếm cho các thuật toán xuôi dòng.

Tất cả các phương pháp phân cụm đều có mục tiêu là xác định các nhóm bản ghi sao cho độ tương tự trong một nhóm là rất cao, trong khi độ tương tự với các bản ghi trong các nhóm khác là rất thấp. Nói cách khác, như thể hiện trong Hình 10.1, các thuật toán phân cụm tìm cách xây dựng các cụm bản ghi sao cho biến thể giữa các cụm là lớn so với biến thể bên trong cụm.

10.2 GIỚI THIỆU VỀ THUẬT TOÁN K-MEANS CLUSTERING

Có nhiều phương pháp phân cụm khác nhau, bao gồm phân cụm theo thứ bậc, phân cụm mạng Kohonen và phân cụm BIRCH.1 Ở đây, chúng ta sẽ tập trung vào thuật toán phân cụm k-mean. Thuật toán phân cụm k-means2 là một

1Để biết chi tiết về các phương pháp phân cụm này, hãy xem Khai thác dữ liệu và phân tích dự đoán, Phiên bản thứ hai, của Daniel T. Larose và Chantal D. Larose, John Wiley và Sons, Inc., 2015.

2James B. MacQueen, Một số phương pháp phân loại và phân tích các quan sát đa biến. Trong Kỷ yếu của Hội nghị chuyên đề Berkeley lần thứ 5 về Xác suất và Thống kê Toán học, Tập. 1, tr.
281–297, Nhà xuất bản Đại học California, Berkeley, CA, 1967.

Machine Translated by Google

thuật toán đơn giản và hiệu quả để tìm các cụm trong dữ liệu. Các thuật toán tiến hành như sau:

- Bước 1: Hỏi người dùng tập dữ liệu nên được phân chia thành bao nhiêu cụm k .
- Bước 2: Gán ngẫu nhiên k bản ghi làm vị trí trung tâm cụm ban đầu.
- Bước 3: Đối với mỗi bản ghi, tìm trung tâm cụm gần nhất. Do đó, theo một nghĩa nào đó, mỗi trung tâm cụm "sở hữu" một tập hợp con các bản ghi, do đó đại diện cho một phân vùng của tập dữ liệu. Do đó, chúng ta có k cụm, C1 , C2 , ., Ck .
- Bước 4: Với mỗi k cụm, tìm tâm cụm và cập nhật vị trí của mỗi tâm cụm thành giá trị mới của tâm.
- Bước 5: Lặp lại các bước 3–5 cho đến khi hội tụ hoặc kết thúc.

Tiêu chí "gần nhất" trong bước 3 thường là khoảng cách Euclide, mặc dù các tiêu chí khác cũng có thể được áp dụng. Tâm cụm trong bước 4 được tìm thấy như sau. Giả sử rằng chúng ta có n điểm dữ liệu (a1 , b1 , c1), (a2 , b2 , c2), ., (an , bn , cn), cen troid của các điểm này là trọng tâm của các điểm này và nằm tại điểm (ai /n, bi /n, ci /n,). Ví dụ: các điểm (1,1,1), (1,2,1), (1,3,1) và (2,1,1) sẽ có trọng tâm

$$\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4}\right) = \left(1.25, 1.75, 1.0\right)$$

Thuật toán kết thúc khi trọng tâm không còn thay đổi. Trong tất cả các hồ sơ khác từ, thuật toán kết thúc khi đối với tất cả các cụm C1 , C2 , ., Ck , "sở hữu" của mỗi trung tâm cụm vẫn còn trong cụm đó. Để biết thêm thông tin về phân cụm k-means, hãy xem Khai thác dữ liệu và Phân tích dự đoán.

10.3 ỨNG DỤNG K-MEANS CLUSTERING

Chúng tôi áp dụng thuật toán phân cụm k-means cho white_wine_training và white_ bộ dữ liệu wine_test . Các bộ dữ liệu này được điều chỉnh từ bộ dữ liệu Chất lượng Rượu tại UCI.3 Dữ liệu bao gồm các đặc tính hóa học và chất lượng của một bộ sưu tập rượu vang trắng Bồ Đào Nha. Các yếu tố dự đoán là rượu và đường. Biến số mục tiêu là chất lượng, thước đo mức độ ngon của rượu, theo một người nếm thử chuyên nghiệp. Khi xây dựng các cụm, điều quan trọng là không bao gồm biến mục tiêu làm đầu vào cho thuật toán phân cụm. Làm như vậy sẽ làm sai lệch kết quả nếu sau này chúng ta sử dụng các cụm để dự đoán mục tiêu. Điều quan trọng nữa là chuẩn hóa hoặc bình thường hóa tất cả các yếu tố dự đoán, sao cho độ biến thiên lớn hơn của một yếu tố dự đoán không chi phối quá trình xây dựng cụm.

Bây giờ, thuật toán k-means yêu cầu nhà phân tích chỉ định số lượng cụm mong muốn. Để đơn giản ta xác định k = 2 cụm và tiến hành áp dụng

3https://archive.ics.uci.edu/ml/datasets/wine+quality. Ngoài ra P. Cortez, A. Cerdeira, F. Almeida, T. Matos, và J. Reis, Lập mô hình các sở thích về rượu bằng cách khai thác dữ liệu từ các đặc tính hóa lý, Hệ thống hỗ trợ quyết định, Elsevier, 47(4), 547–553, 2009.

Machine Translated by Google

BẢNG 10.1 Giá trị biến trung bình, theo cụm, đối với tập dữ liệu white_wine_training

Biến đổi	Cụm 1 : 712 loại rượu "Rượu ngọt"	Cụm 2:1097 loại rượu "Rượu khô"
đường_z	0,96	0,62
Rượu_z	0,76	0,49

Các kết quả Python trong Hình 10.2 đã được sao chép ở đây trong Bảng 10.1

thuật toán phân cụm k-means cho các biến dự đoán là rượu và đường. Python đã được sử dụng. Bảng 10.1 cho thấy giá trị trung bình của biến đối với mỗi yếu tố dự đoán, đối với hai cụm kết quả. Cụm 1 chứa 712 loại rượu vang, với hàm lượng đường trung bình cao hơn 0,96 độ lệch chuẩn so với hàm lượng đường trung bình tổng thể đối với tất cả các loại rượu vang trắng trong tập dữ liệu huấn luyện. (Nghĩa là, giá trị trung bình của Cụm 1 đối với biến sugar_z thông thường tiêu chuẩn là 0,96.) Tuy nhiên, nồng độ cồn trung bình của Cụm 1 nằm 0,76 độ lệch chuẩn dưới mức cồn trung bình đối với tất cả các loại rượu. Mặt khác, Cụm 2 chứa 1097 loại rượu vang, với hàm lượng đường trung bình thấp hơn Cụm 1 (0,62 độ lệch chuẩn dưới mức trung bình tổng thể) nhưng có hàm lượng cồn cao hơn, ở mức 0,49 độ lệch chuẩn trên mức trung bình tổng thể.

Vì vậy, chúng tôi có thể xác định Cụm 1 có chứa "Rượu ngọt", nhiều đường nhưng ít cồn, trong khi Cụm 2 chứa "Rượu khô", ít đường nhưng có nồng độ cồn cao hơn. Có lẽ, phương pháp xác thực cụm quan trọng nhất là thu được các cụm có ý nghĩa đối với một chuyên gia về chủ đề. Một tìm kiếm nhanh trên Internet cho thấy cách chúng ta thu được những loại rượu khác nhau này trong thế giới thực: "Trong quá trình lên men, thành của chúng ta biến đổi đường. ethanol [rượu]."4 Vì vậy, không có gì ngạc nhiên khi men thuật toán phân cụm đã phát hiện ra hai cụm "tự nhiên" này rượu vang trắng: rượu vang ngọt và rượu vang khô. Nhìn chung, rượu vang khô rõ ràng đã trải qua quá trình lên men lâu hơn so với rượu vang ngọt.

10.4 XÁC NHẬN CỤM

Các giải pháp cụm nên được xác nhận. Vì không có dự đoán nào được đưa ra bằng cách sử dụng tập dữ liệu huấn luyện, nên chúng tôi chỉ cần áp dụng lại thuật toán k-means, lần này là đối với white_wine_test tập dữ liệu và so sánh kết quả thu được với tập huấn luyện. Bảng 10.2

BẢNG 10.2 Giá trị biến trung bình, theo cụm, đối với bộ dữ liệu white_wine_test

Biến đổi	Cụm 1: 638 loại rượu "Rượu ngọt"	Cụm 2: 1122 rượu "Rượu khô"
đường_z	1,07	0,61
Rượu_z	0,80	0,46

Các kết quả Python trong Hình 10.3 được sử dụng cho bảng này. Các nhãn cụm "Cụm 1" và "Cụm 2" đã được đảo ngược để dễ hiểu. Các nhãn cụm này được thuật toán gán tùy ý

4https://en.wikipedia.org/wiki/Fermentation_in_winemaking

Machine Translated by Google

BẢNG 10.3 Sự khác biệt về trung bình biến, tập huấn luyện trừ tập kiểm tra, theo cụm

Biến đổi	Đào tạo – Kiểm tra "Rượu ngọt"	Đào tạo – Kiểm tra "Rượu khô"
đường_z	0,96 1,07 = 0,11	0,62(0,61) = 0,01
Rượu_z	0,76(0,80) = 0,04	0,49 0,46 = 0,03

chứa các giá trị biến trung bình kết quả, theo cụm. Như thể hiện trong Bảng 10.3, sự khác biệt về giá trị trung bình (tập huấn luyện trừ tập kiểm tra) là tương đối nhỏ. Các nhà phân tích muốn xác nhận thêm có thể thực hiện các bài kiểm tra t hai mẫu tại đây.

10.5 CÁCH THỰC HIỆN K-MEANS CLUSTERING SỬ DỤNG PYTHON

Tải các gói cần thiết.

```
nhập gấu trúc dưới dạng pd
từ số liệu thống kê nhập khẩu scipy
từ sklearn.cluster nhập KMeans
```

Đọc trong bộ dữ liệu white_wine_training là wine_train.

```
wine_train = pd.read_csv("C:/.../white_wine_training")
```

Để đơn giản, chúng ta hãy cô lập các biến dự đoán và lưu chúng dưới dạng X.

```
X = wine_train[['rượu', 'đường']]
```

Sau khi chúng tôi có các biến dự đoán, hãy chuẩn hóa chúng bằng cách sử dụng phép biến đổi điểm số z và lưu kết quả dưới dạng khung dữ liệu.

```
Xz = pd.DataFrame(stats.zscore(X), cột=['rượu', 'đường'])
```

Như ở chương 3, lệnh stats.zscore sẽ chuyển đổi các biến trong X vào điểm số z của họ. Chúng tôi lưu các biến được chuẩn hóa mới dưới dạng khung dữ liệu bằng cách sử dụng lệnh DataFrame() . Các cột đầu vào tùy chọn sử dụng tên đã cho làm tên cột. Chúng tôi lưu kết quả dưới dạng Xz.

Bây giờ, chúng tôi chạy phân cụm k-mean trên tập dữ liệu huấn luyện.

```
kmeans01 = KMeans(n_clusters = 2).fit(Xz)
```

Lệnh KMeans() thiết lập các tham số của thuật toán k-means. Trong trường hợp của chúng tôi, đầu vào n_cluster = 2 chỉ định rằng chúng tôi muốn có hai cụm. phù hợp() lệnh chạy thuật toán k-mean được chỉ định trên dữ liệu của chúng tôi, với đầu vào Xz đưa ra tập dữ liệu chúng tôi muốn phân cụm. Kết quả phân cụm được lưu dưới tên kmeans01.

Machine Translated by Google

Để điều tra kết quả phân cụm, chúng ta cần lưu tư cách thành viên cụm như biến riêng của nó.

```
cụm = kmeans01.nhãn_
```

Thông tin thành viên cụm được chứa trong nhãn_ trong kết quả kmeans01 mà chúng tôi đã lưu trước đó. Để đơn giản, chúng tôi lưu nó dưới dạng đối tượng riêng của nó, cụm.

Khi chúng tôi có thông tin thành viên cụm, chúng tôi có thể tách các bản ghi thành hai nhóm dựa trên tư cách thành viên của cụm.

```
Cụm1 = Xz.loc[cụm == 0]
Cụm2 = Xz.loc[cụm == 1]
```

Kết quả là hai bộ dữ liệu, một cho các bản ghi trong Cụm 1 và một cho các bản ghi trong Cụm 2.

Cuối cùng, chúng tôi tính toán số liệu thống kê tóm tắt của hai cụm bằng cách sử dụng lệnh description() .

```
Cluster1.describe()
Cluster2.describe()
```

Lệnh description() in các số liệu thống kê khác nhau cho các biến trong mỗi cụm, được minh họa trong Hình 10.2. Các phương tiện từ Hình 10.2 được sao chép vào Bảng 10.1.

Để xác thực kết quả phân cụm, hãy chạy thuật toán phân cụm k-means trên tập dữ liệu thử nghiệm. Mã được đưa ra dưới đây và tương tự với trường hợp tập huấn luyện. Các kết quả mô tả cho tập dữ liệu thử nghiệm được thể hiện trong Hình 10.3, với phương tiện được sao chép vào Bảng 10.2.

```
wine_test = pd.read_csv("C:/.../white_wine_test")
X_test = wine_test[['rượu', 'đường']]
Xz_test = pd.DataFrame(stats.zscore(X_test), cột=['rượu',
'đường'])
```

```
In [106]: Cluster1.describe()
Out[106]:
        alcohol_z     sugar_z
count  712.000000  712.000000
mean    -0.755428    0.961034
std      0.580989    0.818726
min     -1.826971   -0.908740
25%     -1.158911    0.354160
50%     -0.908388    0.867883
75%     -0.407343    1.488630
max      2.014374    5.512788
```

```
In [107]: Cluster2.describe()
Out[107]:
        alcohol_z     sugar_z
count  1097.000000  1097.000000
mean      0.490305    -0.623752
std       0.905663     0.475694
min      -1.576448    -1.122791
25%      -0.156821    -0.951551
50%       0.427732    -0.844525
75%       1.179299    -0.352208
max       2.891203     1.477928
```

Hình 10.2 Các mô tả Python của Cụm 1 (trái) và Cụm 2 (phải) từ tập dữ liệu huấn luyện.

Machine Translated by Google


```
In [115]: Cluster1_test.describe()
Out[115]:
         alcohol_z      sugar_z
count  1122.000000  1122.000000
mean      0.456397    -0.605782
std       0.903287     0.459740
min      -1.675754    -1.089453
25%      -0.218729    -0.945241
50%       0.368129    -0.821632
75%       1.157351    -0.285988
max       2.776268     1.423949
```

```
In [116]: Cluster2_test.describe()
Out[116]:
         alcohol_z     sugar_z
count  638.000000  638.000000
mean    -0.802630    1.065341
std      0.561207    0.779670
min     -2.080483   -1.037949
25%     -1.190079    0.396441
50%     -0.947241    1.032518
75%     -0.542512    1.583612
max      1.562080    3.298700
```

Hình 10.3 Các mô tả Python của Cụm 1 (trái) và Cụm 2 (phải) từ tập dữ liệu thử nghiệm.

```
kmeans_test = KMeans(n_clusters = 2).fit(Xz_test)
cluster_test = kmeans_test.labels_ # Thành viên cụm
Cluster1_test = Xz_test.loc[cluster_test == 0]
Cluster2_test = Xz_test.loc[cluster_test == 1]
Cluster1_test.describe()
Cluster2_test.describe()
```

10.6 CÁCH THỰC HIỆN K-MEANS CLUSTERING SỬ DỤNG R

Đọc tập dữ liệu white_wine_training dưới dạng wine_train và đặt các biến dự đoán thành ma trận riêng của chúng.

```
X <- subset(wine_train, select = c("alcohol", "sugar"))
```

Lệnh subset() sẽ chọn hai biến có tên là rượu và đường từ tập dữ liệu wine_train và lưu trữ chúng dưới tên riêng của chúng, X.

Bây giờ, chúng tôi chuẩn hóa cả hai biến dự đoán và lưu đầu ra dưới dạng dữ liệu khung. Cần có định dạng khung dữ liệu để chạy lệnh kmeans() .

```
Xs <- as.data.frame(scale(X))
colnames(Xs) <- c("alcohol_z", "sugar_z")
```

Lệnh scale() biến các biến trong X thành điểm số z tương ứng của chúng, trong khi as.data.frame lưu kết quả dưới dạng khung dữ liệu. Kết quả được lưu dưới dạng Xs. Chúng tôi chỉnh sửa tên cột bằng cách sử dụng colnames() để nhấn mạnh rằng các biến hiện đã được chuẩn hóa.

Lệnh kmeans() được bao gồm trong bản cài đặt cơ sở của R. Tuy nhiên, nếu bạn gặp lỗi cho biết "Không thể tìm thấy chức năng 'kmeans'," hãy cài đặt và mở gói thống kê bằng cách sử dụng install. gói ("số liệu thống kê"); thư viện (số liệu thống kê).

Chạy thuật toán phân cụm k-means.

```
kmeans01 <- kmeans(Xs, tâm = 2)
```

Machine Translated by Google

Các đầu vào bắt buộc là Xs, khung dữ liệu và center = 2, số lượng cụm mà thuật toán sẽ tìm kiếm. Lưu ý rằng chúng tôi lưu đầu ra của thuật toán phân cụm dưới dạng kmeans01.

Chúng ta cần lưu tư cách thành viên cụm của mỗi bản ghi dưới dạng biến riêng của nó.

```
cụm <- as.factor(kmeans01$cluster)
```

Mã kmeans01$cluster sẽ nhận tư cách thành viên cụm của mỗi bản ghi. Vì chúng ta có hai cụm trong ví dụ này nên các giá trị của kmeans01$cluster sẽ là 1 hoặc 2. Lệnh as.factor() sẽ lưu chuỗi số đó làm thừa số.

Bây giờ chúng ta hãy xem số liệu thống kê mô tả của từng cụm. Đầu tiên, chúng tôi tách các bản ghi thành hai nhóm, dựa trên cụm mà chúng thuộc về.

```
Cluster1 <- Xs[ which(cụm == 1), ]
Cluster2 <- Xs[ which(cụm == 2), ]
```

Lệnh which() , được sử dụng trong ký hiệu ngoặc ở phía bên trái của dấu phẩy, chỉ chọn những bản ghi có tư cách thành viên cụm là 1 (đối với Cụm 1) hoặc 2 (đối với Cụm 2). Sau đó, chúng tôi chạy lệnh summary() trên từng nhóm riêng lẻ và lưu ý đầu ra quan tâm.

```
tóm tắt(Cụm1)
tóm tắt(Cụm2)
```

Kết quả được thể hiện trong Hình 10.4.

Để xác thực các cụm, hãy nhập tập dữ liệu white_wine_test là wine_test và tập hợp con các biến rượu và đường. Thực hiện tiêu chuẩn hóa biến và phân cụm k-means trên tập dữ liệu thử nghiệm. Mã được đưa ra dưới đây. Kết quả được thể hiện trong Hình 10.5.

```
X_test <- subset(wine_test, select = c("alcohol", "sugar"))
Xs_test <- as.data.frame(scale(X_test))
colnames(Xs_test) <- c("alcohol_z", "sugar_z")
kmeans01_test <- kmeans(Xs_test, center = 2)
cluster_test <- as.factor(kmeans01_test$cluster)
Cluster1_test <- Xs[ which(cluster_test == 1), ]
Cluster2_test <- Xs[ which(cluster_test == 2), ]
tóm tắt(Cluster1_test); tóm tắt(Cluster2_test)
```

```
> summary(Cluster1)
   alcohol_z          sugar_z
 Min.   :-1.5760   Min.   :-1.1225
 1st Qu.:-0.1568   1st Qu.:-0.9513
 Median : 0.4276   Median :-0.8443
 Mean   : 0.4902   Mean   :-0.6236
 3rd Qu.: 1.1790   3rd Qu.:-0.3521
 Max.   : 2.8904   Max.   : 1.4775
```

```
> summary(Cluster2)
   alcohol_z          sugar_z
 Min.   :-1.8265   Min.   :-0.9085
 1st Qu.:-1.1586   1st Qu.: 0.3541
 Median :-0.9081   Median : 0.8676
 Mean   :-0.7552   Mean   : 0.9608
 3rd Qu.:-0.4072   3rd Qu.: 1.4882
 Max.   : 2.0138   Max.   : 5.5113
```

Hình 10.4 Các mô tả R của Cụm 1 và Cụm 2 từ tập dữ liệu huấn luyện.

Machine Translated by Google


```
> summary(Cluster1_test)
   alcohol_z          sugar_z
 Min.   :-1.8265   Min.   :-1.1011
 1st Qu.:-1.2421   1st Qu.:-0.7801
 Median :-0.9081   Median : 0.3327
 Mean   :-0.8267   Mean   : 0.3606
 3rd Qu.:-0.5742   3rd Qu.: 1.2100
 Max.   : 1.4294   Max.   : 5.5113
```

```
> summary(Cluster2_test)
   alcohol_z          sugar_z
 Min.   :-1.7430   Min.   :-1.1225
 1st Qu.:-0.2403   1st Qu.:-0.9245
 Median : 0.3441   Median :-0.5019
 Mean   : 0.3697   Mean   :-0.1869
 3rd Qu.: 1.0120   3rd Qu.: 0.3327
 Max.   : 1.9303   Max.   : 3.5853
```

Hình 10.5 Mô tả R của Cụm 1 (trái) và Cụm 2 (phải) từ bộ dữ liệu thử nghiệm.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Giải thích việc phân cụm đang cố gắng đạt được điều gì, sử dụng các khái niệm về sự thay đổi giữa các cụm và sự thay đổi bên trong cụm.
2. Phân cụm tìm cách nhóm những bản ghi hay biến nào?
3. Tại sao việc áp dụng phân cụm khá sớm trong quá trình lập mô hình lại hữu ích?
4. Đúng hay sai: phân cụm k-means tự động chọn số lượng cụm tối ưu.
5. Tại sao chúng ta bỏ biến mục tiêu làm đầu vào cho thuật toán phân cụm?
6. Giải thích cách chúng tôi tiến hành thực hiện xác thực cụm.
7. Tại sao chúng ta chuẩn hóa các yếu tố dự đoán số trước khi phân cụm?
8. Phương pháp xác nhận cụm quan trọng nhất có lẽ là gì?
9. Trọng tâm của các điểm (1, 5), (2, 4) và (3, 3) là gì?
10. Cung cấp một ví dụ về phân cụm trong thế giới hàng ngày không được thảo luận trong phần này chương.

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu white_wine_training và white_wine_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Nhập và chuẩn hóa cả tập dữ liệu huấn luyện và kiểm tra.
12. Chạy phân cụm k-mean trên tập dữ liệu huấn luyện, sử dụng hai cụm.
13. Cho giá trị trung bình của từng biến trong mỗi cụm và sử dụng giá trị trung bình để xác định cụm “Rượu vang khô” và cụm “Rượu vang ngọt”.
14. Xác thực kết quả phân cụm bằng cách chạy phân cụm k-means trên tập dữ liệu thử nghiệm, sử dụng hai cụm và xác định cụm “Rượu vang khô” và cụm “Rượu vang ngọt”.

Machine Translated by Google

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu ngũ cốc . Sử dụng Python hoặc R để giải quyết từng vấn đề.

15. Tập hợp các biến Chất béo và Natri vào khung dữ liệu riêng của chúng, X. Chuẩn hóa tập dữ liệu.

16. Chạy phân cụm k-means trên tập dữ liệu, sử dụng ba cụm.

17. Lấy tóm tắt của từng biến trong mỗi cụm và sử dụng tóm tắt để nhận dạng:

    Một. Một cụm ít chất béo, ít natri.

    b. Một cụm ít chất béo, natri cao.

    c. Một cụm natri cao, chất béo cao.

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu Framingham_training và Framingham_test . Chỉ sử dụng các trường Giới tính và Tuổi . Chuẩn hóa Tuổi.

18. Chạy phân cụm k-means trên tập dữ liệu Framingham_training , yêu cầu k = 2 cụm.

19. Lập bảng thống kê tóm tắt các cụm của bạn. Mô tả những gì hai clus ters bao gồm.

20. Thực hiện phân cụm k-means trên tập dữ liệu Framingham_test , yêu cầu k = 2 cụm.

21. Báo cáo kết quả từ bộ kiểm tra của bạn. Các cụm của bạn đã được xác thực chưa?

22. Chạy lại phân cụm k-means trên tập dữ liệu Framingham_training , lần này chỉ định k = 3 cụm.

23. Lập bảng thống kê tóm tắt các cụm của bạn. Mô tả hồ sơ nào thuộc về đến từng cụm.

24. Thực hiện phân cụm k-means trên tập dữ liệu Framingham_test , chỉ định k = 3 cụm.

25. Báo cáo kết quả từ bộ kiểm tra của bạn. Các cụm của bạn đã được xác thực chưa?

26. Chạy phân cụm k-means trên tập dữ liệu Framingham_training . Chỉ định k = 4 cụm.

27. Lập bảng thống kê tóm tắt bốn cụm của bạn. Mô tả rõ ràng bốn của bạn cụm.

28. Thực hiện phân cụm k-means trên tập dữ liệu Framingham_test , yêu cầu k = 4 cụm.

29. Báo cáo kết quả từ tập kiểm tra của bạn. Các cụm của bạn đã được xác thực chưa?

30. Bạn thích giải pháp phân cụm nào, k = 2, 3 hoặc 4, và tại sao?

Machine Translated by Google

chương 11

MÔ HÌNH HỒI QUY

11.1 NHIỆM VỤ DỰ TOÁN

Cho đến nay trong Giai đoạn lập mô hình, chúng tôi đã đề cập đến các nhiệm vụ sau:

- Nhiệm vụ phân loại
- Nhiệm vụ phân cụm

Vẫn còn hai nhiệm vụ phải hoàn thành:

- Nhiệm vụ lập dự toán
- Nhiệm vụ hiệp hội

Trong chương này, chúng tôi đề cập đến nhiệm vụ ước lượng; sau đó, trong Chương 14, chúng ta sẽ đề cập đến nhiệm vụ liên kết.

Phương pháp phổ biến nhất để thực hiện nhiệm vụ ước tính là hồi quy tuyến tính. Hồi quy tuyến tính đơn giản ước tính mối quan hệ giữa một công cụ dự đoán số và mục tiêu liên tục, sử dụng một đường thẳng. Mô hình hồi quy bội xấp xỉ mối quan hệ giữa một tập hợp các yếu tố dự đoán p>1 và một mục tiêu liên tục duy nhất, sử dụng mặt phẳng p-chiều hoặc siêu phẳng.

11.2 MÔ HÌNH HỒI QUY MÔ TẢ

Mô hình hồi quy bội thông thường là một mô hình tham số, được xác định bởi phương trình sau:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad + \beta_{x_{trang}} + \varepsilon$$

trong đó x đại diện cho các biến dự đoán và β′ đại diện cho các tham số mô hình chưa biết, có giá trị được ước tính bằng cách sử dụng dữ liệu.[1] Bây giờ, hãy ước tính

[1]Để biết thêm thông tin về mô hình hồi quy suy luận, hãy xem Khai thác dữ liệu và Phân tích dự đoán.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

tham số mô hình sử dụng dữ liệu mẫu đại diện cho suy luận thống kê cổ điển. Tuy nhiên, Phương pháp khoa học dữ liệu được nêu trong Chương 1 sử dụng chéo xác thực hơn là suy luận thống kê cổ điển để xác thực kết quả mô hình. Vì vậy, trong cuốn sách này, chúng ta sẽ bỏ qua phương trình hồi quy tham số ở trên, ủng hộ cách tiếp cận mô tả đối với mô hình hồi quy, sử dụng phương trình hồi quy sau:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + \quad + b_{trang}x$$

Trong phương trình hồi quy này, yˆ đại diện cho giá trị ước tính của biến mục tiêu y, b đại diện cho các giá trị đã biết của hệ số hồi quy và x′s đại diện cho các biến dự đoán.

11.3 MỘT ỨNG DỤNG CỦA NHIỀU MÔ HÌNH HỒI QUY

Để minh họa hồi quy bội, chúng tôi chuyển sang bộ dữ liệu clothing_sales_training và clothing_sales_test . Khách hàng có một số dữ liệu về chi tiêu của khách hàng và muốn ước tính Sales_per_Visit, dựa trên ba yếu tố dự đoán:

- Số ngày giữa các lần mua (“Ngày,” Liên tục: Số ngày trung bình giữa các lần mua.)
- Thẻ tín dụng (“CC,” Cờ: Khách hàng có thẻ tín dụng của cửa hàng không?)
- Tài khoản Web ("Web," Cờ: Khách hàng có tài khoản web không?)

Vì vậy, phương trình hồi quy tạm thời của chúng tôi sẽ là:

$$\text{Doanh số mỗi lượt truy cập} = b_0 + b_1(\text{Ngày giữa các lần mua}) + b_2(\text{Thẻ tín dụng}) + b_3(\text{Tài khoản Web})$$

Bởi vì chỉ có một yếu tố dự đoán liên tục, nên không cần thiết phải loại bỏ các yếu tố dự đoán. Kết quả hồi quy của Doanh số bán hàng trên mỗi lượt truy cập so với ba yếu tố dự đoán cho tập hợp đào tạo được hiển thị trong Hình 11.1. Chúng tôi sử dụng các giá trị p làm hướng dẫn để cho chúng tôi biết biến nào thuộc về mô hình. Lưu ý rằng chúng tôi không thực hiện suy luận như vậy (miền thông thường của giá trị p), bởi vì chúng tôi sẽ cẩn thận xác thực chéo các kết quả này với tập dữ liệu thử nghiệm. Ngưỡng giá trị p thông thường để giữ lại các biến trong mô hình hồi quy là khoảng 0,05, mặc dù các giá trị ngưỡng khác nhau giữa các trường. Các biến có giá trị p thấp hơn ngưỡng được giữ lại trong mô hình.

Từ Hình 11.1, chúng ta thấy rằng Tài khoản Web, với giá trị p là 0,533, không thuộc mô hình. Kết quả hồi quy cho tập dữ liệu thử nghiệm trong Hình 11.2 đồng tình rằng Tài khoản Web không thuộc về mô hình. Điều này để lại cho chúng tôi phương trình hồi quy của chúng tôi là:

$$\text{Doanh số mỗi lượt truy cập} = b_0 + b_1(\text{Ngày giữa các lần mua}) + b_2(\text{Thẻ tín dụng}))$$

Machine Translated by Google

	coef	std err	t	P>|t|	[0.025	0.975]
const	73.3654	4.676	15.689	0.000	64.192	82.538
CC	21.8175	4.766	4.578	0.000	12.468	31.167
Days	0.1644	0.017	9.802	0.000	0.131	0.197
Web	7.2755	11.658	0.624	0.533	-15.593	30.144

Hình 11.1 Kết quả hồi quy Python cho tập dữ liệu huấn luyện.

	coef	std err	t	P>|t|	[0.025	0.975]
const	80.2877	4.000	20.071	0.000	72.441	88.135
CC	20.8955	4.170	5.011	0.000	12.716	29.075
Days	0.1261	0.014	9.120	0.000	0.099	0.153
Web	12.4811	9.054	1.378	0.168	-5.280	30.242

Hình 11.2 Xác thực Python của kết quả hồi quy với tập dữ liệu thử nghiệm.

	coef	std err	t	P>|t|	[0.025	0.975]
const	73.6209	4.657	15.808	0.000	64.485	82.757
CC	22.1357	4.738	4.672	0.000	12.842	31.429
Days	0.1637	0.017	9.784	0.000	0.131	0.197

Hình 11.3 Mô hình hồi quy cuối cùng cho tập dữ liệu huấn luyện bằng Python.

	coef	std err	t	P>|t|	[0.025	0.975]
const	80.7656	3.986	20.260	0.000	72.946	88.586
CC	21.5262	4.146	5.192	0.000	13.393	29.659
Days	0.1254	0.014	9.071	0.000	0.098	0.152

Hình 11.4 Xác thực mô hình hồi quy cuối cùng với bộ dữ liệu thử nghiệm bằng Python.

Do đó, chúng tôi chạy lại mô hình hồi quy, lần này bỏ qua Tài khoản Web từ mô hình. Kết quả của tập huấn luyện và tập kiểm tra được thể hiện trong Hình 11.3 và 11.4. Sử dụng các hệ số từ tập huấn luyện, chúng tôi thu được mô hình hồi quy cuối cùng là:

$$(\text{Doanh số trên mỗi lượt truy cập}) = 73.6209 + 0.1637(\text{Số ngày giữa các lần mua hàng}) + 22.1357(\text{Thẻ tín dụng})$$

Nghĩa là, doanh thu ước tính trên mỗi lượt truy cập đối với cơ sở khách hàng của chúng tôi là 73,6209 đô la cộng với 0,1637 đô la nhân với số ngày giữa các lần mua hàng cộng với 22,1357 đô la nếu họ có thẻ tín dụng của cửa hàng. Chúng tôi thấy rằng khách hàng có xu hướng chi tiêu nhiều hơn nếu họ có thẻ tín dụng của cửa hàng. Ngoài ra, thời gian giữa các lần truy cập càng lâu, khách hàng càng có xu hướng chi tiêu nhiều hơn.

Machine Translated by Google

Chúng tôi diễn giải các hệ số hồi quy này như sau:

- Thẻ tín dụng. Mức tăng doanh số trên mỗi lượt truy cập ước tính đối với khách hàng có thẻ tín dụng của cửa hàng (so với khách hàng không có thẻ tín dụng của cửa hàng) là 22,1357 USD, khi số ngày giữa các lần mua không đổi.
- Số ngày mua hàng. Đối với mỗi lần tăng thêm một ngày trong số ngày trung bình giữa các lần mua hàng, mức tăng doanh số trên mỗi lượt truy cập ước tính là $0,1637, khi Thẻ tín dụng không đổi. Điều này có thể hiểu rõ hơn nếu chúng ta so sánh hai người mua sắm, Khách hàng A và Khách hàng B, trong đó Khách hàng A có số ngày trung bình giữa các lần mua hàng dài hơn một tháng (30 ngày) so với Khách hàng B. Khi đó, Doanh số trên mỗi lượt truy cập của Khách hàng A là 30×0,1637 đô la = $4,91 nhiều hơn Khách hàng B, giữ Thẻ tín dụng không đổi.

11.4 CÁCH THỰC HIỆN MÔ HÌNH HỒI QUY ĐA SỐ BẰNG PYTHON

Đầu tiên, như mọi khi, chúng tôi tải các gói cần thiết.

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
nhập statsmodels.api dưới dạng sm
```

Tiếp theo, chúng tôi nhập bộ dữ liệu clothing_sales_training và clothing_sales_test lần lượt là sales_train và sales_test .

```
sales_train = pd.read_csv("C:/.../clothing_sales_
đào tạo.csv")
sales_test = pd.read_csv("C:/.../clothing_sales_test.csv")
```

Để đơn giản, chúng tôi tách các biến dự đoán và biến mục tiêu. Chúng tôi gọi khung dữ liệu của các biến dự đoán X và biến mục tiêu y.

```
X = pd.DataFrame(sales_train[['CC', 'Days', 'Web']])
y = pd.DataFrame(sales_train[['Doanh số mỗi lượt truy cập']])
```

Để có một số hạng không đổi b0 trong mô hình hồi quy của chúng ta, chúng ta cần thêm một biến không đổi vào các biến dự đoán của mình.

```
X = sm.add_constant(X)
```

Chạy lệnh add_constant() trên biến X sẽ thêm một cột vào khung dữ liệu chứa giá trị một (1.0).

Cuối cùng, chúng tôi chạy mô hình hồi quy bội.

```
model01 = sm.OLS(y, X).fit()
```

OLS là viết tắt của “Bình phương nhỏ nhất thông thường”, là phương pháp được sử dụng để phù hợp với mô hình hồi quy này. Lưu ý rằng hai đầu vào của lệnh OLS() là đích

Machine Translated by Google

biến y và các biến dự đoán X. Lưu mô hình được trang bị dưới dạng model01. Để có kết quả của mô hình hồi quy, hãy chạy lệnh summary() trên model01.

```
model01.summary()
```

Một đoạn trích từ đầu ra của lệnh summary() được hiển thị trong Hình 11.1. Các hệ số hồi quy được đặt trong cột coef .

Để xác minh kết quả mô hình hồi quy, chúng tôi chạy mã tương tự trên sales_test bộ dữ liệu. Mã được đưa ra dưới đây, các giải thích tương đương với các giải thích được đưa ra trước đó trong phần này.

```
X_test = pd.DataFrame(sales_test[['CC', 'Days', 'Web']])
y_test = pd.DataFrame(sales_test[['Doanh số mỗi lượt truy cập']])
X_test = sm.add_constant(X_test)
model01_test = sm.OLS(y_test, X_test).fit()
model01_test.summary()
```

Đoạn trích từ kết quả từ lệnh summary() chạy trên model01_ kiểm tra được đưa ra trong Hình 11.2. Các kết quả xác nhận các kết quả từ model01.

Để loại bỏ biến Web khỏi mô hình hồi quy, chúng tôi xác định lại X khung dữ liệu để chỉ bao gồm hai biến dự đoán còn lại. Sau khi làm như vậy, chúng ta cũng cần thêm lại thuật ngữ không đổi vào khung dữ liệu biến dự đoán của mình.

```
X = pd.DataFrame(sales_train[['CC', 'Days']])
X = sm.add_constant(X)
```

Khi khung dữ liệu biến dự đoán đã sẵn sàng, chúng tôi chạy OLS() và fit() các lệnh trên biến mục tiêu y và khung dữ liệu X mới một lần nữa. Lưu ý rằng chúng tôi đã không thay đổi đầu vào y , vì chỉ cần thay đổi đầu vào X. Lưu mô hình mới với tên model02 và chạy lệnh summary() trên model02 để xem kết quả.

```
model02 = sm.OLS(y, X).fit()
model02.summary()
```

Một đoạn trích từ đầu ra của lệnh model02.summary() được hiển thị trong Hình 11.3.

Để xác minh mô hình nhỏ hơn này, chúng tôi chạy mã tương tự trên dữ liệu thử nghiệm. Một lần nữa, những lời giải thích tương tự như những lời giải thích cho ví dụ trước.

```
X_test = pd.DataFrame(sales_test[['CC', 'Days']])
X_test = sm.add_constant(X_test)
model02_test = sm.OLS(y_test, X_test).fit()
model02_test.summary()
```

Một đoạn trích của đầu ra từ lệnh model02_test.summary() được hiển thị trong Hình 11.4.

Machine Translated by Google

11.5 CÁCH THỰC HIỆN MÔ HÌNH HỒI QUY ĐA SỐ BẰNG R

Tải tập dữ liệu clothing_sales_training và clothing_sales_test dưới dạng sales_train và sales_test, tương ứng. Tiếp theo, hãy đảm bảo các biến nhị phân là thừa số trong cả hai tập dữ liệu.

```
sales_train$CC <- as.factor(sales_train$CC)
sales_train$Web <- as.factor(sales_train$Web)
sales_test$CC <- as.factor(sales_test$CC)
sales_test$Web <- as.factor(sales_test$Web)
```

Bây giờ, hãy chạy toàn bộ mô hình cho tập dữ liệu huấn luyện.

```
model01 <- lm(công thức = Sales.per.Visit ~ Days + Web + CC, data =
sales_train)
```

Lưu ý hai phần đầu vào được yêu cầu: công thức và dữ liệu. Công thức có cùng dạng Mục tiêu ~ Dự đoán mà chúng ta đã thấy trước đây.
Đầu vào data = sales_train chỉ định tập dữ liệu mà các biến của chúng ta đến từ đó. Chúng tôi lưu kết quả của mô hình hồi quy dưới tên model01.
Để xem tóm tắt các kết quả của mô hình, hãy chạy lệnh summary() trên model01.

```
tóm tắt(model01)
```

Một đoạn trích của đầu ra được tạo bởi bản tóm tắt (model01) được hiển thị trong Hình 11.5.

Để xác thực mô hình, hãy thay đổi dữ liệu đầu vào để chỉ định rằng các biến hiện đến từ tập dữ liệu sales_test .

```
model01_test <- lm(công thức = Sales.per.Visit ~ Days + Web + CC, data
= sales_test)
```

Để xem tóm tắt hồi quy của mô hình mới này, hãy chạy tóm tắt (model01_test). Một đoạn trích của đầu ra được tạo bởi lệnh này được hiển thị trong Hình 11.6.

Để xóa các biến khỏi mô hình, hãy xóa tên của chúng khỏi chuỗi biến dự đoán ở bên phải dấu ngã trong lệnh lm() .

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 73.36537    4.67621  15.689  < 2e-16 ***
Days         0.16438    0.01677   9.802  < 2e-16 ***
Web1         7.27550   11.65786   0.624    0.533
CC1         21.81750    4.76607   4.578  5.1e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hình 11.5 Kết quả hồi quy từ R cho tập dữ liệu huấn luyện.

Machine Translated by Google


```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 80.28768    4.00016  20.071  < 2e-16 ***
Days         0.12610    0.01383   9.120  < 2e-16 ***
Web1        12.48109    9.05412   1.378    0.168
CC1         20.89548    4.16987   5.011 6.11e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hình 11.6 Xác thực kết quả hồi quy từ R với tập dữ liệu thử nghiệm.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 73.62090    4.65727  15.808  < 2e-16 ***
Days         0.16374    0.01674   9.784  < 2e-16 ***
CC1         22.13570    4.73772   4.672 3.26e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hình 11.7 Kết quả hồi quy cuối cùng từ R cho tập dữ liệu huấn luyện.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 80.76564    3.98640  20.260  < 2e-16 ***
Days         0.12538    0.01382   9.071  < 2e-16 ***
CC1         21.52618    4.14603   5.192 2.39e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hình 11.8 Xác thực kết quả hồi quy cuối cùng từ R với tập dữ liệu thử nghiệm.

Các lệnh để chạy các mô hình mới và tạo đầu ra tóm tắt được đưa ra bên dưới.

```
model02 <- lm(công thức = Sales.per.Visit ~ Days + CC, data = sales_train)

tóm tắt(model02)
model02_test <- lm(công thức = Sales.per.Visit ~ Days + CC, data = sales_test)

tóm tắt(model02_test)
```

Lưu ý rằng các đoạn trích của đầu ra được tạo bởi tóm tắt (model02) và summary(model02_test) được thể hiện tương ứng trong Hình 11.7 và 11.8.

11.6 ĐÁNH GIÁ MÔ HÌNH ĐỂ ƯỚC LƯỢNG

Chúng ta có thể sử dụng phương trình hồi quy để đưa ra dự đoán (ước tính) về doanh số bán hàng trên mỗi lượt truy cập. Ví dụ: hãy xem xét Khách hàng 1 trong tập huấn luyện, người đã mua hàng trong 333 ngày và không có thẻ tín dụng của cửa hàng (Thẻ tín dụng = 0). Cắm các giá trị này vào phương trình hồi quy, chúng tôi thu được:

Doanh số trên mỗi lượt truy cập 73 62 0 16 37 333 22 14 0$ 0,13

Machine Translated by Google


```
In [194]: np.sqrt(model02.scale)
Out[194]: 87.54136112817613
```

Hình 11.9 Sai số chuẩn của ước tính từ Python.

Nghĩa là, bằng cách sử dụng mô hình hồi quy, chúng tôi sẽ ước tính doanh số bán hàng trung bình trên mỗi lượt truy cập của khách hàng này là $\hat{y}$ = $128,13. Tuy nhiên, doanh thu thực tế trên mỗi lượt truy cập của khách hàng này là y = $184,23. Vì vậy, lỗi dự đoán (dư) cho khách hàng này là:

$$\text{sai số dự đoán } y - \hat{y} = 184.23 - 128.13 = 56.10 \text{ \$}.$$

Vì vậy, khách hàng này chi tiêu nhiều hơn $56,10 so với dự kiến, dựa trên số ngày của họ giữa các lượt truy cập và tình trạng thẻ tín dụng.

Kích thước điển hình của lỗi dự đoán được đưa ra bởi thống kê s, lỗi tiêu chuẩn của ước tính.

$$s = \sqrt{MSE} = \sqrt{\frac{SSE}{n-p-1}} = \sqrt{\frac{\sum(y-\hat{y})^2}{n-p-1}}$$

Ở đây, s = $87,54, nghĩa là kích thước của sai số dự đoán điển hình của mô hình là khoảng $87,54, như có thể thấy trong Hình 11.9. Giá trị lớn này là do dữ liệu của chúng tôi được trích xuất từ một tập dữ liệu lớn hơn nhiều, bao gồm hàng chục yếu tố dự đoán khác, hữu ích để ước tính mô hình của chúng tôi chính xác hơn. Tuy nhiên, thông thường, s là thước đo rất quan trọng để đo lường hiệu quả của mô hình hồi quy.

Trong dẫn xuất của nó, s bình phương các sai số dự đoán, do đó có thể tạo ra các ngoại lệ có ảnh hưởng quá mức đến độ lớn của thống kê. Do đó, các nhà khoa học dữ liệu nên so sánh s với Sai số tuyệt đối trung bình (MAE), được đưa ra bởi

$$\text{Có nghĩa là lỗi tuyệt đối } = \frac{\sum|y-\hat{y}|}{N}$$

MAE lấy khoảng cách giữa giá trị thực tế và giá trị dự đoán của y và tìm giá trị trung bình của các khoảng cách này. Không có bình phương đang diễn ra. Đối với bất kỳ thống kê đánh giá mô hình nào, chúng ta nên làm như sau:

1. Xây dựng mô hình hồi quy sử dụng tập dữ liệu huấn luyện.
2. Tính toán MAE bằng cách chuyển tập dữ liệu thử nghiệm qua mô hình được huấn luyện trên tập dữ liệu huấn luyện.

   Đối với tập dữ liệu huấn luyện, chúng ta có MAE = $53,39.

Số liệu mô hình ước tính

Khi đánh giá các mô hình ước tính, hãy luôn báo cáo cả s và MAE.

Machine Translated by Google


```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Sales per Visit   R-squared:                       0.065
Model:                            OLS   Adj. R-squared:                  0.064
Method:                 Least Squares   F-statistic:                     50.72
Date:                Mon, 13 Aug 2018   Prob (F-statistic):           5.12e-22
Time:                        12:33:41   Log-Likelihood:                -8546.4
No. Observations:                1451   AIC:                         1.710e+04
Df Residuals:                    1448   BIC:                         1.711e+04
Df Model:                           2
Covariance Type:            nonrobust
```

Hình 11.10 Radj^2 từ Python cho mô hình hồi quy cuối cùng.

Cuối cùng, R2 là một thước đo hồi quy nổi tiếng. Nó được hiểu là tỷ lệ của sự thay đổi trong phản ứng được tính bởi các yếu tố dự đoán trong mô hình. mà phạt R2 cho Đối với nhiều mô hình hồi sử dụng Radj^2, quy, các nhà phân tích nên có quá nhiều yếu tố dự đoán không hữu ích trong mô hình. Mô hình hồi quy của chúng tôi có $\text{Radj}^2 = 0\ 064$ ·, như trong Hình 11.10. Nghĩa là, 6,4% sự thay đổi trong Doanh số bán hàng trên mỗi lượt truy cập được tính bởi các yếu tố dự đoán Số ngày kể từ khi Mua hàng và Thẻ tín dụng. Tỷ lệ nhỏ này không có gì đáng ngạc nhiên, vì có nhiều yếu tố khác ảnh hưởng đến số tiền khách hàng chi tiêu.

11.6.1 Cách thực hiện đánh giá mô hình ước lượng bằng Python

Để sử dụng mô hình hồi quy nhằm dự đoán doanh số bán hàng của khách hàng trên mỗi lượt truy cập, trước tiên chúng ta cần chỉ định các giá trị biến cho khách hàng đầu tiên cho mô hình hồi quy Python. Vì các biến trong mô hình theo thứ tự Hằng số, CC, Ngày, đây là thứ tự mà chúng tôi chỉ định các giá trị.

```
cust01 = np.column_stack((1, 0, 333))
```

Đầu vào đầu tiên trong lệnh column_stack() là 1, cho hằng số trong mô hình.

Khi bạn đã xây dựng được khách hàng đang được đề cập, hãy chạy dự đoán() lệnh trên cust01. Vì chúng tôi đang dự đoán doanh số bán hàng bằng cách sử dụng các kết quả được lưu trữ trong model02, nên chúng tôi sử dụng model02.predict().

```
model02.predict(cust01)
```

Để có được các giá trị dự đoán của tất cả khách hàng trong tập dữ liệu thử nghiệm, hãy thay đổi đầu vào của lệnh predict() thành khung dữ liệu biến dự báo dữ liệu thử nghiệm, X_test.

```
ypred = model02.predict(X_test)
```

Kết quả là một cột dự đoán, một cột cho mỗi bản ghi trong tập dữ liệu thử nghiệm. Các giá trị này sẽ cho phép chúng tôi tính toán MAE sau trong phần này.

Machine Translated by Google


```
In [191]: met.mean_absolute_error(y_true = ytrue, y_pred = ypred)
Out[191]: 53.38639553029432
```

Hình 11.11 MAE từ Python.

Python không tự động cung cấp lỗi tiêu chuẩn của ước tính. Tuy nhiên, nó có thể được tính bằng căn bậc hai của tham số tỷ lệ của mô hình.

```
np.sqrt(model02.scale)
```

Để tính MAE, chúng ta cần cả giá trị dự đoán và giá trị thực tế của y. Các giá trị thực tế là các giá trị của biến mục tiêu, được đổi tên bên dưới thành ytrue cho rõ ràng. Các giá trị ypred đến từ đoạn mã trên.

```
ytrue = sales_train[['Doanh số mỗi lượt truy cập']]
met.mean_absolute_error(y_true = ytrue, y_pred = ypred)
```

Đầu ra cuối cùng từ mã là khoảng 53,39, giá trị của MAE. Mã, với kết quả này, được hiển thị trong Hình 11.11.

Để có được giá trị của R^2_{adj} ,kiểm tra đầu ra từ lệnh tóm tắt() minh họa trong phần Python trước đó và được hiển thị trong Hình 11.10.

11.6.2 Cách thực hiện đánh giá mô hình ước tính bằng R

Để sử dụng mô hình của chúng tôi nhằm dự đoán doanh số trên mỗi lượt truy cập của một khách hàng cụ thể, chúng tôi xây dựng khung dữ liệu chứa thông tin của khách hàng đó.

```
cust01 <- data.frame(CC = as.factor(0), Days = 333)
```

Lệnh data.frame() sẽ tạo khung dữ liệu bằng cách sử dụng nội dung đầu vào. Tên của các biến phải khớp chính xác với tên của các biến dự đoán trong mô hình. Vì biến thẻ tín dụng là các yếu tố khi chúng tôi xây dựng mô hình, hãy đảm bảo rằng chúng là các yếu tố khi tạo khách hàng mới này. Lưu dữ liệu khách hàng mới này với tên cust01. Lưu ý rằng chúng tôi không bao gồm biến mục tiêu.

```
dự đoán (đối tượng = model02, newdata = cust01)
```

Khi chúng tôi chạy lệnh predict() bằng cách sử dụng object = model02 và newdata = cust01, đầu ra là số lượng bán hàng được dự đoán trên mỗi lượt truy cập.

Lỗi tiêu chuẩn của ước tính được đưa ra như một phần của đầu ra được tạo bởi lệnh tóm tắt() . Hình 11.12 hiển thị một đoạn trích của đầu ra được tạo bởi bản tóm tắt (model02). Thống kê quan trọng s được gọi là “Sai số dư chuẩn” trong đầu ra này, và được báo cáo cho mô hình này là 87,54 trong Hình 11.12, cùng với giá trị của R^2_{adj} , được gọi là "R-bình phương đã điều chỉnh."

Để tính toán MAE, chúng tôi cần các giá trị thực tế và dự đoán cho tất cả các bản ghi trong tập dữ liệu thử nghiệm bằng mô hình dữ liệu huấn luyện.

```
X_test <- data.frame(Days = sales_test$Days, CC = sales_
kiểm tra $CC)
ypred <-predict(object = model02, newdata = X_test)
```

Machine Translated by Google


```
Residual standard error: 87.54 on 1448 degrees of freedom
Multiple R-squared:  0.06547,   Adjusted R-squared:  0.06418
F-statistic: 50.72 on 2 and 1448 DF,  p-value: < 2.2e-16
```

Hình 11.12 Sai số chuẩn và R-bình phương được điều chỉnh từ R.

```
> MAE(y_pred = ypred, y_true = ytrue)
[1] 53.3864
```

Hình 11.13 MAE từ R.

```
ytrue <- sales_test$Sales.per.Visit
```

Chúng tôi cũng cần cài đặt và mở gói MLmetrics .

```
install.packages("MLmetrics"); thư viện (MLmetrics)
```

Sau khi gói được mở, bạn có thể tính toán MAE.

```
MAE(y_pred = ypred, y_true = ytrue)
```

Hai đầu vào của lệnh MAE() là y_pred và y_true. Đặt y_pred = ypred, các giá trị bạn thu được từ mô hình hồi quy; và đặt y_true = ytrue, biến mục tiêu từ tập dữ liệu huấn luyện. Kết quả của việc chạy lệnh này là MAE, mã và đầu ra được thể hiện trong Hình 11.13.

11.7 HỒI QUY TỪNG BƯỚC

Trong ví dụ nhỏ này, chúng tôi chỉ có ba yếu tố dự đoán. Tuy nhiên, hầu hết các dự án khoa học dữ liệu sử dụng hàng chục nếu không muốn nói là hàng trăm yếu tố dự đoán. Do đó chúng ta cần một phương pháp để dễ dàng lựa chọn mô hình hồi quy tốt nhất. Phương pháp này được gọi là hồi quy từng bước. Trong hồi quy từng bước, các yếu tố dự đoán hữu ích được nhập vào mô hình tại một thời điểm, bắt đầu với yếu tố dự đoán hữu ích nhất. Do tuyến tính nhiều màu hoặc các hiệu ứng khác, khi một số biến hữu ích được nhập vào, một trong số chúng có thể không còn được coi là hữu ích nữa và nên bị loại bỏ.
Vì lý do này, hồi quy từng bước thêm từng yếu tố dự đoán hữu ích nhất vào mô hình và sau đó kiểm tra xem liệu tất cả chúng có còn phù hợp hay không. Cuối cùng, thuật toán từng bước không thể tìm thấy các yếu tố dự đoán hữu ích nào nữa và hội tụ thành một mô hình cuối cùng.

Việc áp dụng hồi quy từng bước (không hiển thị) đối với clothing_sales_ tập dữ liệu training và clothing_sales_test hội tụ trên các mô hình cuối cùng được hiển thị trong Hình 11.3 và 11.4. Điều quan trọng là phải hiểu rằng hồi quy từng bước không đảm bảo sẽ khám phá ra mô hình tối ưu vì thuật toán tìm kiếm của nó không thực hiện tất cả các hồi quy có thể. Để đảm bảo mô hình tối ưu, bạn có thể sử dụng hồi quy tập hợp con tốt nhất, [2] mặc dù phần mềm có thể giới hạn số lượng bộ dự đoán đối với thuật toán tập hợp con tốt nhất.

2Xem Khai thác dữ liệu và phân tích dự đoán.

Machine Translated by Google


```
Start:  AIC=12982.67
Sales.per.Visit ~ Days + Web + CC

       Df Sum of Sq      RSS   AIC
- Web   1      2986 11096733 12981
<none>              11093747 12983
- CC    1    160657 11254404 13002
- Days  1    736574 11830321 13074

Step:  AIC=12981.06
Sales.per.Visit ~ Days + CC

       Df Sum of Sq      RSS   AIC
<none>              11096733 12981
- CC    1    167291 11264025 13001
- Days  1    733593 11830326 13072
```

Hình 11.14 Đầu ra từ hồi quy từng bước trong R.

11.7.1 Cách thực hiện hồi quy từng bước bằng R

Để chạy hồi quy từng bước, trước tiên bạn cần cài đặt và mở gói MASS .

```
install.packages("MASS"); thư viện(MASS)
```

Chạy mô hình hồi quy, bao gồm tất cả các biến đang được xem xét. Lưu mô hình dưới một tên. Đối với ví dụ này, chúng tôi sẽ sử dụng model01, mà chúng tôi biết là có một biến, Web, không thuộc về mô hình.

Khi bạn đã lưu mô hình, đã đến lúc hồi quy từng bước.

```
model01_step <- stepAIC(object = model01)
```

Lệnh stepAIC() sẽ chạy hồi quy từng bước trên đối tượng được chỉ định. Ví dụ của chúng tôi, chúng tôi muốn chạy hồi quy từng bước trên model01. Chúng tôi lưu kết quả dưới tên model01_step.

Ngay cả việc lưu đầu ra stepAIC() dưới tên cũng sẽ hiển thị một số đầu ra, được đưa ra trong Hình 11.14. Đầu ra cho thấy các bước được thực hiện để hội tụ trên một mô hình. Di chuyển từ nửa trên xuống nửa dưới của đầu ra cho thấy thuật toán từng bước đã thực hiện một bước. Cụ thể, nó đã loại bỏ biến Web.

Nếu bạn tự chạy tên model01_step , nó sẽ cung cấp cho bạn các hệ số hồi quy của mô hình cuối cùng. Nếu bạn chạy tóm tắt(model01_step), nó sẽ đưa ra tóm tắt đầy đủ của mô hình cuối cùng, sẽ khớp với đầu ra được cung cấp bởi tóm tắt(model02), vì mô hình cuối cùng được hội tụ theo từng bước là mô hình hồi quy mà chúng tôi đã lưu dưới dạng model02.

11.8 CÁC MÔ HÌNH CƠ BẢN CHO HỒI QUY

Mô hình cơ sở thông thường để so sánh mô hình hồi quy của bạn với mô hình yy đơn giản . Nếu bất kỳ yếu tố dự đoán nào hữu ích trong việc ước tính phản hồi, thì mô hình đó sẽ đánh bại mô hình yy . Tuy nhiên, chúng ta vẫn nên chính thức

Machine Translated by Google

xác minh rằng mô hình hồi quy của chúng tôi (hoặc bất kỳ mô hình ước tính nào) đang hoạt động tốt hơn mô hình năm , như sau:[3]

So sánh mô hình cơ sở cho các mô hình ước tính

1. Tính toán sai số của mô hình cơ sở. Chúng có dạng Lỗi y y.
2. Tính toán MAE cho mô hình cơ sở, như sau:

$$\text{MAE}_{\text{đường cơ sở}} = \frac{\sum |y - \bar{y}|}{N}$$

3. So sánh MAEBaseline với MAE cho mô hình ước lượng.

$$\text{MAE}_{\text{hồi quy}} = \frac{\sum |y - \hat{y}|}{N}$$

4. Mô hình ước tính tốt hơn mô hình cơ sở khi

MAERegression < MAEBaseline

Chúng tôi áp dụng So sánh mô hình cơ sở cho mô hình hồi quy cuối cùng của chúng tôi (Hình 11.3) như sau:

1. Chúng tôi tính toán các lỗi cho mô hình cơ sở bằng cách sử dụng y $ 112 57 được cung cấp bởi tập dữ liệu thử nghiệm.
2. Chúng tôi tính MAEBaseline = $55,53.
3. MAERegression mà chúng tôi thu được bằng cách chuyển tập dữ liệu thử nghiệm qua mô hình được phát triển bởi tập dữ liệu huấn luyện là $53,39.
4. Vì $53,39 = MAERegression<MAEBaseline = $55,53 nên mô hình hồi quy của chúng tôi đã đánh bại mô hình cơ sở.

NGƯỜI GIỚI THIỆU

Gói MASS chia sẻ cùng một ấn phẩm cốt lõi mà chúng ta đã thấy đối với gói nnet : WN Venables và BD Ripley, Modern Applied Statistics with S, Fourth Edition, Springer, New York, 2002.

Gói MLmetrics , cho phép chúng tôi có được MAE trong R, có thể được khám phá thêm tại đây: Yachen Yan, MLmetrics: Số liệu đánh giá học máy. Phiên bản gói R 1.1.1. 2016. https://CRAN.R-project.org/package=MLmetrics

[3] Một số nhà khoa học dữ liệu có thể thích so sánh MSERegression với MSEBaseline.

Machine Translated by Google

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Làm thế nào để hồi quy bội xấp xỉ mối quan hệ giữa một tập hợp hai yếu tố dự đoán và một mục tiêu số duy nhất?
2. Giải thích cách chúng ta bỏ qua phương pháp suy luận thống kê cổ điển để hồi quy.
3. Giải thích tại sao không cần chuẩn hóa các yếu tố dự báo khi chỉ có một dự đoán liên tục và những cái khác là cờ?
4. Đúng hay sai: Việc chúng tôi sử dụng giá trị p làm hướng dẫn để xác định việc đưa vào mô hình có nghĩa là chúng tôi đang sử dụng suy luận thống kê. Nếu sai, giải thích tại sao không.
5. Đối với kết quả của tập huấn luyện trong Hình 11.3, giả sử cả hai khách hàng đều có thẻ tín dụng của cửa hàng, nhưng Khách hàng A có 100 ngày giữa các lần mua hàng nhiều hơn so với Khách hàng B. Hãy mô tả sự khác biệt về doanh thu ước tính trên mỗi lần ghé thăm của hai khách hàng.
6. Đối với kết quả tập huấn luyện trong Hình 11.3, giả sử hai khách hàng đều có cùng ngày giữa các lần mua hàng, nhưng Khách hàng C có thẻ tín dụng của cửa hàng còn Khách hàng D thì không. Mô tả sự khác biệt về doanh thu ước tính trên mỗi lượt truy cập của hai khách hàng.
7. Tính toán sai số dự đoán cho Khách hàng 2 trong tập huấn luyện.
8. Tính s cho tập dữ liệu thử nghiệm.
9. Tính toán MAE cho tập dữ liệu thử nghiệm.
10. Đúng hay sai: Hồi quy từng bước luôn tìm ra bộ dự đoán tối ưu.

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với thử nghiệm clothing_sales_training và clothing_sales_test bộ dữ liệu. Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Sử dụng tập huấn luyện để chạy mô hình hồi quy nhằm dự đoán Doanh số mỗi lượt truy cập sử dụng Số ngày giữa các lần mua hàng, Thẻ tín dụng và Tài khoản web. Xác định biến dự đoán nào không nên có trong mô hình.
12. Xác thực mô hình từ bài tập trước, bằng cách chạy hồi quy bằng phép thử bộ dữ liệu.
13. Giả sử ai đó nói: “Không có bằng chứng nào cho thấy mối quan hệ giữa Doanh số trên mỗi lượt truy cập và liệu khách hàng có thẻ tín dụng của cửa hàng hay không.” Làm thế nào bạn sẽ trả lời?
14. Giả sử ai đó nói: “Không có bằng chứng nào cho thấy mối quan hệ giữa Doanh số trên mỗi lượt truy cập và liệu khách hàng có tài khoản web của cửa hàng hay không.” Làm thế nào bạn sẽ trả lời?
15. Chạy mô hình hồi quy để dự đoán Doanh số trên mỗi lượt truy cập, chỉ sử dụng các biến được tìm thấy là đáng kể trong mô hình hồi quy trước đó.
16. Xác nhận mô hình từ bài tập trước.
17. Sử dụng phương trình hồi quy để hoàn thành câu sau: “Doanh thu ước tính trên mỗi lượt truy cập bằng..”

Machine Translated by Google

18. Tính và diễn giải sai số chuẩn của ước lượng cho hồi quy.

19. Tìm và giải thích R^2_{adj}.

20. Tính toán và diễn giải MAE cho mô hình hồi quy. So sánh nó với tiêu chuẩn lỗi.

21. Thực hiện hồi quy từng bước trên mô hình trong Bài tập 11. Xác nhận rằng nó hội tụ với mô hình trong Bài tập 13.

22. Tính MAEBaseline.

23. Tính toán MAERegression.

24. Xác định xem mô hình hồi quy có vượt trội so với mô hình cơ sở của nó hay không.

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với tập dữ liệu dành cho người lớn . Sử dụng Python hoặc R để giải quyết từng vấn đề.

25. Phân vùng tập dữ liệu thành tập huấn luyện và tập kiểm tra, mỗi tập chứa khoảng một nửa Hồ sơ.

26. Chạy mô hình hồi quy để dự đoán Số giờ mỗi tuần bằng cách sử dụng Độ tuổi và Giáo dục Num. Có được một bản tóm tắt của mô hình. Có bất kỳ biến dự đoán nào không nên có trong mô hình không?

27. Xác nhận mô hình từ bài tập trước.

28. Sử dụng phương trình hồi quy để hoàn thành câu sau: “Số giờ ước tính mỗi tuần bằng..”

29. Diễn giải hệ số Tuổi.

30. Giải thích hệ số cho Giáo dục Num.

31. Tìm và giải thích giá trị của s.

32. Tìm và giải thích R^2_{adj}.

33. Tìm MAEBaseline và MAERegression, đồng thời xác định xem mô hình hồi quy có hoạt động tốt hơn mô hình cơ sở của nó hay không.

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu bank_reg_training và bank_reg_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

34. Sử dụng tập huấn luyện để chạy hồi quy dự đoán Điểm tín dụng, dựa trên Tỷ lệ Nợ trên Thu nhập và Số tiền Yêu cầu. Có được một bản tóm tắt của mô hình. Cả hai yếu tố dự đoán có thuộc về mô hình không?

35. Xác nhận mô hình từ bài tập trước.

36. Sử dụng phương trình hồi quy để hoàn thành câu sau: “Điểm tín dụng ước tính bằng..”

37. Giải thích hệ số cho Tỷ lệ nợ trên thu nhập.

38. Giải thích hệ số cho Số tiền yêu cầu.

Machine Translated by Google

39. Tìm và giải thích giá trị của s.

40. Tìm và giải thích R^2_{adj}. Bình luận.

41. Tìm MAEBaseline và MAERegression và xác định xem mô hình hồi quy có hoạt động tốt hơn mô hình cơ sở của nó hay không.

42. Xây dựng mô hình hồi quy để dự đoán Lãi suất, sử dụng Số tiền yêu cầu. Có được một bản tóm tắt của mô hình.

43. Giải thích điều gì bất thường với kết quả của bạn từ bài tập trước.

44. Xây dựng biểu đồ phân tán Lãi suất theo Số tiền yêu cầu. Mô tả mối quan hệ giữa các biến. Giải thích cách mối quan hệ này giải thích các kết quả bất thường từ mô hình hồi quy của bạn.

Đối với các bài tập sau, hãy làm việc với Framingham_training và Framingham_test bộ dữ liệu. Thể hiện lại Giới tính sao cho nó là một biến cờ với 0 cho nam và 1 cho nữ. Sử dụng Python hoặc R để giải quyết từng vấn đề.

45. Sử dụng tập huấn luyện để chạy hồi quy dự đoán Tuổi, dựa trên Giới tính và Trình độ học vấn. Có được một bản tóm tắt của mô hình. Cả hai yếu tố dự đoán có thuộc về mô hình không?

46. Xác nhận mô hình từ bài tập trước.

47. Sử dụng phương trình hồi quy để hoàn thành câu sau: “Tuổi ước tính bằng..”

48. Giải thích hệ số cho Giới tính.

49. Giải thích hệ số cho Giáo dục.

50. Tìm và giải thích giá trị của s.

51. Tìm và giải thích R^2_{adj}.

52. Tìm MAEBaseline và MAERegression và xác định xem mô hình hồi quy có vượt trội so với mô hình cơ sở của nó hay không.

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu white_wine_training và white_wine_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

53. Sử dụng tập huấn luyện để chạy hồi quy dự đoán Chất lượng, dựa trên Rượu và Đường. Có được một bản tóm tắt của mô hình. Cả hai yếu tố dự đoán có thuộc về mô hình không?

54. Xác nhận mô hình từ bài tập trước.

55. Sử dụng phương trình hồi quy để hoàn thành câu này: “Chất lượng ước tính bằng..”

56. Giải thích hệ số cho Rượu.

57. Diễn giải hệ số Đường.

58. Tìm và giải thích giá trị của s.

59. Tìm và giải thích R^2_{adj}.

60. Tìm MAEBaseline và MAERegression, và xác định xem mô hình hồi quy có vượt quá hình thành mô hình cơ sở của nó.

Machine Translated by Google

chương 12

GIẢM KÍCH THƯỚC

12.1 NHU CẦU GIẢM KÍCH THƯỚC

Tính chiều cao trong khoa học dữ liệu đề cập đến khi có một số lượng lớn các yếu tố dự đoán trong tập dữ liệu. Ví dụ: 100 yếu tố dự đoán mô tả một không gian 100 chiều. Vì vậy, tại sao chúng ta cần giảm kích thước trong khoa học dữ liệu?

1. Đa cộng tuyến. Thông thường, cơ sở dữ liệu lớn có nhiều yếu tố dự đoán. Không chắc rằng tất cả các yếu tố dự đoán này đều không tương quan với nhau. Đa cộng tuyến, xảy ra khi có mối tương quan đáng kể giữa các yếu tố dự đoán, có thể dẫn đến các mô hình hồi quy không ổn định.
2. Đếm kép. Việc đưa vào các yếu tố dự đoán có tương quan cao có xu hướng nhấn mạnh quá mức một khía cạnh cụ thể của mô hình, nghĩa là về cơ bản là nhân đôi tính khía cạnh này. Ví dụ: giả sử chúng ta đang cố gắng ước tính tuổi của những người trẻ tuổi bằng cách sử dụng kiến thức toán học, chiều cao và cân nặng. Vì chiều cao và cân nặng có mối tương quan với nhau nên mô hình này về cơ bản là tính hai lần thành phần thể chất của trẻ so với thành phần trí tuệ.
3. Lời nguyền của thứ nguyên. Khi số chiều tăng lên, thể tích của không gian tiền đoán tăng theo cấp số nhân, tức là nhanh hơn số lượng chính số tiền dự đoán. Do đó, ngay cả đối với kích thước mẫu lớn, không gian chiều cao vẫn thưa thớt. Ví dụ: quy tắc thực nghiệm cho biết khoảng 68% dữ liệu được phân phối bình thường nằm trong một độ lệch chuẩn của giá trị trung bình. Nhưng, đây là cho một chiều. Đối với 10 chiều, chỉ có 2% dữ liệu nằm trong siêu cầu tương tự.
4. Vi phạm Tiết chế. Việc sử dụng quá nhiều yếu tố dự đoán cũng vi phạm nguyên tắc tiết kiệm, nguyên tắc khoa học mà người ta thấy trong nhiều ngành khoa học, rằng mọi thứ thường hành xử theo một cách khá tiết kiệm. Trong khoa học dữ liệu, nên xem xét tính đơn giản (parsimony) khi so sánh các mô hình, giữ cho số lượng yếu tố dự đoán ở mức có thể dễ dàng diễn giải.
5. Trang bị quá nhiều Giữ quá nhiều yếu tố dự đoán trong mô hình có xu hướng dẫn đến quá phù hợp, trong đó tính tổng quát của các phát hiện bị cản trở do dữ liệu mới không hoạt động giống như dữ liệu huấn luyện cho tất cả nhiều yếu tố dự đoán.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

6. Bỏ lỡ bức tranh lớn hơn. Hơn nữa, các nhà phân tích nên cố gắng theo dõi bức tranh toàn cảnh và chỉ phân tích ở cấp độ thay đổi có thể bỏ sót các nguyên tắc cơ bản làm cơ sở cho mối quan hệ giữa các yếu tố dự đoán. Thay vào đó, một số yếu tố dự đoán có thể tự nhiên rơi vào một nhóm duy nhất (một yếu tố hoặc thành phần), nhóm này giải quyết một khía cạnh duy nhất của dữ liệu. Ví dụ: các biến số dư tài khoản tiết kiệm, số dư tài khoản séc, vốn chủ sở hữu nhà, giá trị danh mục đầu tư chứng khoán và số dư 401k đều có thể cùng nhau nằm trong một thành phần duy nhất, tài sản.

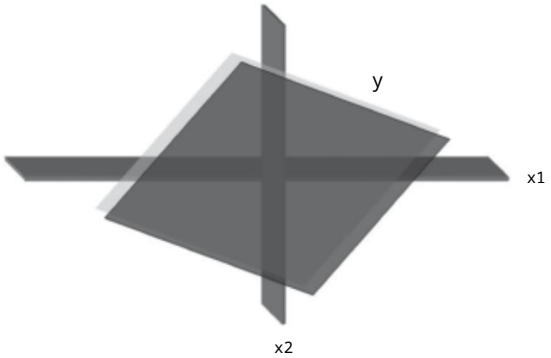
Tóm lại, các phương pháp giảm kích thước sử dụng cấu trúc tương quan giữa các biến dự đoán để thực hiện những điều sau:

1. Giảm số lượng các mục dự đoán.
2. Giúp đảm bảo rằng các mục dự đoán này không tương quan với nhau.
3. Cung cấp một khuôn khổ để diễn giải các kết quả.

12.2 ĐA CỘNG TUYẾN

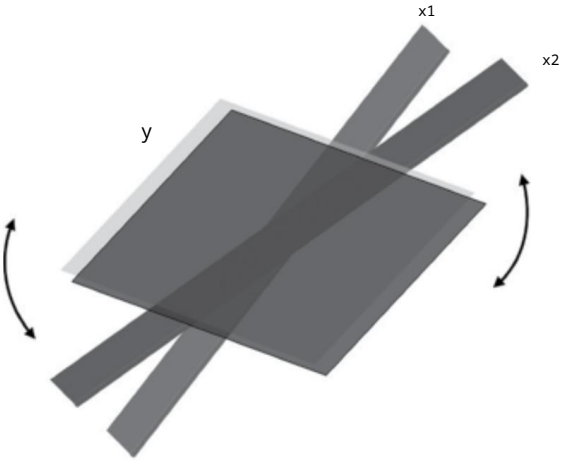
Các nhà khoa học dữ liệu cần đề phòng đa cộng tuyến, một tình trạng trong đó một số biến dự đoán có tương quan với nhau. Đa cộng tuyến dẫn đến sự không ổn định trong không gian giải pháp, ví dụ, dẫn đến các hệ số hồi quy mà bạn không thể tin tưởng, vì độ biến thiên của hệ số quá lớn. Đa cộng tuyến là một rủi ro nghề nghiệp đối với các nhà khoa học dữ liệu, bởi vì nhiều bộ dữ liệu có hàng chục, nếu không muốn nói là hàng trăm yếu tố dự đoán, một số trong số đó thường tương quan với nhau.

Xét Hình 12.1 và 12.2. Hình 12.1 minh họa một tình huống trong đó các yếu tố dự đoán x1 và x2 không tương quan với nhau; nghĩa là chúng trực giao hoặc độc lập. Trong trường hợp như vậy, các yếu tố dự đoán tạo thành một cơ sở vững chắc, trên đó bề mặt phản hồi y có thể nằm chắc chắn, do đó cung cấp các ước tính hệ số ổn định b1 và b2 mỗi ước tính với độ biến thiên nhỏ. Mặt khác, Hình 12.2 minh họa một tình huống đa cộng tuyến trong đó các yếu tố dự đoán x1 và x2 tương quan với nhau



Hình 12.1 Khi các yếu tố dự đoán x1 và x2 không tương quan với nhau, bề mặt phản hồi y dựa trên cơ sở vững chắc, cung cấp các ước tính hệ số ổn định.

Machine Translated by Google



Hình 12.2 Đa cộng tuyến: khi các yếu tố dự đoán tương quan với nhau, bề mặt phản hồi không ổn định, dẫn đến các ước tính hệ số không rõ ràng và có tính thay đổi cao.

khác, sao cho khi một trong số chúng tăng lên, thì cái kia cũng vậy. Trong trường hợp này, các yếu tố dự đoán không còn tạo thành cơ sở vững chắc mà bề mặt phản ứng có thể dựa vào đó một cách chắc chắn. Thay vào đó, khi các yếu tố dự đoán tương quan với nhau, bề mặt phản hồi không ổn định, cung cấp các ước tính hệ số biến đổi cao b1 và b2 .

Độ biến thiên cao liên quan đến các ước tính có nghĩa là các mẫu khác nhau có thể tạo ra các ước tính hệ số với các giá trị rất khác nhau. Ví dụ: một mẫu có thể tạo ước tính hệ số dương cho x1 , trong khi mẫu thứ hai có thể tạo ước tính hệ số âm. Tình huống này là không thể chấp nhận được khi nhiệm vụ phân tích yêu cầu giải thích về mối quan hệ giữa phản hồi và các yếu tố dự đoán, riêng lẻ.

Chúng ta hãy xem một ví dụ về đồ chơi để giúp chúng ta hiểu vấn đề. Hãy xem xét dân số cực nhỏ trong Bảng 12.1.

Rõ ràng, x1 và x2 có tương quan với nhau, với r = 0,938, giá trị p gần bằng 0.

Bây giờ, hãy chia thành hai mẫu, như trong Bảng 12.2 và 12.3.

Phương trình hồi quy cho mẫu 1 là:

$$\hat{y} = -0.542 + 2.552x_1 - 0.206x_2$$

Phương trình hồi quy cho mẫu 2 là:

$$\hat{y} = -10.8 + 0.547x_1 + 17.92x_2$$

Lưu ý rằng hệ số của x2 là âm trong mẫu 1 và dương trong mẫu 2.

Điều này thể hiện hành vi không ổn định trong các hệ số hồi quy, gây ra bởi mối tương quan giữa các yếu tố dự đoán. Về cơ bản, chúng ta không thể tin tưởng vào các giá trị, hoặc thậm chí là các dấu hiệu của các hệ số hồi quy. Vì vậy, giải thích các hệ số hồi quy cho khách hàng của chúng tôi

Machine Translated by Google

BẢNG 12.1 Quần thể tí hon cho ví dụ về đồ chơi của chúng ta

Dân số		
x1	x2	Mục tiêu, y
1	1	2.0693
1	2	2.6392
2	2	3.7501
2	3	5.6432
3	3	5,8925
3	4	6.4308
4	4	8.3950
4	5	8.4947
5	5	11.3236
5	5	10.1562

BẢNG 12.2 Mẫu 1 từ dân số nhỏ bé của chúng ta

mẫu 1		
x1	x2	Mục tiêu, y
1	1	2.0693
2	2	3.7501
3	4	6.4308
4	5	8.4947
5	5	11.3236

BẢNG 12.3 Mẫu 2 từ dân số nhỏ bé của chúng ta

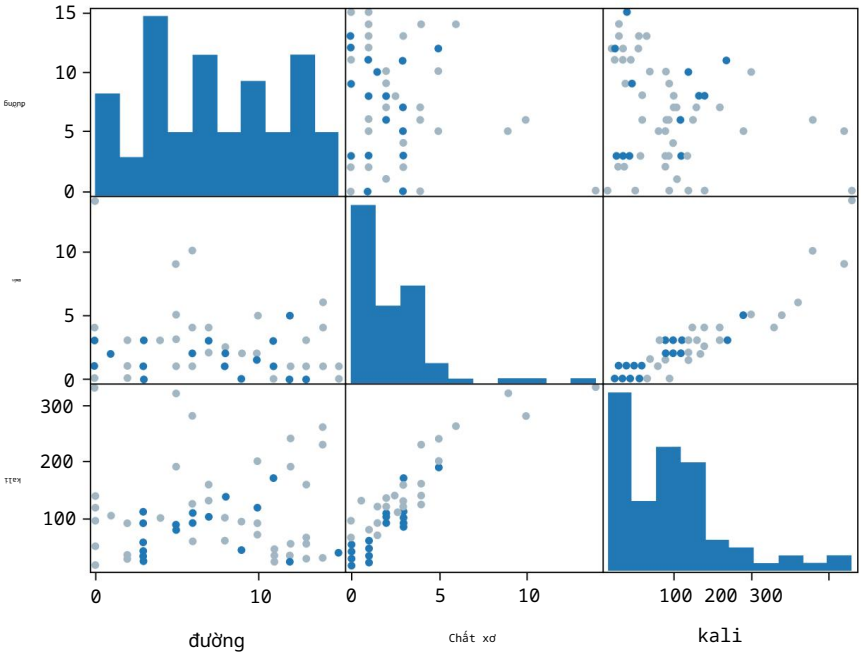
mẫu 2		
x1	x2	Mục tiêu, y
1	2	2.6392
2	3	5.6432
3	3	5,8925
4	4	8.3950
5	5	10.1562

có lẽ không phải là một ý tưởng tốt. Các giá trị và dấu hiệu có thể thay đổi từ mẫu này sang mẫu khác, do đó, việc diễn giải các hệ số chỉ từ một mẫu có thể khiến khách hàng của bạn phải trả nhiều tiền. Đây là lý do tại sao chúng ta cần các phương pháp để xử lý hiện tượng đa cộng tuyến này.

Chúng tôi sẽ sử dụng một tập hợp con của bộ dữ liệu Ngũ cốc , trong đó chúng tôi ước tính xếp hạng dinh dưỡng của ngũ cốc ăn sáng, sử dụng chất xơ, kali và đường. Phương trình hồi quy của chúng tôi như sau:

$$\text{đánh giá} = b_0 + b_1 \cdot \text{chất xơ} + b_2 \cdot \text{kali} + b_3 \cdot \text{đường}$$

Machine Translated by Google



Hình 12.3 Biểu đồ ma trận của các yếu tố dự đoán từ Python. Chất xơ và kali có mối tương quan với nhau.

Để nhận thức được khả năng đa cộng tuyến có thể xảy ra, nhà phân tích nên điều tra cấu trúc tương quan giữa các biến dự đoán. Hình 12.3 cung cấp sơ đồ ma trận của các yếu tố dự báo. Rõ ràng, kali và chất xơ có mối tương quan tích cực. Trên thực tế, hệ số tương quan của chúng là r = 0,912 (không hiển thị). Mối tương quan chặt chẽ này sẽ làm tăng độ biến thiên của các hệ số hồi quy, làm cho mô hình hồi quy của chúng ta không ổn định.

12.3 XÁC ĐỊNH ĐA CỘNG TUYẾN SỬ DỤNG CÁC YẾU TỐ LẠM PHÁT PHƯƠNG BIẾN

Tuy nhiên, giả sử chúng ta không kiểm tra sự hiện diện của mối tương quan giữa các yếu tố dự đoán của mình và vẫn tiếp tục thực hiện hồi quy. Có cách nào để kết quả hồi quy có thể cảnh báo chúng ta về sự hiện diện của đa cộng tuyến không?
Câu trả lời là có: Chúng tôi có thể yêu cầu báo cáo các yếu tố lạm phát phương sai (VIF).

VIF cho yếu tố dự báo thứ i được cho bởi:

$$\text{VIF}_i = \frac{1}{1 - r_i^2}$$

Machine Translated by Google


```
> vif(model03)
   Fiber    Potass    Sugars
6.850050 6.693982 1.158761
```

Hình 12.4 Kết quả hồi quy, với hệ số lạm phát phương sai từ R cho thấy vấn đề đa cộng tuyến.

trong đó ^{2} đại diện cho giá trị R2 thu được bằng cách hồi quy xi trên bộ dự đoán khác các biến Ri . Lưu ý rằng ^{2} Ri sẽ lớn khi xi có tương quan cao với các tiền tố khác, do đó làm cho VIFi lớn.

Một nguyên tắc chung để giải thích giá trị của VIF là coi VIFi ≥ 5 là một chỉ báo về đa cộng tuyến vừa phải và coi VIFi ≥ 10 là một chỉ báo về đa cộng tuyến nghiêm trọng. VIF năm tương ứng với Ri = 0,80, trong khi VIFi = 10 tương ứng với Ri ^{2} = 0,80. Đối với hồi quy đánh giá dinh

dưỡng về chất xơ, kali và đường, chúng ta có kết quả được cung cấp trong Hình 12.4. VIF đối với chất xơ là 6,85 và VIF đối với kali là 6,69, với cả hai giá trị đều biểu thị tính đa cộng tuyến từ trung bình đến mạnh.

12.3.1 Cách xác định đa cộng tuyến bằng Python

Đầu tiên, chúng tôi tải các gói cần thiết và đọc trong bộ dữ liệu ngũ cốc dưới tên ngũ cốc.

```
nhập gấu trúc dưới dạng pd
nhập statsmodels.api dưới dạng sm nhập
statsmodels.stats.outliers_influence dưới dạng inf ngũ cốc =
pd.read_csv("C:/.../cereals.csv")
```

Sau khi tập dữ liệu ở dạng Python, hãy rút ra ba biến dự đoán và đặt chúng vào khung dữ liệu của riêng chúng. Gọi khung dữ liệu X.

```
X = pd.DataFrame(ngũ cốc[['Đường', 'Chất xơ', 'Kali']])
```

Bây giờ chúng ta có tất cả các biến dự đoán cùng nhau, hãy sử dụng phân tán_ matrix() với X làm đầu vào để tạo ma trận biểu đồ phân tán.

```
pd.plotting.scatter_matrix(X)
```

Kết quả của lệnh scatter_matrix() được hiển thị trong Hình 12.3 ở trên. Lưu ý rằng lệnh tạo cả biểu đồ phân tán và biểu đồ.

Để có được các giá trị VIF, trước tiên chúng ta cần làm sạch một chút dữ liệu. Sử dụng lệnh dropna() trên khung dữ liệu X để xóa mọi bản ghi có giá trị bị thiếu.

```
X = X.dropna()
```

Sau đó, đảm bảo bạn thêm thuật ngữ không đổi vào khung dữ liệu X.

```
X = sm.add_constant(X)
```

Machine Translated by Google

Cuối cùng, hãy chạy lệnh Variance_inflation_factor() như được đưa ra bên dưới để lấy các giá trị VIF cho cả bốn cột trong khung dữ liệu X.

```
[inf.variance_inflation_factor(X.values, i) cho i trong phạm vi(X.shape[1])]
```

Đầu ra sẽ bao gồm một giá trị VIF cho thuật ngữ không đổi mà chúng tôi đã thêm vào; Phớt lờ nó đi. Các giá trị quan tâm của VIF dành cho ba biến dự đoán, là các số thứ hai, thứ ba và thứ tư được xuất ra bởi phương sai_inflation_factor()
yêu cầu.

12.3.2 Cách xác định đa cộng tuyến trong R

Để xây dựng ma trận biểu đồ phân tán trong R, trước tiên chúng ta cần xác định cột nào trong tập dữ liệu chứa các biến dự đoán của chúng ta.

```
tên (ngũ cốc)
```

Bạn có thể thấy rằng các biến Đường, Kali và Chất xơ lần lượt nằm trong các cột 10, 8 và 11. Đây là những cột bạn sẽ đặt trong ma trận biểu đồ phân tán trong Hình 12.5.

```
cặp(x = ngũ cốc[, c(10, 8, 11)], pch = 16)
```

Lệnh cho ma trận biểu đồ phân tán là cặp(). Đầu vào bắt buộc là x, yêu cầu đưa các cột vào ma trận biểu đồ phân tán. Chúng tôi chỉ định các cột sử dụng ngũ cốc[, c(10, 8, 11)]. Một đầu vào tùy chọn pch = 16 sẽ thay đổi các điểm biểu đồ phân tán từ vòng tròn mở thành vòng tròn khép kín.

Để tính toán VIF, trước tiên chúng ta cần cài đặt và mở gói xe .

```
install.packages("xe"); thư viện (ô tô)
```

Khi gói ô tô được mở, chúng tôi xây dựng mô hình có các hệ số mà chúng tôi muốn kiểm tra tính đa cộng tuyến và lưu đầu ra của mô hình.

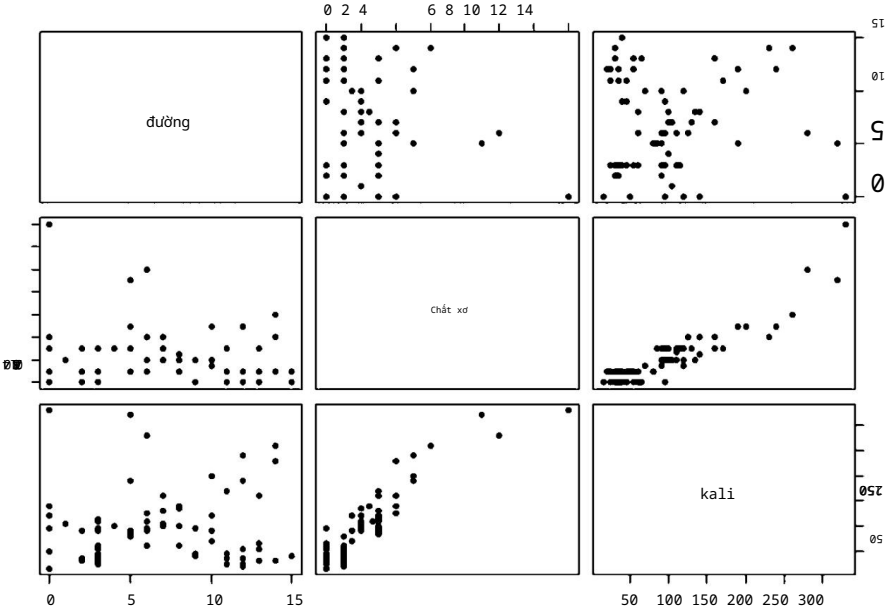
```
model03 <- lm(công thức = Xếp hạng ~ Chất xơ + Kali +
Đường, dữ liệu = ngũ cốc)
```

Cuối cùng, chúng tôi sử dụng lệnh vif() trên mô hình.

```
vif(model03)
```

Đầu vào bắt buộc duy nhất cho lệnh vif() là tên mà chúng tôi đã lưu mô hình của mình BẰNG. Đầu ra, như trong Hình 12.4, là VIF cho mỗi biến dự báo.

Machine Translated by Google



Hình 12.5 Ma trận biểu đồ phân tán của các yếu tố dự báo trong R.

Machine Translated by Google

12.4 PHÂN TÍCH CÁC THÀNH PHẦN CHÍNH

Vì vậy, bây giờ chúng tôi đã xác định được tính đa cộng tuyến giữa các yếu tố dự đoán của chúng tôi, chúng tôi phải làm gì bây giờ?

Một giải pháp là áp dụng phân tích thành phần chính. Phân tích thành phần chính (PCA) tìm cách giải thích cấu trúc tương quan của một tập hợp các biến xác định trước, sử dụng một tập hợp nhỏ hơn các kết hợp tuyến tính không tương quan của các biến này, được gọi là các thành phần . Tổng độ biến thiên được tạo ra bởi tập hợp đầy đủ m yếu tố dự đoán thường có thể được giải thích chủ yếu bằng một tập hợp k<m thành phần nhỏ hơn. Điều này có nghĩa là có gần như nhiều thông tin trong k thành phần như trong m biến ban đầu. Ngoài ra, các thành phần k không tương quan với nhau, không giống như các yếu tố dự đoán tương quan ban đầu. Nếu muốn, nhà phân tích sau đó có thể thay thế m biến ban đầu bằng k<m thành phần, sao cho tập dữ liệu làm việc hiện bao gồm n bản ghi trên k thành phần, thay vì n bản ghi trên m phần tử dự đoán. Đây là giảm kích thước!

Nhà phân tích cần lưu ý rằng PCA chỉ hoạt động dựa trên các biến dự đoán và bỏ qua biến mục tiêu. Ngoài ra, các yếu tố dự đoán phải được tiêu chuẩn hóa hoặc chuẩn hóa. Về mặt toán học, các thành phần chính là các tổ hợp tuyến tính không tương quan Yi của các yếu tố dự đoán, với các đặc điểm sau:

- Thành phần chính đầu tiên thường là quan trọng nhất. Nó chiếm sự thay đổi lớn hơn giữa các yếu tố dự đoán so với bất kỳ thành phần nào khác.
- Thành phần chính thứ hai chiếm tỷ lệ biến động nhiều thứ hai và không tương quan với thành phần thứ nhất.
- Thành phần chính thứ ba chiếm tỷ lệ thay đổi nhiều thứ ba và là không tương quan với hai cái đầu tiên, v.v.

12.5 ỨNG DỤNG PHÂN TÍCH CÁC THÀNH PHẦN CHÍNH

Để minh họa ứng dụng của PCA, chúng ta chuyển sang trang clothing_store_PCA_training và bộ dữ liệu clothing_store_PCA_test . Chúng tôi quan tâm đến việc ước tính doanh số phản hồi trên mỗi lượt truy cập bằng cách sử dụng các công cụ dự đoán Lượt truy cập mua hàng, Số ngày trên hồ sơ, Số ngày giữa các lần mua, Các mặt hàng khác nhau đã mua và Số ngày kể từ khi mua. Tuy nhiên, Hình 12.6 cho thấy có mối tương quan đáng kể giữa các yếu tố dự báo. Ngoài ra, Hình 12.7, hiển thị hồi quy Doanh số bán hàng trên mỗi lượt truy cập so với các công cụ dự đoán, cho biết một số chỉ số VIF bị thổi phồng vừa phải.

Do đó, chúng tôi thực hiện PCA trên các yếu tố dự đoán này, sử dụng phép quay varimax trên tập dữ liệu huấn luyện.

Xoay vòng giải pháp PCA giúp hiểu được các thành phần.
Xem xét các thành phần được xoay trong Hình 12.8, chúng tôi thấy rằng, nếu chúng ta chỉ trích xuất thành phần chính đầu tiên, chúng ta sẽ chỉ chiếm 31,3% phương sai giữa các yếu tố dự báo. Nếu chúng ta trích xuất hai thành phần, chúng ta chiếm khoảng 52,2% (xem Cumulative Var trong Hình 12.8), v.v. Vì vậy, câu hỏi đặt ra, chúng ta nên trích xuất bao nhiêu thành phần?

Machine Translated by Google

	Days.since.Purchase.Z	Purchase.Visits.Z	Days.on.File.Z
Days.since.Purchase.Z	1.000	-0.440	-0.159
Purchase.Visits.Z	-0.440	1.000	0.364
Days.on.File.Z	-0.159	0.364	1.000
Days.between.Purchases.Z	0.573	-0.453	0.203
Diff.Items.Purchased.Z	-0.379	0.821	0.303

	Days.between.Purchases.Z	Diff.Items.Purchased.Z
Days.since.Purchase.Z	0.573	-0.379
Purchase.Visits.Z	-0.453	0.821
Days.on.File.Z	0.203	0.303
Days.between.Purchases.Z	1.000	-0.371
Diff.Items.Purchased.Z	-0.371	1.000

Hình 12.6 Ma trận tương quan từ R cho thấy mối tương quan đáng kể giữa các yếu tố dự báo.

Days.since.Purchase.Z	Purchase.Visits.Z	Days.on.File.Z
1.701947	3.793173	1.536395
Days.between.Purchases.Z	**Diff.Items.Purchased.Z**	
2.145807	3.076706	

Hình 12.7 Hồi quy từ R cho thấy một số xu hướng hướng tới các VIF lớn vừa phải.

Loadings:

	RC1	RC2	RC3	RC4	RC5
Days.since.Purchase.Z			0.935		
Purchase.Visits.Z	0.725				0.573
Days.on.File.Z		0.971			
Days.between.Purchases.Z				0.910	
Diff.Items.Purchased.Z	0.965				

	RC1	RC2	RC3	RC4	RC5
SS loadings	1.566	1.045	1.035	1.006	0.348
Proportion Var	0.313	0.209	0.207	0.201	0.070
Cumulative Var	0.313	0.522	0.729	0.930	1.000

Hình 12.8 Đoạn trích từ kết quả PCA của R.

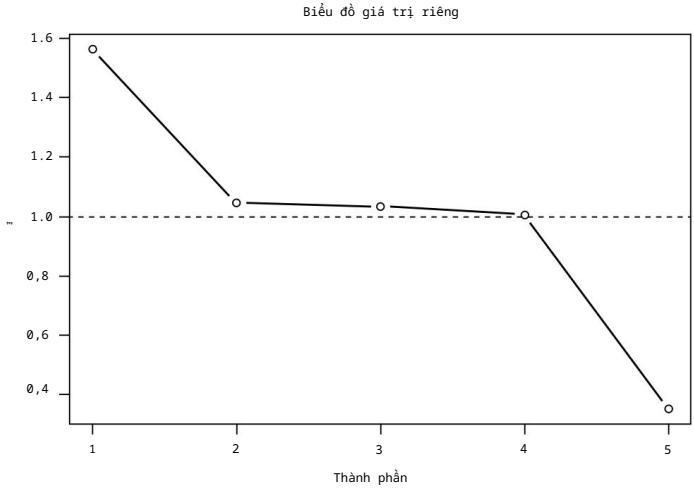
12.6 CHÚNG TÔI NÊN CHIẾT XUẤT BAO NHIÊU THÀNH PHẦN?

Nhớ lại rằng một trong những động cơ thúc đẩy PCA là giảm kích thước. Câu hỏi đặt ra là "Làm cách nào để chúng tôi xác định có bao nhiêu thành phần cần trích xuất?" Ví dụ: chúng ta có nên chỉ giữ lại hai thành phần chính đầu tiên không, vì chúng giải thích hơn một nửa (52% Var tích lũy) của tổng biến thiên? Hay chúng ta nên giữ lại tất cả năm thành phần, vì chúng giải thích 100% sự thay đổi? Chà, rõ ràng là việc giữ lại tất cả năm thành phần không giúp chúng ta giảm số chiều. Như thường lệ, câu trả lời nằm ở đâu đó giữa hai thái cực này.

12.6.1 Tiêu chí Giá trị bản địa

Trong Hình 12.8, các giá trị riêng được dán nhãn là "tải SS". Giá trị riêng là 1,0 có nghĩa là thành phần này sẽ giải thích về "giá trị của một yếu tố dự báo" của tổng biến thiên. Lý do căn bản cho việc sử dụng tiêu chí giá trị riêng là mỗi

Machine Translated by Google



Hình 12.9 Đồ thị của các giá trị riêng từ R, với đường chấm tại giá trị riêng là 1.

thành phần phải giải thích ít nhất một giá trị biến thiên của một yếu tố dự báo, và do đó, tiêu chí giá trị riêng chỉ ra rằng chỉ các thành phần có giá trị riêng lớn hơn một mới được giữ lại. Lưu ý rằng, nếu có ít hơn 20 yếu tố dự đoán, tiêu chí giá trị riêng có xu hướng khuyến nghị trích xuất quá ít thành phần, trong khi, nếu có hơn 50 biến, tiêu chí này có thể khuyến nghị trích xuất quá nhiều.

Từ Hình 12.9, chúng ta thấy rằng bốn trong số các thành phần được xoay có giá trị riêng lớn hơn 1, và do đó được giữ lại. Thành phần 5 có giá trị riêng xoay vòng thấp hơn nhiều so với 1.0, vì vậy chúng tôi không bao gồm Thành phần 5. Do đó, tiêu chí giá trị riêng gợi ý rằng chúng tôi trích xuất k = 4 thành phần chính.

12.6.2 Tiêu chí giải thích tỷ lệ phương sai

Đối với tiêu chí giải thích về tỷ lệ phương sai, trước tiên, khách hàng hoặc nhà phân tích chỉ định tỷ lệ nào trong tổng độ biến thiên mà họ muốn các thành phần chính tính đến. Sau đó, nhà phân tích chỉ cần chọn từng thành phần một cho đến khi đạt được tỷ lệ biến đổi mong muốn được giải thích. Ví dụ: giả sử chúng tôi muốn các thành phần của mình giải thích khoảng 70% sự thay đổi trong các yếu tố dự đoán. Sau đó, từ Hình 12.8, chúng tôi sẽ chọn các thành phần 1–3, cùng nhau giải thích 72,9% độ biến thiên. Mặt khác, nếu chúng ta muốn các thành phần của mình giải thích được 90% tính biến thiên, thì chúng ta cũng cần đưa vào thành phần 4, thành phần này cùng với ba thành phần đầu tiên sẽ giải thích 93% tính biến thiên. Nếu không có thêm thông tin đầu vào từ khách hàng, chúng tôi có thể chỉ nói rằng tỷ lệ phương sai được giải thích theo tiêu chí cho thấy chúng tôi sử dụng các thành phần k = 3 hoặc k = 4.

Vì tiêu chí giá trị riêng đề xuất k = 4 thành phần và tỷ lệ tiêu chí được giải thích phương sai là ổn với k = 3 hoặc k = 4 thành phần, do đó, chúng tôi đồng thuận giải quyết việc trích xuất k = 4 thành phần.

Machine Translated by Google

12.7 THỰC HIỆN PCA VỚI k = 4

Hình 12.10 cho thấy các ma trận thành phần (i) không quay và (ii) xoay kết quả để trích xuất ba thành phần. Đầu tiên chúng ta hãy kiểm tra ma trận được xoay trong Hình 12.10b. Lưu ý rằng các trọng số thành phần nhỏ hơn 0,5 đã bị loại bỏ để tăng cường khả năng diễn giải. Thành phần chính đầu tiên (RC1 cho Thành phần được Xoay vòng 1) là sự kết hợp của Các mặt hàng khác nhau đã mua và Lượt truy cập mua hàng, có mối tương quan thuận với nhau, do trọng số thành phần của chúng có cùng dấu. Các thành phần có thể chứa tổ hợp các yếu tố dự đoán có tương quan tích cực hoặc tiêu cực với nhau. Nếu chính xác một trong các trọng số thành phần là âm, thì đó sẽ là dấu hiệu cho thấy Các mặt hàng khác nhau đã mua và Lượt truy cập mua hàng có mối tương quan nghịch. Các thành phần chính còn lại là "singleton", mỗi thành phần chỉ chứa một yếu tố dự báo duy nhất.

Bây giờ, giả sử chúng ta không xoay ma trận thành phần, kết thúc bằng ma trận thành phần không xoay trong Hình 12.10a. Lưu ý rằng các giải thích của các thành phần chính ít rõ ràng hơn. Thành phần 1 rất lớn, chứa bốn trong số năm yếu tố dự báo, với cả mối tương quan tích cực và tiêu cực xen lẫn vào nhau. Việc giải thích ma trận thành phần được xoay vòng rõ ràng hơn nhiều.

12.8 XÁC NHẬN CÁC THÀNH PHẦN CHÍNH

Như với bất kỳ phương pháp khoa học dữ liệu nào khác, kết quả của PCA phải được xác thực bằng cách sử dụng bộ dữ liệu thử nghiệm. Hình 12.11 cho thấy tỷ lệ phương sai được giải thích bởi cả năm thành phần, với tỷ lệ phần trăm không khác nhiều so với tập huấn luyện

```
Loadings:
                        PC1    PC2    PC3    PC4
Days.since.Purchase    -0.718         0.569
Purchase.Visits         0.898
Days.on.File                   0.825
Days.between.Purchases -0.662  0.647
Diff.Items.Purchased    0.849
```

```
Loadings:
                        RC1    RC2    RC3    RC4
Days.since.Purchase                   0.933
Purchase.Visits         0.862
Days.on.File                   0.965
Days.between.Purchases                        0.898
Diff.Items.Purchased    0.948
```

hình 12.10 (a) trọng lượng thành phần không quay, từ r. (b) trọng số thành phần với phép quay varimax, từ r.

```
                  RC1   RC2   RC3   RC4   RC5
SS loadings     1.805 1.035 1.033 0.993 0.134
Proportion Var  0.361 0.207 0.207 0.199 0.027
Cumulative Var  0.361 0.568 0.775 0.973 1.000
```

Hình 12.11 Tỷ lệ phương sai được giải thích từ R cho tập dữ liệu thử nghiệm.

Machine Translated by Google


```
Loadings:
                        RC1    RC2    RC3    RC4
Days.since.Purchase                   0.932
Purchase.Visits         0.891
Days.on.File                   0.965
Days.between.Purchases                        0.901
Diff.Items.Purchased    0.944
```

Hình 12.12 Các trọng số thành phần từ R cho tập dữ liệu thử nghiệm.

```
> round(cor(pca02_rot$scores),2)
    RC1 RC2 RC3 RC4
RC1   1   0   0   0
RC2   0   1   0   0
RC3   0   0   1   0
RC4   0   0   0   1
```

Hình 12.13 Đầu ra R cho thấy các thành phần chính không tương quan với nhau.

```
> vif(model.pca)
PC1 PC2 PC3 PC4
  1   1   1   1
```

Hình 12.14 Đầu ra R cho thấy hồi quy sử dụng các thành phần chính loại bỏ đa cộng tuyến.

kết quả ở hình 12.8. Bốn thành phần xoay cho tập kiểm tra, được minh họa trong Hình 12.12, tương tự như đối với tập huấn luyện trong Hình 12.10b.

Vì vậy, PCA có làm giảm bớt vấn đề đa cộng tuyến của chúng ta không? Chúng ta có thể kiểm tra bằng cách kiểm tra

1. Mối tương quan giữa bốn thành phần.
2. Biến dự đoán VIF cho hồi quy của phản ứng trên các thành phần.

Ma trận tương quan cho các thành phần chính được thể hiện trong Hình 12.13. Tất cả các tương quan đều bằng 0, nghĩa là các thành phần không tương quan với nhau. Cuối cùng, chúng tôi thu được các VIF cho hồi quy Doanh số bán hàng trên mỗi lượt truy cập trên bốn thành phần chính được trích xuất thay thế cho các yếu tố dự đoán ban đầu. Các kết quả VIF này được hiển thị trong Hình 12.14 chỉ ra rằng tất cả các VIF đều bằng 1, mức tối thiểu.

12.9 CÁCH THỰC HIỆN PHÂN TÍCH CÁC THÀNH PHẦN CHÍNH BẰNG PYTHON

Tải các gói cần thiết.

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
từ sklearn.tiền xử lý nhập StandardScaler
từ sklearn.decomposition nhập PCA
```

Machine Translated by Google

	Days since Purchase	Purchase Visits	Days on File
Days since Purchase	1.000000	-0.439821	-0.158718
Purchase Visits	-0.439821	1.000000	0.363729
Days on File	-0.158718	0.363729	1.000000
Days between Purchases	0.573090	-0.453024	0.202890
Diff Items Purchased	-0.378658	0.821257	0.302624

	Days between Purchases	Diff Items Purchased
Days since Purchase	0.573090	-0.378658
Purchase Visits	-0.453024	0.821257
Days on File	0.202890	0.302624
Days between Purchases	1.000000	-0.371018
Diff Items Purchased	-0.371018	1.000000

Hình 12.15 Ma trận tương quan từ Python.

Đọc trong hai bộ dữ liệu, clothing_store_PCA_training và clothing_store_PCA_test, dưới dạng clothes_train và clothes_test.

```
clothes_train = pd.read_csv("C:/.../clothing_store_PCA_training")
clothes_test = pd.read_csv("C:/.../clothing_store_PCA_test")
```

Tách các biến dự đoán khỏi phần còn lại của tập dữ liệu huấn luyện bằng lệnh drop() . Lưu ý rằng chúng tôi loại bỏ biến mục tiêu Doanh số trên mỗi lượt truy cập, vì vậy chúng tôi chỉ còn lại các biến dự đoán. Cách tiếp cận này phù hợp nhất khi biến mục tiêu và biến dự báo quan tâm là các biến duy nhất trong dữ liệu của bạn.
Lưu các biến dưới dạng X.

```
X = clothes_train.drop('Doanh số mỗi lượt truy cập', 1)
```

Lấy ma trận tương quan của các biến X bằng cách sử dụng lệnh corr() .

```
X.corr()
```

Kết quả từ lệnh corr() được thể hiện trong Hình 12.15.

Bây giờ, chúng tôi sẽ chạy PCA với năm thành phần. Trước tiên, chúng tôi sẽ chỉ định số lượng thành phần bằng cách sử dụng n_components trong lệnh PCA() , sau đó khớp PCA với dữ liệu bằng cách sử dụng fit_transform() với X làm đầu vào.

```
pca01 = PCA(n_components=5)
PrincipComp = pca01.fit_transform(X)
```

Sau khi chạy PCA, nhiệm vụ tiếp theo của chúng ta là xem xét mức độ biến thiên được giải thích bởi từng thành phần và giải thích biến thiên tích lũy tương ứng. Sử dụng đối tượng pca01 và giải thích_variance_ratio_ để có được độ biến thiên được giải thích bởi từng thành phần.

```
pca01.explained_variance_ratio_
```

Machine Translated by Google

Có được biến thiên tích lũy được giải thích bằng cách chạy cumsum() lệnh trên pca01.explained_variance_ratio_.

```
np.cumsum(pca01.explained_variance_ratio_)
```

Kết quả của chúng tôi trong phần này là dành cho các thành phần gốc, chưa được xoay. Khi viết bài này, gói sklearn của Python không có lệnh "ngoài luồng" để thực hiện xoay varimax trên các thành phần.

12.10 CÁCH THỰC HIỆN PHÂN TÍCH CÁC THÀNH PHẦN CHÍNH BẰNG R

Nhập tập dữ liệu clothing_store_PCA_training và clothing_store_PCA_test lần lượt là clothes_train và clothes_test . Để đơn giản hóa mã sau này, chúng tôi sẽ tách dữ liệu đào tạo và kiểm tra thành các biến X và y.

```
y <- clothes_train$Sales.per.Visit
X <- clothes_train[, c(1:5)]
X_test <- clothes_test[, c(1:5)]
```

Hãy nhớ chuẩn hóa các biến dự đoán.

```
X_z <- as.data.frame(tỷ lệ(X))
colnames(X_z) <- c("Days.since.Purchase.Z", "Purchase.
Lượt truy cập.Z", "Ngày.on.File.Z",
      "Days.between.Purchases.Z", "Diff.Items.Purchased.Z")
```

Để có được ma trận tương quan, chúng ta sử dụng lệnh cor() .

```
tròn(cor(X_z), 3)
```

Lệnh cor() , với các biến dự đoán X_z làm đầu vào, được đặt bên trong lệnh round() . Đầu vào thứ hai của lệnh round() là số chữ số có nghĩa mà câu trả lời sẽ được làm tròn thành. Trong trường hợp của chúng tôi, chúng tôi đã chỉ định ba chữ số có nghĩa. Đầu ra từ lệnh round() được hiển thị trong Hình 12.6.

Để có được các giá trị VIF, trước tiên chúng ta cần chạy mô hình hồi quy, sau đó sử dụng lệnh vif() từ gói xe hơi như được trình bày chi tiết trong phần R trước đó.

```
model01 <- lm(formula = y ~ Days.since.Purchase.Z +
Purchase.Visits.Z + Days.on.File.Z +
        Days.between.Purchases.Z + Diff.Items.Purchased.Z, dữ liệu =
        X_z)
vif(model01)
```

Đầu ra của lệnh vif() được hiển thị trong Hình 12.7.

Machine Translated by Google

Để chạy PCA, trước tiên chúng ta phải cài đặt và mở gói psych .

```
install.packages("tâm lý"); thư viện (tâm lý)
```

Khi gói được mở, chúng tôi sử dụng lệnh hiệu trưởng () để thực hiện PCA.

```
pca01 <- hiệu trưởng(r = X_z, rotate = "varimax", nfactors = 5)
```

Chúng tôi đang sử dụng lệnh principal() với ba giá trị đầu vào. Đầu vào r = X_z chỉ định các biến chúng tôi muốn phân tích. Đầu vào rotate = “varimax” yêu cầu R thực hiện xoay varimax trên các thành phần trước khi trình bày kết quả.
Cuối cùng, trạng thái đầu vào nfactors = 5 mà chúng tôi muốn có năm thành phần. Chúng tôi lưu đầu ra PCA dưới dạng pca01.

Chúng tôi quan tâm đến việc tải kết quả pca01 .

```
in(pca01$loadings, ngưỡng = 0,49)
```

Chúng tôi sử dụng ngưỡng = 0,49 để triệt tiêu các trọng số PCA nhỏ. Đầu ra do đoạn mã trên tạo ra được thể hiện trong Hình 12.8.

Để tạo biểu đồ của các giá trị riêng, chúng ta cần các giá trị riêng của các thành phần được xoay. Chúng được đặt dưới “SS Loadings” trong Hình 12.8. Sau khi bạn lưu các giá trị này dưới dạng vectơ riêng của chúng, hãy vẽ chúng bằng cách sử dụng lệnh plot() .

```
ss.load <- c(1.566, 1.045, 1.035, 1.006, 0.348)
plot(ss.load, type = "b", main = "Lô đồ giá trị riêng", ylab = "Giá trị",

             xlab = "Thành phần"); abline(h = 1, lty =2)
```

Loại đầu vào = “b” vẽ các giá trị riêng bằng cách sử dụng cả điểm và đường kết nối. Đầu vào chính, xlab và ylab tùy chỉnh tiêu đề và nhãn trục. Lệnh abline() thêm một dòng vào biểu đồ. Giá trị đầu vào h = 1 thêm đường có vị trí nằm ngang 1 và đầu vào lty = 2 chỉ định đường đứt nét. Kết quả được thể hiện trong hình 12.9.

Để so sánh kết quả của việc không xoay với xoay varimax, hãy chạy lệnh principal() bằng cách sử dụng rotate = “none” cho trường hợp không xoay và rotate = “varimax”
cho phép quay varimax.

```
pca02_norot <- hiệu trưởng(r = X, rotate = "none", nfactors = 4)

in(pca02_norot$loadings, ngưỡng = 0,5)
pca02_rot <- hiệu trưởng(r = X, xoay = "varimax", nfactors = 4)

in(pca02_rot$loadings, ngưỡng = 0,5)
```

Một đoạn trích từ đầu ra print() của pca02_norot$loadings được hiển thị trong Hình 12.10a, trong khi một đoạn trích từ đầu ra print() của pca02_rot$loadings được hiển thị trong Hình 12.10b. Lưu ý rằng cả hai lệnh đều sử dụng cutoff = 0,5 để triệt tiêu các trọng số dưới 0,5.

Machine Translated by Google

Để xác thực kết quả PCA, hãy chạy thuật toán trên dữ liệu thử nghiệm. Đầu tiên, đứng loại bỏ dữ liệu.

```
X_test_z <- tỷ lệ(X_test)
```

Sau đó, xác nhận rằng bốn thành phần được khuyến nghị.

```
pca02_test <- hiệu trưởng(r = X_test_z, rotate =
"varimax", nfactors = 5)
pca02_test$loadings
```

Một đoạn trích của đầu ra được hiển thị trong Hình 12.11. Sau khi xác nhận rằng bốn thành phần được khuyến nghị, hãy kiểm tra trọng lượng thành phần.

```
pca02_test <- hiệu trưởng(r = X_test_z, rotate =
"varimax", nfactors = 4)
in(pca02_test$loadings, ngưỡng = 0,5)
```

Một đoạn trích của đầu ra được hiển thị trong Hình 12.12.

Để có được mối tương quan của các thành phần trong tập dữ liệu huấn luyện, hãy chạy round(cor()) trên điểm số từ pca02_rot.

```
vòng(cor(pca02_rot$scores),2)
```

Trong trường hợp này, chúng tôi làm tròn đến hai chữ số có nghĩa. Kết quả được thể hiện trong hình 12.13.

Để hồi quy biến mục tiêu trên các thành phần, bạn có thể muốn lưu từng thành phần dưới dạng biến riêng của nó. Điều này được hiển thị dưới đây.

```
PC1 <-pca02_rot$scores[,1]; PC2 <- pca02_rot$scores[,2]
PC3 <- pca02_rot$scores[,3]; PC4 <- pca02_rot$scores[,4]
```

Bây giờ bạn có thể chạy mô hình hồi quy và lấy VIF. Các mã được hiển thị dưới đây.

```
model.pca <- lm(y ~ PC1 + PC2 + PC3 + PC4); vif(model.pca)
```

Đầu ra từ lệnh vif() được hiển thị trong Hình 12.14.

12.11 KHI NÀO KHÔNG CÓ ĐA CỘNG TUYẾN TÍNH VẤN ĐỀ?

Bây giờ, tùy thuộc vào nhiệm vụ mà nhà phân tích phải đối mặt, đa cộng tuyến trên thực tế có thể không phải là một khiếm khuyết chết người. Weiss1 lưu ý rằng đa cộng tuyến "không ảnh hưởng bất lợi đến khả năng của phương trình hồi quy mẫu trong việc dự đoán biến phản ứng." Ông nói thêm rằng đa cộng tuyến không ảnh hưởng đáng kể đến các ước tính điểm của biến mục tiêu, khoảng tin cậy cho giá trị phản hồi trung bình hoặc dự đoán

1Weiss, Introductory Statistics, Ninth Edition, Pearson, London, 2010.

Machine Translated by Google

khoảng thời gian cho một giá trị phản hồi được chọn ngẫu nhiên. Tuy nhiên, do đó, nhà khoa học dữ liệu phải hạn chế nghiêm ngặt việc sử dụng mô hình đa cộng tuyến để ước tính và dự đoán biến mục tiêu. Việc giải thích mô hình sẽ không phù hợp, vì các hệ số riêng lẻ có thể không có ý nghĩa khi có sự tuyến tính nhiều màu. Tóm lại, các mô hình không tính đến đa cộng tuyến có thể được sử dụng để ước tính, nhưng không dùng để mô tả hoặc diễn giải.

NGƯỜI GIỚI THIỆU

Tên của gói ô tô là Đồng hành với hồi quy ứng dụng và có thể tìm thấy thông tin chi tiết tại đây: John Fox và Sanford Weisberg, An {R} Companion to Applied Regression, Ấn bản thứ hai, Sage, Thousand Oaks, CA, 2011.

Gói psych, cho phép chúng tôi PCA với phép quay varimax, được ghi lại ở đây: W. Revelle, Psych: Quy trình nghiên cứu tính cách và tâm lý, Đại học Northwestern, Evanston, IL, 2018.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Ý nghĩa của tính chiều cao trong khoa học dữ liệu là gì?
2. Tại sao chúng ta cần các phương pháp giảm kích thước?
3. Các thành phần chính thay thế bộ dự đoán m ban đầu bằng gì?
4. Thành phần chính nào có tính biến đổi nhiều nhất?
5. Thành phần chính nào khác tương quan với thành phần chính thứ nhất?
6. Tại sao chúng ta sử dụng phép quay?
7. Giải thích tiêu chí giá trị bản địa?
8. Tỷ lệ phương sai được giải thích theo tiêu chí là gì?
9. Đúng hay sai: Không cần thiết phải thực hiện xác thực các thành phần chính.
10. Khi chúng ta sử dụng các thành phần chính làm yếu tố dự đoán trong mô hình hồi quy, giá trị nào các VIF có lấy không? Điều này chỉ ra điều gì?

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với clothing_store_PCA_training và clothing_ bộ dữ liệu store_PCA_test . Sử dụng Python hoặc R để giải quyết từng vấn đề.

11. Tiêu chuẩn hóa hoặc bình thường hóa các yếu tố dự báo.
12. Xây dựng ma trận tương quan cho các biến dự đoán Lượt truy cập mua hàng, Số ngày trên tệp, Số ngày giữa các lần mua, Các mặt hàng khác nhau đã mua và Số ngày kể từ khi mua. Biến nào có tương quan cao?

Machine Translated by Google

13. Tính toán VIF cho từng biến dự đoán. Biến dự báo nào VIFs chỉ ra đa cộng tuyến là một vấn đề?

14. Chạy PCA bằng phép quay varimax và năm thành phần. Bao nhiêu phần trăm của sự thay đổi được giải thích bởi một thành phần? Bởi hai thành phần? Bởi tất cả năm thành phần?

15. Vẽ đồ thị của các giá trị riêng. Sử dụng tiêu chí giá trị riêng, bạn sẽ giữ lại bao nhiêu thành phần?

16. Giả sử chúng ta muốn giải thích ít nhất 80% sự thay đổi. Có bao nhiêu thành phần sẽ bạn giữ lại?

17. Chạy PCA bằng phép quay varimax và bốn thành phần. Bốn thành phần giải thích bao nhiêu phần trăm của sự thay đổi?

18. Biến hoặc các biến nào được chứa trong mỗi thành phần?

19. Sử dụng bốn thành phần làm biến dự đoán trong mô hình hồi quy để ước tính Doanh số trên mỗi lượt truy cập. Các hệ số hồi quy của bốn thành phần là gì?

20. VIF của 4 thành phần trong mô hình hồi quy là gì?

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu ngũ cốc . Sử dụng Python hoặc R để giải quyết từng vấn đề.

21. Chuẩn hóa hoặc chuẩn hóa các yếu tố dự đoán Đường, Chất xơ và Kali.

22. Xây dựng ma trận tương quan cho Đường, Chất xơ và Kali. Biến nào có tương quan cao?

23. Xây dựng mô hình hồi quy để ước tính Rating dựa trên Đường, Chất xơ và Kali. Lấy VIF từ mô hình. VIF nào chỉ ra rằng đa cộng tuyến là một vấn đề?

24. Chạy PCA bằng phép quay varimax và ba thành phần. Bao nhiêu phần trăm của sự thay đổi được giải thích bởi một thành phần? Bởi hai thành phần? Bởi cả ba thành phần?

25. Vẽ đồ thị giá trị riêng của ba thành phần. Sử dụng tiêu chí giá trị riêng, bạn sẽ giữ lại bao nhiêu thành phần?

26. Giả sử chúng ta muốn giải thích ít nhất 70% độ biến thiên. Có bao nhiêu thành phần sẽ bạn giữ lại?

27. Chạy PCA sử dụng xoay varimax và hai thành phần. bao nhiêu phần trăm của sự thay đổi làm hai thành phần giải thích?

28. Biến hoặc nhiều biến nào được chứa trong Thành phần 1? Biến hoặc các biến là gì có trong Hợp phần 2?

29. Sử dụng hai thành phần này làm biến dự đoán trong mô hình hồi quy để ước tính Xếp hạng. Các hệ số hồi quy của hai thành phần là gì?

30. VIF của hai thành phần trong mô hình hồi quy là bao nhiêu?

Đối với các bài tập sau, hãy làm việc với red_wine_PCA_training và red_wine_PCA_test bộ dữ liệu. Sử dụng Python hoặc R để giải quyết từng vấn đề. Biến mục tiêu là chất lượng rượu vang. Các yếu tố dự đoán là rượu, lượng đường còn lại, độ pH, mật độ và độ axit cố định.

Machine Translated by Google

31. Tiêu chuẩn hóa hoặc chuẩn hóa các yếu tố dự báo.

32. Xây dựng ma trận tương quan cho các yếu tố dự đoán. Giữa những yếu tố dự đoán nào bạn tìm mối tương quan cao nhất?

33. Xây dựng mô hình hồi quy để ước lượng chất lượng dựa trên các yếu tố dự báo. Lấy VIF từ mô hình. VIF nào chỉ ra rằng đa cộng tuyến là một vấn đề? So sánh các biến có VIF cao với các biến tương quan từ bài tập trước.

34. Thực hiện PCA bằng phép quay varimax. Hiển thị tỷ lệ phương sai đã xoay được giải thích để trích xuất tối đa năm thành phần. Bao nhiêu phần trăm của sự thay đổi được giải thích bởi một thành phần? Bởi hai thành phần? Bởi ba thành phần? Bởi bốn thành phần? Bởi tất cả năm thành phần?

35. Giả sử chúng ta muốn giải thích ít nhất 90% sự thay đổi. Có bao nhiêu thành phần tỷ lệ phương sai giải thích tiêu chí gợi ý chúng tôi trích xuất?

36. Vẽ biểu đồ giá trị riêng của năm thành phần. Theo giá trị riêng cri terion, chúng ta nên trích xuất bao nhiêu thành phần?

37. Kết hợp các khuyến nghị từ hai tiêu chí để đạt được sự đồng thuận về việc chúng ta nên trích xuất bao nhiêu thành phần.

38. Lập hồ sơ từng thành phần của bạn, nêu rõ những biến nào được bao gồm và lưu ý mối tương quan bên trong thành phần của chúng (tích cực hoặc tiêu cực). Để đơn giản, chỉ xem xét các trọng số thành phần lớn hơn 0,5.

39. Tạo ma trận tương quan cho các thành phần. Những giá trị này nghĩa là gì?

40. Tiếp theo, chỉ sử dụng các thành phần bạn đã trích xuất để ước tính chất lượng rượu bằng phương pháp hồi quy người mẫu. Không bao gồm các yếu tố dự đoán ban đầu.

    Một. So sánh giá trị của s và Radj ^{2} giữa hồi quy PCA và mô hình hồi quy ban đầu.

    b. Giải thích tại sao mô hình ban đầu vượt trội hơn một chút so với mô hình PCA.

    c. Giải thích làm thế nào mô hình PCA có thể được coi là vượt trội, mặc dù hơi vượt trội?

41. Trong hồi quy của bạn từ bài tập trước, VIF của hai thành phần trong mô hình hồi quy là gì? Những giá trị này nghĩa là gì?

Machine Translated by Google

chương 13

MÔ HÌNH TUYẾN TÍNH TỔNG HỢP

13.1 TỔNG QUAN VỀ MÔ HÌNH TUYẾN TÍNH TỔNG QUÁT

Trong Chương 11, các mô hình hồi quy tuyến tính mà chúng ta đã xem xét đều có một biến phản hồi liên tục. Tuy nhiên, điều gì sẽ xảy ra nếu chúng ta muốn xây dựng một mô hình hồi quy cho một phản hồi nhị phân? Hoặc cho một phản ứng rời rạc số? May mắn thay, có một nhóm các mô hình tuyến tính bao gồm cả ba trường hợp – liên tục, số rời rạc và nhị phân – của các biến phản ứng hồi quy: Mô hình tuyến tính tổng quát (GLM).

Để giải thích cách hồi quy cho ba loại phản hồi khác nhau có thể liên quan với nhau như thế nào, chúng ta sẽ xem xét ngắn gọn lại các phương trình hồi quy tham số cho từng trường hợp. Một khi chúng ta thiết lập được chúng có quan hệ với nhau như thế nào, thì chúng ta sẽ sử dụng các phiên bản mô tả của chúng, giống như chúng ta đã làm trong Chương 11.

Nhớ lại mô hình tham số cho hồi quy bội, được đưa ra ở đây.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad + \beta_{x_{trang}} + \varepsilon$$

Tổng β0+β1 x1+β2 x2++βp xp được gọi là công cụ dự đoán tuyến tính. Để cho ngắn gọn, chúng ta sẽ viết bộ dự đoán tuyến tính là Xβ. Công thức kết nối tiền tố tuyến tính với giá trị trung bình μ của biến y tại một tập hợp các giá trị đã cho của biến dự đoán được gọi là hàm liên kết, g(μ).

Các hàm liên kết khác nhau đòi hỏi các mô hình hồi quy khác nhau, với mỗi hàm liên kết được liên kết với một loại phản hồi cụ thể. Đối với mỗi loại phản hồi khác nhau mà chúng tôi thảo luận, chúng tôi sẽ chỉ định một g(μ) cụ thể sao cho Xβ = g(μ) và giải quyết cho μ để có được hình thức cuối cùng của mô hình.

Chúng tôi bắt đầu bằng cách chứng minh cách GLM hoạt động bằng cách chỉ ra cách hồi quy tuyến tính có thể được biểu thị dưới dạng GLM. Sau đó, chúng tôi chứng minh hai mô hình hồi quy mới dưới dạng GLM: hồi quy logistic và hồi quy Poisson.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

13.2 HỒI QUY TUYẾN TÍNH NHƯ MỘT MÔ HÌNH TUYẾN TÍNH TỔNG QUÁT

Khi biến phản hồi có phân phối Chuẩn tại mỗi tập hợp các giá trị biến dự đoán nhất định, thì chúng ta quay lại lĩnh vực hồi quy tuyến tính. Trong lĩnh vực này, chức năng liên kết chỉ đơn giản là chức năng nhận dạng, trong đó

$$g(\mu) = \mu$$

Đặt Xβ bằng g(μ) bằng cách sử dụng liên kết nhận dạng này sẽ cho chúng ta Xβ = g(μ) = μ.

Khi chúng ta có mối quan hệ chức năng giữa Xβ và μ, chúng ta có thể làm việc ngược lại để có được dạng cuối cùng của mô hình hồi quy bằng cách mở rộng ký hiệu viết tắt của chúng ta. Làm như vậy sẽ cho chúng ta phương trình dân số cho hồi quy tuyến tính:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad + \beta_{x_{trang}} + \varepsilon$$

từ đó chúng ta có thể có được hình thức mô tả

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \quad + b_{x_{trang}}$$

mà chúng ta đã làm việc trong Chương 11.

13.3 HỒI QUY LOGISTIC NHƯ MỘT MÔ HÌNH TUYẾN TÍNH TỔNG QUÁT

Tiếp theo, giả sử chúng ta đang cố gắng dự đoán một phản hồi nhị phân, chẳng hạn như liệu khách hàng có thẻ tín dụng của cửa hàng hay không. Trong trường hợp này, phân phối của biến phản hồi có thể là nhị phân: 1 hoặc 0, biểu thị Có hoặc Không.

Hàm liên kết cho biến phản hồi nhị phân là $g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$. chúng tôi đặt hàm này bằng với công cụ dự đoán tuyến tính Xβ của chúng tôi để có được

$$X\beta = \ln\left(\frac{\mu}{1-\mu}\right)$$

Để cô lập μ, chúng tôi sử dụng thực tế là eln(x) = x và thu được

$$\mu = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

Công thức trên đảm bảo giá trị trung bình của biến phản hồi, μ, sẽ luôn nằm trong khoảng từ 0 đến 1. Nói cách khác, giá trị mà mô hình hồi quy có thể được sử dụng để ước tính xác suất mà y = 1.

Để làm rõ rằng các giá trị dự đoán của chúng tôi từ hồi quy logistic là xác suất, thay vì giá trị nhị phân, chúng ta hãy viết mô hình hồi quy dưới dạng dự đoán p(y),

Machine Translated by Google

xác suất mà y = 1. Nếu chúng ta làm việc ngược từ ký hiệu viết tắt của mình, chúng ta sẽ nhận được dạng tham số của mô hình

$$p_{y}(\quad) = \frac{{}_{\text{kinh nghiệm}}\left(\beta_0 + \beta_1 x_1 + \beta_{x\ 2}\ {}_2 + \quad + \beta_{\text{trang}}\ x\right)}{1 + {}_{\text{kinh nghiệm}}\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad + \beta_{\text{trang}}\ x\right)} + \varepsilon$$

và có thể viết các hình thức mô tả như

$$\hat{p}_{y}(\quad) = \frac{{}_{\text{kinh nghiệm}}\left(b_0 x + b x_1\ {}_1 + \ {}_2\ {}_2 + \quad + b x_{\text{trang}}\right)}{1 + {}_{\text{kinh nghiệm}}\left(b_0 x + b x_1\ {}_1 + \ {}_2\ {}_2 + \quad + b x_{\text{trang}}\right)}$$

13.4 ỨNG DỤNG MÔ HÌNH HỒI QUY LOGISTIC

Chúng ta hãy xem lại bộ dữ liệu clothing_sales_training và clothing_sales_test . Lần này, mục tiêu của chúng tôi là xác định xem khách hàng có thẻ tín dụng của cửa hàng hay không, để nhóm tiếp thị của chúng tôi có thể gửi quảng cáo đến những người không có thẻ, lôi kéo họ đăng ký thẻ. Biến phản hồi của chúng tôi trong trường hợp này là nhị phân: Có, khách hàng có thẻ; hoặc Không, khách hàng không. Vì biến phản hồi là nhị phân nên chúng tôi sẽ sử dụng hồi quy logistic.

Mô hình hồi quy logistic tạm thời của chúng tôi sẽ là

$$\hat{p}\left(\text{thẻ tín dụng}\right) = \frac{{}_{\text{kinh nghiệm}}\left(b_0 + \text{Số}\ {}_1\left(\text{ngày giữa các lần mua b Tài}\ {}_2\ \text{khoản web}\quad \text{nt}\right)\right)}{1 + {}_{\text{kinh nghiệm}}\left(b_0 + \text{Số}\ {}_1\left(\text{ngày giữa các lần mua b Tài}\ {}_2\ \text{khoản web}\quad\right)\right)}$$

Kết quả hồi quy Thẻ tín dụng trên hai biến dự báo được thể hiện trong Hình 13.1. Các giá trị p được hiển thị trong đầu ra cho chúng ta biết rằng cả hai biến đều thuộc về mô hình. Khi chúng tôi xác thực chéo các kết quả với tập dữ liệu thử nghiệm, chúng tôi thu được kết quả như trong Hình 13.2.

Mô hình kiểm định xác nhận rằng cả hai biến đều thuộc về mô hình. Sử dụng các hệ số từ tập dữ liệu huấn luyện, chúng tôi thu được mô hình hồi quy logistic cuối cùng:

$$\hat{p}\left(\text{thẻ tín dụng}\right) = \frac{{}_{\text{kinh nghiệm}}\left(0.496 - 0.004\ \text{Số}\left(\text{ngày giữa các lần mua hàng}\right) + 1.254\left(\text{Tài khoản web}\right)\right)}{1 + {}_{\text{kinh nghiệm}}\left(0.496 - 0.004\ \text{Ngày giữa các lần mua}\right) + 1.254\left(\text{Tài khoản web}\right)}$$

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	0.4962	0.0887	5.5968	0.0000	0.3224	0.6699
Days	-0.0037	0.0004	-8.4491	0.0000	-0.0046	-0.0028
Web	1.2537	0.3307	3.7914	0.0001	0.6056	1.9018

Hình 13.1 Kết quả hồi quy logistic Python cho tập dữ liệu huấn luyện.

Machine Translated by Google

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	0.4634	0.0873	5.3105	0.0000	0.2924	0.6345
Days	-0.0035	0.0004	-8.2261	0.0000	-0.0043	-0.0026
Web	1.0973	0.2830	3.8780	0.0001	0.5427	1.6519

Hình 13.2 Kết quả hồi quy logistic Python cho tập dữ liệu thử nghiệm.

Vì vậy, làm thế nào để chúng ta giải thích các hệ số hồi quy logistic? Mỗi hệ số hồi quy mô tả sự thay đổi ước tính trong tỷ lệ cược log của biến phản hồi khi biến dự đoán của hệ số tăng thêm một. Để minh họa, hãy xem xét Tài khoản Web biến dự đoán nhị phân. Hệ số hồi quy cho Tài khoản Web là 1,254. Bằng cách tính toán e1,254 = 3,504, chúng tôi nhận thấy rằng một khách hàng có khả năng có thẻ tín dụng của cửa hàng cao hơn khoảng 3,5 lần nếu họ có tài khoản web so với khi họ không có tài khoản web.

Chúng tôi có thể thực hiện một hoạt động tương tự cho hệ số Ngày giữa các lần mua hàng. Bằng cách tính toán e 0,004 = 0,996, chúng tôi nhận thấy rằng cứ mỗi ngày đặt cược thêm vào thời điểm mua hàng, khả năng khách hàng có thẻ tín dụng tại cửa hàng sẽ thấp hơn 0,4%. Vì việc đếm số ngày riêng lẻ có thể là phép đo quá hẹp, nên chúng ta có thể nhân hệ số này với 30 để thu được e30× 0,004 = 0,89 và phát hiện ra rằng cứ sau 30 ngày không mua hàng, khả năng khách hàng mua hàng sẽ giảm đi 11% để có một cửa hàng thẻ tín dụng.[1]

13.4.1 Cách thực hiện hồi quy logistic bằng Python

Tải các gói cần thiết và nhập quần áo_bán hàng_đào tạo và quần áo_ bộ dữ liệu sales_test lần lượt là sales_train và sales_test .

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
nhập statsmodels.api dưới dạng sm
từ số liệu thống kê nhập khẩu scipy
sales_train = pd.read_csv("C:/.../clothing_sales_
đào tạo.csv")
sales_test = pd.read_csv("C:/.../clothing_sales_test.
csv")
```

Để đơn giản, chúng tôi tách các biến thành biến dự đoán X và biến phản hồi y. Thêm một hằng số vào khung dữ liệu X để đưa một hằng số vào mô hình hồi quy của chúng ta.

```
X = pd.DataFrame(sales_train[['Days', 'Web']])
X = sm.add_constant(X)
y = pd.DataFrame(sales_train[['CC']])
```

[1]Để biết thêm về hồi quy logistic, hãy xem Khai thác dữ liệu và Phân tích dự đoán.

Machine Translated by Google

Để thực hiện hồi quy logistic, hãy sử dụng các lệnh Logit() và fit() . Lưu đầu ra của mô hình và chạy lệnh summary2() trên đầu ra của mô hình đã lưu để xem kết quả của mô hình.

```
logreg01 = sm.Logit(y, X).fit()
logreg01.summary2()
```

Một đoạn trích của các kết quả được hiển thị trong Hình 13.1.

Để xác thực mô hình, hãy thực hiện các bước tương tự trên tập dữ liệu thử nghiệm. Mã được đưa ra dưới đây.

```
X_test = pd.DataFrame(sales_test[['Days', 'Web']])
X_test = sm.add_constant(X_test)
y_test = pd.DataFrame(sales_test[['CC']])
logreg01_test = sm.Logit(y_test, X_test).fit()
logreg01_test.summary2()
```

Một đoạn trích kết quả được hiển thị trong Hình 13.2.

13.4.2 Cách thực hiện hồi quy logistic bằng R

Nhập bộ dữ liệu clothing_sales_training và clothing_sales_test dưới dạng sales_train và sales_test, tương ứng.

Để chạy mô hình hồi quy logistic, chúng ta sẽ sử dụng lệnh glm() .

```
logreg01 <- glm(công thức = CC ~ Ngày + Web, dữ liệu = sales_
xe lửa, gia đình = nhị thức)
```

Phần lớn mã tương tự như trong Chương 11; đầu vào công thức liệt kê các biến phản hồi và biến dự đoán và đầu vào data = sales_train chỉ định tập dữ liệu. Những thay đổi duy nhất là lệnh glm() và thêm family = nhị thức
đầu vào. Lệnh glm() sẽ chạy phân tích GLM và family = nhị thức xác định mô hình hồi quy logistic. Lưu đầu ra của mô hình dưới dạng logreg01.

Để xem tóm tắt mô hình, hãy chạy lệnh summary() với tên của mô hình đã lưu làm đầu vào duy nhất. Một đoạn trích từ đầu ra được hiển thị trong Hình 13.3.

```
tóm tắt(logreg01)
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.4961706  0.0886529   5.597 2.18e-08 ***
Days        -0.0037016  0.0004381  -8.449  < 2e-16 ***
Web          1.2536955  0.3306672   3.791  0.00015 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hình 13.3 Kết quả hồi quy logistic từ R cho tập dữ liệu huấn luyện.

Machine Translated by Google


```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.4634478  0.0872706   5.310 1.09e-07 ***
Days        -0.0034721  0.0004221  -8.226  < 2e-16 ***
Web          1.0972994  0.2829570   3.878 0.000105 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hình 13.4 Kết quả hồi quy logistic từ R cho tập dữ liệu thử nghiệm.

Để xác nhận tính hợp lệ của mô hình, hãy chạy cùng một mô hình trên dữ liệu thử nghiệm và thu được bản tóm tắt của mô hình (Hình 13.4). Mã được đưa ra dưới đây.

```
logreg01_test <- glm(công thức = CC ~ Days + Web, data = sales_test,
family = nhị thức)
tóm tắt (logreg01_test)
```

13.5 KHỬ ĐỘC

Có nhiều loại mô hình hồi quy khác nằm dưới sự bảo trợ của GLM. Chúng ta sẽ xem xét một cái khác: Hồi quy Poisson. Hồi quy Poisson được sử dụng khi bạn muốn dự đoán số lượng sự kiện, chẳng hạn như số lần khách hàng sẽ liên hệ với dịch vụ khách hàng. Phân phối của biến phản hồi sẽ là số lần xuất hiện, với giá trị tối thiểu bằng 0.

Hàm liên kết cho biến phản hồi đếm là g(μ) = ln(μ). Chúng tôi đặt liên kết chức năng bằng với dự đoán tuyến tính của chúng tôi để có được

$$\beta = \ln(\mu)$$

Sau khi cô lập μ, chúng ta có

$$\mu = e^{\beta}$$

Làm việc ngược lại từ ký hiệu viết tắt của chúng tôi, chúng tôi tìm thấy tham số phiên bản của phương trình hồi quy Poisson

$$\text{các bạn} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{\text{trang}} x_{\text{trang}} + \varepsilon$$

từ đó chúng ta có thể viết các hình thức mô tả

$$\widehat{\text{các bạn}} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_{\text{trang}} x_{\text{trang}}$$

13.6 ỨNG DỤNG MÔ HÌNH HỒI QUY ĐỘC

Chúng tôi sẽ sử dụng tập dữ liệu rời bỏ để xây dựng một mô hình ước tính số lượng cuộc gọi dịch vụ khách hàng dựa trên việc khách hàng có rời bỏ hay không. Biến phản hồi của chúng tôi là một biến có giá trị nguyên, đó là lý do tại sao chúng tôi sử dụng hồi quy Poisson thay vì hồi quy tuyến tính cho ước tính này.

Machine Translated by Google

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3714	0.016	23.877	0.000	0.341	0.402
Churn = True	0.4305	0.034	12.582	0.000	0.363	0.498

Hình 13.5 Kết quả hồi quy Poisson của Python để dự đoán số lượng cuộc gọi dịch vụ khách hàng.

Cấu trúc của mô hình hồi quy Poisson của chúng tôi sẽ là

$$\text{Cuộc gọi CustServ} = \exp\left(\beta_0 + \beta_1 \text{Churn} \left(\quad \right)\right)$$

Kết quả phân tích hồi quy được đưa ra trong Hình 13.5. Sử dụng các hệ số đã cho ở trên, chúng ta có thể xây dựng mô hình hồi quy Poisson

$$\text{Cuộc gọi CustServ} = \exp\left(0.3714 + 0.4305 \text{ Churn} \left(\text{True} = \quad \right)\right)$$

Bây giờ, chúng ta diễn giải các hệ số hồi quy Poisson như thế nào? Khi được sử dụng làm số mũ của e, hệ số hồi quy mô tả sự thay đổi cấp số nhân ước tính trong biến trả lời khi biến dự đoán của hệ số tăng thêm một. Trong trường hợp của chúng tôi, hệ số hồi quy là 0,4305, mang lại cho chúng tôi e0,4305 = 1,538.
Yếu tố dự đoán của hệ số, Churn, bằng 0 nếu khách hàng không rời bỏ và bằng 1 nếu họ rời đi. Do đó, việc chuyển từ khách hàng không rời bỏ sang khách hàng rời bỏ làm tăng số lượng cuộc gọi dịch vụ khách hàng dự đoán mà khách hàng thực hiện lên 1,538 lần, tương đương 53,8%.

13.6.1 Cách thực hiện hồi quy Poisson bằng Python

Đương nhiên, để xác thực kết quả lập mô hình của chúng tôi, trước tiên bạn sẽ chia dữ liệu thành tập dữ liệu huấn luyện và kiểm tra. Vì kiểm định chéo đã được trình bày trong các chương khác cũng như đối với hồi quy logistic, chúng tôi giới hạn phần này để minh họa cách xây dựng mô hình hồi quy Poisson.

Tải các gói cần thiết.

```
nhập gấu trúc dưới dạng pd
nhập numpy dưới dạng np
nhập statsmodels.api dưới dạng sm
nhập statsmodels.tools.tools dưới dạng stattools
```

Đọc trong bộ dữ liệu churn vào Python dưới dạng churn.

```
churn = pd.read_csv("C:/.../churn")
```

Biến dự đoán của chúng tôi là Churn, là biến phân loại. Tương tự như các tác vụ lập mô hình trước đó, chúng ta cần thay đổi các giá trị phân loại của Churn thành giả

Machine Translated by Google

biến. Đối với bài tập này, chúng tôi sẽ sử dụng một biến giả, bằng một nếu khách hàng rời bỏ.

```
churn_ind = pd.get_dummies(churn['Churn'], drop_first = True)
```

Lệnh get_dummies() tạo hai biến chỉ báo, một biến cho mỗi giá trị phân loại trong Churn. Đầu vào drop_first = True sẽ loại bỏ biến giả đầu tiên, tương ứng với Churn = False trong trường hợp của chúng tôi và giữ lại biến giả còn lại, tương ứng với Churn = True.

Các lệnh còn lại lưu biến giả mới dưới dạng khung dữ liệu, thêm một hằng số để mô hình hồi quy của chúng ta có một hằng số và đổi tên các cột để đầu ra dễ đọc hơn.

```
X = pd.DataFrame(churn_ind)
X = sm.add_constant(X)
X.columns = ['const', 'Churn = True']
```

Chúng tôi cũng chuẩn bị biến phản ứng.

```
y = pd.DataFrame(churn[['CustServ Calls']])
```

Cuối cùng, chúng tôi chạy hồi quy Poisson bằng lệnh GLM() .

```
poisreg01 = sm.GLM(y, X, family = sm.families.
Poisson()).fit()
```

Lưu ý ba giá trị đầu vào của lệnh GLM() . Hai biến đầu tiên, y và X, lần lượt chỉ định biến phản hồi và biến dự báo. Thứ ba, family = sm.families.Poisson(), xác định rằng nên sử dụng hồi quy Poisson.

Lệnh fit() sẽ khớp mô hình với dữ liệu của chúng ta. Chúng tôi lưu kết quả dưới dạng poisreg01.

Sử dụng lệnh summary() để xem kết quả của mô hình.

```
poisreg01.summary()
```

Một đoạn trích từ đầu ra của summary() được hiển thị trong Hình 13.5.

13.6.2 Cách thực hiện hồi quy Poisson bằng R

Như với ví dụ Python của chúng tôi, chúng tôi giới hạn phần này để minh họa cách xây dựng mô hình hồi quy Poisson. Chúng ta quay lại lệnh glm() mà trước đây chúng ta đã sử dụng để xây dựng mô hình hồi quy logistic, bây giờ là xây dựng mô hình hồi quy Poisson.

```
poisreg01 <- glm(công thức = CustServ.Calls ~ Churn, data = churn,
family = poisson)
```

Machine Translated by Google


```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.37377    0.01638   22.82   <2e-16 ***
ChurnTrue    0.42795    0.03602   11.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hình 13.6 Kết quả hồi quy Poisson từ R.

Đầu vào công thức hiện chỉ định Cuộc gọi CustServ làm biến phản hồi và Churn làm biến dự đoán. Họ đầu vào = poisson xác định rằng nên áp dụng hồi quy Poisson cho dữ liệu. Lưu đầu ra hồi quy dưới dạng poisreg01.

Sử dụng lệnh summary() để xem thông tin chi tiết về mô hình.

```
tóm tắt(poisreg01)
```

Một đoạn trích từ lệnh summary() được hiển thị ở đây trong Hình 13.6.

Để có toàn văn về khám phá thế giới GLM: P. McCullagh và JA Nelder, Generalized Linear Models, Second Edition, Chapman & Hall, London, 1992.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Ba trường hợp biến phản ứng hồi quy được thảo luận trong chương này là gì?
2. Loại mô hình hồi quy nào bao gồm cả ba trường hợp của các biến phản ứng?
3. Chúng ta gọi bộ dự đoán tuyến tính là gì? Làm thế nào để chúng tôi viết nó ở dạng viết tắt của nó?
4. Chức năng liên kết kết nối hai điều gì? Làm thế nào để chúng ta viết nó ở dạng viết tắt của nó?
5. Hàm liên kết cho hồi quy tuyến tính là gì?
6. Chúng ta nên sử dụng loại hồi quy nào khi cố gắng dự đoán phản hồi nhị phân Biến đổi?
7. Chức năng liên kết cho hồi quy logistic là gì?
8. Các giá trị dự đoán từ xác suất hồi quy logistic hay giá trị nhị phân?
9. Dạng mô tả của mô hình hồi quy logistic là gì?
10. Loại hồi quy nào chúng ta nên sử dụng khi cố gắng dự đoán phản hồi đếm Biến đổi?
11. Hàm liên kết cho hồi quy Poisson là gì?
12. Dạng mô tả của mô hình hồi quy Poisson là gì?

Machine Translated by Google

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với thử nghiệm clothing_sales_training và clothing_sales_test bộ dữ liệu. Sử dụng Python hoặc R để giải quyết từng vấn đề.

13. Tạo mô hình hồi quy logistic để dự đoán liệu khách hàng có thẻ tín dụng của cửa hàng hay không, dựa trên việc họ có tài khoản web hay không và số ngày giữa các lần mua hàng. Có được bản tóm tắt của mô hình.

14. Có biến nào cần loại bỏ khỏi mô hình không? Nếu vậy, loại bỏ chúng và chạy lại mô hình.

15. Viết dạng mô tả của mô hình hồi quy logistic sử dụng các hệ số thu được từ Câu hỏi 1.

16. Xác thực mô hình bằng bộ dữ liệu thử nghiệm.

17. Nhận các giá trị dự đoán của biến phản hồi cho mỗi bản ghi trong tập dữ liệu.

Đối với các bài tập sau, hãy làm việc với bộ dữ liệu khuấy . Sử dụng Python hoặc R để giải quyết từng vấn đề.

18. Tạo mô hình hồi quy Poisson để dự đoán số lượng cuộc gọi dịch vụ khách hàng mà một người thực hiện, dựa trên việc khách hàng đó có rời bỏ hay không. Có được một bản tóm tắt của mô hình.

19. Viết dạng mô tả của mô hình hồi quy Poisson từ bài tập trước.

PHÂN TÍCH THỰC TẾ

Đối với các bài tập sau, hãy làm việc với tập dữ liệu dành cho người lớn . Sử dụng Python hoặc R để giải quyết từng vấn đề.

20. Xây dựng mô hình hồi quy logistic để dự đoán thu nhập của một người dựa trên độ tuổi, trình độ học vấn của họ (dưới dạng số, với giáo dục.num có thể thay đổi) và số giờ làm việc mỗi tuần. Có được bản tóm tắt của mô hình.

21. Có biến nào cần loại bỏ khỏi mô hình từ biến trước không? bài tập? Nếu vậy, loại bỏ các biến và chạy lại mô hình.

22. Viết dạng mô tả của mô hình hồi quy logistic cuối cùng từ mô hình trước bài tập.

23. Diễn giải hệ số của biến tuổi .

24. Tìm tác động đối với xác suất để một người có thu nhập cao cứ sau 10 năm lớn hơn.

25. Diễn giải hệ số của biến education.num .

26. Tìm tác động đối với xác suất có thu nhập cao sau mỗi bốn năm học thêm của một người.

27. Giải thích hệ số của biến số giờ.per.tuần .

28. Tìm tác động đối với xác suất có thu nhập cao cứ sau 5 giờ làm việc tuần một người làm việc.

Machine Translated by Google

29. Lấy các giá trị dự đoán bằng cách sử dụng mô hình từ bài tập trước. So sánh các giá trị dự đoán với các giá trị thực tế.

30. Xây dựng mô hình hồi quy Poisson để dự đoán số năm đi học của một người (sử dụng biến education.num) dựa trên tuổi của một người và số giờ họ làm việc mỗi tuần. Có được bản tóm tắt của mô hình.

31. Có biến nào cần loại bỏ khỏi mô hình từ biến trước đó không? bài tập? Nếu vậy, loại bỏ các biến và chạy lại mô hình.

32. Viết dạng mô tả của mô hình hồi quy Poisson cuối cùng từ mô hình trước bài tập.

33. Lấy các giá trị dự đoán bằng cách sử dụng mô hình từ bài tập trước. So sánh các giá trị dự đoán với các giá trị thực tế.

Machine Translated by Google

chương 14

QUY CHẾ HỘI

14.1 GIỚI THIỆU QUY TẮC HỘI

Các luật kết hợp tìm cách phát hiện ra các liên kết giữa các biến và có dạng "Nếu tiền đề, thì hệ quả", cùng với thước đo độ hỗ trợ và độ tin cậy liên quan đến luật. Ví dụ: một siêu thị cụ thể có thể thấy rằng trong số 1000 khách hàng mua sắm vào tối thứ Năm, có 200 người mua tã và trong số 200 người mua tã, 50 người mua bia. Do đó, quy tắc kết hợp sẽ là: "Nếu mua bỉm, sau đó mua bia," với độ hỗ trợ 50/1000 = 5% và độ tin cậy 50/200 = 25%.

Vấn đề nan giải đang chờ đợi bất kỳ thuật toán nào như vậy là lời nguyền của tính không thứ nguyên: Số lượng các luật kết hợp có thể tăng theo cấp số nhân về số lượng thuộc tính. Cụ thể, nếu có k thuộc tính, chúng ta tự giới hạn ở các thuộc tính nhị phân, chúng ta chỉ tính đến các trường hợp dương (ví dụ: mua tã = có), có thứ tự k · 2k 1 luật kết hợp khả dĩ.1 Giả sử rằng một điển hình ứng dụng của luật kết hợp là phân tích rổ thị trường và có thể có hàng nghìn thuộc tính nhị phân (mua bia? mua bỏng ngô? mua sữa? mua bánh mì? v.v.), vấn đề tìm kiếm thoạt nhìn có vẻ hoàn toàn vô vọng. Ví dụ: giả sử rằng một cửa hàng tiện lợi nhỏ chỉ có 100 mặt hàng khác nhau và khách hàng có thể mua hoặc không mua bất kỳ sự kết hợp nào của 100 mặt hàng đó. Sau đó, có 2100 1,27×1030 quy tắc kết hợp có thể đang chờ thuật toán tìm kiếm gan dạ của bạn.
Tuy nhiên, may mắn thay, thuật toán tiên nghiệm để khai thác các luật kết hợp tận dụng cấu trúc bên trong chính các luật đó để giảm vấn đề tìm kiếm xuống kích thước dễ quản lý hơn.2

1David J. Hand, Heikki Mannila, và Padhraic Smyth, Nguyên tắc khai thác dữ liệu, MIT Press, Cambridge, 2001.

2Để biết chi tiết về cách hoạt động của thuật toán tiên nghiệm, hãy xem Khai thác dữ liệu và Phân tích dự đoán.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

14.2 VÍ DỤ ĐƠN GIẢN VỀ KHAI THÁC QUY TẮC KẾT HỢP

Chúng tôi bắt đầu với một ví dụ đơn giản. Giả sử rằng một nông dân địa phương đã dựng một quầy rau bên đường và đang rao bán những mặt hàng sau: {măng tây, đậu, bông cải xanh, ngô, ớt xanh, bí và cà chua}. Biểu thị tập hợp các mục này là I.

Từng người một, khách hàng tấp vào lề, nhấc giỏ và mua nhiều loại com khác nhau. sự kết hợp của các mục này, tập hợp con của I.

Gọi D là tập hợp các giao dịch được trình bày trong Bảng 14.1, trong đó mỗi hành động giao dịch T trong D đại diện cho một tập hợp các mục chứa trong I.

Giả sử rằng chúng ta có một tập hợp cụ thể các mặt hàng A (ví dụ: đậu và bí) và một tập hợp các mặt hàng B khác (ví dụ: măng tây). Sau đó, xác định một quy tắc kết hợp như sau:

Một luật kết hợp có dạng:

$$\text{nếu Athen B tôi}\left(\text{e. . } AB \Rightarrow \quad \right),$$

trong đó tiền đề A và B hệ quả là các tập hợp con riêng của I và A và B loại trừ lẫn nhau.

Định nghĩa này sẽ loại trừ các quy tắc tầm thường như đậu và bí thì đậu.

BẢNG 14.1 Các giao dịch được thực hiện tại quầy rau ven đường

Giao dịch	Mặt hàng đã mua
1	Bông cải xanh, ớt xanh, ngô
2	Măng tây, bí, ngô
3	Bắp, cà chua, đậu, bí
4	Ớt xanh, ngô, cà chua, đậu
5	Đậu, măng tây, bông cải xanh
6	Bí, măng tây, đậu, cà chua
7	cà chua, ngô
[illegible]	Bông cải xanh, cà chua, ớt xanh
9	Bí, măng tây, đậu
10	Đậu, ngô
11	Ớt xanh, bông cải, đậu, bí
12	Măng tây, đậu, bí
13	Bí, ngô, măng tây, đậu
14	Bắp, ớt xanh, cà chua, đậu, bông cải xanh

14.3 HỖ TRỢ, TỰ TIN VÀ NÂNG CẤP

Độ đo mức độ tốt của một luật kết hợp bao gồm độ hỗ trợ, độ tin cậy và độ nâng.
Độ hỗ trợ cho một luật kết hợp cụ thể A B là tỷ lệ các giao dịch trong D chứa cả A và B.
Nghĩa là,

Machine Translated by Google

$$\text{hỗ trợ} = P(A \cap B) = \frac{\text{số giao dịch c chứa cả A và B}}{\text{tổng số giao dịch}}$$

Độ tin cậy của luật kết hợp A B là thước đo độ chính xác của luật, được xác định bằng tỷ lệ phần trăm giao dịch trong D chứa A cũng chứa B. Nói cách khác,

$$\text{tự tin} = P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$

$$= \frac{\text{số lượng giao dịch chứa cả A và B}}{\text{số giao dịch chứa A}}$$

Trong ngôn ngữ xác suất, độ tin cậy biểu thị khả năng xác suất có điều kiện của B, cho trước A.

Ví dụ: xem xét quy tắc kết hợp, “Nếu mua bí, thì hãy mua đậu,” trong đó A đại diện cho bí và B đại diện cho đậu. Từ Bảng 14.1, chúng ta có bảy giao dịch mua bí sau đây (Bảng 14.2). Sáu hành động chuyển đổi mà đậu cũng được mua được in đậm.

Như vậy:

$$\text{hỗ trợ} = P(A \cap B) = \frac{6}{14} = 42.9\% \quad \frac{\text{giao dịch chứa cả A và B}}{\text{tổng giao dịch}}$$

$$\text{tự tin} = P(B \mid A) = \frac{6}{7} = 85.7\% \quad \frac{\text{giao dịch chứa cả A và B}}{\text{giao dịch chứa A}}$$

Một biện pháp khác để định lượng tính hữu dụng của luật kết hợp là thang máy. Thang máy so sánh độ tin cậy trong việc sử dụng quy tắc kết hợp của chúng tôi với xác suất chỉ chọn hệ quả một cách ngẫu nhiên mà không cần viện đến quy tắc kết hợp. Ta định nghĩa thang máy như sau:

$$\text{Thang máy} = \frac{\text{Quy tắc tự tin}}{\text{tỷ lệ p trước của hậu quả}}$$

BẢNG 14.2 Các giao dịch mua bí đao

Giao dịch	Mặt hàng đã mua
2	Măng tây, bí, ngô
3	Bắp, cà chua, đậu, bí
6	Bí, măng tây, đậu, cà chua
9	Bí, măng tây, đậu
11	Ớt xanh, bông cải, đậu, bí
12	Măng tây, đậu, bí
13	Bí, ngô, măng tây, đậu

Machine Translated by Google

Nhớ lại ví dụ về siêu thị, trong số 1000 khách hàng, 200 người mua đường và trong số 200 khách hàng đã mua bỉm này, 50 khách hàng cũng đã mua bia.

Giả sử 100 trong số 1000 khách hàng đã mua bia. Do đó, tỷ lệ trước đây của những người đã mua bia là 100/1000 = 10%. Độ tin cậy của quy tắc là 50/200 = 25%. Do đó, mức tăng của quy tắc kết hợp, "Nếu mua bỉm, thì hãy mua bia," là

$$\text{Thang máy} = \frac{\text{Quy tắc tự tin}}{\text{tỷ lệ p trước của hậu quả}} = \frac{0\ 2.\ 5}{0\ .10} = 2\ .5$$

Điều này có thể được hiểu là "Khách hàng mua bỉm có khả năng mua bia cao gấp 2,5 lần so với khách hàng trong toàn bộ tập dữ liệu". Rõ ràng, quy tắc kết hợp này sẽ hữu ích cho người quản lý cửa hàng muốn bán nhiều bia hơn.

Đối với luật kết hợp "Nếu mua bí (A) thì mua đậu (B)" ta có:

$$\text{Thang máy} = \frac{\text{Quy tắc tự tin}}{\text{tỷ lệ p trước của hậu quả}} = \frac{6\ 7/}{/\ 9\ 14} = 1\ .33$$

Khách hàng mua bí có khả năng mua đậu cao hơn 33% so với nhóm khách hàng chung.

14.4 QUY ĐỊNH CỦA HIỆP HỘI KHAI THÁC

Vì vậy, chúng ta hãy bắt tay vào khai thác các luật kết hợp bằng cách sử dụng Churn_Training_ Tập dữ liệu tập tin. Chuẩn bị bằng cách làm như sau:

- Tập hợp các biến sau vào khung dữ liệu riêng của chúng: VMail Plan, Intl Plan, CustServ Calls và Churn.
- Đặt CustServ Calls là một yếu tố thứ tự.

Chúng ta hãy bắt đầu bằng cách tìm các tỷ lệ "đường cơ sở" cho các biến khác nhau, để sau này chúng ta có thể kiểm tra mức độ tin cậy của các luật kết hợp của chúng ta so với các mức đường cơ sở này. Những tỷ lệ này có thể được tìm thấy trong Hình 14.1 và 14.2. Ví dụ, tỷ lệ khách hàng rời bỏ là 14,53%.

Bây giờ, chúng ta hãy tạo một số luật kết hợp, sử dụng các cài đặt sau:

- Chỉ định loại liên kết để lấy dưới dạng "quy tắc"
- Hỗ trợ tối thiểu bằng 0,01 (1%)
- Độ tin cậy tối thiểu bằng 0,4 (40%)
- Số tiền đề tối đa 1

Sau khi các quy tắc được tạo, bạn có thể phải xóa các quy tắc có chứa Churn trong tiền đề. Các quy tắc thu được sau khi làm như vậy được hiển thị trong Hình 14.3, được sắp xếp theo thang máy.

Machine Translated by Google


```
> t11
            Intl.Plan = no Intl.Plan = yes
Count           2705.0000        295.0000
Proportion         0.9017          0.0983
> t22
            VMail.Plan = no VMail.Plan = yes
Count            2170.0000         830.0000
Proportion          0.7233           0.2767
> t33
            Churn = False Churn = True
Count           2564.0000     436.0000
Proportion         0.8547       0.1453
```

Hình 14.1 Tỷ lệ cho Gói quốc tế, Gói thư thoại và Rời khỏi R.

```
> t44
              CSC = 0  CSC = 1   CSC = 2   CSC = 3   CSC = 4
Count       626.00000 1068.000 679.00000 383.00000 149.00000
Proportion    0.20867     0.356   0.22633   0.12767   0.04967
              CSC = 5  CSC = 6 CSC = 7 CSC = 8 CSC = 9
Count       61.00000 22.00000 8.00000 2.00000 2.00000
Proportion   0.02033  0.00733 0.00267 0.00067 0.00067
```

Hình 14.2 Tỷ lệ cuộc gọi dịch vụ khách hàng từ R.

```
     lhs                    rhs              support    confidence lift      count
[1]  {CustServ.Calls=5} => {Churn=True}     0.01200000 0.5901639  4.0607610   36
[2]  {CustServ.Calls=4} => {Churn=True}     0.02266667 0.4563758  3.1402007   68
[3]  {Intl.Plan=yes}    => {Churn=True}     0.04233333 0.4305085  2.9622143  127
[4]  {CustServ.Calls=3} => {VMail.Plan=no} 0.09933333 0.7780679  1.0756699  298
[5]  {VMail.Plan=yes}   => {Churn=False}    0.25200000 0.9108434  1.0657294  756
[6]  {CustServ.Calls=1} => {Churn=False}    0.32000000 0.8988764  1.0517275  960
[7]  {CustServ.Calls=3} => {Churn=False}    0.11433333 0.8955614  1.0478487  343
[8]  {CustServ.Calls=4} => {VMail.Plan=no} 0.03733333 0.7516779  1.0391860  112
[9]  {CustServ.Calls=2} => {Churn=False}    0.20066667 0.8865979  1.0373611  602
[10] {Intl.Plan=no}     => {Churn=False}    0.79866667 0.8857671  1.0363890 2396
```

Hình 14.3 10 luật kết hợp đầu tiên được phát hiện bởi R, được sắp xếp theo thang máy.

Quy tắc kết hợp có mức tăng lớn nhất là quy tắc đầu tiên, với ID quy tắc [1]:

Nếu dịch vụ khách hàng gọi 5 ~~thì~~ Churn True =

Mức tăng cho ID quy tắc [1] là khoảng 4,06.

14.4.1 Cách khai thác quy tắc liên kết bằng R

Đọc trong tập dữ liệu Churn_Training_File và đặt tên là churn. Bước đầu tiên là tập hợp con từ tập dữ liệu chỉ các cột mà chúng ta muốn có quy tắc kết hợp.

```
min.churn <-
  tập hợp con (churn, select = c("Intl.Plan", "VMail.Plan",
 "CustServ.Calls",
            "Khuấy động"))
```

Machine Translated by Google

Như chúng ta đã thấy trước đây, lệnh subset() lấy một tập dữ liệu và trích xuất các hàng hoặc cột đã chỉ định từ đó. Vì chúng tôi muốn có bốn cột, chúng tôi đặt tên của chúng trong một vectơ bên dưới lựa chọn đầu vào. Chúng tôi đặt tên cho khung dữ liệu mới của mình là tối thiểu. khuấy đảo.

Để thay đổi Cuộc gọi dịch vụ khách hàng thành một yếu tố, chúng tôi sử dụng phương thức đã đặt hàng () yêu cầu.

```
min.churn$CustServ.Calls <- đã đặt hàng(as.factor(min.
churn$CustServ.Calls))
```

Chúng tôi làm việc với hai lệnh lồng nhau ở đây. Đầu tiên, as.factor() lấy biến CustServ.Calls và tạo thành hệ số cho các giá trị khác nhau. Tuy nhiên, biến sẽ được coi là danh nghĩa, không phải thứ tự. Để đặt các mức là thứ tự, chúng ta đưa lệnh as.factor() vào bên trong lệnh có thứ tự() . Thứ tự của các mức trong yếu tố CustServ.Calls giờ đây sẽ được đặt theo thứ tự tăng dần.

Để có được các phân phối cơ sở của bốn biến, chúng tôi sử dụng các bảng. Mã cho bảng đầu tiên được hiển thị bên dưới. Ba bảng còn lại là một bài tập.

```
t1 <- table(min.churn$Intl.Plan)
t11 <- rbind(t1, round(prop.table(t1), 4))
```

Các lệnh table(), prop.table(), và round() đã được thảo luận trong các chương trước. Bảng t1 chứa số lượng khách hàng có và không có Gói quốc tế, trong khi lệnh prop.table() trả về một bảng có tỷ lệ cho các danh mục giống nhau. Lệnh rbind() tạo một ma trận với số lượng và tỷ lệ cùng nhau. Lưu ma trận là t11.

Để làm cho bảng dễ đọc hơn, chúng tôi thêm tên cột và hàng.

```
colnames(t11) <- c("Intl.Plan = no", "Intl.Plan = yes")
rownames(t11) <- c("Count", "Proportion")
t11
```

Để biết thứ tự đặt các giá trị colnames() , hãy xem t1. Cột đầu tiên có các giá trị “không”. Chúng tôi sử dụng thông tin này để thông báo thứ tự của các giá trị colnames() . Cụ thể, giá trị “Intl.Plan = no” xuất hiện trước, tiếp theo là “Intl.Plan = yes”. Kết quả, t11, được thể hiện như bảng đầu tiên trong Hình 14.1.

Khi chúng tôi đã hoàn tất việc thiết lập dữ liệu và lấy thông tin phân phối cơ sở, hãy cài đặt và tải gói cho các quy tắc kết hợp: gói arules .

```
install.packages("arules"); thư viện(arules)
```

Để có được các quy tắc kết hợp, hãy chạy lệnh apriori() từ arules bưu kiện.

```
all.rules <- apriori(data = min.churn, tham số = list(supp = 0,01, target
= "rules",
            conf = 0,4, minlen = 2, maxlen = 2))
```

Machine Translated by Google


```
     lhs                    rhs              support    confidence lift     count
[1]  {CustServ.Calls=5} => {Churn=True}     0.01200120 0.6060606  4.182195   40
[2]  {CustServ.Calls=4} => {Churn=True}     0.02280228 0.4578313  3.159321   76
[3]  {Intl.Plan=yes}    => {Churn=True}     0.04110411 0.4241486  2.926889  137
[4]  {Churn=True}       => {VMail.Plan=no} 0.12091209 0.8343685  1.153443  403
[5]  {VMail.Plan=yes}   => {Churn=False}    0.25262526 0.9132321  1.068001  842
[6]  {CustServ.Calls=3} => {VMail.Plan=no} 0.09930993 0.7715618  1.066618  331
[7]  {CustServ.Calls=3} => {Churn=False}    0.11551155 0.8974359  1.049528  385
[8]  {CustServ.Calls=1} => {Churn=False}    0.31773177 0.8966977  1.048664 1059
[9]  {CustServ.Calls=2} => {Churn=False}    0.20162016 0.8853755  1.035423  672
[10] {Churn=False}      => {Intl.Plan=no}  0.79927993 0.9347368  1.035042 2664
```

Hình 14.4 Top 10 luật từ tất cả. quy tắc từ R, được sắp xếp theo giá trị thang máy.

Mặc dù đầu vào bắt buộc duy nhất là data = min.churn, nhưng chúng tôi cần chỉ định cài đặt tham số đã nêu trước đó. Đầu tiên, supp = 0,01 sẽ đặt mức hỗ trợ tối thiểu là 1%. Đầu vào target = “rules” chỉ định rằng chúng ta muốn các quy tắc kết hợp.
Các giá trị đầu vào minlen = 2 và maxlen = 2 xác định rằng chúng ta muốn tiền đề có chính xác một mục, vì quy tắc có tiền đề trống được coi là có độ dài là một. Cuối cùng, conf = 0,4 sẽ đặt mức hỗ trợ tối thiểu là 40%. Lưu kết quả của thuật toán này với tên all.rules.

Để xem 10 quy tắc hàng đầu mà chúng tôi đã thu được, được sắp xếp theo giá trị mức tăng của chúng, hãy sử dụng các lệnh kiểm tra () và đầu ().

```
kiểm tra (đầu (all.rules, by = "nâng", n = 10))
```

Các giá trị by và n chỉ định tiêu chí để sắp xếp theo và số quy tắc tối đa sẽ trả về. Kết quả được thể hiện trong hình 14.4.

Lưu ý rằng Hình 14.4 chứa một số quy tắc có Churn ở tiền đề (“lhs,” hoặc “bên trái”). Chúng tôi không muốn những quy tắc này. Điều này có nghĩa là bước tiếp theo của chúng tôi là tập hợp con từ all.rules chỉ các quy tắc không chứa Churn trong tiền đề.

Để bắt đầu, chúng ta cần xác định quy tắc nào có Churn ở tiền đề, lhs.
Để làm việc với lhs, chúng ta cần định dạng các quy tắc dưới dạng khung dữ liệu. Tuy nhiên, thuật toán apriori() không trả về đầu ra được định dạng dưới dạng khung dữ liệu. Để chuyển đổi định dạng của lhs sang data frame, chúng ta sử dụng 2 lệnh as() .

```
all.rules.ant.df <- as(as(attr(all.rules, "lhs"), "giao dịch"), "data.frame")
```

Mã lõi, attr(all.rules, “lhs”), xác định rằng chúng ta đang làm việc với tiền đề (lhs) của các quy tắc có trong all.rules. Lệnh as() đầu tiên , lấy mã attr() làm đầu vào và thêm "giao dịch" đầu vào bổ sung , thay đổi tiền đề thành một định dạng dành riêng cho gói arules() được gọi là giao dịch. Bước này là bắt buộc, vì một đối tượng thuộc loại giao dịch sau đó có thể được chuyển đổi thành định dạng khung data.frame bằng cách sử dụng lệnh as() thứ hai , lần này sử dụng giá trị đầu vào thứ hai “data.frame”. Chúng tôi lưu kết quả dưới dạng all.rules.
ant.df để biểu thị rằng tiền đề (“ant”) của đối tượng all.rules đã được chuyển thành khung dữ liệu (“df”).

Machine Translated by Google

Bây giờ chúng tôi đã cô lập các tiền đề trong một hình thức mà chúng tôi có thể làm việc với, chúng tôi kiểm tra chúng để xem cái nào chứa Churn = True hoặc Churn = False.

```
t1 <- rules.dataframe$items == "{Churn=True}"
t2 <- rules.dataframe$items == "{Churn=False}"
non.churn.ant <-abs(t1+t2-1)
```

Các vectơ t1 và t2 là một chuỗi các số 0 và 1, trong đó số 0 có nghĩa là tiền đề không đáp ứng điều kiện và một có nghĩa là nó đã đáp ứng điều kiện. Khi chúng ta lấy giá trị tuyệt đối của t1+t2-1 bằng cách sử dụng lệnh abs() , kết quả là một vectơ duy nhất gồm các số 0 và 1, trong đó các số 1 biểu thị các tiền đề không chứa Churn. Chúng tôi lưu vectơ nhị phân này dưới dạng non.churn.ant.

Cuối cùng, chúng tôi chỉ tập hợp con từ all.rules những quy tắc có non.churn.ant bằng một. Nói cách khác, chúng tôi chỉ tập hợp con những quy tắc không có Churn trong tiền đề.

```
good.rules <- all.rules[non.churn.ant == 1]
```

Lưu các quy tắc kết quả là good.rules.

Chúng ta có thể xem good.rules được sắp xếp theo giá trị mức tăng giảm dần bằng cách sử dụng lệnh com yêu cầu kiểm tra() và head() một lần nữa.

```
kiểm tra (đầu (good.rules, by = "nâng", n = 28))
```

10 hàng đầu tiên của kết quả từ lệnh này được hiển thị trong Hình 14.3.

Để tạo bảng dự phòng cho các Cuộc gọi Dịch vụ Khách hàng và Khách hàng, trong đó sẽ được sử dụng bên dưới, hãy chạy đoạn mã sau:

```
t.csc.churn <- table(min.churn$Churn, min.churn$CustServ.
cuộc gọi)
colnames(t.csc.churn) <- c("CSC = 0", "CSC = 1", "CSC =
2", "CSC = 3", "CSC = 4",
        "CSC = 5", "CSC = 6", "CSC = 7", "CSC = 8", "CSC = 9")
rownames(t.csc.churn) <- c("Churn = Sai", "Churn = True")
addmargins(A = t.csc.churn, FUN = danh sách (Tổng cộng = tổng), yên tĩnh = TRUE)
```

Kết quả được cung cấp trong Hình 14.5.

	CSC = 0	CSC = 1	CSC = 2	CSC = 3	CSC = 4
Churn = False	540	960	602	343	81
Churn = True	86	108	77	40	68
Total	626	1068	679	383	149

	CSC = 5	CSC = 6	CSC = 7	CSC = 8	CSC = 9	Total
Churn = False	25	8	4	1	0	2564
Churn = True	36	14	4	1	2	436
Total	61	22	8	2	2	3000

Hình 14.5 Bảng dự phòng các cuộc gọi Dịch vụ Khách hàng và Khách hàng từ R.

Machine Translated by Google

14.5 XÁC NHẬN CÁC ĐO LƯỜNG CỦA CHÚNG TÔI

Chúng tôi sẽ gọi ID quy tắc [1] là "Quy tắc 1". Tiếp theo, chúng ta hãy xác nhận các giá trị sau cho Quy tắc 1, sử dụng những gì chúng ta đã học cho đến nay:

1. Hỗ trợ,
2. Tự tin,
3. Thang máy.

1. Hỗ trợ.

$$\text{s hỗ trợ P CSC và Churn True} = \left(= 5 \quad = \quad \right)$$

$$= \frac{\text{giao dịch với cả hai CSC} = 5 \text{ và Churn True}}{\text{tổng số giao dịch}} = \frac{36}{3000} = 12 \ . \ \%$$

Làm thế nào mà chúng tôi có được 36? Hỗ trợ yêu cầu giao điểm của hai sự kiện, có thể được tìm thấy bằng cách tạo bảng dự phòng các cuộc gọi dịch vụ khách hàng và rời bỏ, được hiển thị trong Hình 14.5. Lưu ý rằng ô cho CSC = 5 và Churn = True chứa Count = 36, chiếm 1,2% tổng số bản ghi.

2. Tự tin. Sử dụng bảng dự phòng để xác nhận rằng điều này bằng với xác suất có điều kiện P(B A).

$$\text{tự tin P Churn True CSC} = \left(\quad | \quad = 5 \right) = \frac{P\left(\text{Churn True và CSC} \quad = 5\right)}{P\left(\text{CSC} \quad = 5\right)}$$

$$= \frac{\text{số lượng giao dịch chứa cả Churn True thứ 5 của CSC} \quad =}{\text{số lượng giao dịch chứa CSC 5} \quad =}$$

Cả hai đại lượng này được tìm thấy trong Hình 14.5. Như vậy,

$$\text{tự tin P Churn True C|} \quad = \quad \text{5SC} = \quad \right) = \frac{36}{61} = 59\ 016 \ . \ \%$$

3. Thang máy. Diễn giải giá trị này.

$$\text{Thang máy} = \frac{\text{Quy tắc tự tin cho Quy tắc } 1}{\text{Tỷ lệ trước của Churn Tr} \quad =} = \text{uể oải}$$

Từ (2) ta có độ tin cậy = 59,016%. Từ Hình 14.1, chúng ta có tỷ lệ Churn = True trước đây là 14,53%. Vì thế,

$$\text{Thang máy} = \frac{\text{Độ tin cậy của quy tắc đối với ID quy tắc } 2}{\text{Tỷ lệ trước của Churn} \quad = \text{ĐÚNG VẬY}} = \frac{0\ .59016}{0\ .1453} = 4\ .061$$

Machine Translated by Google

Nói cách khác, những khách hàng đã thực hiện năm cuộc gọi đến dịch vụ khách hàng là khả năng rời bỏ cao gấp 4,061 lần so với nhóm khách hàng chung của chúng tôi.

14.6 TIÊU CHUẨN SỰ KHÁC BIỆT CỦA ĐỘ TIN CẬY

Các luật kết hợp ở trên được tạo ra bằng cách sử dụng rion crite độ tin cậy tối thiểu. Tuy nhiên, các tiêu chí khác tồn tại để tạo luật kết hợp. Tiếp theo, chúng tôi xem xét tiêu chí chênh lệch độ tin cậy. Thước đo đánh giá sự khác biệt về độ tin cậy đưa ra sự khác biệt tuyệt đối giữa xác suất trước của kết quả (ở đây, trạng thái rời bỏ) và độ tin cậy của quy tắc. Vì vậy, các quy tắc sẽ chỉ được bao gồm trong trường hợp này nếu:

$$\left|\text{Khả năng xảy ra trước p của quy tắc hệ quả Độ tin cậy 0}\right| \geq 40.$$

Hình 14.6 cho thấy luật kết hợp duy nhất được tạo bằng cách sử dụng giới hạn dưới của chênh lệch độ tin cậy là 40 (cùng với độ hỗ trợ tiền đề tối thiểu là 1%, độ tin cậy của quy tắc tối thiểu là 5% và tiền đề tối đa là 1).

Quy tắc kết hợp này có độ tin cậy Quy tắc 0,59016 và xác suất trước của Churn = True của 0,14533, mang lại cho chúng tôi:

$$\left|\text{Trước p cướp của c sau đó Quy tắc tự tin}\right|$$

$$= \left|0.14533\ 0.59016\ 0\ 44483\right| . \geq .\ 0\ 40$$

Thống kê khác biệt từ Hình 14.6 bằng với chênh lệch tuyệt đối này, 0,44483.

Các quy tắc có độ tin cậy tương tự như xác suất trước của hệ quả có thể không thú vị. Ví dụ: bằng cách chọn ngẫu nhiên một giao dịch từ dữ liệu, xác suất có được một khách hàng rời bỏ là 0,14533. Nếu quy tắc được tạo ra đưa ra thước đo độ tin cậy không xa 0,14533, thì lựa chọn ngẫu nhiên cũng có thể được sử dụng. Do đó, sự khác biệt về độ tin cậy có thể chỉ ra các quy tắc đi chệch khỏi lựa chọn ngẫu nhiên. Trong trường hợp ở đây, có thể thấy rằng những khách hàng đã thực hiện năm cuộc gọi dịch vụ khách hàng có tỷ lệ rời bỏ xuất hiện khá khác với những khách hàng ngẫu nhiên được chọn từ dữ liệu. Thước đo sự khác biệt về độ tin cậy giúp loại bỏ các quy tắc hiển nhiên, chẳng hạn như "Nếu mang thai, thì là nữ". Nó cũng giải thích cho sự phân phối sai lệch hoặc không đồng đều.

14.6.1 Cách áp dụng tiêu chí chênh lệch độ tin cậy bằng R

Để bao gồm tiêu chí khác nhau về độ tin cậy trong cài đặt quy tắc kết hợp của chúng tôi, chúng tôi quay lại lệnh apriori() và thêm các giá trị đầu vào bổ sung.

```
    lhs                  rhs          support confidence diff      lift     count
[1] {CustServ.Calls=5} => {Churn=True} 0.012   0.5901639  0.4448306 4.060761 36
```

Hình 14.6 Tìm thấy luật kết hợp sử dụng chênh lệch độ tin cậy cận dưới = 40.

Machine Translated by Google


```
rules.confdiff <- apriori(data = min.churn, tham số = danh sách(arem = "diff", aval
= TRUE,
          minval = 0,4, supp = 0,01, target = "quy tắc", conf = 0,05, minlen = 2,
          maxlen = 2))
```

Lưu ý ba cài đặt đầu vào mới trong tham số = list(). Cái đầu tiên, arem = "diff", chỉ định rằng nên sử dụng tiêu chí chênh lệch độ tin cậy.
Thứ hai, aval = TRUE, nói rằng giá trị của tiêu chí phải được báo cáo khi kết quả được hiển thị. Thứ ba, minval = 0,4, đặt giới hạn dưới của chênh lệch độ tin cậy là 40. Lưu đầu ra dưới dạng rules.confdiff.

Để xem các quy tắc mới, lần này hãy sử dụng lệnh kiểm tra() và lệnh head() sử dụng rules.confdiff làm đầu vào chính.

```
kiểm tra (đầu (rules.confdiff, by = "nâng", n = 10))
```

14.7 TIÊU CHUẨN CHỈ SỐ TIN CẬY

Để chứng minh tiêu chí chỉ số tin cậy, chúng tôi tạo quy tắc sử dụng giới hạn dưới là 40 (cùng với hỗ trợ tiền đề tối thiểu là 1%, độ tin cậy quy tắc tối thiểu là 5% và tiền đề tối đa là 1). Sau khi loại bỏ bất kỳ luật nào có Churn ở tiền đề, chúng ta thu được ba luật kết hợp như trong Hình 14.7.

Thước đo đánh giá chỉ số tin cậy đưa ra tỷ lệ tuyệt đối giữa xác suất trước của hệ quả (Churn = True) và độ tin cậy của quy tắc. Vì vậy, các quy tắc sẽ chỉ được bao gồm trong trường hợp này nếu:

$$\begin{cases} 1 - \dfrac{\text{Quy tắc tự tin}}{\text{Tỷ lệ trước của hệ quả}} \geq 0.40 \\ \text{hoặc} \\ 1 - \dfrac{\text{pri hoặc tỷ lệ hệ quả}}{\text{Quy tắc tự tin}} \geq 0.40 \end{cases}$$

nào không âm.

```
    lhs                  rhs          support    confidence quot      lift     count
[1] {CustServ.Calls=5} => {Churn=True} 0.01200000 0.5901639  0.7537407 4.060761  36
[2] {CustServ.Calls=4} => {Churn=True} 0.02266667 0.4563758  0.6815490 3.140201  68
[3] {Intl.Plan=yes}    => {Churn=True} 0.04233333 0.4305085  0.6624147 2.962214 127
```

Hình 14.7 Luật kết hợp trong R được tìm thấy bằng cách sử dụng giới hạn dưới của thương số tin cậy = 40.

3Đầu ra sẽ có các quy tắc có Churn ở phần trước. Sau khi chỉ đặt lại các quy tắc đó mà không có Churn trong phần trước, như đã trình bày trong phần R trước đó, đầu ra sẽ khớp với kết quả được hiển thị trong Hình 14.6.

Machine Translated by Google

Hãy để chúng tôi xác nhận các tính toán cho Quy tắc [3] từ Hình 14.5. Quy tắc này có độ tin cậy 0,43051. Từ Hình 14.1, tỷ lệ trước của hệ quả (Churn = True) bằng 0,14533. Như vậy,

$$1 - \frac{\text{Tỷ lệ trước của hậu quả}}{\text{Quy tắc tự tin}} = 1 - \frac{0.14533}{0.43051} = 0.66242 \geq 0.40 .$$

Cho phép sai số làm tròn, thống kê quot từ Hình 14.7 bằng với giá trị chúng tôi thu được ở trên, 0,66242.

Giống như Sự khác biệt về Niềm tin, phương pháp này tính đến sự phân bố không đồng đều. Nó đặc biệt giỏi trong việc tìm ra các quy luật dự đoán các sự kiện hiếm gặp. Tất nhiên, giống như bất kỳ nhiệm vụ khoa học dữ liệu nào khác sử dụng xác thực chéo, việc khai thác quy tắc kết hợp cần phải được xác thực. Chúng tôi cho thấy làm thế nào để làm điều này trong các bài tập.

14.7.1 Cách áp dụng tiêu chí chỉ số tin cậy bằng R

Để bao gồm tiêu chí khác nhau về độ tin cậy trong cài đặt quy tắc kết hợp của chúng tôi, chúng tôi quay lại lệnh apriori() và thay đổi các giá trị đầu vào.

```
rules.confquot <- apriori(data = min.churn, tham số = danh sách(arem = "quot", aval
= TRUE,
          minval = 0,4, supp = 0,01, target = "quy tắc", conf = 0,05, minlen = 2,
maxlen = 2))
kiểm tra (đầu (rules.confquot, by = "nâng", n = 10))
```

Đầu ra của đoạn mã trên, không được hiển thị, bao gồm các quy tắc với Churn trong tiền đề.

Bước tiếp theo của chúng tôi là tập hợp con từ các quy tắc mà chúng tôi đã thu được chỉ những quy tắc không có Churn trong tiền đề. Chúng tôi làm theo các bước chung giống như trong Phần 14.4.1.

```
rules.confquot.ant.df <- as(as(attr(rules.confquot, "lhs"), "giao dịch"),
"data.frame")
t1 <-rules.confquot.ant.df$items == "{Churn=True}"
t2 <-rules.confquot.ant.df$items == "{Churn=False}"
non.churn.ant <-abs(t1+t2-1)
good.rules.confquot <- rules.confquot[non.churn.ant == 1]
kiểm tra (good.rules.confquot)
```

Kết quả được thể hiện trong hình 14.7.

GIÁ TRỊ

Các tác giả xin cảm ơn bạn đã tham gia cùng chúng tôi khi chúng tôi tìm hiểu về Khoa học dữ liệu bằng Python và R. Bạn nên dành một chút thời gian để thưởng thức thành quả của mình! Hãy xem bạn đã học được bao nhiêu! Chúng tôi chúc bạn mọi điều tốt đẹp nhất khi tiếp tục cuộc hành trình của mình trong cuộc sống và thông qua khoa học dữ liệu.

Chantal Larose
Daniel Larose

Machine Translated by Google

NGƯỜI GIỚI THIỆU

Công việc của chúng tôi với các luật kết hợp chủ yếu dựa vào gói arules, có các trích dẫn sau: Michael Hahsler, Christian Buchta, Bettina Gruen, và Kurt Hornik, arules: Mining Association Rules và Frequent Itemset. Phiên bản gói R 1.6-1, 2018. https://CRAN.R-project.org/package=arules

Michael Hahsler, Bettina Gruen, và Kurt Hornik, arules – A Computational Environment for Mining Association Rules and Frequent Item Sets, Tạp chí Phần mềm Thống kê, 14(15), 2005. doi:https://doi.org/10.18637/jss.v014.i15.

Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, và Christian Buchta, The arules R-hệ sinh thái gói: phân tích các mẫu thú vị từ bộ dữ liệu giao dịch lớn, Tạp chí Nghiên cứu Máy học, 12, 1977–1981, 2011.

BÀI TẬP

LÀM RÕ CÁC KHÁI NIỆM

1. Làm thế nào để lời nguyền về chiều không gian khiến việc khám phá sự liên kết trở thành một thách thức quy tắc?
2. Luật kết hợp có dạng gì?
3. Bằng cách diễn đạt của bạn, hãy giải thích ý nghĩa của độ hỗ trợ cho luật kết hợp.
4. Độ tin cậy của một luật tương đương với xác suất nào?
5. Giải thích ý nghĩa của thang máy.
6. Tiêu chí khác biệt về độ tin cậy là gì và tại sao nó được sử dụng?
7. Mô tả tiêu chí tỷ lệ tin cậy.

   Sử dụng bảng 14.1 để trả lời các câu hỏi sau.

8. Tính độ hỗ trợ cho quy tắc “Nếu là ngô thì là cà chua”.
9. Tìm độ tin cậy cho quy tắc “Nếu là ngô thì là cà chua”.
10. Tính toán lực nâng cho quy tắc “Nếu là ngô thì là cà chua”.

LÀM VIỆC VỚI DỮ LIỆU

Đối với các bài tập sau, hãy làm việc với tập dữ liệu Churn_Training_File . Sử dụng R để giải quyết từng vấn đề.

11. Tập hợp các biến VMail Plan, Int'l Plan, CustServ Calls, và Churn into own khung dữ liệu. Thay đổi Cuộc gọi CustServ thành một yếu tố có thứ tự.
12. Tạo các bảng cho từng biến trong bốn biến. Bao gồm cả số lượng và tỷ lệ trong mỗi bảng. Sử dụng các bảng để thảo luận về phân phối "đường cơ sở" của từng biến.
13. Lấy các luật kết hợp bằng cách sử dụng các cài đặt được nêu trong Phần 14.4.
14. Đặt con các quy tắc từ bài tập trước để không có tiền đề nào chứa Churn Biến đổi. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

Machine Translated by Google

15. Có được các luật kết hợp bằng cách sử dụng tiêu chí chênh lệch độ tin cậy được nêu trong Mục 14.6.

16. Xác nhận bằng tay rằng giá trị của tiêu chí khác nhau về độ tin cậy đối với quy tắc đầu tiên trong bài tập trước là đúng.

17. Thu được các luật kết hợp bằng cách sử dụng tiêu chí thương số tin cậy được nêu trong Mục 14.7. Chỉ tập hợp con những quy tắc đó mà không có Churn trong tiền đề.

18. Xác nhận bằng tay rằng giá trị của tiêu chí thương số tin cậy cho quy tắc đầu tiên trong bài tập trước là đúng.

PHÂN TÍCH THỰC TẾ

Đối với phần sau, hãy làm việc với tập dữ liệu Người lớn . Sử dụng R để giải quyết từng vấn đề.

19. Đặt các biến giáo dục, hôn nhân.status và thu nhập vào khung dữ liệu của riêng chúng.

20. Tạo bảng cho từng biến. Bao gồm cả số lượng và tỷ lệ trong mỗi bảng. Các bảng này sẽ được sử dụng để có được tỷ lệ trước của các giá trị khác nhau.

21. Thu được các luật kết hợp sử dụng độ hỗ trợ tối thiểu là 2%, độ tin cậy tối thiểu là 50% và tiền đề tối đa là 1.

22. Tập hợp con các luật từ bài tập trước sao cho không có tiền đề nào chứa biến thu nhập . Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

23. Có được các luật kết hợp bằng cách sử dụng tiêu chí chênh lệch độ tin cậy, với giới hạn dưới của chênh lệch độ tin cậy là 30, độ hỗ trợ tiền đề tối thiểu là 2%, độ tin cậy của luật tối thiểu là 50% và tiền đề tối đa là 1.

24. Tập hợp con các quy tắc từ bài tập trước để không có tiền đề nào chứa biến thu nhập . Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

25. Xác nhận bằng tay rằng giá trị của tiêu chí khác nhau về độ tin cậy đối với quy tắc đầu tiên trong bài tập trước là đúng.

26. Lấy các luật kết hợp sử dụng tiêu chí thương số tin cậy, với giới hạn dưới của đại lượng tin cậy là 30, độ hỗ trợ tiền đề tối thiểu là 2%, độ tin cậy của luật tối thiểu là 50% và tiền đề tối đa là 1.

27. Tập hợp con các quy tắc từ bài tập trước sao cho không có tiền đề nào chứa biến thu nhập . Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

28. Xác nhận bằng tay rằng giá trị của tiêu chí thương số tin cậy cho quy tắc đầu tiên trong bài tập trước là đúng.

Đối với các bài tập sau, hãy sử dụng bộ dữ liệu AR_Training và AR_Test . Phản hồi là biến mục tiêu, vì vậy chỉ xem xét các quy tắc trong đó nó là hệ quả duy nhất có thể xảy ra. Các biến khác là yếu tố dự đoán, vì vậy chỉ xem xét các quy tắc trong đó yếu tố dự đoán là tiền đề.
Sử dụng tập huấn luyện cho đến khi có thông báo khác.

29. Tạo bảng cho từng biến. Bao gồm cả số lượng và tỷ lệ trong mỗi bảng. Các bảng này sẽ được sử dụng để có được tỷ lệ trước của các giá trị khác nhau.

Machine Translated by Google

30. Tạo quy tắc kết hợp bằng cách sử dụng độ hỗ trợ tối thiểu là 5%, độ tin cậy tối thiểu là 5% và tiền đề tối đa là 1. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

31. Chọn quy tắc từ bài tập trước với lực nâng lớn nhất. Giải thích giá trị mức tăng này cho người không quen thuộc với khoa học dữ liệu.

32. Tiếp tục với quy tắc kết hợp từ bài tập trước. Tìm bất kỳ tỷ lệ nào trước đó và xây dựng bất kỳ bảng dự phòng nào bạn cần để xác nhận bằng tay các giá trị bạn thu được cho các đại lượng sau:

    Một. Ủng hộ

    b. Sự tự tin

    c. Thang máy

33. Tạo quy tắc kết hợp bằng cách sử dụng độ hỗ trợ tối thiểu là 5%, độ tin cậy tối thiểu là 5% và tiền đề tối đa là 2. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

34. Chọn quy tắc từ bài tập trước với lực nâng lớn nhất. So sánh quy tắc này với quy tắc nâng cao nhất cho tiền đề = 1.

    Một. Quy tắc nào có mức tăng tốt hơn?

    b. Quy tắc nào có sự hỗ trợ lớn hơn?

    c. Nếu bạn là người quản lý tiếp thị và chỉ có thể tài trợ cho một trong những quy tắc này, nó sẽ được, và tại sao?

35. Lấy các luật kết hợp bằng cách sử dụng tiêu chí chênh lệch độ tin cậy, với giới hạn dưới của chênh lệch độ tin cậy là 30, độ hỗ trợ tối thiểu là 5%, độ tin cậy tối thiểu là 5% và tiền đề tối đa là 1. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

36. Chọn quy tắc từ bài tập trước với lực nâng lớn nhất. Xác nhận bằng tay giá trị chênh lệch độ tin cậy.

37. Nhận các luật kết hợp bằng cách sử dụng tiêu chí chênh lệch độ tin cậy, với giới hạn dưới của chênh lệch độ tin cậy là 10, độ hỗ trợ tối thiểu là 5%, độ tin cậy tối thiểu là 5% và tiền đề tối đa là 2. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

38. Chọn quy tắc từ bài tập trước với lực nâng lớn thứ hai. So sánh quy tắc này với quy tắc nâng cao nhất cho tiền đề = 1.

    Một. Quy tắc nào có mức tăng tốt hơn?

    b. Quy tắc nào có sự hỗ trợ lớn hơn?

    c. Nếu bạn là người quản lý tiếp thị và chỉ có thể tài trợ cho một trong những quy tắc này, nó sẽ được, và tại sao?

39. Lấy các luật kết hợp bằng cách sử dụng tiêu chí thương số tin cậy, với giới hạn dưới của thương số tin cậy là 30, độ hỗ trợ tiền đề tối thiểu là 5%, độ tin cậy của luật tối thiểu là 5% và tiền đề tối đa là 3. Hiển thị các luật, được sắp xếp theo giá trị thang máy giảm dần .

40. Chọn quy tắc từ bài tập trước với lực nâng lớn nhất. Xác nhận bằng tay giá trị cho thương số tin cậy.

Machine Translated by Google

Đối với tập Bài tập tiếp theo, chúng ta sẽ kiểm tra các luật kết hợp mà chúng ta đã tìm thấy trước đó. Sử dụng bộ dữ liệu AR_Test .

41. Tạo bảng cho từng biến. Bao gồm cả số lượng và tỷ lệ trong mỗi bảng. Các bảng này sẽ được sử dụng để có được tỷ lệ trước của các giá trị khác nhau.

42. Tạo quy tắc kết hợp bằng cách sử dụng độ hỗ trợ tối thiểu là 5%, độ tin cậy tối thiểu là 5% và tiền đề tối đa là 1. Hiển thị các quy tắc, được sắp xếp theo giá trị thang máy giảm dần.

43. So sánh các luật bạn thu được trong bài tập trước với các luật bạn thu được bằng cách sử dụng cùng tiêu chí từ tập dữ liệu huấn luyện. Bạn sẽ nói rằng các quy tắc kết hợp của chúng tôi đã được xác thực?

Machine Translated by Google

TÓM TẮT DỮ LIỆU VÀ HÌNH ẢNH HÓA

Ở đây chúng tôi trình bày một đánh giá rất ngắn gọn về các phương pháp tóm tắt và trực quan hóa dữ liệu. Để có thông tin chi tiết hơn, vui lòng xem Khám phá số liệu thống kê của Daniel T. Larose (WH Freeman, ấn bản thứ hai, 2013).

PHẦN 1: TỔNG KẾT PHẦN 1: CÁC KHỐI XÂY DỰNG PHÂN TÍCH DỮ LIỆU

- Thống kê mô tả đề cập đến các phương pháp tóm tắt và sắp xếp các thông tin trong một tập dữ liệu.

  Xem xét Bảng A.1, chúng tôi sẽ sử dụng bảng này để minh họa một số khái niệm thống kê.
- Các thực thể mà thông tin được thu thập được gọi là các phần tử. TRONG Bảng A.1, các yếu tố là 10 ứng viên. Các phần tử còn được gọi là các trường hợp hoặc các môn học.
- Biến là một đặc trưng của một phần tử, nó nhận các giá trị khác nhau đối với các phần tử khác nhau . Các biến số trong Bảng A.1 là tình trạng hôn nhân, thế chấp, thu nhập, cấp bậc, năm và rủi ro. Biến còn được gọi là thuộc tính.
- Tập hợp các giá trị biến cho một phần tử cụ thể là một quan sát. Các quan sát còn được gọi là hồ sơ. Quan sát đối với Người nộp đơn 2 là:

Người nộp đơn	Tình trạng hôn nhân	Thu nhập thế chấp ($)	Thu nhập	Xếp hạng	Năm	Rủi ro
2	Đã cưới	Đúng	32.000	7	2010	Tốt

- Các biến có thể là định tính hoặc định lượng.
  - Một biến định tính cho phép các phần tử được phân loại hoặc phân loại theo một số đặc tính. Các biến định tính trong Bảng A.1 là tình trạng hôn nhân, thế chấp, cấp bậc và rủi ro. Biến định tính còn được gọi là biến phân loại.

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose.
© 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

Bảng A.1 Đặc điểm của 10 người xin vay

Người xin việc	Tình trạng hôn nhân	Thu nhập thế chấp ($)		Thứ hạng thu nhập	Năm	Rủi ro
1	Đơn	Đúng	38.000	2	2009	Tốt
2	Đã cưới	Đúng	32.000	7	2010	Tốt
3	Khác	KHÔNG	25.000	9	2011	Tốt
4	Khác	KHÔNG	36.000	3	2009	Tốt
5	Khác	Đúng	33.000	4	2010	Tốt
6	Khác	KHÔNG	24.000	10	2008	Xấu
7	Đã cưới	Đúng	25.100	...	2010	Tốt
...	Đã cưới	Đúng	48.000	1	2007	Tốt
9	Đã cưới	Đúng	32.100	6	2009	Xấu
10	Đã cưới	Đúng	32.200	5	2010	Tốt

- Một biến định lượng nhận các giá trị số và cho phép thực hiện phép tính số học một cách có ý nghĩa trên nó. Các biến định lượng trong Bảng A.1 là thu nhập và năm. Biến định lượng còn được gọi là biến số.

- Dữ liệu có thể được phân loại theo bốn cấp độ đo lường: danh nghĩa, thứ tự, khoảng và tỷ lệ. Dữ liệu danh nghĩa và thứ tự là phân loại; dữ liệu khoảng và tỷ lệ là số.

  - Dữ liệu danh nghĩa đề cập đến tên, nhãn hoặc danh mục. Không có thứ tự tự nhiên, cũng như không thể thực hiện phép tính số học trên dữ liệu danh nghĩa. Các biến danh nghĩa trong Bảng A.1 là tình trạng hôn nhân, thế chấp và rủi ro.
  - Dữ liệu thứ tự có thể được kết xuất thành một thứ tự cụ thể. Tuy nhiên, số học không thể được thực hiện một cách có ý nghĩa trên dữ liệu thứ tự. Biến thứ tự trong Bảng A.1 là thứ hạng thu nhập.
  - Dữ liệu khoảng bao gồm dữ liệu định lượng được xác định trên một khoảng không có số 0 tự nhiên. Phép cộng và phép trừ có thể được thực hiện trên dữ liệu khoảng. Biến khoảng thời gian trong Bảng A.1 là năm. (Lưu ý rằng không có "năm 0". Lịch đi từ 1BCE đến 1CE)
  - Dữ liệu tỷ lệ là dữ liệu định lượng có thể thực hiện phép cộng, phép trừ, phép nhân và phép chia. Số 0 tự nhiên tồn tại đối với dữ liệu tỷ lệ. Biến khoảng trong Bảng A.1 là thu nhập.

- Một biến số có thể nhận một số giá trị hữu hạn hoặc có thể đếm được là một biến rời rạc , trong đó mỗi giá trị có thể được biểu thị dưới dạng một điểm tốc độ riêng biệt, với khoảng cách giữa mỗi điểm. Biến rời rạc trong Bảng A.1 là năm.

- Một biến số có thể nhận vô số giá trị là một biến liên tục, các giá trị có thể của nó tạo thành một khoảng trên trục số, không có khoảng cách giữa các điểm. Biến liên tục trong Bảng A.1 là thu nhập.

Machine Translated by Google

- Tổng thể là tập hợp tất cả các phần tử quan tâm cho một vấn đề cụ thể. Một tham số là một đặc tính của dân số. Ví dụ: dân số là tập hợp tất cả các cử tri Mỹ và tham số là tỷ lệ dân số ủng hộ mức thuế carbon 1 USD/tấn.

  - Giá trị của một tham số thường không xác định, nhưng nó là một hằng số.

- Một mẫu bao gồm một tập hợp con của tổng thể. Một đặc trưng của một mẫu được gọi là một thống kê. Ví dụ: mẫu là tập hợp các cử tri Mỹ trong lớp học của bạn và thống kê là tỷ lệ mẫu ủng hộ mức thuế carbon 1 USD/tấn.

  - Giá trị của một thống kê thường được biết trước, nhưng nó thay đổi từ mẫu này sang mẫu khác.

- Điều tra dân số là việc thu thập thông tin từ mọi thành phần trong dân số. Ví dụ, cuộc điều tra dân số ở đây sẽ là tìm kiếm từ mọi cử tri Mỹ xem họ có ủng hộ mức thuế carbon 1 USD/tấn hay không. Điều tra dân số như vậy là không thực tế, vì vậy chúng tôi chuyển sang suy luận thống kê.

- Suy luận thống kê đề cập đến các phương pháp ước tính hoặc rút ra kết luận về các đặc điểm của dân số dựa trên các đặc điểm của một mẫu của dân số đó. Ví dụ: giả sử 50% cử tri trong lớp học của bạn ủng hộ thuế; sử dụng suy luận thống kê, chúng tôi sẽ suy luận rằng 50% tổng số cử tri Mỹ ủng hộ thuế. Rõ ràng, có vấn đề với điều này. Mẫu không ngẫu nhiên cũng không đại diện. Ước tính không có mức độ tin cậy, v.v.

- Khi chúng ta lấy một mẫu mà mỗi phần tử có cơ hội được chọn như nhau, chúng ta có một mẫu ngẫu nhiên.
- Biến dự đoán là một biến có giá trị được sử dụng để giúp dự đoán giá trị của biến phản hồi. Các biến dự đoán trong Bảng A.1 là tất cả các biến ngoại trừ rủi ro.

- Một biến phản hồi là một biến quan tâm có giá trị có lẽ được xác định ít nhất một phần bởi tập hợp các biến dự đoán. Biến phản hồi có thể có trong Bảng A.1 là rủi ro.

PHẦN 2: HÌNH ẢNH: BIỂU ĐỒ VÀ BẢNG
ĐỂ TỔNG HỢP VÀ TỔ CHỨC DỮ LIỆU

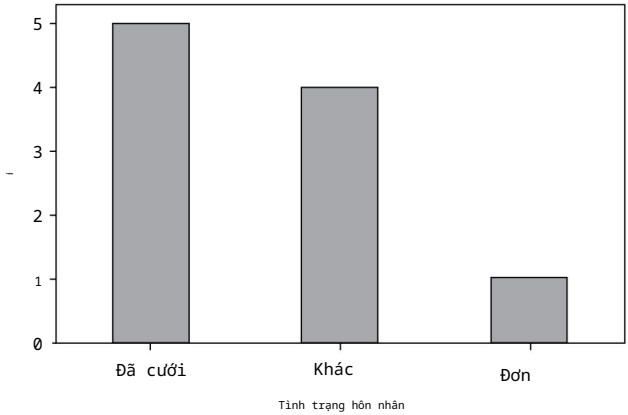
A.1 Biến phân loại

- Tần suất (hoặc số lượng) của một danh mục là số lượng giá trị dữ liệu trong mỗi danh mục. Tần suất tương đối của một danh mục cụ thể đối với một biến phân loại bằng tần số của nó chia cho số trường hợp.
- Phân bố tần suất (tương đối) cho một biến phân loại bao gồm tất cả các loại mà biến đó giả định, cùng với tần suất (tương đối) cho mỗi giá trị. Các tần số tổng hợp với số lượng các trường hợp; tổng các tần số tương đối bằng 1.

Machine Translated by Google

Bảng A.2 Phân bố tần số và phân bố tần số tương đối

Hạng mục Tình trạng hôn nhân	Tần thường xuyên	Tần số tương đối
Đã cưới	5	0,5
Khác	4	0,4
Đơn	1	0,1
Tổng cộng	10	1.0



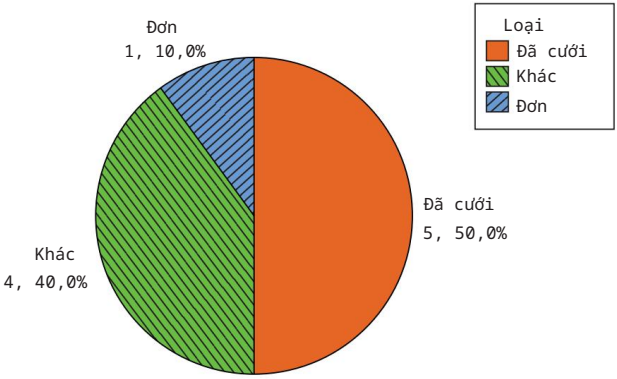
Hình A.1 Biểu đồ thanh cho tình trạng hôn nhân.

- Ví dụ, Bảng A.2 chứa phân bố tần suất và phân bố tần số tương đối cho tình trạng hôn nhân thay đổi đối với dữ liệu từ Bảng A.1.
- Biểu đồ thanh là biểu đồ được sử dụng để biểu thị tần số hoặc tần số tương đối cho một biến phân loại. Lưu ý rằng các thanh không chạm vào nhau.
  - Biểu đồ Pareto là biểu đồ thanh trong đó các thanh được sắp xếp theo thứ tự giảm dần. Hình A.1 là một ví dụ về biểu đồ Pareto.
- Biểu đồ hình tròn là một vòng tròn được chia thành các lát cắt, với kích thước của mỗi lát cắt tỷ lệ thuận với tần suất tương đối của danh mục được liên kết với lát cắt đó. Hình A.2 thể hiện một biểu đồ hình tròn về tình trạng hôn nhân.

A.2 Biến định lượng

- Dữ liệu định lượng được nhóm thành các lớp. Giới hạn lớp dưới (trên) của một lớp bằng giá trị nhỏ nhất (lớn nhất) trong lớp đó . Chiều rộng lớp là sự khác biệt giữa các giới hạn lớp dưới liên tiếp.
- Đối với dữ liệu định lượng, phân phối tần suất (tương đối) chia dữ liệu thành các lớp không chồng lấp có độ rộng lớp bằng nhau. Bảng A.3 cho thấy

Machine Translated by Google



Hình A.2 Biểu đồ hình tròn về tình trạng hôn nhân.

Bảng A.3 Phân bố tần suất và phân bố tần suất tương đối của thu nhập

Loại thu nhập ($)	Tính thường xuyên	Tần số tương đối
24.000–29.999	3	0,3
30.000–35.999	4	0,4
36.000–41.999	2	0,2
42.000–48.999	1	0,1
Tổng cộng	10	1.0

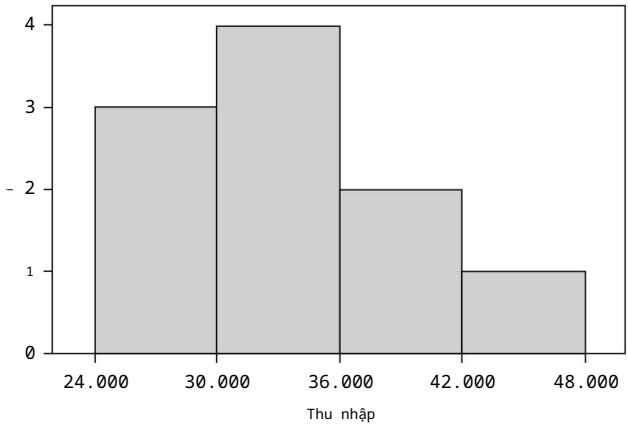
Bảng A.4 Phân bố tần suất tích lũy và phân bố tần suất tương đối tích lũy của thu nhập

Loại thu nhập ($)	Tần suất tích lũy	Tần suất tương đối tích lũy
24.000–29.999	3	0,3
30.000–35.999	7	0,7
36.000–41.999	9	0,9
42.000–48.999	10	1.0

phân bố tần suất và phân bố tần suất tương đối của thu nhập biến đổi liên tục từ Bảng A.1.

- Phân phối tần suất tích lũy (tương đối) cho biết tổng số (tần suất tương đối) của các giá trị dữ liệu nhỏ hơn hoặc bằng giới hạn loại trên (Bảng A.4).

Machine Translated by Google



Hình A.3 Biểu đồ thu nhập.

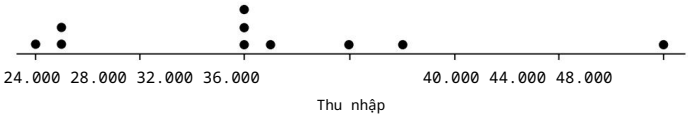
- Phân phối của một biến là một biểu đồ, bảng hoặc công thức chỉ định các giá trị và tần số của biến cho tất cả các phần tử trong tập dữ liệu. Ví dụ, Bảng A.3 thể hiện sự phân phối của thu nhập khả biến.
- Biểu đồ tần số là biểu diễn đồ họa của phân bố tần số (tương đối) cho một biến định lượng (Hình A.3). Lưu ý rằng các biểu đồ biểu diễn một phiên bản đơn giản của việc làm mịn dữ liệu và do đó có thể thay đổi hình dạng tùy thuộc vào số lượng và độ rộng của các lớp. Do đó, biểu đồ nên được giải thích một cách thận trọng. Xem Khám phá thống kê của Daniel T. Larose (WH Freeman) Phần 2.4 để biết ví dụ về tập dữ liệu được trình bày dưới dạng cả hai đối xứng và lệch phải bằng cách thay đổi số lượng và độ rộng của các lớp biểu đồ.
- Màn hình gốc và lá hiển thị hình dạng của phân phối dữ liệu trong khi vẫn giữ lại các giá trị dữ liệu gốc trong màn hình, chính xác hoặc gần đúng. Đơn vị lá được xác định bằng lũy thừa 10 và đơn vị thân gấp 10 lần đơn vị lá. Sau đó, mỗi lá đại diện cho một giá trị dữ liệu, thông qua sự kết hợp giữa thân và lá. Ví dụ, trong Hình A.4, đơn vị lá (cột bên phải) là 1000 giây và đơn vị gốc (cột bên trái) là 10.000 giây. Vì vậy, "2 4" đại diện cho 2×10.000+4×1000 = 24.000 đô la, trong khi "2 55" đại diện cho hai khoản thu nhập bằng nhau là 25.000 đô la (một trong số đó là chính xác, còn lại là gần đúng – 25.100 đô la). Lưu ý rằng Hình A.4, quay 90° sang trái, thể hiện hình dạng của phân bố dữ liệu.
- Trong một dotplot, mỗi dấu chấm đại diện cho một hoặc nhiều giá trị dữ liệu, được đặt phía trên trục số (Hình A.5).
- Một phân bố là đối xứng nếu tồn tại một trục đối xứng (một đường thẳng) chia phân bố thành hai nửa gần như là ảnh phản chiếu của nhau (Hình A.6a).

Machine Translated by Google

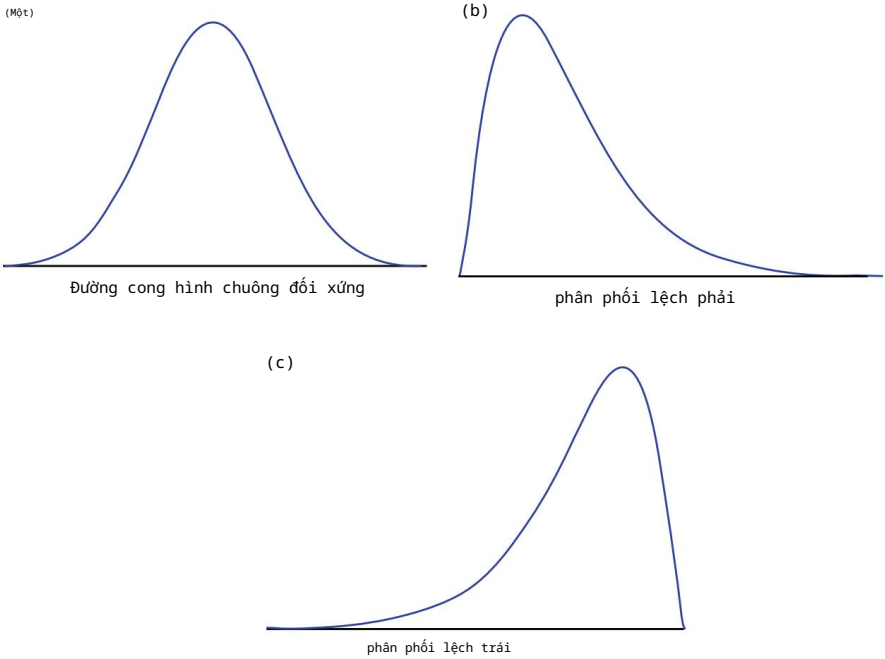

```
Stem-and-leaf of Income
Leaf Unit = 1000

     2  4
     2  55
     3  2223
     3  68
     4
     4  8
```

Hình A.4 Hiển thị thu nhập từ thân và lá .



Hình A.5 Biểu đồ thu nhập.



Hình A.6 (a) Đường cong hình chuông đối xứng. (b) Phân phối lệch phải. (c) Phân phối lệch trái.

Machine Translated by Google

- Dữ liệu lệch phải có đuôi bên phải dài hơn bên trái (Hình A.6b).
  Dữ liệu lệch trái có đuôi bên trái dài hơn bên phải (Hình A.6c).

PHẦN 3: TỔNG KẾT 2: BIỆN PHÁP TRUNG TÂM, TÍNH BIẾN ĐỔI VÀ VỊ TRÍ

- Ký hiệu tổng x có nghĩa là cộng tất cả các giá trị dữ liệu x. Cỡ mẫu là n và cỡ dân số là N.
- Số đo tâm cho biết phần trung tâm của dữ liệu nằm ở đâu trên trục số. Các biện pháp của trung tâm chúng ta sẽ tìm hiểu là giá trị trung bình, trung vị, chế độ và tầm trung.
  - Giá trị trung bình là trung bình cộng của một tập dữ liệu. Để tính giá trị trung bình, hãy cộng các giá trị và chia cho số lượng giá trị. Thu nhập trung bình từ Bảng A.1 là:

$$\frac{38{,}000 + 32{,}000 + \ldots + 32{,}200}{10} = \frac{325{,}400}{10} = \$32{,}540$$

  - Trung bình mẫu là trung bình cộng của một mẫu và được ký hiệu là $\bar{x}$ ("x-bar").
  - Trung bình dân số là trung bình số học của một dân số và là ký hiệu là μ ("myu," chữ cái Hy Lạp cho m).
  - Trung vị là giá trị dữ liệu ở giữa, khi có một số lẻ các giá trị dữ liệu và dữ liệu đã được sắp xếp theo thứ tự tăng dần. Nếu có một số chẵn, trung vị là giá trị trung bình của hai giá trị dữ liệu ở giữa.
    Khi dữ liệu thu nhập được sắp xếp theo thứ tự tăng dần, hai giá trị ở giữa là $32.100 và $32.200, giá trị trung bình là thu nhập trung bình, $32.150.
  - Chế độ là giá trị dữ liệu xảy ra với tần suất lớn nhất. Cả hai biến định lượng và phân loại đều có thể có các chế độ, nhưng chỉ các biến định lượng mới có thể có phương tiện hoặc trung vị. Mỗi giá trị thu nhập chỉ xảy ra một lần, vì vậy không có chế độ. Chế độ cho năm là 2010, với tần suất là 4.
  - Dải trung bình là giá trị trung bình của các giá trị lớn nhất và nhỏ nhất trong một tập dữ liệu. Thu nhập tầm trung là

$$\text{thu nhập tầm trung} = \frac{(\text{tối đa}(\text{thu nhập}) + \text{tối thiểu}(\text{thu nhập}))}{2} = \frac{48{,}000 + 24{,}000}{2} = \$36{,}000$$

- Skewness và các biện pháp của trung tâm. Sau đây là những xu hướng, và không quy tắc nghiêm ngặt.

Machine Translated by Google

- Đối với dữ liệu đối xứng, giá trị trung bình và trung vị xấp xỉ bằng nhau.
- Đối với dữ liệu lệch phải, giá trị trung bình lớn hơn giá trị trung vị.
- Đối với dữ liệu lệch trái, trung vị lớn hơn trung bình.

• Các thước đo về tính biến đổi định lượng mức độ biến đổi, độ lan truyền hoặc độ phân tán có trong dữ liệu. Các phép đo độ biến thiên mà chúng ta sẽ học là phạm vi, phương sai, độ lệch chuẩn và sau đó là phạm vi liên vùng

(chỉ số thông minh).

- Phạm vi của một biến bằng hiệu giữa giá trị lớn nhất và giá trị nhỏ nhất. Phạm vi thu nhập là: phạm vi = max(thu nhập)
  tối thiểu(thu nhập) = 48.000 24.000 = $24.000.
- Độ lệch là sự khác biệt có dấu giữa giá trị dữ liệu và giá trị trung bình. Đối với Người đăng ký 1, độ lệch trong thu nhập bằng xx 38 000 32 540 5460 Đối với bất kỳ tập dữ liệu có thể hiểu được nào, độ lệch trung bình luôn bằng 0 vì tổng các độ lệch bằng 0.
- Phương sai tổng thể là giá trị trung bình của các độ lệch bình phương, được ký hiệu là σ2 (“sigma-squared”):

$$\sigma^2 = \frac{\sum (x - \mu)^2}{N}$$

- Độ lệch chuẩn của tổng thể là căn bậc hai của tổng thể phương sai: $\sigma = \sqrt{\sigma^2}$.
- Phương sai mẫu xấp xỉ giá trị trung bình của các độ lệch bình phương, với n được thay thế bằng n – 1 ở mẫu số để làm cho nó trở thành một ước lượng không chệch của σ2 . (Công cụ ước tính không chệch là một thống kê có giá trị kỳ vọng bằng tham số mục tiêu của nó.)

$$s^2 = \frac{\sum (x - \bar{x})^2}{N - 1}$$

- Độ lệch chuẩn mẫu là căn bậc hai của phương sai mẫu:
  $s = \sqrt{s2}$.
- Phương sai được biểu thị bằng đơn vị bình phương, một cách giải thích có thể không rõ ràng đối với những người không chuyên. Vì lý do này, độ lệch chuẩn, được biểu thị bằng đơn vị ban đầu, được ưu tiên sử dụng khi báo cáo kết quả. Ví dụ: phương sai mẫu của thu nhập là s2 = 51.860.444 đô la bình phương, ý nghĩa của nó có thể không rõ ràng đối với khách hàng. Tốt hơn nên báo cáo độ lệch chuẩn mẫu s = $7201.
- Độ lệch chuẩn mẫu s được hiểu là kích thước của độ lệch điển hình, nghĩa là kích thước của sự khác biệt điển hình giữa các giá trị dữ liệu và

Machine Translated by Google

giá trị dữ liệu trung bình. Ví dụ: thu nhập thường lệch khỏi giá trị trung bình là $7201.

- Số đo vị trí cho biết vị trí tương đối của một giá trị dữ liệu cụ thể trong phân phối dữ liệu. Các thước đo vị trí mà chúng tôi đề cập ở đây là phần trăm, thứ hạng phần trăm, điểm Z và phần tư.
  - Phân vị thứ p của tập dữ liệu là giá trị dữ liệu sao cho p phần trăm giá trị trong tập dữ liệu bằng hoặc thấp hơn giá trị này. Phần trăm thứ 50 là trung vị. Ví dụ: thu nhập trung bình là 32.150 đô la và 50% giá trị dữ liệu nằm ở hoặc thấp hơn giá trị này.
  - Xếp hạng phần trăm của một giá trị dữ liệu bằng tỷ lệ phần trăm của các giá trị trong tập dữ liệu bằng hoặc thấp hơn giá trị đó. Ví dụ: xếp hạng phần trăm thu nhập của Người nộp đơn 1 là $38.000 là 90%, vì đó là tỷ lệ phần trăm thu nhập bằng hoặc thấp hơn $38.000.
  - Điểm Z cho một giá trị dữ liệu cụ thể biểu thị có bao nhiêu độ lệch chuẩn mà giá trị dữ liệu nằm trên hoặc dưới giá trị trung bình. Đối với một mẫu, điểm Z là:

$$\text{điểm Z} = \frac{x - \bar{x}}{s}$$

Đối với Người nộp đơn 6, điểm Z là

$$\frac{24{,}000 - 32{,}540}{7201} \approx -1.2$$

Thu nhập của Ứng viên 6 nằm dưới mức trung bình 1,2 độ lệch chuẩn.

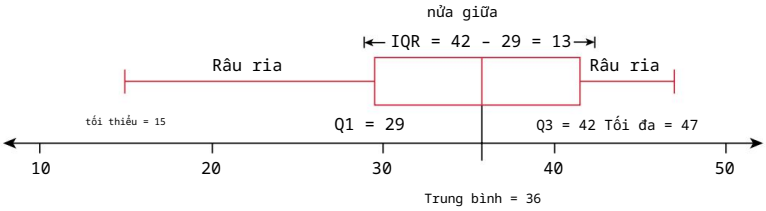
  - Chúng tôi cũng có thể tìm thấy các giá trị dữ liệu, được cho điểm Z. Giả sử sẽ không có khoản vay nào được cấp cho những người có thu nhập thấp hơn 2 độ lệch chuẩn so với mức trung bình. Ở đây, Z-score = 2 và thu nhập tối thiểu tương ứng là:

$$\text{thu nhập} = \bar{x} - z \cdot \text{độ lệch chuẩn} = 32{,}540 - (2)(7201) = 18{,}138 \text{ đô la}$$

Sẽ không có khoản vay nào được cung cấp cho những người nộp đơn có thu nhập dưới $18,138.

  - Nếu phân phối dữ liệu là bình thường, thì Quy tắc Thực nghiệm nêu rõ:
    - Khoảng 68% dữ liệu nằm trong một độ lệch chuẩn của giá trị trung bình.
    - Khoảng 95% dữ liệu nằm trong hai độ lệch chuẩn của giá trị trung bình.
    - Khoảng 99,7% dữ liệu nằm trong ba độ lệch chuẩn của giá trị trung bình.
  - Phần tư thứ nhất (Q1) là phần trăm thứ 25 của tập dữ liệu; phần tư thứ hai (Q2) là phần trăm thứ 50 (trung vị); và phần tư thứ ba (Q3) là phân vị thứ 75.

Machine Translated by Google



Hình A.7 Biểu đồ hộp của dữ liệu lệch trái.

- IQR là thước đo độ biến thiên không nhạy cảm với sự hiện diện của các giá trị ngoại lệ. IQR = Q3 Q1.
- Trong phương pháp IQR để phát hiện ngoại lệ, giá trị dữ liệu x là ngoại lệ nếu một trong hai
  - x ≤Q1 1,5(IQR) hoặc
  - x ≥Q3+1,5(IQR).

- Tóm tắt năm số của một tập dữ liệu bao gồm mức tối thiểu, Q1 , trung bình, Q3, và tối đa.
- Boxplot là một biểu đồ dựa trên tóm tắt năm số, hữu ích để nhận biết tính đối xứng và độ lệch . Giả sử đối với một tập dữ liệu cụ thể (không phải từ Bảng A.1), chúng ta có Min = 15, Q1 = 29, Median = 36, Q3 = 42 và Max = 47.
  Sau đó, boxplot được hiển thị trong Hình A.7.
  - Hộp bao gồm "nửa giữa" của dữ liệu từ Q1 đến Q3.
  - Râu bên trái kéo dài xuống giá trị tối thiểu không phải là giá trị ngoại lệ.
  - Râu bên phải kéo dài đến giá trị tối đa không phải là giá trị ngoại lệ.
  - Khi râu bên trái dài hơn râu bên phải thì sự phân bố lệch trái và ngược lại.
  - Khi râu có chiều dài bằng nhau, sự phân bố là đối xứng. Phân bố trong Hình A.7 cho thấy bằng chứng về sự lệch trái.

PHẦN 4: TỔNG KẾT VÀ HÌNH ẢNH HÓA CÁC MỐI QUAN HỆ LƯỠI BIẾN

- Mối quan hệ hai biến là mối quan hệ giữa hai biến.
- Mối quan hệ giữa hai biến phân loại được tóm tắt bằng cách sử dụng bảng dự phòng, là bảng chéo của hai biến và chứa một ô cho mọi tổ hợp giá trị biến (nghĩa là cho mọi

Machine Translated by Google

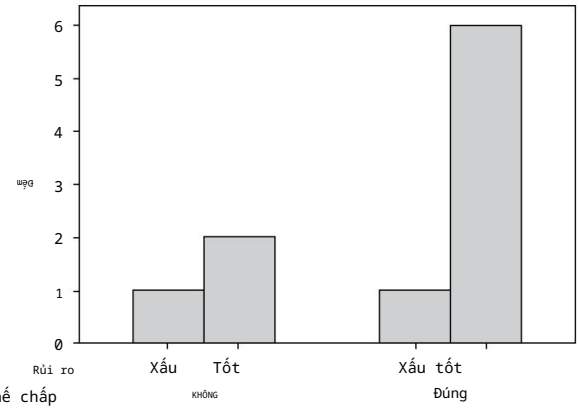
dự phòng). Bảng A.5 là bảng dự phòng cho các biến thế chấp và rủi ro. Cột tổng chứa phân phối cận biên cho rủi ro, nghĩa là phân phối tần suất cho riêng biến này. Tương tự, hàng tổng đại diện cho phân phối cận biên cho thế chấp.

- Có thể học được nhiều điều từ bảng dự phòng. Tỷ lệ cơ bản của rủi ro xấu là 2/10 = 20%. Tuy nhiên, tỷ lệ rủi ro xấu đối với người nộp đơn không có thế chấp là 1/3 = 33%, cao hơn mức cơ sở; và tỷ lệ rủi ro xấu đối với người nộp đơn có thế chấp chỉ là 1/7 = 1%, thấp hơn so với mức cơ sở. Do đó, việc người nộp đơn có thế chấp hay không rất hữu ích để dự đoán rủi ro.

- Biểu đồ thanh nhóm là biểu diễn đồ họa của bảng dự phòng. Hình A.8 cho thấy biểu đồ thanh nhóm rủi ro, được nhóm theo thế chấp. Lưu ý rằng sự chênh lệch giữa hai nhóm là rõ ràng ngay lập tức.
- Để tóm tắt mối quan hệ giữa biến định lượng và biến phân loại, chúng ta tính thống kê tóm tắt cho biến định lượng ứng với từng cấp độ của biến phân loại. Ví dụ, Minitab cung cấp

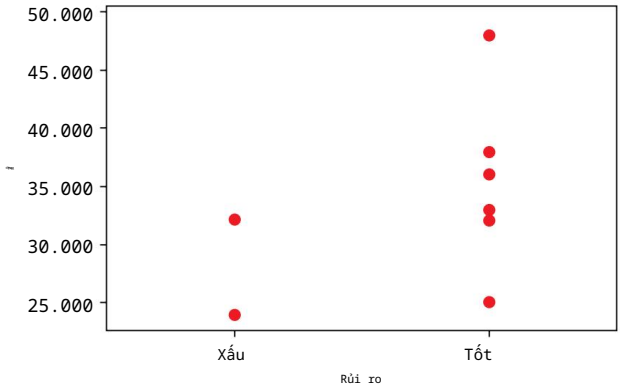
Bảng A.5 Bảng dự phòng cho thế chấp so với rủi ro

		Thế chấp		
		Đúng	KHÔNG	Tổng cộng
Rủi ro	Tốt	6	2	[illegible]
	Xấu	1	1	2
	Tổng cộng	7	3	10



Hình A.8 Biểu đồ thanh nhóm rủi ro, nhóm theo thế chấp.

Machine Translated by Google



Hình A.9 Đồ thị giá trị cá nhân của thu nhập so với rủi ro.

thống kê tóm tắt sau đây về thu nhập, đối với hồ sơ có rủi ro xấu và đối với hồ sơ có rủi ro tốt. Tất cả các biện pháp tóm tắt là lớn hơn cho rủi ro tốt. Sự khác biệt có đáng kể không? Chúng ta cần thực hiện kiểm định giả thuyết để tìm hiểu (Chương 4).

Thống kê mô tả: Thu nhập

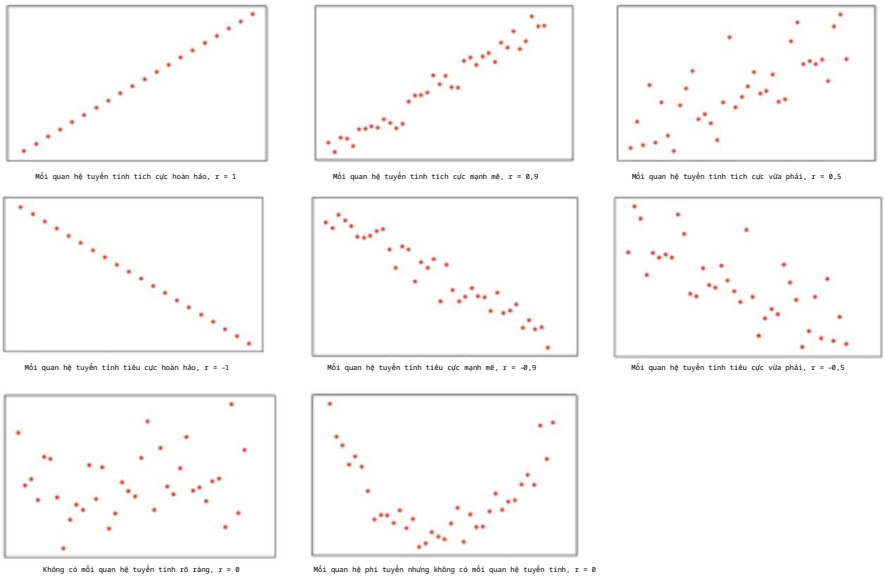
	Rủi ro	Nghĩa là	std dev	tối thiểu	trung bình tối đa	
Biến đổi	Xấu	28.050	5728	24.000	28.050	32.100
Thu nhập	Tốt	33.663	7402	25.000	32.600	48.000

- Để hình dung mối quan hệ giữa một biến định lượng và một biến phân loại, chúng ta có thể sử dụng một biểu đồ giá trị riêng lẻ, về cơ bản là một tập hợp các biểu đồ chấm dọc, một biểu đồ cho mỗi danh mục trong biến phân loại. Hình A.9 thể hiện biểu đồ giá trị cá nhân đối với thu nhập so với rủi ro, cho thấy thu nhập đối với rủi ro tốt có xu hướng lớn hơn.
- Biểu đồ phân tán được sử dụng để hình dung mối quan hệ giữa hai biến định lượng x và y. Mỗi điểm (x, y) được vẽ trên một mặt phẳng Descartes với trục x nằm ngang và trục y nằm trên trục tung. Hình A.10 thể hiện tám đồ thị phân tán, thể hiện một số loại tàu quan hệ có thể có giữa các biến, cùng với giá trị của hệ số tương quan r.

- Hệ số tương quan r lượng hóa độ mạnh và chiều của mối quan hệ tuyến tính giữa hai biến định lượng. Hệ số tương quan được định nghĩa là

$$r = \frac{\sum\left(x_x\overline{y}_y^{-}\right)\left(\quad -^{-}\right)}{\left(N-1\right)s_{x_y}}$$

Machine Translated by Google



Hình A.10 Một số mối quan hệ có thể có giữa x và y.

Machine Translated by Google

trong đó sx và sy tương ứng là độ lệch chuẩn của biến x và biến y. $1 \le r \le 1$. ° Trong khai thác dữ liệu,

nơi có một số lượng lớn các bản ghi (hơn 1000), thậm chí các giá trị nhỏ của r, chẳng hạn như $0,1 \le r \le 0,1$ có thể có ý nghĩa thống kê. ° Nếu r

dương và có ý nghĩa, ta nói rằng x và y tương quan thuận.
Sự gia tăng của x có liên quan đến sự gia tăng của y.

° Nếu r âm và có nghĩa, ta nói rằng x và y tương quan nghịch. Sự gia tăng của x có liên quan đến việc giảm của y.

Machine Translated by Google

Mục lục

thuật toán tiên nghiệm , 47, 70, 199
mục đích kiểm toán kế toán, 142 lệnh
add_constant(), 154 lệnh addmargins(),
53, 123–4, 125 bộ dữ liệu adult_ch6_test , 81 bộ
dữ liệu adult_ch6_training , 81,
82 lệnh aes(), 58 Alcohol_flag, 115, 116,
117, 122, 124 Tất cả Mô
hình Phủ định, 77, 101, 109 Tất cả Mô hình Tích
cực, 77, 109 Gói phần mềm Anaconda, lệnh
9 apriori(), 208, 210 Xác suất A-
priori, gói 125 arules , 204 , 205
lệnh as.factor(), 148 lệnh
as.numeric(), 36 liên kết, 5, 7
quy tắc liên kết, 30

tiêu chí khác biệt về độ tin cậy, độ tin cậy
208–9 của, tiêu chí
chỉ số tin cậy 201, 209–10 xác nhận số liệu
của chúng tôi, mức tăng 207–8, khai
thác 201–2,
hỗ trợ 200, 202–6 cho,
giá trị 200–201, thuộc
tính 210, 215

thuật toán lan truyền ngược, 134 bộ dữ
liệu bổ sung của ngân hàng , 71
bộ dữ liệu của ngân hàng-bổ sung-full.txt ,
29 bộ dữ liệu bank_marketing_test , 29, 31
bộ dữ liệu bank_marketing_training , 22–3, 29, 47,
49, 50
tập dữ liệu bank_train , biểu đồ
thanh 24, 25, 218

lệnh bar(), 55, 57 phương
pháp hay
nhất về biểu đồ
thanh, 49 với lớp phủ phản hồi, 47–51
từ Python của age_binned, 61, 62, 63 từ
R của
age_binned, 59, 60 từ R của
previous_outcome, 47, 48 sử dụng Python, 49–
50 sử dụng R, 50–51 trong
R của biến giáo
dục, 36, 37
hiệu suất mô hình cơ sở, 4, 78 cho Phân
loại nhị phân, 77 phân loại k-nary,
77 cho hồi quy, 162–3 cho các
mô hình ước tính, 163 mô
hình hồi quy cuối cùng, 163

Định lý Bayes xác
suất biên, 114 trước không cung
cấp thông tin, 114 xác suất
sau, 114 giá trị biến dự đoán,
113 xác suất trước, 114 hồi quy tập
hợp con tốt nhất, 161 sử
dụng R, 162 biến thể giữa các cụm,
142

Mô hình danh mục lớn nhất, 77 tạo
thùng dựa
trên giá trị dự đoán bằng
Python, 59–62 sử dụng R,
62–3 phương pháp
hay nhất, 59
Phân cụm BIRCH, 142 mối quan
hệ hai biến, 225 boxplot, 225
nhánh, 81

Khoa học dữ liệu sử dụng Python và R, Phiên bản đầu tiên. Chantal D. Larose và Daniel T. Larose. © 2019 John Wiley & Sons, Inc. Xuất bản năm 2019 bởi John Wiley & Sons, Inc.

Machine Translated by Google

Cây quyết định
thuật toán
C5.0 sử dụng Python,
89–90 sử dụng R, 89,
90–91 R, thước đo đánh giá của, 102,
103 GIỎ HÀNG xem các trường hợp hoặc đối tượng
mô hình cây hồi
quy và phân loại (GIỎI),
lệnh categorical() 215, 84
dữ liệu phân loại dưới dạng số, biểu thị lại
bằng Python, 36–8 sử
dụng R, 38–9
biến phân loại xem thêm biểu đồ thanh biến
định
tính, tần suất
218 (hoặc số đếm), phân bố tần
suất 217, 218 (tương đối), biểu đồ hình tròn
217, 218, 218
lệnh c(), 24, 103 điều
tra dân số,
217 lớp độc lập có điều kiện, 114 phân
loại, 5, 6–7 phân loại
và mô hình cây hồi quy (GIỎI), 11 nút đã kết
hôn, 82–3
nút trạng thái hôn
nhân chưa kết hôn, 83 nút gốc phân
chia, 82, 83–4 sử dụng
Python, 84–6 sử dụng
R, 86–8 bộ dữ
liệu kiểm tra_dữ liệu_dữ liệu_dữ liệu_điều
khiển_quần áo, 101 bộ dữ liệu kiểm
tra_dữ_liệu_quần áo , 103 bộ dữ liệu kiểm
tra_bán hàng_quần
áo, 152, 154, 156, 161, 189, 190, 191 bộ dữ liệu đào tạo bằng dữ liệu
156, 161, 189, 190, 191
bộ dữ liệu kiểm tra quần áo_store_PCA_ ,
175 bộ dữ liệu đào tạo quần áo_store_PCA_ , 175
biểu đồ cột nhóm, 226
nhóm, 5, 7, 30 mục
đích kiểm toán kế toán, 142
BIRCH, 142
biểu hiện gen, 142 thứ
bậc, 142
K-mean, 142–4
Mạng Kohonen, 142 tiếp
thị mục tiêu, 141 xác
nhận, 144–5 lệnh
colnames(), 86 giá trị
colnames(), 204
cột của phần quan tâm, 25 lệnh
column_stack(), 56, 159 hàm kết hợp,
131–3 thành phần, 175 lệnh
concat(), 74 tính
độc lập có điều kiện,
118 xác suất có điều kiện, 125
tiêu chí chênh lệch độ tin cậy,
208–9 sử dụng R, 208–9 độ tin cậy của quy
tắc, 209 tiêu
chí thương số tin cậy sử dụng
R, 209–10 bảng dự phòng, 225,
226 thực tiễn tốt
nhất, 52 để đánh giá mô hình 1,
101 để đánh giá mô
hình 2, 106 để đánh giá mô
hình 3, 108 về giới tính và
cái chết, 136 sử dụng Python,
13–14, 52–3, 123 sử
dụng R, 23–4, 53 biến liên tục,
lệnh 216 cor(),
lệnh 181 corr(), hệ số
tương quan 180, ma
trận tương quan 227,
229 từ Python, thẻ tín dụng 180,
154, 157 lệnh crosstab(), 13, 14, 49,
61 quy trình xác nhận
chéo, 3, 70 phân phối tần số tích lũy
(tương đối), 219 hành vi đường
cong, 133 lệnh cut(), 62
dữ liệu,
tạo bằng dữ liệu
quần áo , 152, 154,
216, nạo vét dữ
liệu 3, lệnh 70 data.frame(), quy trình
khai thác dữ liệu 85, 145,
160, mô hình hóa dữ liệu 142,
tập dữ liệu huấn luyện cân bằng
chuẩn bị, hiệu suất mô hình cơ sở 73–
6, phân vùng dữ liệu 77–8,
69–72 Giai đoạn
thiết lập, 69 xác thực phân
vùng của bạn, 72 phân vùng
dữ liệu, 69–72
chuẩn bị dữ liệu thêm
trường chỉ mục bằng
Python, 31–2 sử
dụng R, 32–3 bộ dữ liệu tiếp thị ngân hàng, 29

Machine Translated by Google

thay đổi giá trị trường gây hiểu nhầm
bằng Python, 34
bằng R, 34–6 xác
định các giá trị ngoại lệ
sử dụng Python, 41–2
sử dụng R, 42–3
giai đoạn,
31 giai đoạn hiểu vấn đề về khoa
học dữ liệu, 30 mục tiêu dự
án, 29–30 biểu hiện lại dữ
liệu phân loại dưới dạng số sử dụng Python,
36–8 sử dụng R, 38–9
chuẩn hóa số các
trường sử dụng Python, 40 sử dụng R,
40 giai đoạn chuẩn
bị dữ liệu,
2 khoa học dữ liệu, 1

Phương pháp khoa học dữ liệu (DSM)
bản chất lặp lại và thích ứng, 1, 2, 5
giai đoạn,
3 giai đoạn chuẩn bị dữ liệu,
2 giai đoạn triển khai,
5 giai đoạn đánh giá, 4–
5 giai đoạn phân tích dữ liệu khám phá,
3 giai đoạn lập mô
hình, 4 giai đoạn tìm hiểu vấn đề, 2
giai đoạn thiết
lập, 3–4 dữ liệu
liên kết nhiệm vụ
khoa học, 5, 7 phân
loại, 5, 6–7 phân
cụm, 5, 7 mô tả,
5, 6 ước tính, 5,
6 dự đoán, 5, 7
nhà khoa học dữ
liệu, 6 tóm tắt dữ
liệu về mối quan hệ hai biến, 225–9
khối xây dựng của phân tích dữ liệu, 215–17
thước đo trung tâm, 222–3
thước đo vị trí, 224–5 thước đo
độ biến thiên, 223–4
ngày kể từ khi mua, 154 ngày_kể
từ_biến_trước, 34, 35, 36 nút quyết định, 81
lệnh
DecisionTreeClassifier(), 85, 90 cây quyết định,
30 thuật toán C5.0,
88–91
cây phân loại và hồi quy, rừng ngẫu nhiên 83–
8, 91–3
giai đoạn triển khai, 5
lệnh description(), 146 mô
tả, 5, 6 mô hình
hồi quy mô tả, 151–2 thống kê mô tả, 6, 215
độ lệch, 223 lệnh dim(), 32, 71
chiều, lời
nguyền, 167, 199 thành phần
giảm kích thước nên trích xuất, 176–7

Tiêu chí Giá trị riêng, 176–7
tiêu chí giải thích về tỷ lệ phương
sai, 177 lời
nguyền về chiều, 167 tính hai
lần, 167 đa cộng tuyến,
167, 168–74, 183–4 quá khớp, 167 thực hiện
PCA với K = 4,
178 phân tích thành phần chính , 175 –
6, 178–83 vi phạm phân tích, biến rời rạc
167,
phân phối 216, lệnh div()
220, 50, 52, 53 dotplot,
đếm kép 220, 221,
lệnh drop() 167, lệnh dropna()
180, 172

DSM xem Phương pháp khoa học dữ liệu (DSM)

EDA xem phân tích dữ liệu khám phá (EDA) giáo
dục_biến số, 36, 37, 38, 39 biến giáo dục, 36, 37

Tiêu chí Giá trị bản địa, 176–7
phần tử, 215
Quy tắc thực nghiệm,
224 phương pháp tập hợp,
91 entropy của X,
88 ước tính, 5, 6
giai đoạn đánh giá, 4–5
phân tích dữ liệu khám phá (EDA), 30
biểu đồ thanh có lớp phủ phản hồi, 47–51 tạo
thành phần dựa trên giá trị dự đoán, 58–63 bảng
dự phòng, 51–3 Nghiên cứu về
tim Framingham, 136, 137 biểu đồ có lớp
phủ phản hồi, 53–8 so với HT, 47

Machine Translated by Google

giai đoạn phân tích dữ liệu khám phá,
lệnh 3 export_graphviz(), 86, 90

mô hình hồi quy logistic cuối cùng, 189
mô hình hồi quy cuối cùng, 153, 159
kết quả hồi quy cuối cùng
từ R, 157
phần tư thứ nhất (Q1),
224 lệnh fit(), 85, 155, 191, 194
tóm tắt năm số, 225
Dữ liệu Nghiên cứu Tim mạch Framingham, 136
Tập dữ liệu Framingham_training, 137
phân phối tần số, 218

phân cụm biểu hiện gen, 142 hồi
quy tuyến tính mô hình tuyến tính
tổng quát (GLM), 188
hồi quy logistic, 188–92 hồi
quy Poisson, 192–5 lệnh
geom_bar(), 51, 58 lệnh
geom_histogram(), 58 lệnh
get_dummies(), 194 ggplot() mã,
124 lệnh ggplot(),
gói 51, 57 ggplot2, tiêu chí
Gini 20–21, Chỉ số Gini
91, 83 GLM xem các
mô hình tuyến
tính tổng quát (GLM)
Lệnh GLM(), 194 phân tích
dữ liệu đồ họa, 47 lệnh
grid.arrange(), 124 gói
gridExtra, 124

lệnh head(), 33, 42, 43, 205, 209 phân cụm
theo cấp bậc, 142 lệnh hist(),
34, 35 biểu đồ, 15, 220
phương pháp hay nhất,
55 trong Python,
35 trong R,
33, 36, 39 với
lớp phủ phản hồi , 53–8 sử dụng
Python, 55–7 sử dụng
R, 54, 58 của
Spyder, 16, 17 tập
dữ liệu lưu giữ, 70
tính đồng nhất của tỷ lệ, thử nghiệm cho, 72
HT xem thử nghiệm giả thuyết
(HT) thử nghiệm giả
thuyết (HT), 47 EDA so với, 47

lệnh ifelse(), 35 Cửa
sổ "Nhập tập dữ liệu", 21, 22 nhập
các gói bằng Python,
11–12 bằng R,
trường chỉ
mục 20–21, thêm một, 31–3 bằng
Python, 31–2 bằng R,
32–3 biểu đồ
giá trị riêng lẻ, tăng thông
tin 227 hoặc giảm entropy, khái niệm, lệnh
88 kiểm tra(),
cài đặt 205, 209. lệnh
packages(), 20 dữ liệu khoảng thời
gian, 216 IQR, 225
Phương
pháp IQR để phát hiện các ngoại lệ, 225

xác thực chéo k-fold, 70
Ứng dụng thuật toán phân cụm K-
mean của, 143–4 sử dụng
Python, 145–7 sử dụng
R, 147–9
Lệnh KMeans(), 145, 147
Phân cụm mạng Kohonen, 142

nút lá, 81 lệnh
legend(), lệnh 55
levels(), lệnh 39
library(), 21 bộ dự báo
tuyến tính, 187 hồi
quy tuyến tính
dưới dạng mô hình tuyến tính
tổng quát, 188 hàm liên
kết, lệnh 187, 188
lm(), 156 hồi quy logistic, 30
ứng dụng của, 189–92 làm
mô hình tuyến tính chung, 188–9
sử dụng Python, 190–191
sử dụng R, 191–2
Lệnh logit(), 191

Lệnh MAE(), phân phối
cận biên 161, giả thuyết hậu
nghiệm tối đa 226, trung bình 114, sai số
tuyệt đối
trung bình (MAE) 222, 158, 160, 161
thước đo
tâm, 222–3 thước đo vị trí,
224–5

Machine Translated by Google

trung bình,
222 số liệu,
208 độ tin cậy,
207
nâng, 207 hỗ
trợ, 207 tầm
trung, 222 quy tắc hiệp
hội khai thác tỷ lệ "cơ sở", 202, 203
Tập dữ liệu Churn_Training_File , 202
ví dụ về, 200 sử
dụng R, 203–6
giá trị trường gây hiểu nhầm,
33–6 biểu đồ của days_since_previous, 33
sử dụng Python, 34
sử dụng R, 34–
6 chế độ,
222 đánh giá mô hình
tính toán chi phí lỗi không đồng đều
ma trận chi phí điều chỉnh,
ma trận chi phí
105, ứng dụng 104, dự đoán
100–104, biện
pháp đánh giá phân loại 100, chi phí 97–9
trên mỗi bản ghi và lợi nhuận
trên mỗi bản
ghi, 106 chi phí lỗi do dữ liệu
ma trận chi phí, 107,
108 đơn giản
hóa, 108 phương
pháp cho, 100 số liệu cho tất
cả các mô hình, 109 độ chính xác, thu hồi
và điểm Fß , 99–100 độ nhạy và
độ đặc hiệu, 99 chi phí lỗi không đồng đều,
so
sánh các mô hình,
106–7 giai đoạn lập mô hình, 4 mô
hình02 lệnh .summary(), lệnh 155
model02_test.summary(), 155 đa
cộng tuyến, 167, 183–4 dự đoán
từ Python, 171 hệ số hồi quy, 169–
70 phương trình hồi quy,
170 hệ số ổn định, 168
mô hình hồi quy không ổn định, 167
sử dụng Python, 172 –3
sử dụng R, 173, 174
sử dụng các hệ số lạm phát phương sai, 171–
4 hệ số biến thiên, lệnh 168–
9 MultinomialNB(), 11 mô hình hồi
quy bội
ứng dụng của, 152–4 sử
dụng Python, 153, 154–5 sử
dụng R, 156–7

Phân loại Naïve Bayes, 30 ứng
dụng, 115–25
Định lý Bayes, 113–14 tính
độc lập có điều kiện của lớp, 114 giả
thuyết hậu nghiệm tối đa, 114 trong
Python, 121–3 trong
R, 123–5
lệnh nawBayes(), 124 hành vi
gần như không đổi, 133 hành vi
gần như tuyến tính, 133 mạng
lưới,
131 mạng lưới thần kinh mô hình,
30, 70 ứng dụng, lan
truyền ngược 134–6, 134
trọng số kết nối và chức năng kết hợp,
131–3 đầu vào dữ
liệu và giá trị ban đầu, 131 ví dụ
về, 130 phân tích
dữ liệu khám phá, 135–6 lớp và nút
ẩn, 130–131 diễn giải trọng số, đầu
vào mạng 136–7, giá trị đầu ra
131–2, 132 trong
R, mô hình nơ-ron
thực và nơ-
ron nhân tạo 137–8, chức năng kích hoạt
sigmoid
129, 130, chức năng sigmoid 133–4, cấu
trúc 132, mạng được kết
nối hoàn toàn 129–
31 , 130 tính chất feedforward, 129
lớp, 129 mô hình nơ-ron,
thực và nhân
tạo, 129, 130 dữ liệu danh nghĩa, 216 phân
phối chuẩn, 188
ntree, 93

Thuật toán NultinonomialNB(), 122
biến số xem biến định lượng gói numpy ,
11, 12, 34

Lệnh OLS(), 154–5 lệnh(),
42–3 lệnh(), 204 dữ liệu
thứ tự, 216

Machine Translated by Google

"Bình phương nhỏ nhất thông thường" (OLS),
154 trang bị thừa, 167

gói, 20 gói gấu
trúc pd, 32 gói gấu trúc ,
tham số 11, 12, 32, 34, biểu đồ Pareto
217, phân tích
218, nguyên tắc, 167
PCA xem phân tích thành phần chính
(PCA)

Lệnh PCA(), thứ hạng phần
trăm 180, biểu đồ hình
tròn 224, lệnh
plot() 218, lệnh 49, 90, 182
plotnet(), ứng dụng mô hình
hồi quy Poisson 138 của, cấu
trúc 192–5 của, 193 sử
dụng Python, 193–4
sử dụng R, 194–5 dân số,
trung bình dân số
217, độ lệch chuẩn
dân số 222, phương sai dân
số 223, phân phối sau 223, lệnh 113, 119,
120 predict(), dự đoán 86,
91, 92, 103, 123, 125, 159, 160 , 5, 7 biến
dự đoán, 154, 217 Previous_outcome, 47, 48, 49,
50 lệnh
principal(), 182
ứng dụng phân tích thành phần chính
(PCA), 175–6 với K = 4, 178 biến dự đoán,
175 kết quả từ R , 176 đầu ra
R, 179 sử dụng Python, 179–81 sử dụng R, 181–
3 xác thực của, 178–9 phân
phối trước, 113,
119 giai đoạn hiểu vấn đề, 2
mục tiêu thành vấn đề
khoa học dữ
liệu, dịch, 30 mục tiêu dự
án, 29–30 lệnh
prop.table(), 63, 204
phương trình hồi quy tạm thời,
phân vị thứ 152 , 224

giá trị p, 152, 189
Python, 40
thêm trường chỉ mục, 31–2 cân
bằng tập dữ liệu đào tạo, 74–5 biểu đồ thanh
với lớp phủ phản hồi, 49–50 kiến thức cơ bản về
mã hóa khi truy cập
các bản ghi và biến trong, nút 14–15 và
văn bản
di chuột, 11 lệnh thực thi,
10–11 nhận dữ liệu vào, 12–13 nhập
gói vào, 11–12 lưu đầu ra
vào, 13–14 thiết lập đồ họa trong,
15–17 sử dụng nhận xét, 9–10
tạo thùng dựa trên dự đoán giá trị,
thuật toán 59–62 C5.0, mô
hình GIỎ HÀNG 89–90, dữ liệu phân loại 84–6 dưới
dạng số, biểu thị lại, 36–8
bảng dự phòng, ma trận
tương quan 52–3 từ, 180 tải xuống, 9 biểu đồ có
lớp phủ
phản hồi, 55 –7 xác định ngoại
lệ, 41–2 nhập gói, 11–12 thuật toán
phân cụm K-means,
145–7 hồi quy logistic, 190–191 giá trị trường
gây hiểu nhầm, 34 đánh giá mô
hình để ước tính, đa cộng tuyến
159–60, mô hình hồi quy bội 172–3 , 153, 154–
5 Phân loại Naïve Bayes, 121–3
PCA, 179–81 Mô hình hồi quy
Poisson, 193–4 khu rừng ngẫu nhiên, 92 tiêu chuẩn
hóa các trường số, 40 Hồi
quy logistic Python, 190

biến định tính, 215 biến định
lượng, 215, 216 lớp, 218 phân phối
tần suất tích
lũy (tương đối), phân phối 219, 220
dotplot, 220, 221
phân phối tần suất
(tương đối), biểu đồ
218–19, dữ liệu sai lệch 220, 221, 222

Machine Translated by Google

hiển thị thân và lá, phân phối
đối xứng 220, 221, lệnh truy vấn
() 220, 221, 41

r

quy tắc kết hợp, 209
biểu đồ thanh có lớp phủ phản hồi, 50–
51 kiến thức cơ
bản về mã hóa truy cập các bản ghi và
biến trong, 24–5 lệnh
thực thi, 20 nhận
dữ liệu, 21–3 nhập gói, 20–
21 lưu đầu ra, 23–4
sử dụng nhận xét,
19 hồi quy tập hợp con tốt
nhất, 162 tạo thùng dựa trên giá trị dự
đoán, 62–3 thuật toán
C5.0, 89, 90–91 dữ liệu phân loại dưới
dạng số,
biểu thức lại, tiêu chí chênh lệch độ
tin cậy 38–9, tiêu chí chỉ số tin cậy
208–9, 209–10 bảng dự
phòng, 53 lượt tải
xuống, 17–19 kết quả hồi
quy cuối cùng, 157 biểu đồ có lớp
phủ phản hồi, 58 gói nhập,
20–21 trường chỉ mục, thêm
thuật toán phân cụm, 32–3 K-means,
hồi quy logistic 147–9,
191–2 quy tắc liên kết khai
thác, giá trị trường sai lệch
203–6, đa cộng tuyến 34–6,
mô hình hồi quy bội 173, 174, phân
loại Naïve Bayes 156–7, mô hình
mạng thần kinh 123–5, trường
số 137–8, chuẩn hóa, 40 ngoại
lệ, xác định, 42–3 PCA,
181–3 Mô
hình hồi quy Poisson, 194–5 khu
rừng ngẫu nhiên, 92–
3 hồi quy từng bước, 162
tập dữ liệu huấn
luyện,
75–6 R2 159 R2 adj (R-squared đã điều
chỉnh), 160, 161 Lệnh
phạm vi,
lệnh 223 phạm vi(),
32 dữ liệu tỷ
lệ, lệnh 216 rbind(), mô
hình 76, 204 R C5.0, thước đo đánh giá,
lệnh 102 read_csv(), hệ số hồi quy
11, 12, 23, 31, đường cơ sở
mô hình hồi quy 154
mô hình hồi quy, mô hình hồi quy mô
tả 162–3, đánh giá mô hình 151–2 để ước
tính số liệu mô hình ước tính,
158 sử dụng Python, 159–60 sử
dụng R, kết quả bội
số 160–161 (xem
mô hình hồi quy bội) từ R, 157
hồi quy
từng bước, 161–2
cây hồi quy, 83–8 phân phối
tần số tương đối,
217, 218–19 lấy mẫu lại, 73 biểu đồ thanh
lớp phủ phản
hồi với, 47–51
biểu đồ với, 53–8 giá
trị phản hồi, 101

sau khi cân bằng
lại, 75 biến phản hồi,
217 nút gốc,
81 ma trận thành phần được xoay,
178 lệnh round(), 53, 181, 204
Rstudio, 19–25
Cửa sổ RStudio, lệnh
runif() 18, 19, 72

bộ dữ liệu sales_test ,
156 mẫu,
217 lệnh sample(), 74
giá trị trung bình
mẫu, 222 độ lệch chuẩn mẫu,
223 phương sai mẫu,
223 lệnh scale(), 147
lệnh scatter_matrix(), 172 biểu
đồ phân tán, 227,
228 phần tư thứ hai (Q2) ,
224 lệnh Series(), 32
lệnh set.seed(), 71
giai đoạn thiết lập,
3–4, 69 chức năng kích hoạt
sigmoid, 132–3 dữ liệu lệch, 223, 222
RandomForestClassifier() , 92 rừng ngẫu nhiên, 30 trong Python, 92 trong R, 92–3, mẫu ngẫu nhiên, 217

Machine Translated by Google

độ lệch, 222–3
sklearn, 121
gói sklearn.naive_bayes , 11 gói
sklearn , 89 lệnh
sort_values(), 41 đầu ra đồ
họa
Spyder trong, 16, 18 biểu
đồ, 15, 17 Cửa sổ
tùy chọn, 16, 17 Trình biên
dịch Spyder, 9 Cửa
sổ Spyder, 10 “SS
loading,” 176 thuật
toán hiện đại, 4 thống kê, 217
suy luận thống
kê, 217 lệnh stats.zscore,
145 hiển thị thân và lá, lệnh
220, 221 stepAIC(), 162 hồi quy
từng bước, 161 –2 sử dụng
lệnh R, 162 subset(), 204
Sugar_flag,
115, 116, 117, 124 lệnh
sum(), 52, 53, 61 lệnh summary(),
148, 155, 156, 160, 191, 195
summary2() lệnh, ký hiệu tổng 191, tổng sai số
bình
phương 222 (SSE), phân phối
đối xứng 134, 220, 221

lệnh table(), 63, 104, 204 tiếp
thị mục tiêu, 141 biến
mục tiêu, 154 tập dữ
liệu thử nghiệm,
70 phần tư thứ ba (Q3),
224 lệnh title(), 55
tập dữ liệu huấn luyện, 70,
73–6 bằng Python,
74– 5 trong
R, 75–6 huấn luyện trừ tập kiểm
tra, 145 t-test, hai
mẫu, 72 kiểm chứng chéo hai lần, 70

ước lượng không chệch,
223 ma trận thành phần chưa quay, 178
lớp vỏ trên, 141

xác thực phân vùng của bạn,
lệnh 72 value_counts(), 74 biến,
215 định tính,
215 định lượng,
215, 216 phương sai,
lệnh
223variance_inflation_factor(),
173 hệ số lạm
phát phương sai (VIF), phương sai 171–2,
173, 176,
tỷ lệ của, 178 phép xoay
varimax, 178 lệnh vif(),
173, 181 trực quan hóa các
mối quan hệ
hai chiều, 225–9 biểu đồ và bảng để
tóm tắt và sắp xếp dữ liệu, 217–22
biến phân loại, 217–18
biến
định lượng, 218–22

Tài khoản Web, 152, 153, 190
lệnh which(), 42, 75 tập dữ
liệu white_ wine_test , 143, 144 tập dữ
liệu white_wine_training , 143, 144, 145,
147 tập
dữ liệu wine_flag_test , 115, 120
tập dữ liệu wine_flag_training , 115, 120

lệnh xlabel(), 55

câu trả lời “có”, tỷ lệ phần trăm, 74–5
lệnh ylabel(), 53, 55 giá
trị ypred, 103, 104, 125, 160

lệnh zscore(), chuyển
đổi 40 điểm z, 145, 224
Thử nghiệm Z, hai mẫu, 72