DỰ ĐOÁN CHẤT LƯỢNG RƯỢU VANG

Nhóm B

Khoa Toán - Tin học, Trường Đại học Khoa học Tự nhiên
Học phần: Xử lý số liệu thống kê - 21TTH_KDL - MTH10513

Ngày 5 tháng 9 năm 2024

Các thành viên thực hiện

Giảng viên: TS. Tô Đức Khánh

Học viên:

Bùi Quang Thắng - 21280048
Huỳnh Thị Thu Thoảng - 21280074
Nguyễn Thị Bích Ngọc - 21280100
Đoàn Thị Mẫn Nhi - 21280102
Nguyễn Thúy Vy - 21280120

Mục lục

Mục lục

Giới thiệu bài toán

Hình 1: Rượu vang đỏ và trắng

- Chất lượng rượu vang là một yếu tố quan trọng quyết định đến giá bán sản phẩm và ảnh hưởng trực tiếp đến doanh thu của công ty.
- Do đó, việc phân loại chất lượng rượu vang là cần thiết để nhà quản lý và phát triển có cái nhìn sâu sắc hơn về sản phẩm của mình.

Giới thiệu bài toán

Làm thế nào để phân loại chất lượng rượu vang?

Giới thiệu dữ liệu

- Dữ liệu gồm hai files: **winequality-red.csv** và **winequality-white.csv**:
  - **winequality-red.csv**: rượu vang đỏ, với 12 cột và 1599 dòng.
  - **winequality-white.csv**: rượu vang trắng, với 12 cột và 4898 dòng.

  $\Longrightarrow$ Công ty đang sản xuất 2 loại rượu, với **chủ yếu là rượu vang trắng** với số lượng dòng gấp hơn 3 lần rượu vang đỏ.

Mô tả dữ liệu

- Hai files đều có cùng tên cột như sau. Các cột đều là các biến định lượng, chỉ có cột **quality** là các biến định tính.

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
7.100000	0.3200000	3.200000e-01	11.0000000	0.03800000	16.000000	66.00000	0.9937000	3.240000	0.4000000	11.500000	3
8.500000	0.2600000	2.100000e-01	16.2000000	0.07400000	41.000000	197.00000	0.9980000	3.020000	0.5000000	9.800000	3
7.100000	0.8750000	5.000000e-02	5.7000000	0.08200000	3.000000	14.00000	0.9980800	3.400000	0.5200000	10.200000	3
8.300000	1.0200000	2.000000e-02	3.4000000	0.08400000	6.000000	11.00000	0.9989200	3.480000	0.4900000	11.000000	3
6.100000	0.2600000	2.500000e-01	2.9000000	0.04700000	289.000000	440.00000	0.9931400	3.440000	0.6400000	10.500000	3
11.600000	0.5800000	6.600000e-01	2.2000000	0.07400000	10.000000	47.00000	1.0008000	3.250000	0.5700000	9.000000	3
7.600000	0.4800000	3.700000e-01	1.2000000	0.03400000	5.000000	57.00000	0.9925600	3.050000	0.5400000	10.400000	3
7.900000	0.6400000	4.600000e-01	10.6000000	0.24400000	33.000000	227.00000	0.9983000	2.870000	0.7400000	9.100000	3
8.600000	0.5500000	3.500000e-01	15.5500000	0.05700000	35.500000	366.50000	1.0001000	3.040000	0.6300000	11.000000	3
6.800000	0.8150000	0.000000e+00	1.2000000	0.26700000	16.000000	29.00000	0.9947100	3.320000	0.5100000	9.800000	3
7.400000	1.1850000	0.000000e+00	4.2500000	0.09700000	5.000000	14.00000	0.9966000	3.630000	0.5400000	10.700000	3
7.100000	0.4900000	2.200000e-01	2.0000000	0.04700000	146.500000	307.50000	0.9924000	3.240000	0.3700000	11.000000	3
9.400000	0.2400000	2.900000e-01	8.5000000	0.03700000	124.000000	208.00000	0.9939500	2.900000	0.3800000	11.000000	3
8.300000	0.3300000	4.200000e-01	1.1500000	0.03300000	18.000000	96.00000	0.9911000	3.200000	0.3200000	12.400000	3
6.900000	0.3900000	4.000000e-01	4.6000000	0.02200000	5.000000	19.00000	0.9915000	3.310000	0.3700000	12.600000	3
7.500000	0.3200000	2.400000e-01	4.6000000	0.05300000	8.000000	134.00000	0.9958000	3.140000	0.5000000	9.100000	3

Hình 2: Các cột dữ liệu

Mục lục

Kiểm tra dữ liệu

- Kiểm tra dữ liệu bị thiếu:

```
> sum(is.na(winequality_red))
[1] 0
> sum(is.na(winequality_white))
[1] 0
```

Hình 3: Kiểm tra và đếm các giá trị bị thiếu trong dữ liệu

$\Longrightarrow$ Không cần handle missing data khi xử lý bài toán.

Kiểm tra dữ liệu

- Kiểm tra dữ liệu bị trùng lặp:

```
> sum(duplicated(winequality_red))
[1] 240
> sum(duplicated(winequality_white))
[1] 937
```

Hình 4: Kiểm tra và đếm các giá trị bị trùng lặp trong dữ liệu

- Sau khi loại bỏ các giá trị trùng lặp bằng hàm *unique*, ta kiểm tra lại số dòng trước và sau khi drop:

```
> winequality_red <- unique(winequality_red)
> winequality_white <- unique(winequality_white)
> nrow(winequality_red)
[1] 1359
> nrow(winequality_white)
[1] 3961
```

Mô tả dữ liệu

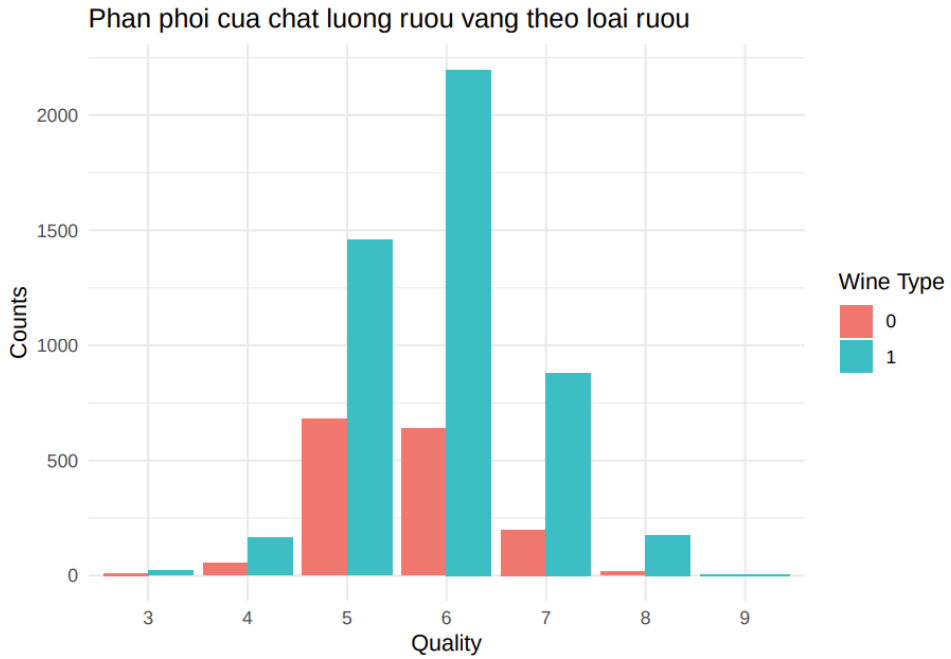
	bien	max	mean	median	min	sd
1	alcohol	14.900	10.49180	10.3000	8.0000	1.192712
2	chlorides	0.611	0.05603	0.0470	0.0090	0.035034
3	citric_acid	1.660	0.31863	0.3100	0.0000	0.145318
4	density	1.039	0.99470	0.9949	0.9871	0.002999
5	fixed_acidity	15.900	7.21531	7.0000	3.8000	1.296434
6	free_sulfur_dioxide	289.000	30.52532	29.0000	1.0000	17.749400
7	p_h	4.010	3.21850	3.2100	2.7200	0.160787
8	residual_sugar	65.800	5.44324	3.0000	0.6000	4.757804
9	sulphates	2.000	0.53127	0.5100	0.2200	0.148806
10	total_sulfur_dioxide	440.000	115.74457	118.0000	6.0000	56.521855
11	volatile_acidity	1.580	0.33967	0.2900	0.0800	0.164636

Hình 5: Bảng tóm tắt các biến định lượng

- Hai biến **free_sulful_dioxide** và **total_sulful_dioxide** có độ lệch chuẩn cao
  $\Longrightarrow$ có sự khác biệt trong quy trình sản xuất rượu vang

Mô tả dữ liệu

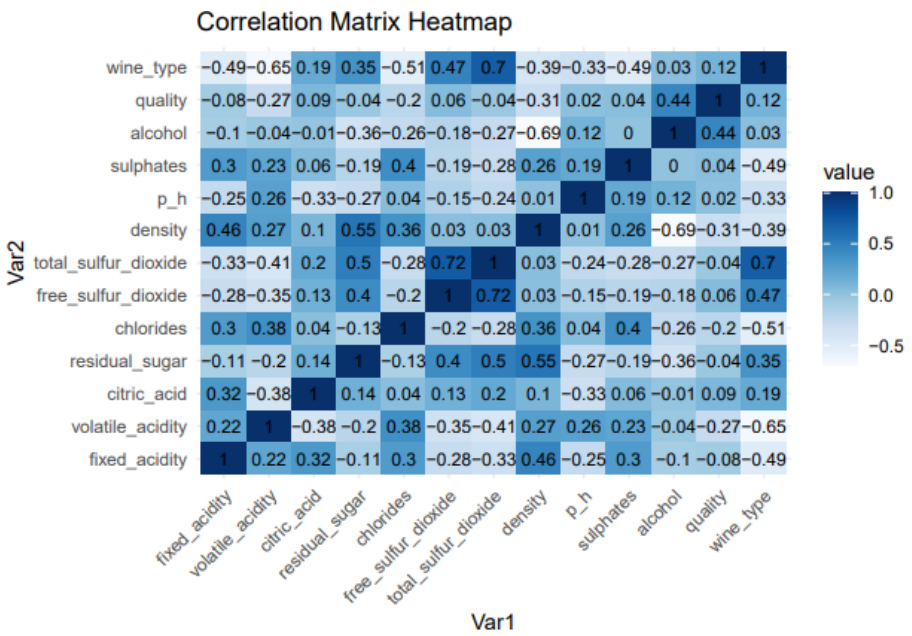
- Vì bài toán đang thực hiện là bài toán phân loại. Do đó, cột **quality** sẽ là biến target. Phân phối của chất lượng rượu vang theo hai loại như trong biểu đồ bên phải.



Hình 6: Bảng tóm tắt biến định tính

- **Nhận xét:** Data bị imblance ở chỗ: có rất ít các chai rượu có chất lượng rất thấp (3,4,...) và rất cao (9,8,...) $\Longrightarrow$ cần có các kỹ thuật để xử lý vấn đề này.

Exploratory data analysis (EDA)

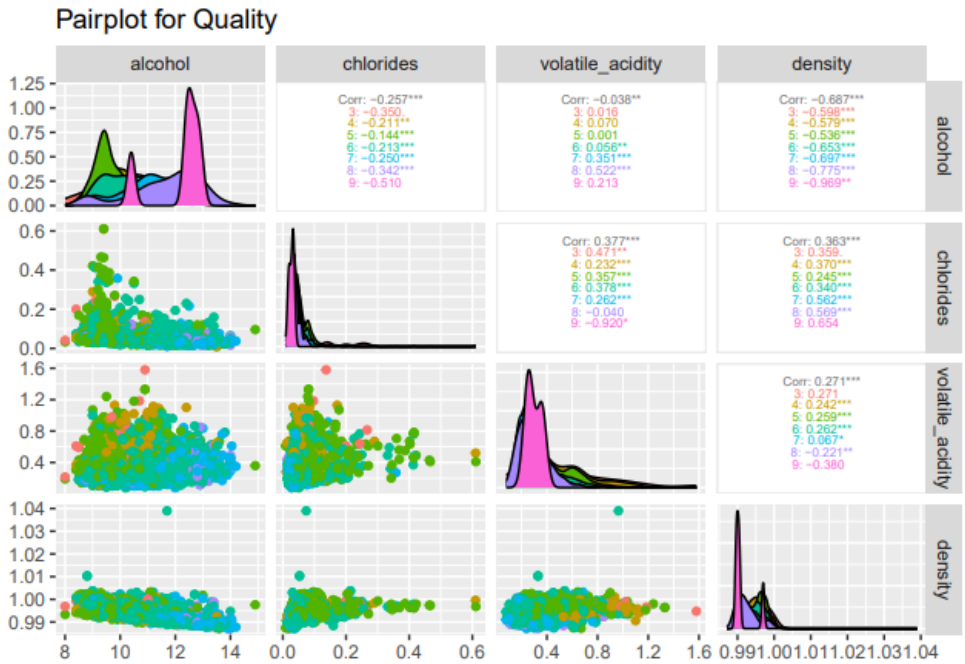


Hình 7: Biểu đồ heatmap biểu diễn sự tương quan giữa các biến

- Biến **quality** có tương quan (correlation) mạnh đến feature **"alcohol"** $(corr = 0.44)$, nên ta phỏng đoán đây là feature quan trọng quyết định nên chất lượng của rượu vang.
- Bên cạnh đó, các biến tương quan:
  - chlorides (-0.13)
  - volatile acidity (-0.27)
  - density (-0.31).
- Các feature còn lại có tương quan với quality không quá mạnh.

Exploratory data analysis (EDA)

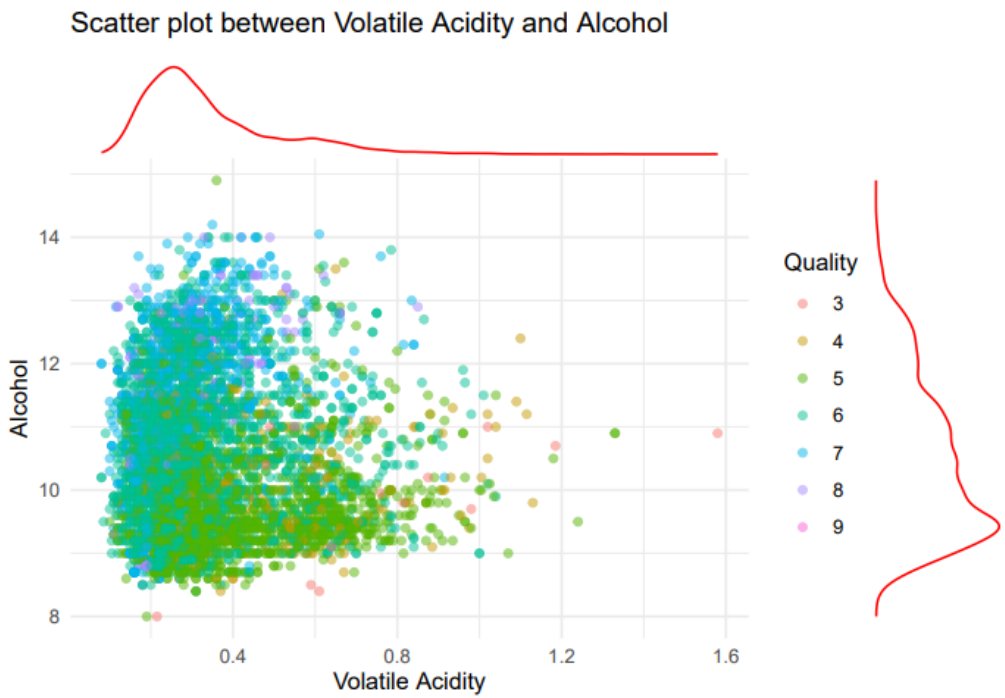
- Vì **alcohol** đã có tương quan dương khá cao đối với biến **quality**, do đó ta vẽ biểu đồ pairplot để tìm hiểu xem biến tương quan âm nào có ảnh hưởng với quality nhiều nhất.



Hình 8: Biểu đồ pair plot giữa các biến với quality

Exploratory data analysis (EDA)

- Xem xét mối quan hệ giữa hai biến có tương quan cao với cột quality

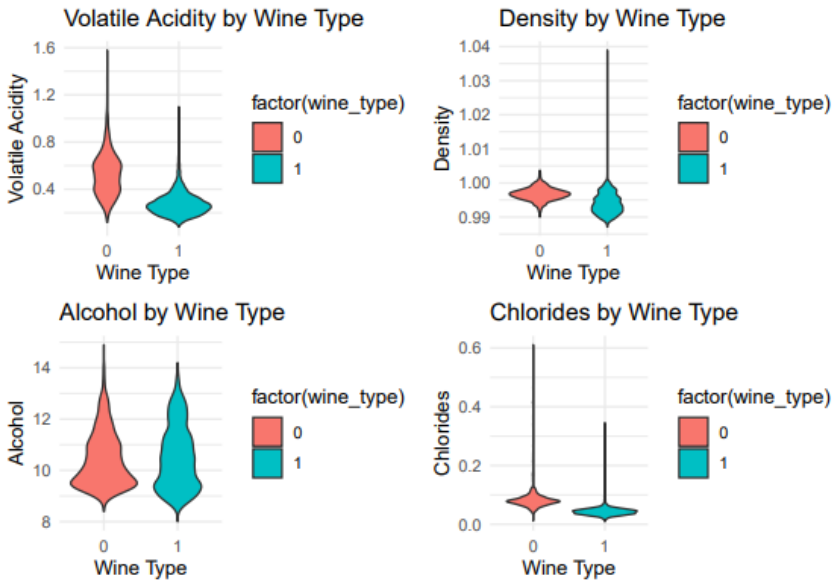


Hình 9: Biểu đồ marginal density

$\Rightarrow$ Hai biến Volatile Acidity và Alcohol đóng vai trò quan trọng trong bài toán phân loại chất lượng rượu vang.

Exploratory data analysis (EDA)

- Xem xét feature range của các cột giữa hai loại rượu vang đỏ và trắng:



Hình 10: Biểu đồ violin

⇒ Hai loại rượu có đặc trưng khác nhau với khoảng giá trị khác nhau.

Mục lục

Feature Engineering

- Kết hợp hai bộ dữ liệu rượu vang đỏ và trắng.
- Tạo cột mới `wine_type` để phân biệt giữa rượu vang đỏ và trắng.

```
# Thêm cột loại rượu
winequality_red$wine_type <- 0
winequality_white$wine_type <- 1
```

- Chuyển cột *quality* thành kiểu dữ liệu factor.

```
# merge lại
combined_wines <- rbind(winequality_red, winequality_white)
combined_wines <- unique(combined_wines)
combined_wines$quality <- as.factor(combined_wines$quality)
```

Feature Engineering

- Relabel lại trên merge data và red, white wine data : Ta sẽ relabel lại các class 3 và 4 thành 4, class 8 và 9 thành 8, để giảm số lượng class bị imbalance quá mức, sau đó chuyển cột quality thành kiểu dữ liệu factor.
- Chia dữ liệu thành tập huấn luyện (train) và kiểm tra (test):
  - Tập train: 80% dữ liệu.
  - Tập test: 20% dữ liệu.

Feature Engineering

- Xử lý dữ liệu imbalance để tăng hiệu suất cho các mô hình bằng phương pháp SMOTE.

```
table(train_data$quality)
```

```
##
##    4    5    6    7    8
##  197 1711 2269  864  159
```

Hình 11: imbalance ở các class

```
train_data_balanced <- custom_smote_all(train_data, train_data$quality,
                                        "quality", 1000, c("5", "6"))

table(train_data_balanced$quality)
```

```
##
##    4    5    6    7    8
## 1000 1711 2269 1000 1000
```

Hình 12: Các label sau khi đã handle trên tập train

Mục lục

Phân loại sử dụng dataset merge giữa red và white wine

Multinomial Logistic Regression

- Accuracy : 0.4857
- 95% CI : (0.4582, 0.5133)
- Confusion Matrix:

–	4	5	6	7	8
4	19	56	23	2	0
5	12	215	113	9	1
6	14	149	358	118	14
7	2	1	18	16	2
8	2	6	55	70	22

- Độ chính xác của mô hình là khá tốt, tuy nhiên vẫn còn một số class bị dự đoán sai, đặc biệt là class 4 và 8 (kể cả khi ta đã resample lại).
- Precision rất tệ ở class 4 và 8, ngược lại với class 5 và 6, điều này có thể là do số lượng mẫu ban đầu của hai class này quá ít, nên khi oversample, model vẫn bị thiếu thông tin về hai class này.

Phân loại sử dụng dataset merge giữa red và white wine

Multinomial Logistic Regression

Ưu điểm:

- Không yêu cầu giả định phân phối của các feature.
- Giải thích được sự biến động của biến mục tiêu (Y).
- Tìm được biến giải thích phù hợp.

Nhược điểm:

- Không hiệu quả trên dữ liệu bị mất cân bằng.
- Không hoạt động tốt khi lượng thông tin không đủ.
- Hoạt động kém hiệu quả nếu mô hình thật sự không có dạng tuyến tính.
- Mô hình yêu cầu các feature là độc lập, hoặc có sự tương quan thấp.

Discriminant Analysis

- Accuracy : 0.3793
- 95% CI : (0.3528, 0.4064)
- Confusion Matrix:

–	4	5	6	7	8
4	21	55	13	1	0
5	12	204	121	7	0
6	7	112	184	50	7
7	2	19	93	63	12
8	7	37	156	94	20

- Mô hình QDA cũng cho kết quả khá tốt, tuy nhiên vẫn còn một số class bị dự đoán sai, đặc biệt là class 4 và 8.
- Precision rất tệ ở class 4 và 8, ngược lại với class 5 và 6.

$\Rightarrow$ Độ chính xác tổng thể của QDA là thấp hơn so với Logistic Regression, điều này có thể là do QDA không phù hợp với dữ liệu của ta.

Phân loại sử dụng dataset merge giữa red và white wine

Naive Bayes

- Naive Bayes là một thuật toán phân lớp được mô hình hoá dựa trên định lý Bayes, rất hiệu quả và đơn giản, đặc biệt đối với các bài toán phân loại có số lượng mẫu nhỏ hoặc dữ liệu bị thiếu.
- Phương pháp Naive Bayes giả định các biến là độc lập. Train model Naive Bayes với phương pháp Laplace Smoothing, để model tự thêm 1 vào các dữ liệu bị thiếu tương ứng với từng class. Tránh việc gây ra các xác suất Bayes bằng 0, điều này sẽ giảm được hiện tượng overfitting.

Naive Bayes - Kết quả mô hình

- Accuracy: 0.2961
- 95% CI : (0.2713, 0.3217)
- Confusion Matrix:

–	4	5	6	7	8
4	9	62	40	5	1
5	11	184	129	20	1
6	18	103	149	37	11
7	1	3	8	16	0
8	10	75	241	137	26

Nhận xét: Độ chính xác của Naive Bayes là rất thấp, bị confuse khá nhiều giữa các class, đặc biệt là class 4 và class 8.

$\Rightarrow$ Đến đây, ta vẫn giữ quyết định chọn MLR là mô hình tốt nhất cho bài toán này.

Naive Bayes - Nguyên nhân cho hiệu suất thấp

1. Phương pháp Naive Bayes dựa trên giả định các biến là độc lập, tuy nhiên nhìn vào biểu đồ tương quan giữa các biến, ta thấy vẫn có nhiều biến có mối tương quan cao => vi phạm giả định về tính độc lập dẫn tới hiệu suất thấp
2. Naive Bayes hoạt động tốt trên các biến rời rạc, tuy nhiên ở đây nhóm sử dụng ước lượng mật độ kernel. Đây là phương pháp phi tham số, linh hoạt hơn so với giả định phân phối cụ thể. Tuy nhiên, với biến liên tục, Naive Bayes vẫn sẽ không hiệu quả bằng một số phương pháp khác như hồi quy logistic hoặc SVM, đặc biệt khi dữ liệu hiện tại có nhiều biến liên tục và mối quan hệ phức tạp giữa các biến.

Phân loại sử dụng dataset merge giữa red và white wine

Random Forest - không sử dụng bootstrap

- Random Forest sử dụng kỹ thuật bagging của Decision Tree
- Độ chính xác cao, khả năng generalize tốt.
- Xử lý data imbalanced hiệu quả.

- Accuracy : 0.6669
- 95% CI : (0.6405, 0.6926)
- Confusion Matrix:

–	4	5	6	7	8
4	17	22	4	0	0
5	19	297	87	5	0
6	11	105	427	84	14
7	2	3	33	107	8
8	0	0	16	19	17

$\Rightarrow$ Độ chính xác ở class 4 và 8 cải thiện rõ rệt so với 2 mô hình trước

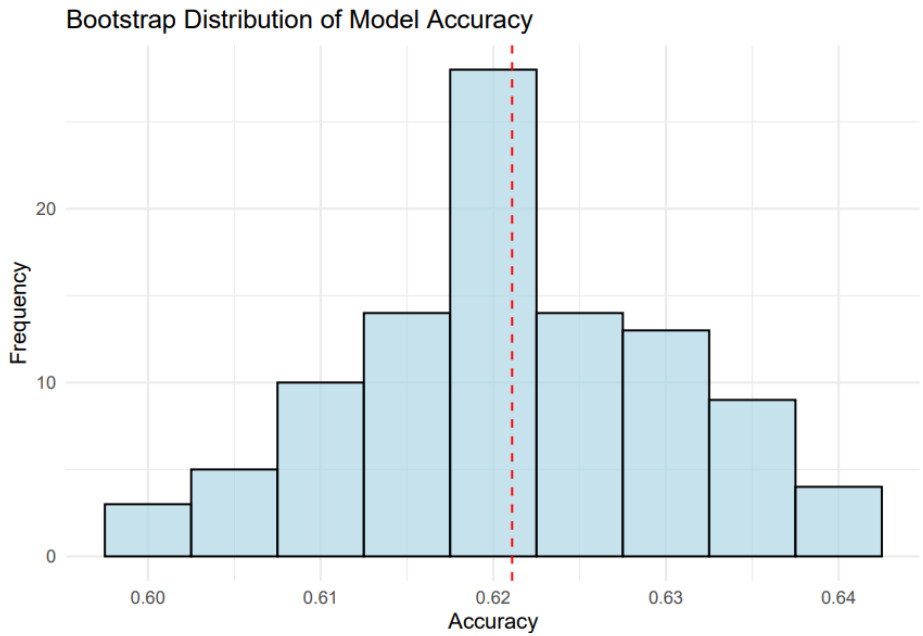
Random Forest - sử dụng bootstrap

- Đánh giá sự ổn định và độ tin cậy của mô hình bằng cách lặp lại quá trình, huấn luyện nhiều lần trên các mẫu con của dữ liệu gốc và đo lường độ chính xác trên tập kiểm tra.

- R = 100

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = train_data_balanced, statistic = bootstrap_accuracy,
##     R = 100)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.6669237 -0.04583655 0.009138086
```

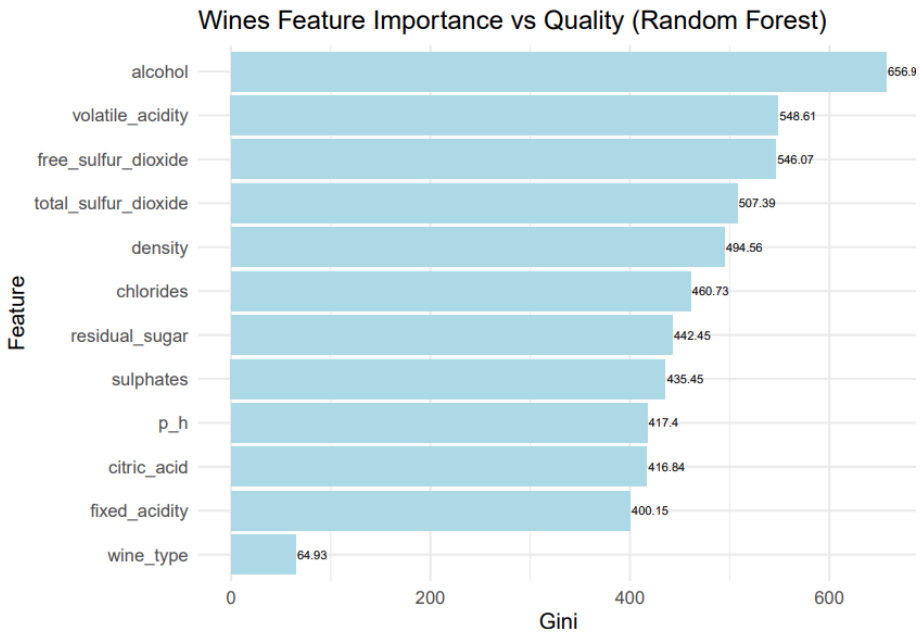
Hình 13: bootstrap results



Hình 14: đồ thị về phân phối của độ chính xác mô hình với bootstrap

⇒ Mô hình có khả năng dự đoán tốt trên dữ liệu chưa từng thấy.

Tầm quan trọng của các features so với quality



Hình 15: Gini của từng feature với quality

- Độ cồn, độ axit bay hơi, hàm lượng sulfur dioxide tự do, tổng hàm lượng sulfur dioxide, mật độ là những features quan trọng nhất.
- Hai features tương quan: nồng độ cồn và độ axit bay hơi là hai yếu tố quan trọng nhất ảnh hưởng đến chất lượng rượu vang, và phải đi kèm với nhau.

$\Rightarrow$ Đề xuất: tập trung vào nồng độ cồn và độ axit bay hơi trong quá trình sản xuất; kiểm soát các features quan trọng: độ pH, hàm lượng đường, độ đặc.

Phân loại sử dụng dataset merge giữa red và white wine

SVM (Further work)

- SVM (Support Vector Machine) là một mô hình học máy mạnh mẽ và linh hoạt, rất phù hợp cho các bài toán phân loại đa lớp.
- Mô hình SVM có khả năng tổng quát hoá tốt, xử lý tốt dữ liệu nhiều chiều và tìm ra biên phân cách tối ưu giữa các lớp.

SVM - Kết quả mô hình

- Accuracy : 0.532
- 95% CI : (0.5044, 0.5594)
- Confusion Matrix:

–	4	5	6	7	8
4	20	52	14	2	0
5	14	240	116	6	0
6	14	128	371	109	16
7	1	4	18	43	7
8	0	3	48	55	16

- Mô hình SVM cho kết quả với độ chính xác khoảng 50%-60%.

$\Rightarrow$ Mô hình SVM có thể cải thiện được nhiều so với Naive Bayes hay Logistic, nhưng vẫn không tốt bằng Random Forest.

SVM - Ưu và nhược điểm

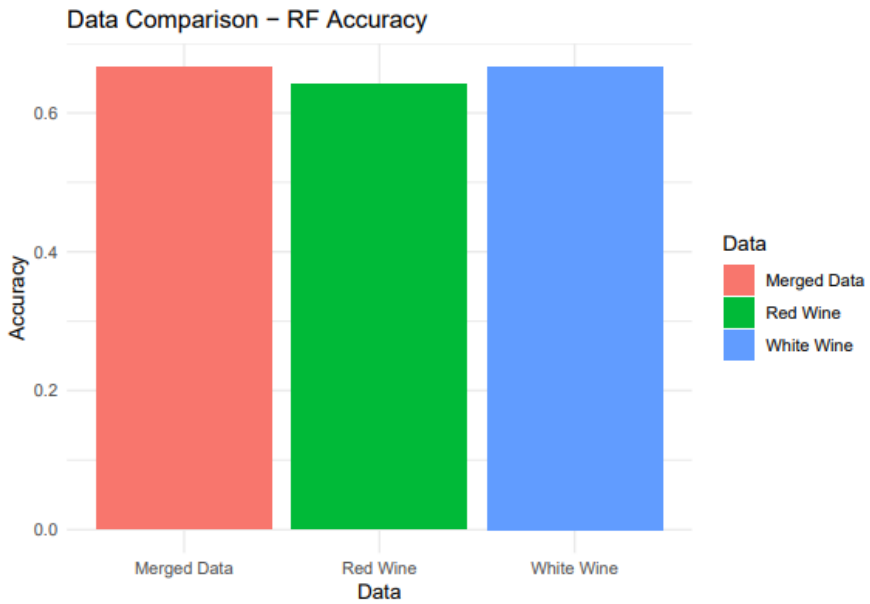
Ưu điểm

- Hoạt động tương đối tốt khi các lớp có sự phân chia rõ ràng.
- Hiệu quả hơn trong không gian nhiều chiều
- Hiệu quả về bộ nhớ

Nhược điểm

- Không phù hợp cho tập dữ liệu lớn
- Không hoạt động tốt khi tập dữ liệu có nhiều nhiễu
- Trong trường hợp số chiều vượt quá số lượng mẫu dữ liệu huấn luyện, SVM sẽ hoạt động kém.
- Không giải thích được xác suất để biến thuộc một class nào đó.

Phân loại sử dụng model riêng cho red và white



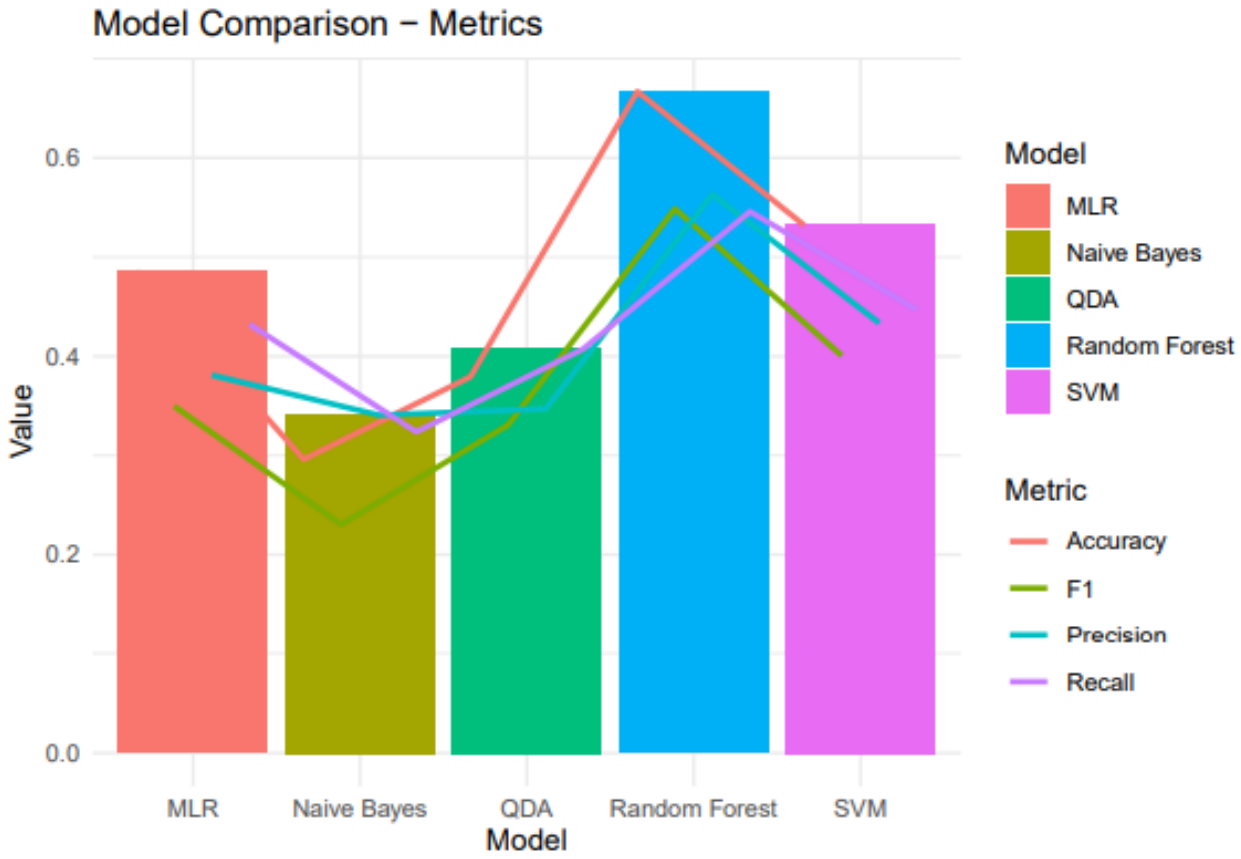
Hình 16: Biểu đồ so sánh Accuracy của dữ liệu merge và dữ liệu của từng loại rượu khi sử dụng RF

Trong phần này, ta xem xét hiệu suất của các mô hình Random Forest khi áp dụng cho từng loại rượu vang riêng biệt (red wine và white wine)
$\Rightarrow$ Như vậy, ta có thể thấy rằng mô hình Random Forest dù áp dụng cho cách xử lý dữ liệu khác nhau (merge, red wine, white wine) đều cho kết quả tương đương nhau, với độ chính xác khoảng 0.6-0.7.

Mục lục

So sánh và lựa chọn mô hình tốt nhất

- Để hiểu rõ hơn về hiệu suất của các mô hình học máy khác nhau, chúng ta cần so sánh các chỉ số như độ chính xác (Accuracy), F1, độ chính xác (Precision) và độ nhạy (Recall).



Hình 17

So sánh và lựa chọn mô hình tốt nhất

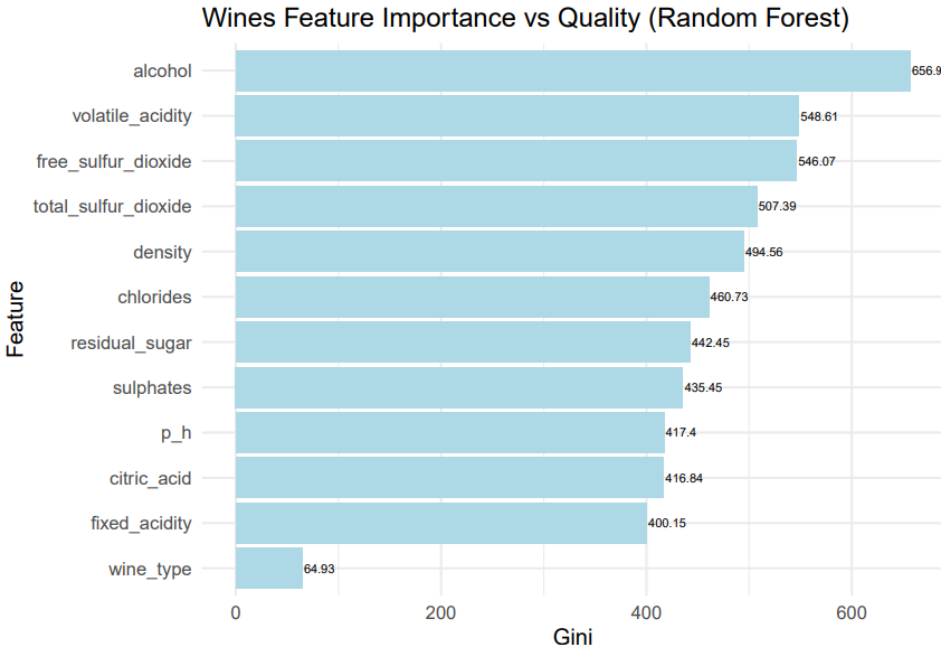
Model	Accuracy	F1	Precision	Recall
MLR	0.4857363	0.3496235	0.3809433	0.4322365
QDA	0.3793369	0.3307033	0.3468991	0.4073364
Random Forest	0.6669237	0.5486088	0.5631412	0.5458295
Naive Bayes	0.2960678	0.2304026	0.3406816	0.3236917
SVM	0.5319969	0.4005731	0.4334528	0.4469603

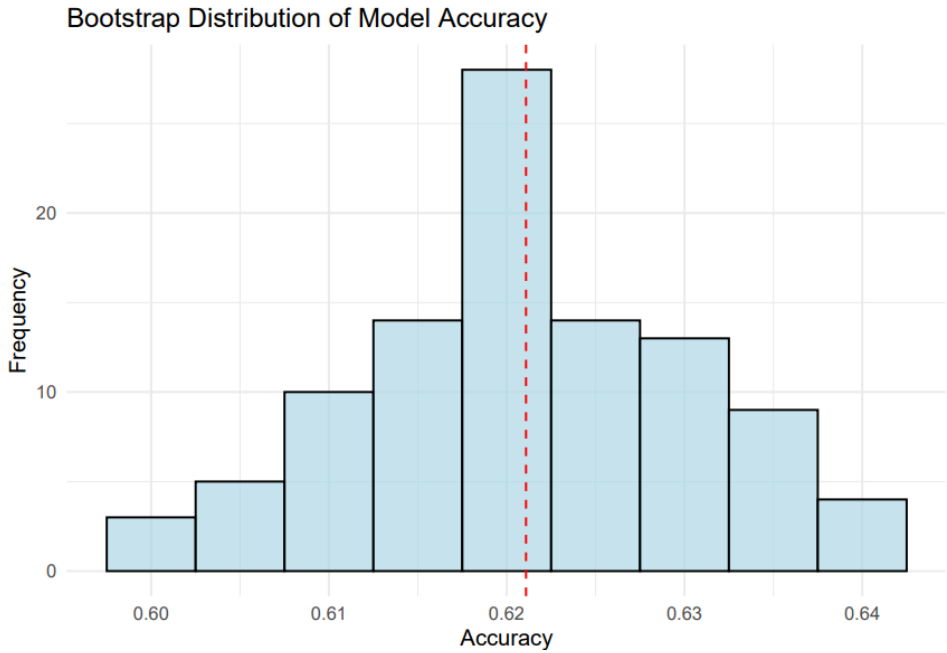
Hình 18

- Trong bài toán hiện tại và các mô hình đã test trên data merge, nhóm quyết định chọn Random Forest là mô hình tốt nhất cho bài toán dự đoán chất lượng rượu vang.

Kết luận của việc lựa chọn mô hình

- Mô hình "Random Forest" đạt được độ chính xác tổng thể khá tốt sau khi được train trên tập dữ liệu đã được xử lý bằng kỹ thuật SMOTE để cân bằng số lượng mẫu giữa các lớp chất lượng rượu vang.
- Phương pháp bootstrap với 100 lần lặp lại cho thấy độ chính xác của mô hình dao động trong khoảng từ 0.60 đến 0.66 $\Rightarrow$ chứng tỏ mô hình có sự ổn định và độ tin cậy khá cao.





Đề xuất cho nhà sản xuất rượu vang

- Nên tập trung đặc biệt vào việc kiểm soát **độ cồn và độ axit bay hơi** trong quá trình sản xuất, vì đây là hai yếu tố quan trọng nhất ảnh hưởng đến chất lượng rượu vang.
- Ngoài ra, việc kiểm soát các yếu tố khác như **độ pH, hàm lượng đường, độ đặc**, là những tính chất cũng quan trọng không kém, đi theo sau nồng độ cồn và axit bay hơi.

Q&A

Thank you for listening!

Any questions for us?