데이터 마이닝 프로젝트

<와인 품질 분류, 예측 모델 수립>

2014170852 산업경영공학부 조영관

목차

<문제 정의>

주제 선정 배경

세상에는 다양한 종류의 술이 있다. 많은 술들이 사람들의 사랑을 받고 있지만, 그 중 고급스러운 이미지를 풍기면서도 오래 전부터 많은 사람들의 사랑을 받아왔던 술이 바로 '와인'이다. 사람들은 오랜 시간동안 '와인'에 대해 연구하고 음미하면서 와인에 등급을 매기기 시작하였다. 와인을 만드는데 사용한 재료, 와인을 구성하는 성분, 숙성 기간 등 다양한 요인들이 와인의 '맛'과 '품질'을 결정하는데 중요하다.

그 중 나는 '와인'의 내적 요소들에 주목하고 싶었다. 숙성 기간, 맛과 같은 주관적인 요소가 아니라 와인을 구성하는 화학적 성분만으로도 사람들이 등급을 매기는 '와인'의 품질을 구분할 수 있을지 궁금하여 위 프로젝트 주제를 선정하고 데이터 분석 프로젝트를 진행하기로 하였다.

탐색하고자 하는 문제

화학적 성분들(입력변수 X)만으로도 인간이 매긴 와인의 품질 등급(입력변수 Y)을 잘 구분하는 분석 모델을 세울 수 있을까?

<데이터 준비>

Data set

Kaggle에서 wine quality data를 가져왔다.

총 1599개의 Observation이 있으며 입력변수 X는 11개, 출력변수 Y는 1개이다.

입력변수의 이름과 출력변수의 이름은 다음과 같다.

Fixed.acidity	Volatile.acidity	Citric.acid	Residual.sugar
Chlorides	Free.sulfur.dioxide	Total.surfur.dioxide	Density
PH	Sulphates	Alcohol	**Quality**

출력변수 X를 이해하기 쉽게 한국어로 번역하면 다음과 같다.

비휘발성 산도	휘발성 산도	구연산	잔당
염화물	비휘발성 이산화황	전체 이산화황	밀도
수소 이온 농도	황산염	알코올	**와인 품질**

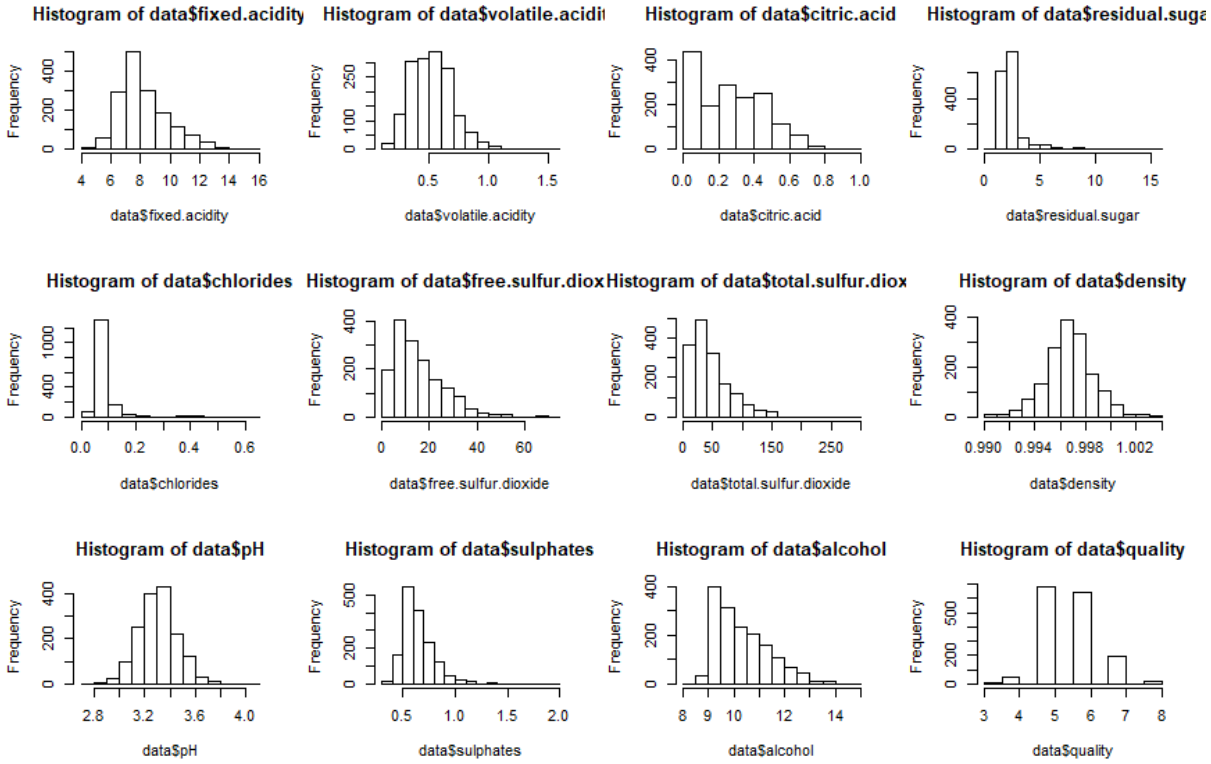
입력변수를 보면 알 수 있듯이, 모두 와인을 구성하는 화학적 성분들이 나열되어 있다.

이 성분들이 와인을 이루어 맛과 품질을 결정한다.

데이터 셋을 자세히 살펴보자.

데이터 특징 살펴보기

입력변수 X와 출력변수 Y가 어떤 분포를 이루는지 확인해보자.

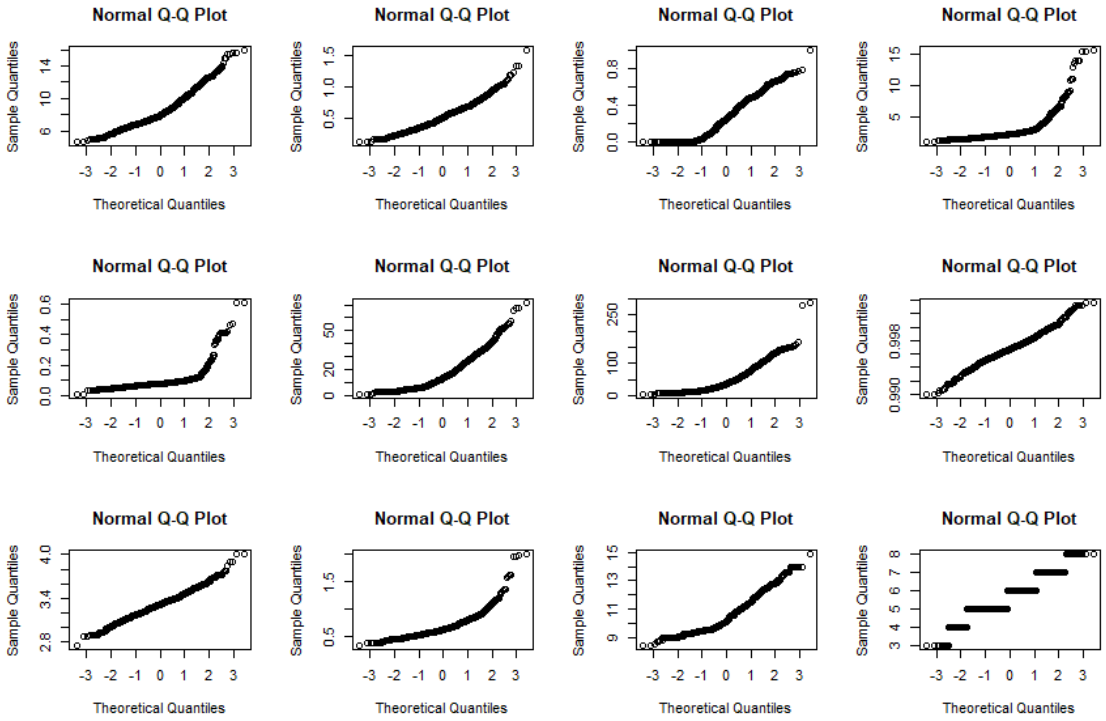


전체적으로 데이터의 분포를 살펴보자.

Fixed.acidity(비휘발성 산도), Volatile.acidity(휘발성 산도), free.sulfur.dioxide(비휘발성 이산화황), total.sulfur.dioxide(전체 이산화황), density(밀도), pH(수소 이온 농도), sulphates(황산염), alcohol(알코올)이 조금은 한쪽으로 치우쳐 있지만 전체적으로 정규분포의 Bell shape을 따름을 확인할 수 있다.

나머지 3개 Chlorides, Residual.sugar, Citric.acid는 Bell shape을 띄지는 않아 정규분포임을 판단하기 어렵다. 위 3개는 qqplot을 통해 다시 확인해보도록 하자.

출력변수 Y인 quality의 경우 연속형 값이 아니라 범주형 값이기 때문에 위와 같은 모양이 나왔지만 분포 자체는 정규분포의 모양을 이룸을 확인할 수 있다.



정규성을 정확히 확인해보기 위해 qqplot을 도시하였다.

변수는 히스토그램과 동일하게 순서대로 나열하였다.

Q-Q plot을 통해 확인한 결과 Quantile 범주 (X축) -2~2에서 (약 95%의 데이터셋을 포함한다) 데이터들이 정규분포를 이루는 선 위에 잘 분포함을 확인할 수 있다.

다음은 입력변수 X간의 상관관계를 파악해보자.

```
> cor(wine_x)
                      fixed.acidity volatile.acidity citric.acid residual.sugar
fixed.acidity           1.00000000     -0.256130895  0.67170343    0.114776724
volatile.acidity       -0.25613089      1.000000000 -0.55249568    0.001917882
citric.acid             0.67170343     -0.552495685  1.00000000    0.143577162
residual.sugar          0.11477672      0.001917882  0.14357716    1.000000000
chlorides               0.09370519      0.061297772  0.20382291    0.055609535
free.sulfur.dioxide    -0.15379419     -0.010503827 -0.06097813    0.187048995
total.sulfur.dioxide   -0.11318144      0.076470005  0.03553302    0.203027882
density                 0.66804729      0.022026232  0.36494718    0.355283371
pH                     -0.68297819      0.234937294 -0.54190414   -0.085652422
sulphates               0.18300566     -0.260986685  0.31277004    0.005527121
alcohol                -0.06166827     -0.202288027  0.10990325    0.042075437
                      chlorides free.sulfur.dioxide total.sulfur.dioxide     density
fixed.acidity         0.093705186        -0.153794193          -0.11318144  0.66804729
volatile.acidity      0.061297772        -0.010503827           0.07647000  0.02202623
citric.acid           0.203822914        -0.060978129           0.03553302  0.36494718
residual.sugar        0.055609535         0.187048995           0.20302788  0.35528337
chlorides             1.000000000         0.005562147           0.04740047  0.20063233
free.sulfur.dioxide   0.005562147         1.000000000           0.66766645 -0.02194583
total.sulfur.dioxide  0.047400468         0.667666450           1.00000000  0.07126948
density               0.200632327        -0.021945831           0.07126948  1.00000000
pH                   -0.265026131         0.070377499          -0.06649456 -0.34169933
sulphates             0.371260481         0.051657572           0.04294684  0.14850641
alcohol              -0.221140545        -0.069408354          -0.20565394 -0.49617977
                             pH    sulphates     alcohol
fixed.acidity        -0.68297819  0.183005664 -0.06166827
volatile.acidity      0.23493729 -0.260986685 -0.20228803
citric.acid          -0.54190414  0.312770044  0.10990325
residual.sugar       -0.08565242  0.005527121  0.04207544
chlorides            -0.26502613  0.371260481 -0.22114054
free.sulfur.dioxide   0.07037750  0.051657572 -0.06940835
total.sulfur.dioxide -0.06649456  0.042946836 -0.20565394
density              -0.34169933  0.148506412 -0.49617977
pH                    1.00000000 -0.196647602  0.20563251
sulphates            -0.19664760  1.000000000  0.09359475
alcohol               0.20563251  0.093594750  1.00000000
```

Fixed.acidity와 citric.acid, Fixed.acidity와 density, Fixed.acidity와 PH가 상관관계가 0.6~0.7 사이로 높은 편임을 확인할 수 있다. 다시 말해, Fixed.acidity는 위 3개의 변수와 관련성이 높은 변수임을 확인할 수 있다. 다만, 예측 모델 성능의 향상을 위해 변수를 제거하지는 않고 그대로 데이터분석을 진행하도록 하겠다. 그 이외의 변수들은 상관관계가 높지 않아 서로 독립적임을 파악할 수 있다.

위와 같이 데이터 분포의 특성을 살펴본 결과, 입력변수 X 11개와 출력변수 Y가 전체적으로 정규성, 등분산성, 독립성을 어느 정도 만족하므로, 다음 단계인 데이터 분석을 진행함에 있어 문제가 없음을 확인할 수 있다.

데이터 전처리

총 6가지 분석모델을 활용하였다. 이 6가지를 크게 3가지로 나눈다.

1) 회귀 – Multiple Linear Regression, KNN, ANN
2) 분류 – Logistic Regression, Classification Decision tree, KNN
3) Unsupervised Learning – Clustering (출력변수 Y 없다고 가정)

각각의 분석 모델의 특징에 맞게 데이터 전처리를 진행하였다.

회귀 모델의 경우, 출력변수 Y를 연속형 값으로 변환하고, 입력변수 11개 모두 활용해 데이터 분석을 진행하였다. 결측치와 이상치는 제거하였다.

분류 모델의 경우, 출력변수 Y를 binary variable로 바꾸었다. Quality 값이 3, 4, 5, 6 인 경우 0 (보통 품질), 7,8인 경우 1 (고급 품질)로 구분해 데이터를 변환한 후 데이터 분석을 진행하였다. 결측치와 이상치는 제거하였다.

마지막으로 출력변수 Y가 없다고 가정하고 (인간이 매긴 품질 등급 없이) 군집 분석을 통해 보통 품질(0)과 고급 품질(1)이 다른 군집으로 비교적 잘 나눠지는지 확인해본다.

<데이터 분석>

Multiple Linear Regression

Multiple Linear Regression 분석을 진행하여 모델을 구축하였다.

그 결과 다음과 같았다.

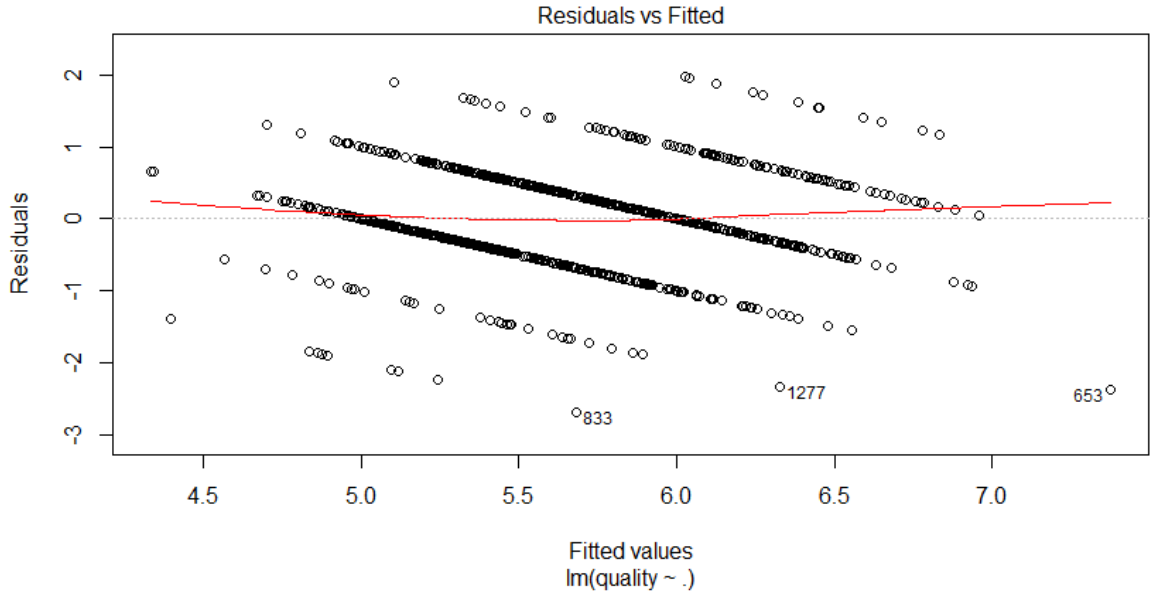
```
Call:
lm(formula = quality ~ ., data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.68181 -0.35756 -0.04335  0.45781  1.97326

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.422e+01  2.572e+01   1.331  0.18356
fixed.acidity         1.773e-02  3.170e-02   0.559  0.57600
volatile.acidity     -1.074e+00  1.413e-01  -7.600 6.27e-14 ***
citric.acid          -4.102e-02  1.752e-01  -0.234  0.81489
residual.sugar        1.568e-02  1.841e-02   0.852  0.39439
chlorides            -1.537e+00  5.246e-01  -2.929  0.00347 **
free.sulfur.dioxide   2.588e-03  2.592e-03   0.999  0.31818
total.sulfur.dioxide -2.574e-03  8.436e-04  -3.051  0.00233 **
density              -3.038e+01  2.627e+01  -1.157  0.24765
pH                   -3.047e-01  2.405e-01  -1.267  0.20544
sulphates             8.876e-01  1.405e-01   6.316 3.89e-10 ***
alcohol               2.627e-01  3.225e-02   8.146 1.01e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6543 on 1107 degrees of freedom
Multiple R-squared:  0.3477,    Adjusted R-squared:  0.3412
F-statistic: 53.63 on 11 and 1107 DF,  p-value: < 2.2e-16
```

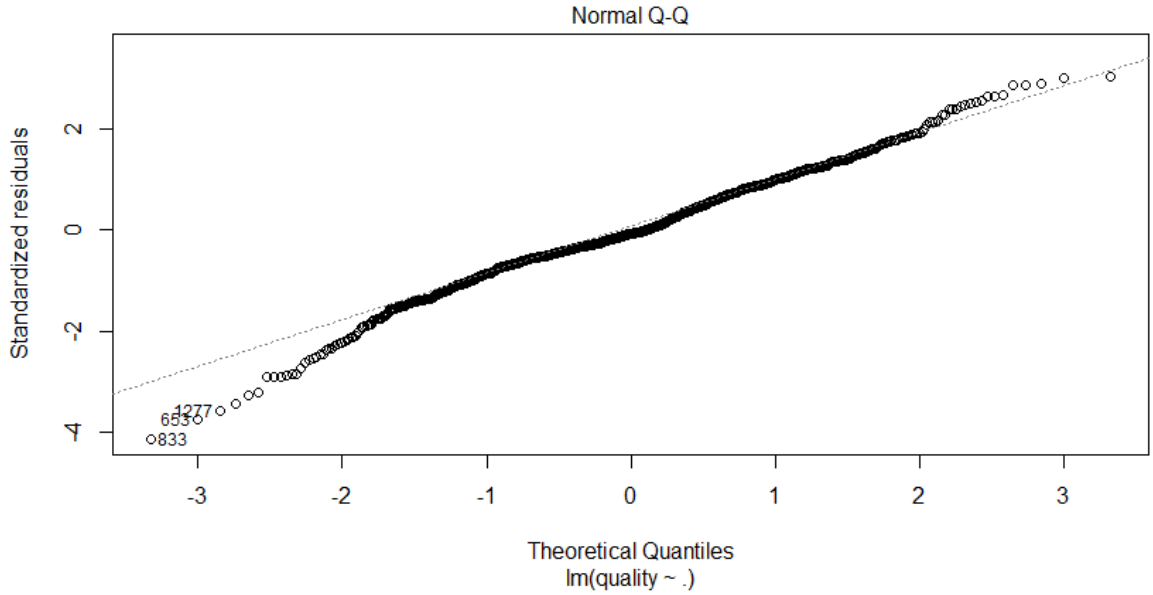
Adjusted R^2 값이 0.3412로 비교적 낮음을 알 수 있다.

즉, Multiple Linear Regression model로는 위 data set에 대해 예측 성능이 좋지 않음을 알 수 있다. 각각 계수의 p-value도 살펴보자.

Volatile.acidity, Chlorides, Total.sulfur.dioxide, Sulphates, Alcohol 이렇게 5가지만이 유의수준 0.05에서 유의하며 나머지 변수들은 Y값에 큰 영향을 미치지 않음을 확인할 수 있다.

성능 향상을 위한 변수 선택을 하기 전에, 모델의 정규성과 등분산성을 먼저 확인하자.





위 두개 plot 중 위인 Residual plot의 경우 Y값이 3,4,5,6,7,8인 이산형 변수라 위와 같은 형태로 나왔지만, 잔차가 불규칙하게 잘 퍼져서 분포해 있음을 확인할 수 있다.

아래 Plot은 구축한 Multiple Linear Regression 모델이 정규성을 띄는지 확인하는 plot이다. 위의 표를 보면 점이 -2~2까지 선 위에 잘 분포되어 있음을 확인할 수 있다.

따라서 정규성, 등분산성을 잘 만족하는 다중 선형 회귀 모델이 구축되었다.

그래서 Forward Selection 방식을 통해 변수선택을 한 후 다시 모델을 구축하였다.

그 결과 다음과 같았다.

```
Call:
lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
    total.sulfur.dioxide + chlorides + pH, data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.64092 -0.36370 -0.05567  0.46433  1.96660

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.9480431  0.4942829   7.987 3.42e-15 ***
alcohol             0.2904538  0.0203726  14.257  < 2e-16 ***
volatile.acidity   -1.0862834  0.1171970  -9.269  < 2e-16 ***
sulphates           0.8302867  0.1347101   6.164 9.94e-10 ***
total.sulfur.dioxide -0.0019421  0.0005844  -3.323  0.00092 ***
chlorides          -1.5624257  0.5034927  -3.103  0.00196 **
pH                 -0.3251251  0.1425231  -2.281  0.02272 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6539 on 1112 degrees of freedom
Multiple R-squared:  0.3455,    Adjusted R-squared:  0.342
F-statistic: 97.86 on 6 and 1112 DF,  p-value: < 2.2e-16
```

위 결과를 살펴보자.

R^2값은 약간 낮아졌음을 확인할 수 있다. R^2는 변수가 추가될수록 높아지므로 당연하다. 하지만 Adjusted R^2값은 오히려 약간 상승했음을 확인할 수 있다.

즉, 유의미한 변수를 제거한 결과 약간 더 예측 성능이 향상된 모델을 확인할 수 있다.

성능을 확인할 수 있는 지표인 오류율도 살펴보자.

	RMSE	MAE	MAPE
MLR	0.6372469	0.5059616	9.251204

RMSE, MAE, MAPE(%) 모두 비교적 작은 오류를 보임을 확인할 수 있다.

결론을 정리하자면, 오류율이 낮은 편이고 학습이 빠른 모델이지만 예측 성능이 좋지 않아 위 Data set에 적합하지 않은 분석 방법임을 확인할 수 있다.

Logistic Regression

위의 Multiple Linear Regression을 학습해본 결과, 회귀 모델의 정확도가 낮음을 확인할 수 있었다. 그래서 분류 모델로 학습시키면 적합할지 궁금하였다.

그래서 로지스틱 회귀분석 모델 구축을 진행하였다.

그 결과 다음과 같았다.

```
Call:
glm(formula = quality ~ ., family = binomial, data = wine_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9102  -0.4409  -0.2276  -0.1118   2.9682

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          -2.832806   0.171327 -16.534  < 2e-16 ***
fixed.acidity         0.299750   0.270744   1.107  0.26824
volatile.acidity     -0.528932   0.168497  -3.139  0.00169 **
citric.acid          -0.002716   0.202271  -0.013  0.98929
residual.sugar        0.200885   0.130628   1.538  0.12409
chlorides            -0.339071   0.179828  -1.886  0.05936 .
free.sulfur.dioxide   0.172160   0.157273   1.095  0.27367
total.sulfur.dioxide -0.719873   0.221927  -3.244  0.00118 **
density              -0.132593   0.254412  -0.521  0.60225
pH                   -0.027138   0.183033  -0.148  0.88213
sulphates             0.604459   0.119340   5.065 4.08e-07 ***
alcohol               1.047612   0.175904   5.956 2.59e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 907.54  on 1118  degrees of freedom
Residual deviance: 623.21  on 1107  degrees of freedom
AIC: 647.21

Number of Fisher Scoring iterations: 6
```

결과를 살펴보자.

유의수준 0.05로 잡고 각각의 변수의 p-value를 확인해보자.

Volatile.acidity, Total.sulfur.dioxide, Sulphates, Alcohol 이렇게 4가지 변수가 유의하고 중요한 변수임을 확인할 수 있다.

선택된 변수의 결과는 Multiple Linear Regression에서 얻어낸 결과와 거의 유사함을 확인할 수 있다.

분류 모델이므로 Confusion Matrix를 통해 분류 성능을 확인해보자.

```
          lr_predicted
lr_target   0   1
        0 400  20
        1  38  22
```

결과는 위와 같다. 전체적으로 accuracy는 높아 보이지만, 실제 target이 1일 때의 경우를 보자. 실제 값이 1임에도 1로 예측하지 않고 0으로 예측한 경우가 38개나 된다.

1로 제대로 분류한 경우는 22개에 불과하다.

즉, Recall 값이 작을 것임을 예측할 수 있다. 실제로 계산한 결과를 살펴보자.

```
                   TPR (Recall) Precision       TNR       ACC       BCR        F1
Logstic Regression    0.3666667 0.5238095 0.952381 0.8791667 0.5909368 0.4313725
```

결과는 위와 같다.

Accuracy는 0.879로 비교적 높은 수치를 보이지만 그 이외의 값들이 높지 않다.

Recall 값이 0.366으로 비교적 작으며 Precision도 0.524로 높지 않다.

BCR과 F1도 각각 0.591, 0.431로 높지 않음을 확인할 수 있다.

결론적으로 위 Logistic regression 모델도 예측 성능이 좋지 않음을 확인할 수 있다.

그래도 Multiple Linear Regression 모델보다는 예측 성능이 조금은 상승했음을 확인할 수 있다.

따라서, 위 두 개의 분석 모델을 비교했을 때, 위의 data set은 회귀 모델인 Multiple Linear Regression을 사용하는 것보다 Logistic Regression 모델을 사용하는 것이 더 적합함을 확인할 수 있다.

Clustering

Y값이 없다고 가정하고 Data set을 준비한 후에 Clustering을 진행하였다.

보통 품질과 고급 품질로 잘 분류될지, 아니면 다른 어떤 특징으로 구분이 될지를 확인해보기 위해 일단 군집을 크게 두 개로 나누어 진행해보았다.

Clustering 방식 중 내가 진행했던 방식은 K-means Clustering이다.

K-means clustering의 특성상 직접 하이퍼 파라미터인 k를 지정해주어야 한다.

기대하는 결과가 있어 결론적으로는 K=2로 진행을 할 것이지만, validation 지표 internal과 stability를 활용했을 때 최적 k를 찾아주는 여러 개의 결정 방식들은 얼마가 최적의 k인지를 추출해주는지 궁금하여 확인해보기로 하였다.

그 결과 다음과 같았다.

```
Optimal Scores:

             Score    Method Clusters
APN            0.0855 kmeans 2
AD             3.3054 kmeans 10
ADM            0.4298 kmeans 2
FOM            0.8583 kmeans 8
Connectivity 297.9528 kmeans 3
Dunn           0.0630 kmeans 10
Silhouette     0.2119 kmeans 3
```

위 결과를 살펴보자.

10개와 8개의 클러스터는 사실 상 의미가 없다.

K=2, 3 정도가 최적의 K임을 확인할 수 있다.

K=2로 설정하여 K-means Clustering을 진행해보자.

	cluster 1	cluster 2
fixed.acidity	0.12398064	-0.22504233
volatile.acidity	-0.04075037	0.07396766
citric.acid	0.00758886	-0.01377485
residual.sugar	-0.11320475	0.20548256
chlorides	-0.07645659	0.13877947
free.sulfur.dioxide	-0.54268504	0.98504978
total.sulfur.dioxide	-0.56763228	1.03033254
density	-0.01930591	0.03504294
pH	-0.01977544	0.03589521
sulphates	-0.07634085	0.13856940
alcohol	0.12657747	-0.22975593

위는 클러스터가 두 개로 분류된 후에, 각각의 클러스터의 개별 변수 별 중심 점을 보여주는 결과이다.

```
> wine_kmc$size
[1] 1031  568
```

위 결과는 두 개의 군집에 분류된 관측치의 개수이다.

군집 1은 1031개의 관측치가 있으며, 군집 2는 568개의 관측치가 있다.

각각의 군집에는 Y값에 따라 얼마나 분류되는지 확인해보자.

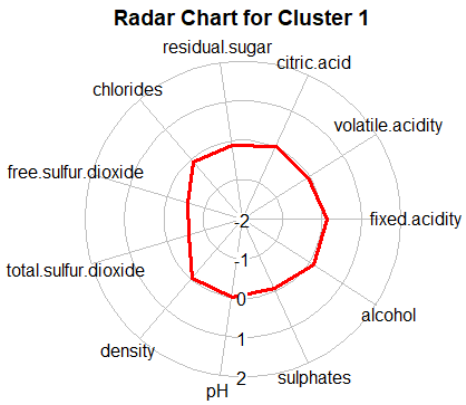
real_class	kmc_cluster 1	2
3	8	2
4	40	13
5	370	311
6	442	196
7	157	42
8	14	4

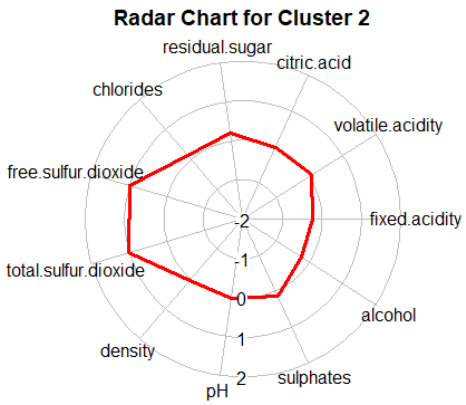
위의 결과를 살펴보면, 명확하지는 않더라도 구분되는 특징을 하나 발견할 수 있다.

군집 1의 경우, 고급 품질인 7과 8이 더 많이 분포함을 확인할 수 있다. 군집 1이 군집 2에 비해 관측치 수가 2배임을 감안해도 더 많다.

즉, 군집 1이 품질 등급이 높은 관측치의 비율이 많으며 군집 2는 상대적으로 비율이 적음을 파악할 수 있다.

Radar Chart에 두 군집을 도시해보자.





결과는 위와 같다.

두 군집의 차이를 개별 변수들을 확인하며 살펴보자.

대체로 대부분의 변수들이 군집1과 2에서 비슷한 특성을 보인다.

그런데 Free.sulfur.dioxide와 Total.sulfur.dioxide이 두가지에서 큰 차이를 보임을 확인할 수 있다. 흥미로운 점은 Total.sulfur.dioxide 변수가 앞에서 진행했던 두 모델에서도 와인의 품질을 결정하는데 중요한 변수라는 것이다.

(Multiple Linear Regression과 Logistic Regression에서도 Total.sulfur.dioxide 변수가 유의수준 0.05 내에서 중요한 변수였다.)

그래서 위의 Y값에 따라 분류했을 때 군집 1이 품질 등급이 높은 (6,7,8) 관측치의 비율이 비교적 더 높았던 것이다.

군집이 유의하게 잘 분류되었는지 성능을 확인해보자.

T-test를 통해 군집 1과 2를 비교하였다.

그 결과 다음과 같았다.

	V1
1	2.421747e-13
2	2.398864e-02
3	6.695208e-01
4	2.433359e-07
5	5.197858e-04
6	3.285799e-166
7	1.101989e-175
8	3.031902e-01
9	2.856202e-01
10	2.372389e-04
11	3.482688e-12

위의 값은 개별 변수의 T-test에 대한 p-value 값이다.

유의수준 0.05에서 11개의 변수 모두 유의함을 확인할 수 있다.

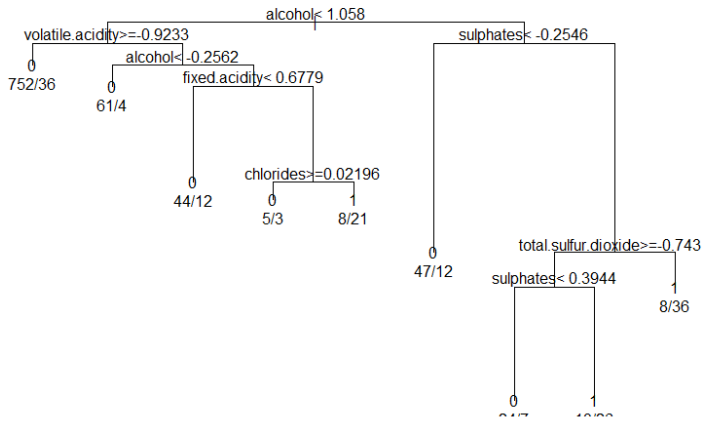
따라서 군집1과 군집2가 유의하게 잘 분류되었음을 확인할 수 있다.

결론을 정리하면, 출력변수 Y가 없다고 가정하고 군집분석을 진행한 결과 Total.sulfur.dioxide 변수를 기준으로 군집1과 군집2로 유의하게 분류되었음을 확인할 수 있었다. 그리고 군집 1은 품질 등급이 높은(6, 7, 8) 관측치의 비율이 높고 군집 2는 그렇지 않았음을 확인할 수 있다.

Decision Tree

다음으로 진행할 분석 모델은 의사결정나무다.

사용한 패키지는 Rpart 패키지이다. (Split 비용함수는 Gini index 이용)



의사결정나무 모델 학습 결과 위와 같았다.

분류 의사결정나무 이므로 출력변수 Y값은 보통 품질(0), 고급 품질(1)로 처리하여 분석을 진행하였다.

위 의사결정나무를 통해 중요 변수를 확인할 수 있다.

Alcohol, volatile.acidity, fixed.acidity, sulphates, chlorides, total.sulfur.dioxide 이렇게 6가지가 중요한 변수이다.

Terminal node는 9개이며, 비교적 잘 분류가 된 불순도가 낮은 terminal node도 있고 불순도(impurity)가 높은 terminal node도 있다.

분류 모델이므로 confusion matrix를 통해 분류 성능을 확인해보자.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 396  34
         1  24  26

               Accuracy : 0.8792
                 95% CI : (0.8466, 0.907)
    No Information Rate : 0.875
    P-Value [Acc > NIR] : 0.4245

                  Kappa : 0.4051

 Mcnemar's Test P-Value : 0.2373

            Sensitivity : 0.9429
            Specificity : 0.4333
         Pos Pred Value : 0.9209
         Neg Pred Value : 0.5200
             Prevalence : 0.8750
         Detection Rate : 0.8250
   Detection Prevalence : 0.8958
      Balanced Accuracy : 0.6881

       'Positive' Class : 0
```

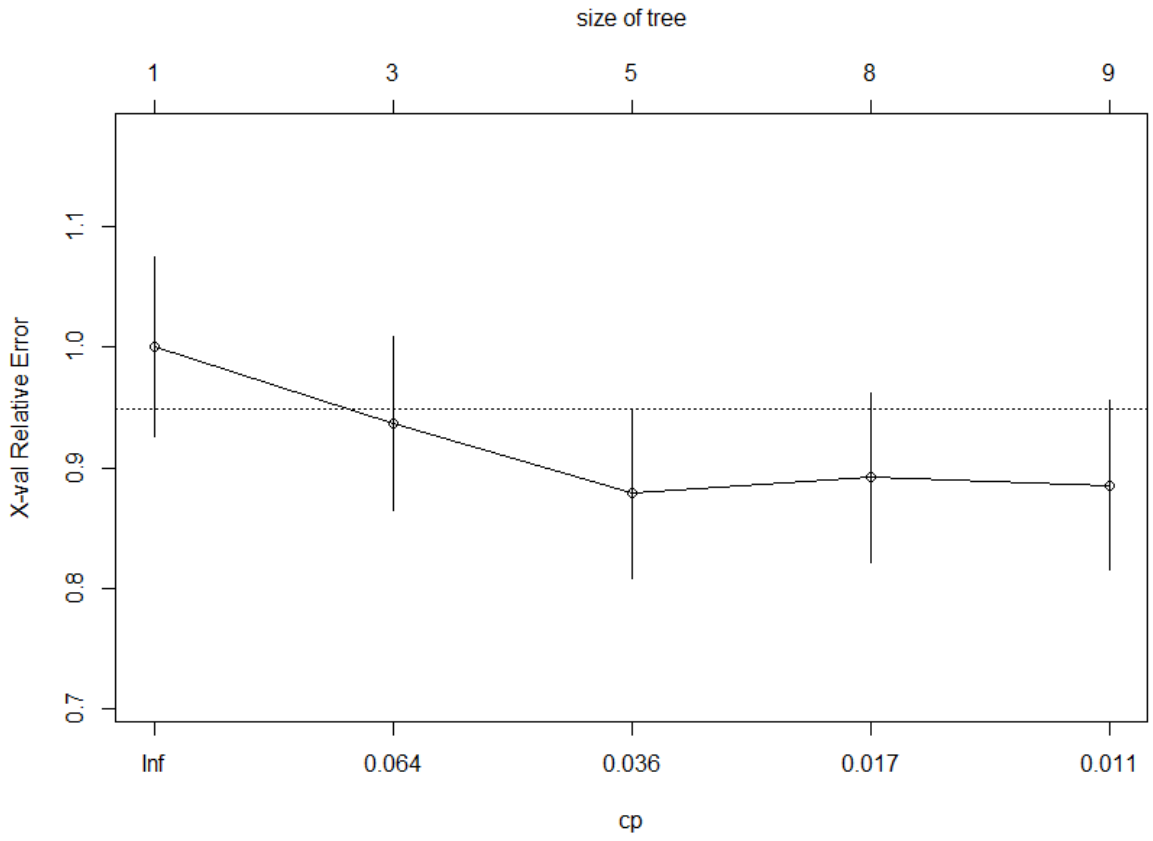
위의 결과를 살펴보자.

전체적인 Accuracy는 0.8792로 높은 편이다.

하지만 다른 지표 값들이 높지 않다.

Recall 값은 0.4333이며 Precision 값은 0.52 이다.

위에서 사용했던 분류 모델인 Logistic Regression과 성능이 비슷함을 확인할 수 있다.

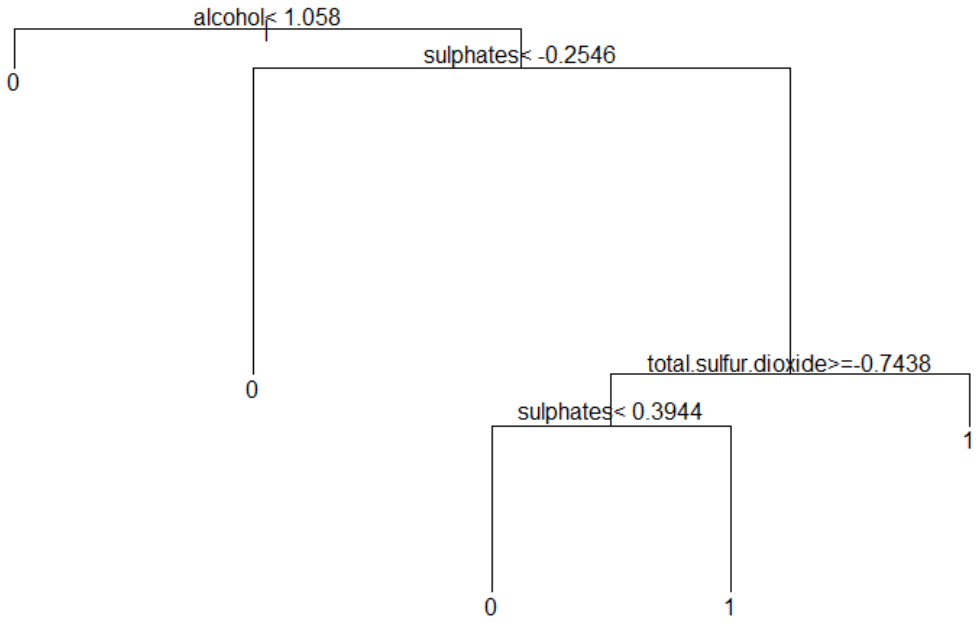


그런데, 위의 tree는 terminal node 수와 가지가 너무 많다. 그래서 과적합(Overfitting)의 위험이 있고 해석의 용이성이 떨어진다. 그래서 Pruning을 진행하도록 한다.

위의 표는 terminal node 수가 몇 개가 적절할지를 확인하는 표이다.

결과를 보면 terminal node 수 5개일 때 Error가 제일 작음을 확인할 수 있다.

따라서 terminal node 수 5개에 맞추어 가지치기를 진행한다.



가지치기를 진행한 결과는 위와 같다.

Terminal node 수가 5개로 줄었고, tree도 훨씬 간결 해졌다.

해석도 더 용이하다. Alcohol이 1.058보다 작으면 보통 품질(0)으로 분류되며, Alcohol이 1.058보다 크지만 sulphates가 -0.2546 보다 작으면 보통 품질(0)으로 분류된다.

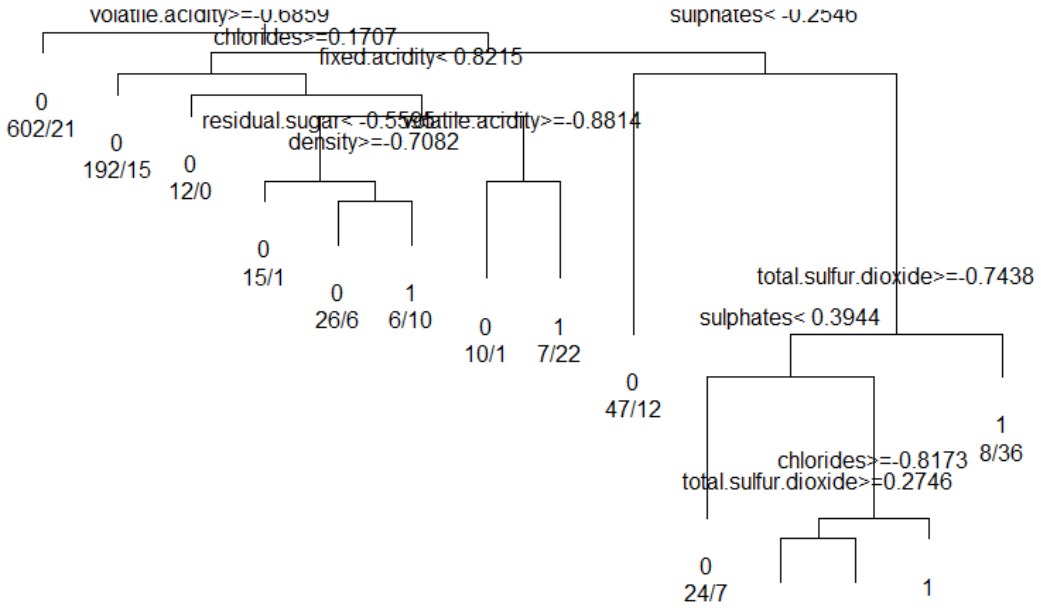
(참고로 분석을 진행하기 전에 입력변수 X들은 모두 Scale (정규화)하였다.)

Sulphates가 -0.2546보다 크고 total.sulfur.dioxide가 -0.7438보다 작으면 고급 품질(1)로 분류된다. Total.sulfur.dioxide가 -0.7438보다 크고 sulphates가 0.3944보다 작으면 보통 품질(0) 크면 고급 품질(1)로 분류된다.

다음으로 Tree를 한 개 더 구축해보았다.

위의 경우에 Split을 할 때의 비용함수를 Gini index를 이용해 측정했지만, 이번에는 Entropy를 이용한 Information gain을 Split 기준으로 설정하여 Tree 모델을 학습하였다.

그 결과는 다음과 같다.



위 결과를 살펴보자.

Entropy를 이용해 split을 한 결과 Gini index를 사용했을 때 보다 더 복잡한 tree가 만들어졌다. Terminal node 수가 14개이다. 위 분류를 진행하는데 사용된 중요 변수는 다음과 같다. Volatile.acidity, chlorides, fixed.acidity, residual.sugar, density, total.sulfur.dioxide, sulphates 총 7가지가 위 분류를 함에 있어 중요한 변수로 선택되었다.

Confusion matrix를 활용해 분류 성능을 확인해보자.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 399  28
         1  21  32

               Accuracy : 0.8979
                 95% CI : (0.8673, 0.9235)
    No Information Rate : 0.875
    P-Value [Acc > NIR] : 0.07072

                  Kappa : 0.5088

 Mcnemar's Test P-Value : 0.39137

            Sensitivity : 0.9500
            Specificity : 0.5333
         Pos Pred Value : 0.9344
         Neg Pred Value : 0.6038
             Prevalence : 0.8750
         Detection Rate : 0.8313
   Detection Prevalence : 0.8896
      Balanced Accuracy : 0.7417

       'Positive' Class : 0
```

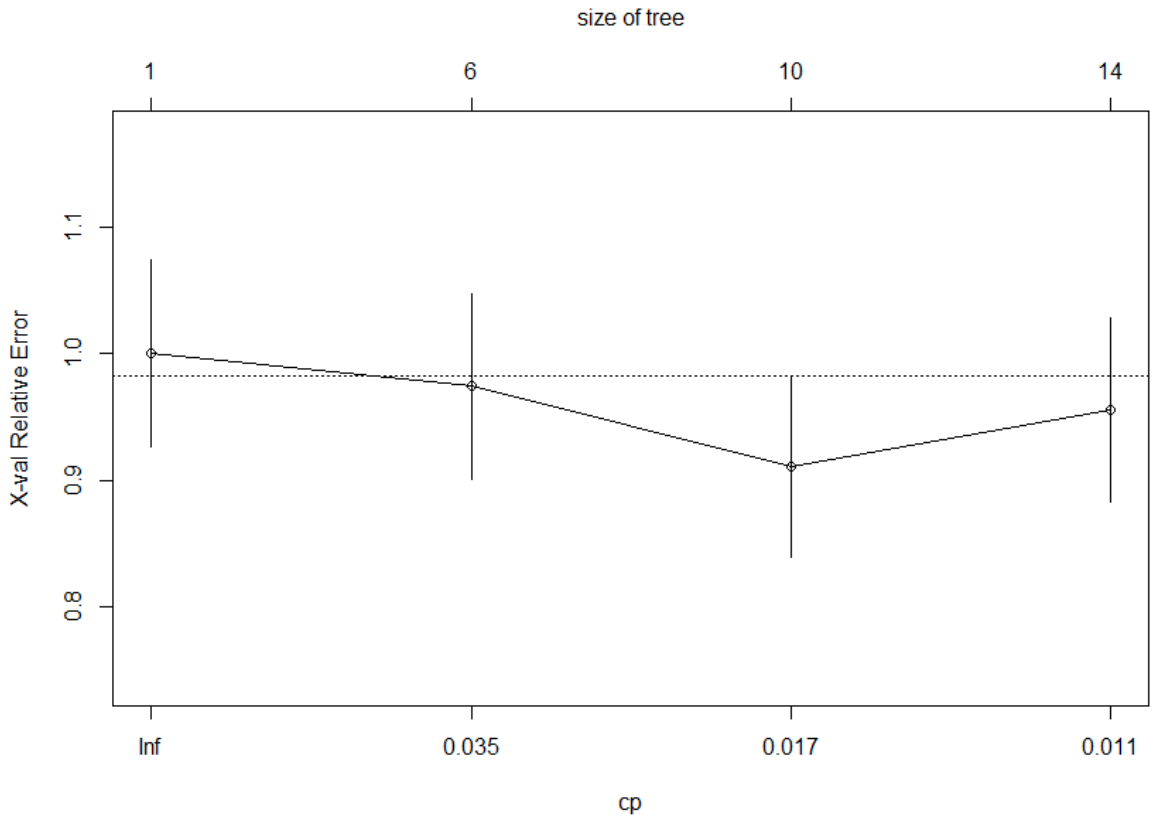
Accuracy 값은 0.8979로 Gini index를 사용했을 때 보다 높다.

Recall 값은 0.533, Precision 값은 0.604로 이 값 역시 Gini index를 사용했을 때 보다 높다. Balanced Accuracy 값도 0.742로 비교적 높은 편이다.

즉, Gini index를 split 함수로 사용했을 때 보다 entropy를 split 함수로 이용했을 때 예측 성능이 더 좋아졌다.

그런데 위 tree도 terminal node가 14개나 되어서 너무 과적합(Overfitting) 되어있는 상태이다. 해석도 용이하지 않다. 따라서 가지치기를 할 필요가 있다.

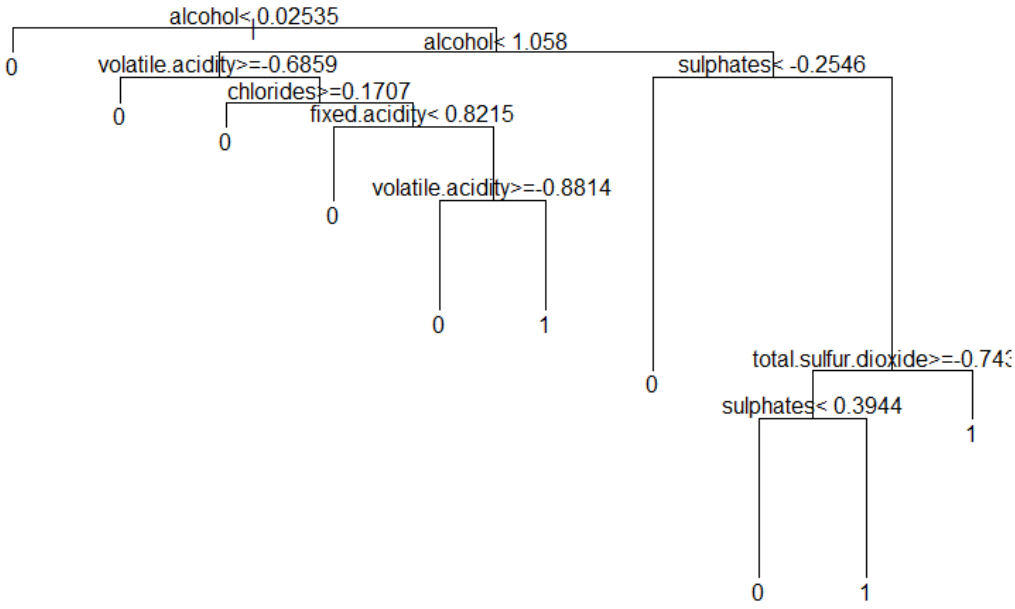
가지치기를 진행하기 위해, 적합한 terminal node 수를 판별하자.



위의 결과를 살펴보자.

가지치기를 통해 적절한 terminal node 수는 10개다.

Terminal node 수가 10개일 때 error가 가장 낮다. 모델을 구축해보자.



모델 구축 결과는 위와 같다.

가지치기 전의 terminal node 14개인 경우보다는 tree가 비교적 단순하다.

그래도, Gini index를 사용한 tree에 가지치기를 했을 때 보다는 복잡하다.

위 tree의 분류 성능을 confusion matrix를 통해 확인해보자.

```
     CART.predict
        0   1
   0  398  22
   1   32  28
```

```
           TPR Precision       TNR Accuracy       BCR F1-Measure
CART 0.4666667      0.56 0.947619   0.8875 0.6649979  0.5090909
```

예측 성능 결과는 위와 같다.

Accuracy는 0.8875로 가지치기 전과 비교해 아주 약간 하락했다.

Recall은 0.467로, Precision은 0.56으로 가지치기 전과 비교해 약간 하락했다.

Pruning 전의 모델이 더 세부적으로 분류하므로 예측 성능이 더 좋은 결과임은 당연하다. 하지만, Pruning 후의 모델 성능이 Pruning 전과 비교해 비교적 작게 하락하였고 tree 모델도 더 단순하므로 Pruning 후의 모델이 더 좋음을 알 수 있다.

이는 Gini index를 적용한 tree에서도 마찬가지다. Pruning 후의 tree 모델이 더 좋은 모델이다.

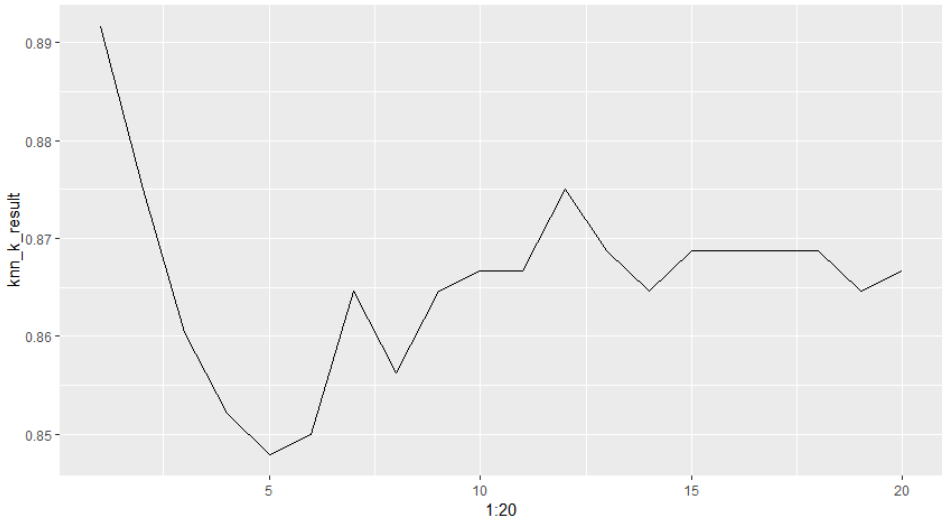
결론을 정리하면, Decision tree의 경우 split 함수를 두 가지를 활용하여 분석을 진행하였다. 그 결과, Gini index로 split을 진행한 tree의 경우 terminal node 수가 5개밖에 되지 않아 해석이 용이하며, 분류를 함에 있어 중요한 변수의 개수를 줄일 수 있었다 (차원 축소). 반면에 entropy index로 split을 진행한 tree의 경우, terminal node 수가 10개로 Gini와 비교해 비교적 많다. 그래서 Gini index를 이용한 tree에 비해 해석이 복잡하지만 예측 성능은 더 좋음을 확인할 수 있었다.

K-Nearest Neighbors

다음은 KNN 분석을 진행하였다.

KNN 분석의 경우 Regression KNN과 Classification KNN 두 가지 다 진행하였다.

먼저 Classification KNN을 진행한 결과를 살펴보자.



위 결과는 K를 1부터 20까지 변화시켰을 때 예측 성능의 변화를 보여주는 결과다.

K=1,2,3 일 때의 경우 사실상 과적합(Overfitting) 모델이라 accuracy가 매우 높을 수밖에 없다. 따라서 위 3개의 경우는 제외하고 생각한다.

위의 그래프의 경우는 accuracy만 보여준 것이므로 다른 지표들도 살펴보자.

	TPR (Recall)	Precision	TNR	ACC	BCR	F1
knn1	0.6500000	0.5571429	0.9261905	0.8916667	0.7759019	0.6000000
knn2	0.3833333	0.6216216	0.9666667	0.8937500	0.6087327	0.4742268
knn3	0.4333333	0.4561404	0.9261905	0.8645833	0.6335213	0.4444444
knn4	0.3166667	0.4750000	0.9500000	0.8708333	0.5484828	0.3800000
knn5	0.4000000	0.4000000	0.9142857	0.8500000	0.6047432	0.4000000
knn6	0.2833333	0.4473684	0.9500000	0.8666667	0.5188127	0.3469388
knn7	0.3833333	0.4509804	0.9333333	0.8645833	0.5981453	0.4144144
knn8	0.3166667	0.4523810	0.9452381	0.8666667	0.5471064	0.3725490
knn9	0.3666667	0.4583333	0.9380952	0.8666667	0.5864881	0.4074074
knn10	0.3333333	0.4651163	0.9452381	0.8687500	0.5613193	0.3883495
knn11	0.3500000	0.4565217	0.9404762	0.8666667	0.5737305	0.3962264
knn12	0.3166667	0.5277778	0.9595238	0.8791667	0.5512252	0.3958333
knn13	0.3500000	0.4666667	0.9428571	0.8687500	0.5744563	0.4000000
knn14	0.2833333	0.4594595	0.9523810	0.8687500	0.5194625	0.3505155
knn15	0.3000000	0.4500000	0.9476190	0.8666667	0.5331845	0.3600000
knn16	0.2833333	0.4857143	0.9571429	0.8729167	0.5207595	0.3578947
knn17	0.2833333	0.4594595	0.9523810	0.8687500	0.5194625	0.3505155
knn18	0.2166667	0.4642857	0.9642857	0.8708333	0.4570871	0.2954545
knn19	0.2333333	0.4242424	0.9547619	0.8645833	0.4719934	0.3010753
knn20	0.1833333	0.3793103	0.9571429	0.8604167	0.4188988	0.2471910

Recall의 경우 5, 7일 때 높다. Precision의 경우 전체적으로 다 비슷하다. BCR의 경우 5,7,9 일 때 높으며 F1의 경우 5,7,9,12가 높다.

결론을 정리하면, Classification KNN은 K= 7, 9일 때 제일 성능이 좋음을 확인할 수 있다.

위에 분석을 진행했던 모델들과 비교해보자.

Logistic Regression 분류 모델과는 성능이 비슷하다. Decision tree 모델보다는 성능이 떨어진다. 회귀 모델인 Multiple Linear Regression 보다는 성능이 좋다.

다음은 Regression KNN을 살펴보자.

모델 학습 후 성능을 살펴본 결과 다음과 같았다.

	RMSE	MAE	MAPE
k-NN	0.7249681	0.5763889	10.444300

위의 결과는 K=12로 했을 때 돌린 결과이다.

(살펴본 결과 K=1~20까지 진행했을 때 12가 성능이 제일 좋았다.)

아쉬운 점은 Classification KNN의 경우 예측 성능이 나쁘지 않았지만, Regression KNN의 경우 회귀 모델이어서 그런지, Multiple Linear Regression과 성능이 비슷하게 나왔다.

결론적으로 위의 Data set의 경우 회귀 모델로는 좋은 성능을 내는 모델을 구축하기 어렵고, 보통 품질(0)과 고급 품질(1)로 나누어 분류 모델을 학습시키는 것이 성능이 더 좋아짐을 확인할 수 있었다.

Artificial Neural Network

마지막으로 진행한 분석은 인공신경망(Artificial Neural Network)이다.

먼저 몇 개의 hidden node가 적합할지를 알아보아야 한다.

Hidden node를 2개부터 20개까지 2개씩 증가시켜가며 모델을 학습시켜보았다.

이 인공신경망 모델의 경우 출력변수 Y는 3~8로 하여 회귀 모델로 진행하였다. (분류 x)

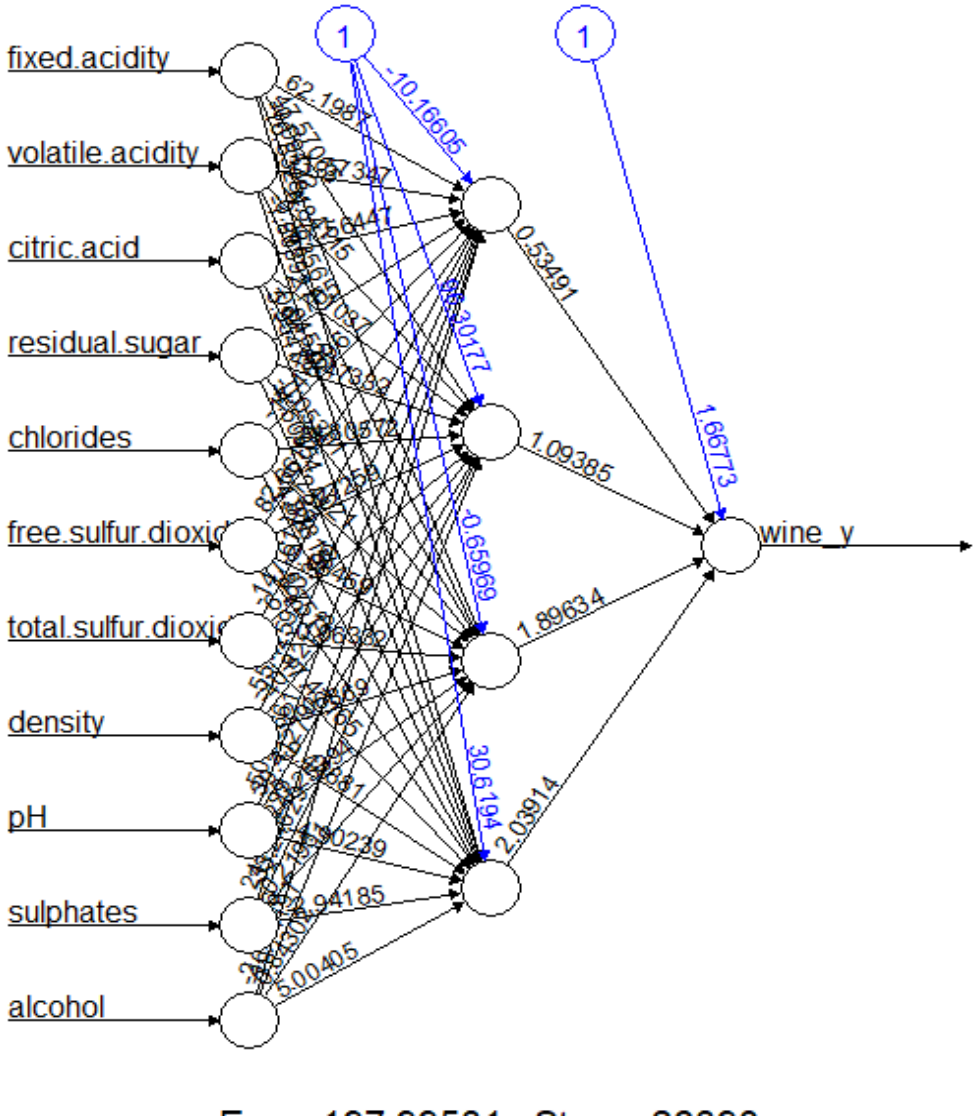
nH	RMSE	MAE	MAPE
4	0.6905885	0.5209527	9.610730
2	0.6846458	0.5290939	9.729201
6	0.8392122	0.5580652	10.265398
12	0.7541372	0.5689950	10.517760
8	0.8921326	0.5901511	10.860753
14	0.8005061	0.6035216	11.160400
10	1.0838350	0.6175410	11.596749
16	0.8186801	0.6250632	11.524814
18	0.8504863	0.6326276	11.780644
20	0.8657853	0.6484661	11.976973

결과는 다음과 같았다.

왼쪽의 nH가 hidden node의 수다. (MAE가 제일 낮은 순으로 나열하였다)

위의 성능 결과를 종합적으로 살펴보면 Hidden node가 4개일 때 제일 성능이 좋음을 확인할 수 있다.

따라서 hidden node를 4개로 설정하고 학습을 진행하였다.



결과를 시각화 하면 위와 같다.

Hidden node 4개로 학습했음을 확인할 수 있다.

결과의 성능을 살펴보자.

	RMSE	MAE	MAPE
ANN	0.6318684	0.4997432	9.023722

Artificial Neural Network를 통해 학습한 결과, 오류율은 위와 같았다.

흥미로운 점은 다른 회귀 모델인 Multiple Linear Regression이나 Regression KNN 보다 성능이 제일 좋았다는 것이다. (물론 차이는 크지 않았지만)

<결론>

🎯모델의 성능 비교

3가지로 구분해 각각의 모델에 대한 성능 평가를 통해 결론을 내릴 수 있다.

첫 번째는 회귀 모델 3가지다.

Multiple Linear Regression, Regression KNN, Artificial Neural Network 다.

	RMSE	MAE	MAPE
MLR	0.6372469	0.5059616	9.251204
k-NN	0.7249681	0.5763889	10.444300
ANN	0.6318684	0.4997432	9.023722

하나로 모아서 확인해보면 위와 같다.

회귀 모델의 경우 각각에 큰 차이는 없었지만 인공신경망 모델이 학습결과가 제일 좋았다. RMSE, MAE, MAPE 세 지표 모두에서 제일 오류율이 작았다.

따라서 위의 Data set에 대해 회귀 모델을 구축할 때는 Artificial Neural Network를 이용해 분석을 진행해야 함을 결론 내릴 수 있다.

두 번째는 분류 모델 3가지다.

Logistic Regression, Decision Tree, KNN이다.

	TPR (Recall)	Precision	TNR	ACC	BCR	F1
Logstic Regression	0.3666667	0.5238095	0.952381	0.8791667	0.5909368	0.4313725

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
CART	0.4666667	0.56	0.947619	0.8875	0.6649979	0.5090909
knn7	0.3833333	0.4509804	0.9333333	0.8645833	0.5981453	0.4144144

세가지 분류 모델을 하나로 모아서 확인해보면 위와 같다.

위 셋 중 Decision Tree가 지표들을 종합적으로 고려했을 때, 성능이 제일 우수함을 확인할 수 있다.

따라서, 보통 품질(0)과 고급 품질(1)로 설정한 후 분류 모델을 사용할 때는 의사결정나무를 사용해야 제일 좋은 예측 성능을 얻을 수 있음을 결론 내릴 수 있다.

마지막은 Clustering이다. 출력변수 Y가 없다고 가정 후 분석을 진행하였다. 그 결과 크게 두개의 유효한 군집으로 나뉘었다. 군집 1은 고급 품질의 와인의 비율이 높은 군집이었으며 군집 2는 보통 품질의 와인의 비율이 높았다. 그리고 그 군집의 유효성도 확인하였다.

종합적인 결론

회귀 모델/분류 모델/클러스터링 구분 없이 모든 분석 모델을 종합적으로 고려해보았을 때, 결론적으로 출력변수 Y인 와인의 품질에 중요한 영향을 미치는 입력변수 X의 목록은 다음과 같다.

Alcohol(알코올), surphates(황산염), Total.sulfur.dioxide(전체 이산화황)이 세 가지가 모든 모델에 포함되는 유의한 변수이다.

따라서 와인 품질을 결정함에 있어 제일 중요한 변수는 위의 세 가지 알코올, 황산염, 전체 이산화황임을 알 수 있다.

우리는 이제 새로운 와인을 만났을 때, 위 세 가지 화학적 성분(변수)에 주목하여 이 와인이 좋은 품질인지 보통 품질인지를 분석 모델을 활용해 구분할 수 있다.

이 때 사용하기에 적합한 분석 모델은 품질 수치를 3~8 사이로 예측하고 싶은 경우 인공신경망 모델, 품질을 고급 품질/보통 품질 두개로 구분하고 싶은 경우 의사결정나무 모델이다.