2018 Artificial Intelligence Spring Semester Final Project

Data Mining Report from Wine Quality Data Set

곽 인 모

2011210036, Computer Science, 고려대학교

kwakinmo@korea.ac.kr

Abstract

이 레포트는 White Wine Quality dataset과 Red Wine Quality dataset 두 개의 public domain을 사용하여, 각각 5개의 Classifer에 대해 기계학습 모델링을 진행하고 각 Classifier의 성능을 비교 분석하였다. 이 실험의 목적은 두 데이터셋을 이용해 두 종류의 와인의 품질을 구분하는 좋은 성능의 모델을 만드는 데에 있다. 이를 위해 데이터 분석에는 모델링이 쉽고 간단한 Weka를 이용하였다.

두 개의 데이터 셋은 포르투갈 북부에 있는 vinho 지역의 vinho verde wine의 white wine, red wine sample으로, 각각 4898, 1599개의 instances와 12개의 attibute로 구성되어 있다. 12개의 attribute는 와인의 특징을 나타내는 11개의 요소와 와인의 품질을 나타내는 1개의 nominal feature이다..

모델링을 진행하기에 앞서 데이터를 살펴보고, UCI에서 제공해주는 데이터 관련 정보와 Attribute에 대한 정보를 찾아봄으로써 데이터에 대한 이해를 높였다. 모델링에 있어서도 중복된 데이터를 삭제하고, outlier 제거와 normalization 등을 통해 모델의 정확도를 끌어올렸다. 또한 각 데이터 셋에서 중요한 Feature들을 선택하여 dimension을 축소시킴으로써, complexity를 낮추고 모델이 overfit되는 것을 방지하고자 하였다.

Data Preprocessing 이후에는 각 데이터셋 마다 5개의 Classifier를 모델링하였다. 이때 각 Classifer의 Parameter값을 변경해 나가면서, Accuracy 뿐만 아니라 전반적으로 좋은 성능을 뛰는 Classifier를 찾고자 하였다. 성능 평가에 있어서는 모델이 Overfit 되는 것을 방지하고자 testing시에 데이터를 나눠서 10번 검증하여 평균값을 이용하는 10-fold Cross Validation을 이용하였다. 또한 두 데이터셋에 대한 각각의 실험을 통해 각기 다른 종의 와인에서 품질을 결정하는 요소들과 각각의 classifier에 대한 비교 분석을 함으로써 insight를 얻고자 하였다.

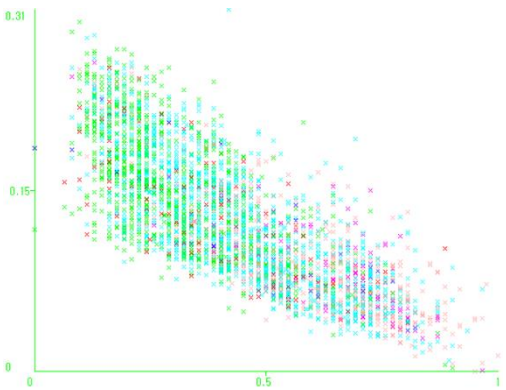
1. Data Information & Exploration

이 레포트는 포르투갈 북부에 있는 vinho 지방의 와인인 vinho verde의 화이트와인과 레드와인 2개의 데이터셋을 사용하였다. White wine 데이터셋은 12개의 attribute와 4898개의 instance, Red wine 데이터셋은 12개의 attribute와 1599개의 instance들로 구성되어 있다.

각 데이터셋의 attribute들은 numerical attribute와 nominal attribute로, 11가지 종류의 물리 화학적인 numerical attribute와 와인의 품질을 나타내는 하나의 nominal attribute로 구성되어 있다.

- Fixed acidity : 포도주 결합산도
- Volatile acidity : 휘발성산
- Citric acid : 구연산
- Residual sugar : 발효 후 와인 속에 남아있는 당분(잔당)
- Chlorides : 염화물
- Free sulfur dioxide : 유리 이산화황
- Total sulfur dioxide : 총 이산화황
- Density : 농도
- pH : 산도
- sulphates : 황산염
- alcohol : 알코올
- quality : 와인을 품질을 나타내는 값으로 0부터 10의 정수 사이값으로 구성되어 있다.

데이터의 attribute간의 관계를 살펴 보았을 때 Density와 Alcoholdms 오른쪽 그림과 같이 음의 관계를 보였다. 또한 Volatile acidity와 fixed acidity는 red wine의 값들이 상대적으로 큰 값을 보였고, 잔당은 상대적으로 단맛을 더 내는 화이트와인이 더 많았다. 또한 두 sulfur dioxide attribute의 경우에는 화이트와인이 약 3배의 큰 값을 보였다. pH는 화이트와인(mean pH 3.1)이 레드와인(mean pH 3.31)보다 살짝 낮았는데, 이는 Vinho Verde의 화이트와인이 레드와인보다 맛이 더 시기 때문으로 볼 수 있다.



white wine의 residuals나 red wine의 citric acid의 경우에는 낮은 값들로 skewed 되어 있었고, 두 데이터 모두 label을 나타내는 quality 값은 순차적이고, 비교적 품질이 안 좋거나 뛰어난 와인들보다 평균적인 품질의 와인이 많았다.

2. Data Preprocessing

데이터 전처리는 기계학습에 있어서 모델의 성능을 결정하는 핵심적인 요소 중 하나이다. 잘못된 데이터가 있거나 값이 없는 데이터 등등 현실의 데이터들은 정교하지 못하다. 그러므로 데이터를 잘 살펴보고 이를 올바르게 처리하는 것이 중요하다.

이 실험에 있어서는 가장 먼저 중복된 데이터나 값이 없는 데이터가 존재하는지 살펴보고 이를 처리하고자 하였다. 우선 두 데이터 셋 모두 값이 없는 데이터는 존재하지 않아서 이를 고려할 필요는 없었다. 그러나 화이트와인 데이터셋의 경우 929개, 레드와인 데이터셋의 경우 240개의 중복된 데이터가 존재하여 이를 삭제하였다.

이후 각각의 데이터셋에 대해 normalization을 진행했다. 데이터들이 완벽히 bell shape의 분포를 보이지는 않아서 standariztion 대신 normalization을 이용했다. 이후 잘못된 값으로 보여지는 outlier를 처리하고자 하였는데 화이트와인에서는 106개, 레드와인에서는 95개의 outlier가 나와 이를 weka 내에서 알고리즘을 이용해 처리하였다. 잘못된 값으로 고려해 볼 수도 있는 값인 extreme value들은 제거 해봤을 때 성능면에서 오히려 좋지 못한 모습을 보여서 제거하지 않고 모델링을 진행하였다.

데이터 전처리 과정 중에 하나인 Feature Selection은 모델링에 중요한 영향을 끼치는 Feature들만 골라서 모델을 만드는 것을 말한다. 이는 모델링 과정의 complexity를 낮춰주고, 데이터에 대한 이해를 좀 더 쉽게 해줄 뿐만 아니라 모델이 overfit되는 것을 방지한다.

이 실험에서는 Feature Selection을 위해서 GainRatio를 이용하였다. 이는 불확실성을 낮추는 정보의 정도를 나타내는 Information Gain을 단점을 보완한 방법이다. 이 값이 클수록 중요한 Feature를 의미하는데, 화이트와인에서는 Alcohol > Density > Chlorides > FreeSulfurDioxide > TotalSulfurDioxide > VolatileAcidity > CitricAcid > FixedAcidity > Residual Sugar > pH > Sulphates의 순서를 보였고, J48를 가지고 모델링을 했을 때 FixedAcidity보다 gain ratio가 낮은 값들을 제거하고 모델링을 했을 때 성능 측면에서 가장 뛰어나 8개의 attribute만 가지고 모델링을 진행하기로 결정하였다. 레드와인에서는 Alcohol > TotalSulfurDioxide > Sulphates > VolatileAcidity > Density > CitricAcid > Chlorides > FreeSulfurDioxide > Residual Sugar > pH > FixedAcidity의 순서를 보였고, freeSulfurDioxide과 그 밑의 낮은 값들은 Gain이 거의 없었기 때문에 7개의 attribute

만을 가지고 모델링을 진행하기로 결정하였다.

이를 통해서 화이트와인과 레드와인에서 중요시 되는 Feature들에 대해 비교해 볼 수 있다. 화이트 와인에서는 Sulphates가 중요시되지 않는데 레드와인에서는 매우 중요한 Feature로 생각된다. 또한 둘다 공통적으로 Alcohol이 와인의 품질을 결정하는 가장 중요한 Feature임을 알 수 있다.

3. Classifier Analysis & Evaluation

3-1. White Wine

White Wine 데이터 셋을 활용하여 와인의 품질을 구분하는 다양한 종류의 Classifer를 만들었다. OneR을 Baseline으로 진행하였고, 이후 Decision Tree, Naïve Bayes, Multiple layer Perceptron, K Nearest Neighborhood, Random Forest 5개의 Classifier를 모델링하고 Cross Validation을 이용해 모델들을 Evaluation했다.

Baseline으로 이용한 OneR은 하나의 룰만을 이용하여 기계학습을 진행한다. 이 경우에, 44%의 낮은 정확도를 보였다.

(1) Decision Tree

Decision Tree는 직사각형의 boundary로 데이터를 구분하는 Classfication으로, 결과값을 트리형태로 볼 수 있고, 직관적으로 이해하기 쉬운 모델링이다.

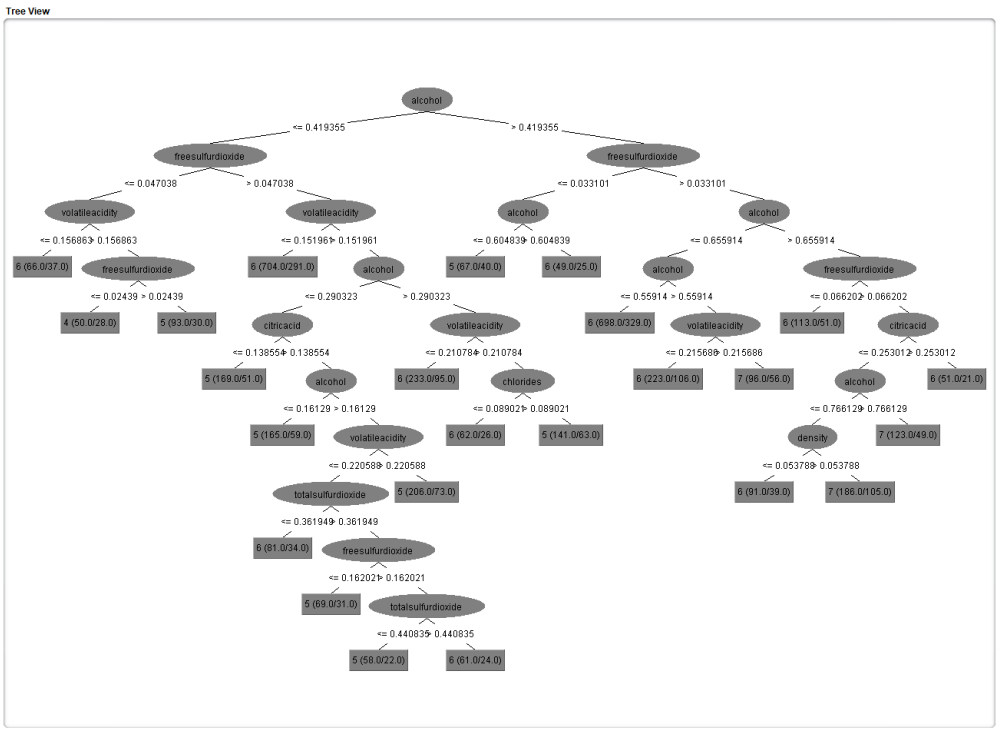
Decision Tree로 모델링을 진행함에 있어서 성능을 올리고자, pruning을 진행하고자 하였다. Pruning은 tree를 각각의 instance들이 완전히 구분될 때까지 트리를 만드는 것이 아니라 leaf마다 그 이전에 모델링을 중단하는 것을 말한다. 이를 위해, 디폴트 값으로 정해져 있는 minNumObj 값을 2에서 10, 20, 40, 60, 50, 45, 42, 43, 41의 순서로 파라미터 값을 변경해가며 다양한 모델을 만들어 보았다. 그 값이 41였던 경우에 Accuracy값이 가장 좋았지만, ROC 값과 Kappa Statistic 값 등이 42일때보다 작아 Accuracy값이 별 차이가 없는 42로 모델링을 정했다.

모델에 대한 평가는 모델이 overfit되는 것을 방지하고자 10-fold Cross Validation으로 하였고, 밑의 그림과 같은 다양한 measure의 Evaluation을 얻을 수 있었다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2042               52.9702 %
Incorrectly Classified Instances      1813               47.0298 %
Kappa statistic                          0.2343
Mean absolute error                      0.1694
Root mean squared error                  0.2942
Relative absolute error                 88.0016 %
Root relative squared error             94.8555 %
Total Number of Instances             3855

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.594     0.005     3
                 0.021    0.003    0.200      0.021    0.039      0.055    0.749     0.139     4
                 0.539    0.174    0.563      0.539    0.551      0.370    0.766     0.535     5
                 0.736    0.552    0.526      0.736    0.614      0.190    0.608     0.528     6
                 0.204    0.052    0.456      0.204    0.281      0.214    0.763     0.370     7
                 0.000    0.000    0.000      0.000    0.000      0.000    0.780     0.096     8
                 0.000    0.000    0.000      0.000    0.000      0.000    0.444     0.001     9
Weighted Avg.    0.530    0.312    0.492      0.530    0.492      0.235    0.692     0.471

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
    0    0    5    8    0    0    0 |    a = 3
    0    3   86   49    2    0    0 |    b = 4
    0   11  610  501    9    0    0 |    c = 5
    0    1  349 1290  113    0    0 |    d = 6
    0    0   30  514  139    0    0 |    e = 7
    0    0    4   88   38    0    0 |    f = 8
    0    0    0    1    4    0    0 |    g = 9
```

그림의 TP rate와 Precision 값을 보면, label값이 5,6인 중간 정도의 품질의 와인에 대해서 그나마 좋은 정확도를 보임을 알 수 있다.

또한 weka의 visualization 기능을 이용하여 해당 Tree의 구조를 밑의 그림과 같이 한 눈에 볼 수 있다. 이를 보면 데이터에 대한 이해를 쉽게 할 수 있고, 새로운 데이터가 오더라도 분류를 직관적으로 진행해 나갈 수 있다.



(2) Naïve Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1769               45.8885 %
Incorrectly Classified Instances      2086               54.1115 %
Kappa statistic                          0.2217
Mean absolute error                      0.1721
Root mean squared error                  0.3168
Relative absolute error                 89.4239 %
Root relative squared error            102.147  %
Total Number of Instances             3855

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.009    0.000      0.000    0.000      -0.006   0.524     0.007     3
                 0.179    0.018    0.275      0.179    0.216      0.198    0.784     0.216     4
                 0.511    0.185    0.534      0.511    0.522      0.330    0.743     0.504     5
                 0.419    0.316    0.525      0.419    0.466      0.106    0.574     0.527     6
                 0.631    0.250    0.352      0.631    0.452      0.313    0.755     0.356     7
                 0.008    0.006    0.045      0.008    0.013      0.005    0.780     0.094     8
                 0.000    0.000    0.000      0.000    0.000      -0.001   0.508     0.003     9
Weighted Avg.    0.459    0.243    0.469      0.459    0.453      0.208    0.670     0.461

=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
   0   1   5   4   3   0   0 |   a = 3
   2  25  45  44  23   1   0 |   b = 4
  22  40 578 382 108   1   0 |   c = 5
  12  22 412 734 565   8   0 |   d = 6
   0   1  38 201 431  11   1 |   e = 7
   0   2   4  33  90   1   0 |   f = 8
   0   0   0   1   4   0   0 |   g = 9
```

위의 그림을 보면, 45.8%의 Accuracy로 52.9%의 Decision Tree Classifier보다 좋지 않음을 알 수 있고, ROC Area 값 (0.670)과 Precision(0.469)를 보더라도, 이전의 모델과 비교해 보았을 때 좋지 못한 성능의 모델임을 알 수 있다.

(3) Multilayer Perceptron

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2050               53.1777 %
Incorrectly Classified Instances      1805               46.8223 %
Kappa statistic                          0.2342
Mean absolute error                      0.1658
Root mean squared error                  0.2942
Relative absolute error                 86.1292 %
Root relative squared error             94.8547 %
Total Number of Instances             3855

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.397     0.003     3
                 0.093    0.004    0.481      0.093    0.156      0.200    0.766     0.212     4
                 0.490    0.153    0.571      0.490    0.527      0.354    0.772     0.533     5
                 0.759    0.578    0.522      0.759    0.619      0.189    0.607     0.526     6
                 0.224    0.049    0.497      0.224    0.309      0.247    0.774     0.396     7
                 0.000    0.001    0.000      0.000    0.000      -0.006   0.788     0.109     8
                 0.000    0.000    0.000      0.000    0.000      0.000    0.352     0.001     9
Weighted Avg.    0.532    0.317    0.511      0.532    0.496      0.241    0.696     0.477

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
    0    1    6    6    0    0    0 |    a = 3
    0   13   70   54    3    0    0 |    b = 4
    0    5  554  567    5    0    0 |    c = 5
    0    6  315 1330  102    0    0 |    d = 6
    0    1   22  503  153    4    0 |    e = 7
    0    1    3   85   41    0    0 |    f = 8
    0    0    0    1    4    0    0 |    g = 9
```

MLP로 모델링을 할 때에, hidden layer 파라미터 값은 디폴트 값인 a로 진행했다. 앞선 linear한 두 모델 보다는 Accuracy(53.1%), Precision(0.511), ROC(0.696) 등으로 모든 Evaluation Measure들의 값이 더 좋았다. 이를 통해 데이터들이 linear한 성격을 띠고 있지 않다는 것을 알 수 있었다.

(4) K Nearest Neighborhood

Model	Accuracy	TP Rate	FP rate	Precision	Recall	F measure	ROC	Kappa statistic
KNN(k=1)	46.1%	0.462	0.280	0.455	0.462	0.456	0.653	0.1884
KNN(k=3)	48.3%	0.484	0.259	0.482	0.484	0.483	0.610	0.2311

KNN은 k개의 가장 가까운 이웃들의 특징을 보고 해당 sample의 클래스를 정하는 classification이다. 디폴트 값으로 k가 1인 KNN에 비해 k값이 3인 모델이 대부분의 Measure에서 좋은 값을 낸 것을 보았을 때, k가 3인 모델이 좀 더 성능이 좋다고 판단된다.

(5) RandomForest

RandomForest는 Decision Tree의 overfit, 불안정성 단점을 보완하기 위해 만들어진 알고리즘으로, n개의 tree를 생성하고 투표를 통해 예측결과를 선택하는 앙상블이다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2080               53.9559 %
Incorrectly Classified Instances      1775               46.0441 %
Kappa statistic                          0.2693
Mean absolute error                      0.1633
Root mean squared error                  0.2895
Relative absolute error                 84.8192 %
Root relative squared error             93.3434 %
Total Number of Instances             3855

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.000    0.000    0.000      0.000    0.000      0.000    0.613     0.006     3
               0.136    0.004    0.543      0.136    0.217      0.259    0.837     0.278     4
               0.546    0.175    0.564      0.546    0.555      0.374    0.787     0.584     5
               0.694    0.489    0.542      0.694    0.609      0.207    0.635     0.567     6
               0.329    0.077    0.480      0.329    0.391      0.295    0.800     0.459     7
               0.023    0.003    0.214      0.023    0.042      0.060    0.787     0.120     8
               0.000    0.000    0.000      0.000    0.000      0.000    0.662     0.004     9
Weighted Avg.  0.540    0.287    0.524      0.540    0.518      0.268    0.721     0.525

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
    0    0    6    7    0    0    0 |    a = 3
    0   19   78   41    2    0    0 |    b = 4
    0   12  617  482   20    0    0 |    c = 5
    0    4  360 1216  169    4    0 |    d = 6
    0    0   31  420  225    7    0 |    e = 7
    0    0    2   76   49    3    0 |    f = 8
    0    0    0    1    4    0    0 |    g = 9
```

이 알고리즘의 Accuracy는 53.9%로 위에서 진행한 4가지 Classifier 모델보다 가장 성능이 좋은 모습을 보였다.

3-2. Red Wine

Red Wine 데이터셋에 대해서도 White Wine과 같이 와인의 품질을 구분하는 다양한 Classfier를 만들었다. OneR을 Baseline으로 진행하였고, 이후 Decision Tree, Naïve Bayes, Multiple layer Perceptron, K Nearest Neighborhood, Random Forest 5개의 Classifier를 모델링하고 Cross Validation을 이용해 모델들을 Evaluation했다.

oneR로 모델링을 진행한 경우 Accuracy는 42.3%로 매우 좋지 않았다.

(1) Decision Tree

White Wine에서와 같이 Decision Tree로 성능을 올리고자, pruning을 진행하고자 하였다. 이를 위해, 디폴트 값으로 정해져 있는 minNumObj 값을 2, 20, 30, 40, 15, 21, 19의 순서로 파라미터 값을 변경해가며 다양한 모델을 만들었다. 그 값이 20일 때 모든 Measure 값들이 좋아 20으로 모델링을 정했다. Accuracy는 White Wine(52.9%)에서 보다 좋은 57.5%가 나왔다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         727               57.5158 %
Incorrectly Classified Instances       537               42.4842 %
Kappa statistic                          0.3093
Mean absolute error                      0.1764
Root mean squared error                  0.3066
Relative absolute error                 82.0359 %
Root relative squared error             93.5856 %
Total Number of Instances             1264

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000     0.000      0.000    0.000      0.000    0.713     0.015     3
                 0.020    0.002     0.333      0.020    0.038      0.074    0.631     0.070     4
                 0.712    0.306     0.631      0.712    0.669      0.402    0.771     0.682     5
                 0.570    0.333     0.534      0.570    0.552      0.235    0.653     0.527     6
                 0.373    0.054     0.483      0.373    0.421      0.358    0.815     0.390     7
                 0.000    0.000     0.000      0.000    0.000      0.000    0.794     0.080     8
Weighted Avg.    0.575    0.269     0.552      0.575    0.556      0.310    0.723     0.550

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   0   5   2   0   0 |   a = 3
   0   1  34  15   0   0 |   b = 4
   0   0 381 149   5   0 |   c = 5
   0   2 168 289  48   0 |   d = 6
   0   0  16  78  56   0 |   e = 7
   0   0   0   8   7   0 |   f = 8
```

(2) Naïve Bayes

Naïve Bayes 모델은 화이트 와인에서는 그 성능이 그다지 좋지 못했으나(Accuracy 45.8%) 레드와인에서는 Accuracy가 56.4%로, Decision Tree Classifier(57.5%)와 크게 차이가 나지

않음을 알 수 있다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         714               56.4873 %
Incorrectly Classified Instances       550               43.5127 %
Kappa statistic                          0.3202
Mean absolute error                      0.1715
Root mean squared error                  0.3126
Relative absolute error                 79.7446 %
Root relative squared error             95.4085 %
Total Number of Instances             1264

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.005    0.000      0.000    0.000      -0.005   0.578     0.029     3
                 0.060    0.025    0.091      0.060    0.072      0.043    0.720     0.093     4
                 0.692    0.262    0.660      0.692    0.675      0.427    0.789     0.741     5
                 0.529    0.296    0.545      0.529    0.537      0.234    0.664     0.516     6
                 0.480    0.076    0.459      0.480    0.469      0.396    0.849     0.423     7
                 0.067    0.011    0.067      0.067    0.067      0.055    0.829     0.067     8
Weighted Avg.    0.565    0.240    0.556      0.565    0.560      0.324    0.742     0.575

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   3   3   1   0   0 |   a = 3
   1   3  33  13   0   0 |   b = 4
   4  14 370 140   7   0 |   c = 5
   1  12 148 268  70   8 |   d = 6
   0   1   7  64  72   6 |   e = 7
   0   0   0   6   8   1 |   f = 8
```

(3) Multi Layer Perceptron

레드와인에서도 Multiple Layer Perceptron을 hidden layer 값을 a로 진행하였고 Learning rate는 0.3, training은 500번으로 하였다. 화이트와인에서도 위의 두 Classfier보다 성능이 좋았는데, 레드와인의 경우에도 Accuracy값(59.0%) 뿐만 아니라 Precision(0.572), ROC(0.746) 등의 값에서 상대적으로 좋은 수치를 냈다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         747               59.0981 %
Incorrectly Classified Instances       517               40.9019 %
Kappa statistic                          0.3289
Mean absolute error                      0.1695
Root mean squared error                  0.3015
Relative absolute error                 78.8364 %
Root relative squared error             92.005  %
Total Number of Instances             1264

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.328     0.004     3
                 0.040    0.004    0.286      0.040    0.070      0.094    0.652     0.109     4
                 0.679    0.226    0.688      0.679    0.683      0.453    0.796     0.719     5
                 0.677    0.404    0.529      0.677    0.593      0.267    0.672     0.528     6
                 0.260    0.037    0.488      0.260    0.339      0.296    0.863     0.414     7
                 0.000    0.000    0.000      0.000    0.000      0.000    0.876     0.057     8
Weighted Avg.    0.591    0.262    0.572      0.591    0.570      0.338    0.746     0.570

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   1   5   1   0   0 |   a = 3
   0   2  28  20   0   0 |   b = 4
   0   2 363 169   1   0 |   c = 5
   0   2 128 343  34   0 |   d = 6
   0   0   4 107  39   0 |   e = 7
   0   0   0   9   6   0 |   f = 8
```

(4) K Nearest Neighborhood

위의 화이트와인에서는 k값을 1과 3으로만 진행해보았는데, 이번에 레드와인에서는 k값을 1, 3, 10, 30, 50, 100, 70, 60, 55, 57, 58, 56 순으로 변경해가면서 모델링을 진행하여 가장 성능이 괜찮은 k값을 찾고자 하였다. 이때 k값이 57일 때 모든 Measure에서 좋은 값을 보여 이 값으로 가장 성능이 좋은 모델로 선택했다.

Accuracy에 있어서는 MLP의 값과 비슷하지만, 다른 Measure 값들이 상대적으로 다 작아서 MLP보다는 조금 미흡한 모델로 판단된다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         746               59.019  %
Incorrectly Classified Instances       518               40.981  %
Kappa statistic                          0.3202
Mean absolute error                      0.1791
Root mean squared error                  0.2995
Relative absolute error                 83.3001 %
Root relative squared error             91.406  %
Total Number of Instances             1264

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.747     0.063     3
                 0.000    0.000    0.000      0.000    0.000      0.000    0.686     0.117     4
                 0.742    0.302    0.643      0.742    0.689      0.435    0.796     0.735     5
                 0.611    0.358    0.534      0.611    0.570      0.249    0.665     0.527     6
                 0.260    0.024    0.591      0.260    0.361      0.343    0.839     0.440     7
                 0.000    0.000    0.000      0.000    0.000      0.000    0.714     0.034     8
Weighted Avg.    0.590    0.274    0.556      0.590    0.563      0.325    0.743     0.580

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   0   6   1   0   0 |   a = 3
   0   0  33  17   0   0 |   b = 4
   0   0 397 137   1   0 |   c = 5
   0   0 174 310  23   0 |   d = 6
   0   0   7 104  39   0 |   e = 7
   0   0   0  12   3   0 |   f = 8
```

(5) Random Forest

화이트와인에서도 Random Forest를 이용한 앙상블 방법이 다른 4가지 모델보다 우수한 모습을 보였는데, 레드와인에서도 마찬가지로 Accuracy 60.2%, TP Rate 0.603, Precision 0.575, ROC curve 0.736으로 다른 4가지 모델에 비해서 모든 면에서 가장 좋은 성능을 보였다.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         762               60.2848 %
Incorrectly Classified Instances       502               39.7152 %
Kappa statistic                          0.3547
Mean absolute error                      0.1695
Root mean squared error                  0.2961
Relative absolute error                 78.8228 %
Root relative squared error             90.378  %
Total Number of Instances             1264

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.001     0.000      0.000     0.000     -0.002    0.641     0.019     3
                 0.020    0.005     0.143      0.020     0.035      0.040    0.723     0.129     4
                 0.729    0.281     0.655      0.729     0.690      0.443    0.807     0.741     5
                 0.611    0.318     0.563      0.611     0.586      0.290    0.694     0.553     6
                 0.407    0.043     0.560      0.407     0.471      0.419    0.855     0.508     7
                 0.000    0.001     0.000      0.000     0.000     -0.003    0.741     0.060     8
Weighted Avg.    0.603    0.252     0.575      0.603     0.585      0.355    0.762     0.601

=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   1   6   0   0   0 |   a = 3
   1   1  35  13   0   0 |   b = 4
   0   1 390 142   2   0 |   c = 5
   0   4 155 310  38   0 |   d = 6
   0   0   9  79  61   1 |   e = 7
   0   0   0   7   8   0 |   f = 8
```

4. Conclusion

<화이트와인>

Model	Accuracy	TP Rate	FP rate	Precision	Recall	F measure	ROC	Kappa statistic
DT	52.9%	0.530	0.312	0.492	0.530	0.493	0.692	0.2343
NB	45.8%	0.459	0.243	0.469	0.459	0.453	0.670	0.2217
MLP	53.1%	0.532	0.317	0.511	0.532	0.496	0.696	0.2342
KNN(k=3)	48.3%	0.484	0.259	0.482	0.484	0.483	0.610	0.2311
RF	**53.9%**	**0.540**	**0.287**	**0.524**	**0.540**	**0.518**	**0.721**	**0.2693**

<레드와인>

Model	Accuracy	TP Rate	FP rate	Precision	Recall	F measure	ROC	Kappa statistic
DT	57.5%	0.575	0.271	0.539	0.575	0.555	0.715	0.3084
NB	56.4%	0.565	0.240	0.556	0.565	0.560	0.742	0.3202
MLP	59.0%	0.591	0.262	0.572	0.591	0.570	0.746	0.3289
KNN(k=57)	59.0%	0.590	0.274	0.556	0.590	0.563	0.743	0.3202
RF	**60.2%**	**0.603**	**0.252**	**0.575**	**0.603**	**0.585**	**0.762**	**0.3547**

화이트와인 데이터 셋과 레드와인 데이터셋을 비교해보았을때, 레드와인 데이터 셋에 대한 성능이 평균 5퍼센트 정도 높았다. 이는 데이터 셋의 양이 3배가량 많은 화이트와인 관련 모델들이 전반적으로 overfit 되었을 수 있다는 판단을 해볼 필요성을 낳는다.

또한 화이트와인, 레드와인 모두 앙상블 방법인 RF에 있어서 가장 좋은 성능을 보였다. 앙상블 방법들은 각 알고리즘의 부족한 점들을 메꾸기 위해 나온 앙상블 방법이기 때문에 이 결과는 당연한 결과로 판단된다.

마지막으로, 분석결과 모델들의 confusion matrix에서 3, 4, 5 품질과 같이 평균적인 품질의 제품들끼리 서로 헷갈려서 잘 구별하지 못하는 경향이 있음도 알 수 있었다.

5. Reference

Modeling wine preferences by data mining from physicochemical properties, Cortez et al.,2009