109-2 EPM 7012 Statistical and Machine Learning: Assignment 2

Yi-Wen Hsiao(d08849010), Institute of Epidemiology and Preventive Medicine, NTU

May 19, 2021

Contents

1. Learning Objectives . . . 1
2. Data Description . . . 1
3. Data Exploration and Visualization . . . 2
4. Class prevalence in train data . . . 3
3. Subsampling for class imbalances . . . 4
4. Modeling . . . 4
  4.1 SVM . . . 5
  4.2 kNN . . . 6
  4.3 Random forest . . . 7
5. Model comparisons . . . 9
  5.1 Confusion matrix . . . 10
  5.2 Performance measurement . . . 10
6. Discussion . . . 11
Session Information . . . 11

---

1. Learning Objectives

- to practice the process of building the classification ML models
  - build classification models (SVM, RF, KNN)
  - split data and turn parameters
  - compare the classification accuracy of above-mentioned models
  - visualize the final classification results
- to learn how to interpret the analytic results

2. Data Description

In this study, we used both red and white variants of the Portuguese "Vinho Verde" wine datasets, a Wine Quality Data Set from UCI Machine Learning repository to construct classification models using below-mentioned physicochemical factors. The basic descriptions of two dataset are as follows:

- Number of Instances for ***red*** and ***white*** wine: 1599 and 4898
- The prevalence of red and white wine: 24.6% and 75.4%
- Number of Attributes: 11 output attribute (two datasets shared same attributes)
  - Input variables (based on physicochemical tests):
    - fixed acidity
    - volatile acidity
    - citric acid
    - residual sugar

    * chlorides
    * free sulfur dioxide
    * total sulfur dioxide
    * density
    * pH
    * sulphates
    * alcohol
  - Output variable (based on label data):
    * type (red and white)

- Missing Attribute Values: None

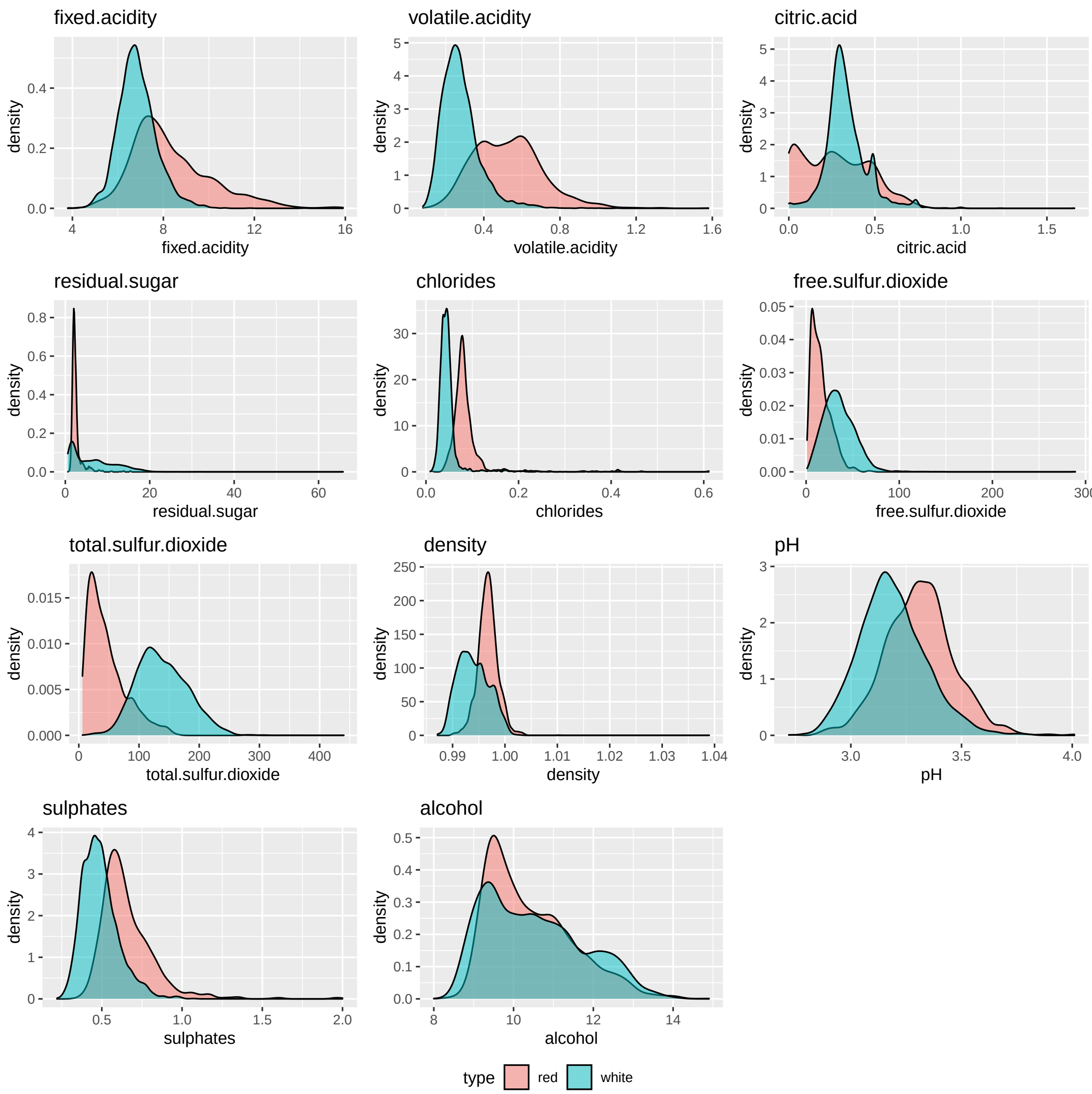
3. Data Exploration and Visualization

In this section, we checked the data structure to look for the possible problems in the datset.

- **Summary statistics**

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
fixed.acidity	3.80000	6.40000	7.00000	7.2153071	7.70000	15.90000
volatile.acidity	0.08000	0.23000	0.29000	0.3396660	0.40000	1.58000
citric.acid	0.00000	0.25000	0.31000	0.3186332	0.39000	1.66000
residual.sugar	0.60000	1.80000	3.00000	5.4432353	8.10000	65.80000
chlorides	0.00900	0.03800	0.04700	0.0560339	0.06500	0.61100
free.sulfur.dioxide	1.00000	17.00000	29.00000	30.5253194	41.00000	289.00000
total.sulfur.dioxide	6.00000	77.00000	118.00000	115.7445744	156.00000	440.00000
density	0.98711	0.99234	0.99489	0.9946966	0.99699	1.03898
pH	2.72000	3.11000	3.21000	3.2185008	3.32000	4.01000
sulphates	0.22000	0.43000	0.51000	0.5312683	0.60000	2.00000
alcohol	8.00000	9.50000	10.30000	10.4918008	11.30000	14.90000

- **Density plot**

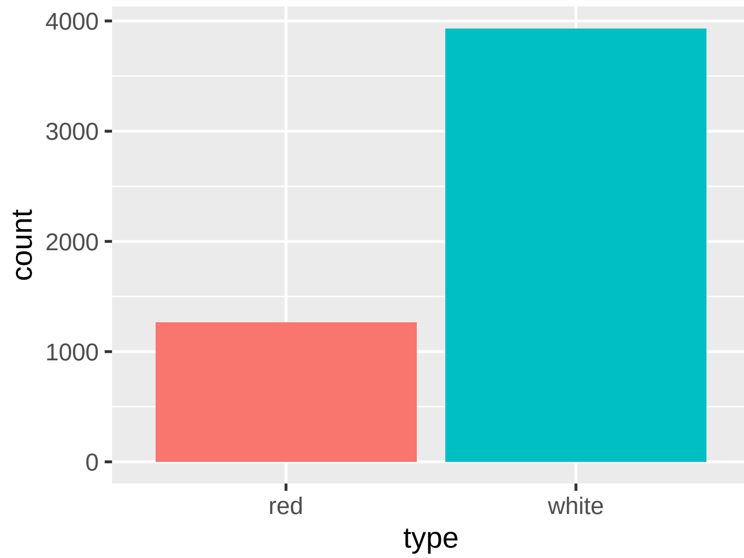


- **Brief summary**
  - Three variables (volatile acid, chlorides and total sulfur dioxide) seem to have distinct peaks for red and white wines, suggesting that these variables may help distinguish two wine types.

4. Class prevalence in train data

Here, we applied a 80/20 split to separate train and test dataset.

- Number of sample in train dataset: 5198
- Number of sample in test dataset: 1299



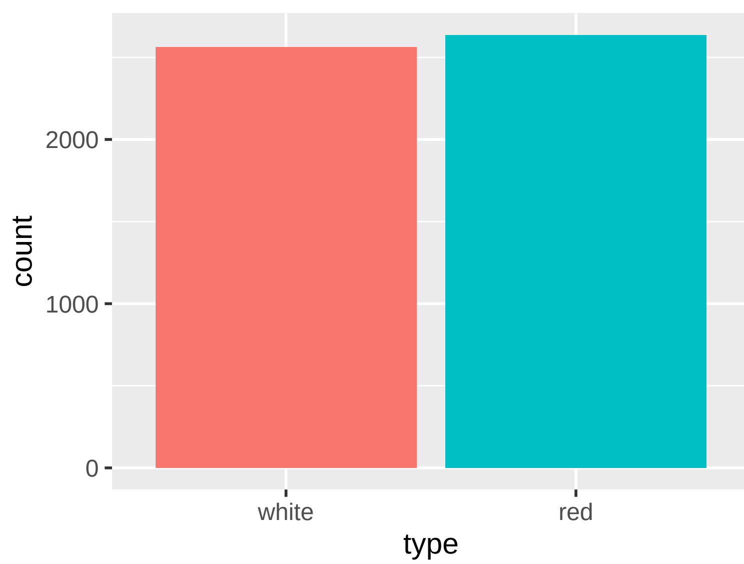
- **Brief summary**
  - There is a clear disparity in the counts of the observed classes in train data, suggesting that this may have negative impact on model fitting.

3. Subsampling for class imbalances

Subsampling the train data in the below-mentioned manners is the one of the technologies to resolve the issue of class imbalance.

- *down-sampling*: randomly subset the majority label to be the same size as the minority label
- *up-sampling*: randomly sample the minority label to be the same size as the majority label
- *hybrid methods*: down-sample the majority label and generate new data for the minority label, such as *SMOTE* and *ROSE*

Here, we used ROSE method, one of the common hybrid methods, to tackle the issue of class imbalances.



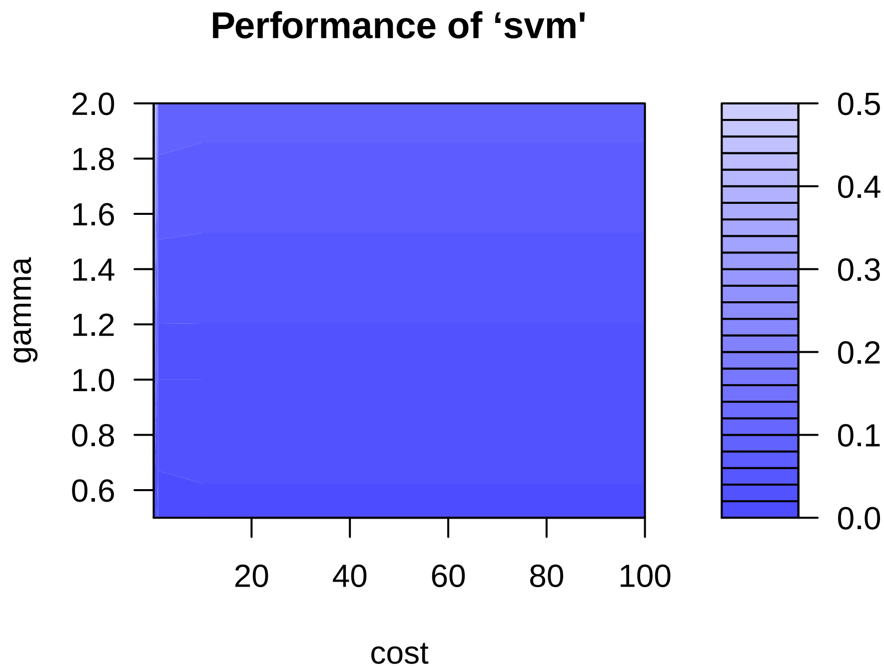
4. Modeling

In this section, we applied three models, including Support Vector Machine (SVM), K Nearest Neighbor (kNN) and Random Forest (RF). Also, we used 10-fold cross validation mode to tune the parameters and then applied these tuned parameters to build the final models.

4.1 SVM

- Here, we used *'e1071'* R package which is based on soft-margin algorithm. Here, we tuned two important parameters: **cost** (range: 0.1, 1.0, 10 and 100) and **gamma** (range: 0.5,1 and 2) while keeped the same kernel setting using .
- The **cost** parameters controls training errors and margins
- The **gamma** parameters defines *how far the influence of a single training example reaches.*

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.5
##
## - best performance: 0.01654439
##
## - Detailed performance results:
##     cost gamma      error  dispersion
## 1    0.1   0.5 0.03193679 0.005465295
## 2    1.0   0.5 0.01654439 0.004813853
## 3   10.0   0.5 0.01750630 0.003323363
## 4  100.0   0.5 0.01750630 0.003323363
## 5    0.1   1.0 0.14680228 0.041358990
## 6    1.0   1.0 0.02674300 0.007001000
## 7   10.0   1.0 0.02751297 0.006555459
## 8  100.0   1.0 0.02751297 0.006555459
## 9    0.1   2.0 0.49269490 0.021579554
## 10   1.0   2.0 0.09235327 0.023562045
## 11  10.0   2.0 0.08869757 0.022855833
## 12 100.0   2.0 0.08869757 0.022855833
```

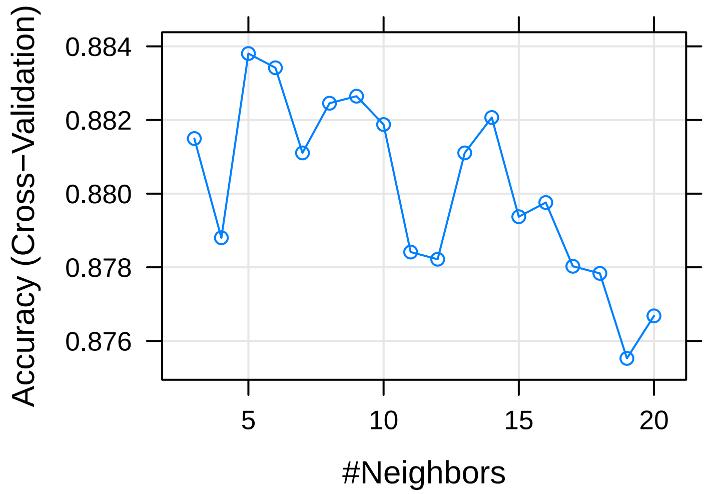


4.2 kNN

- Here, we used *‘caret’* R package and tuned one parameter (k) using GridSearch.
- k represents the number of neighbors to be considered.

```
##     k  Accuracy     Kappa AccuracySD    KappaSD
## 1   3 0.8814962 0.7628803 0.01765332 0.03538194
## 2   4 0.8788021 0.7574692 0.01606999 0.03217240
## 3   5 0.8838043 0.7674937 0.01462779 0.02931070
## 4   6 0.8834178 0.7667000 0.01641443 0.03292530
## 5   7 0.8811064 0.7620964 0.01267075 0.02541756
## 6   8 0.8824552 0.7648051 0.01431432 0.02868246
## 7   9 0.8826467 0.7651885 0.01332821 0.02674430
## 8  10 0.8818768 0.7636647 0.01302720 0.02612280
## 9  11 0.8784141 0.7567379 0.01456201 0.02922524
## 10 12 0.8782196 0.7563507 0.01272572 0.02554417
## 11 13 0.8811064 0.7621273 0.01667541 0.03343924
## 12 14 0.8820687 0.7640762 0.01352214 0.02710480
## 13 15 0.8793742 0.7586708 0.01654204 0.03317499
## 14 16 0.8797599 0.7594481 0.01248826 0.02505817
## 15 17 0.8780276 0.7559823 0.01430554 0.02870408
## 16 18 0.8778350 0.7555897 0.01492835 0.02996989
## 17 19 0.8755262 0.7509665 0.01464784 0.02939671
## 18 20 0.8766815 0.7532960 0.01554614 0.03120301
```

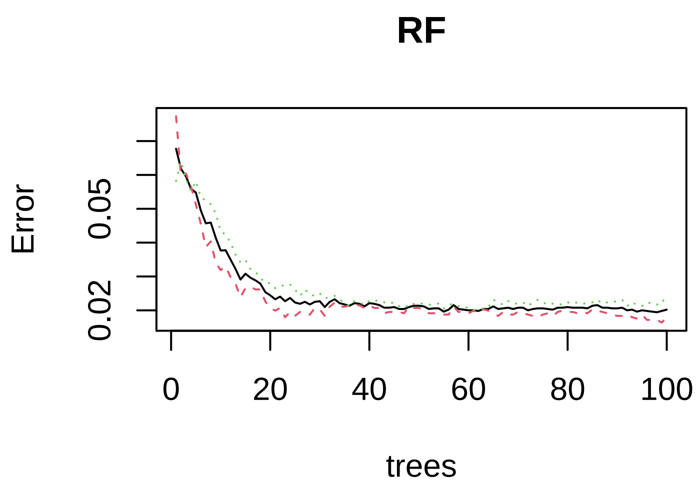
```
##   k
## 3 5
```



4.3 Random forest

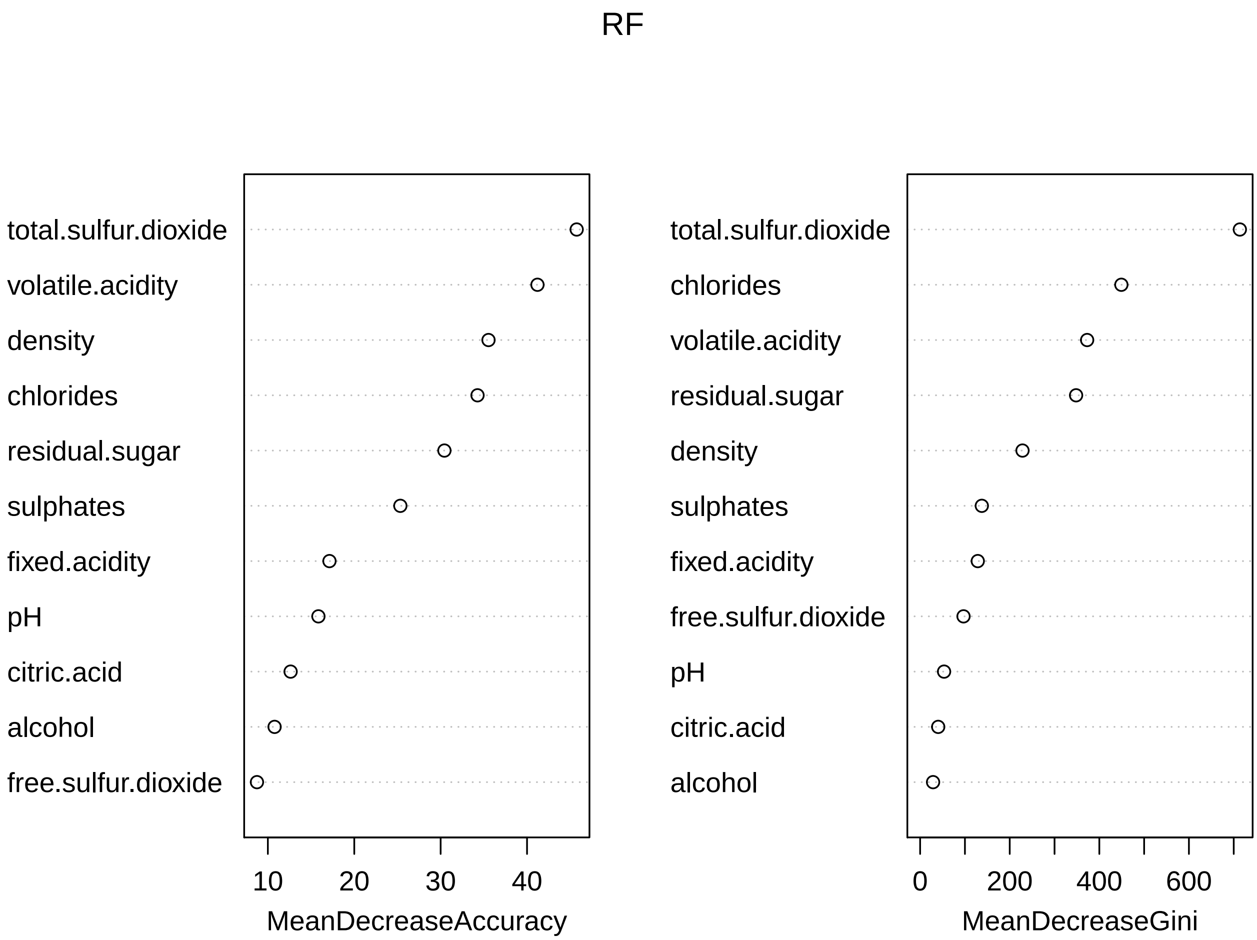
- Here, we used *'randomForest'* R package and tuned *mtry* value with smallest out of bag(OOB) error.
- Plots for variable importance based on either *MeanDesearseAccuracy* or *MeanDesearseGini* were provided for variable selection.

```
##
## Call:
##  randomForest(formula = type ~ ., data = rose_train, importance = T,      ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 2.02%
## Confusion matrix:
##       white  red class.error
## white  2515   46  0.01796173
## red      59 2578  0.02237391
```

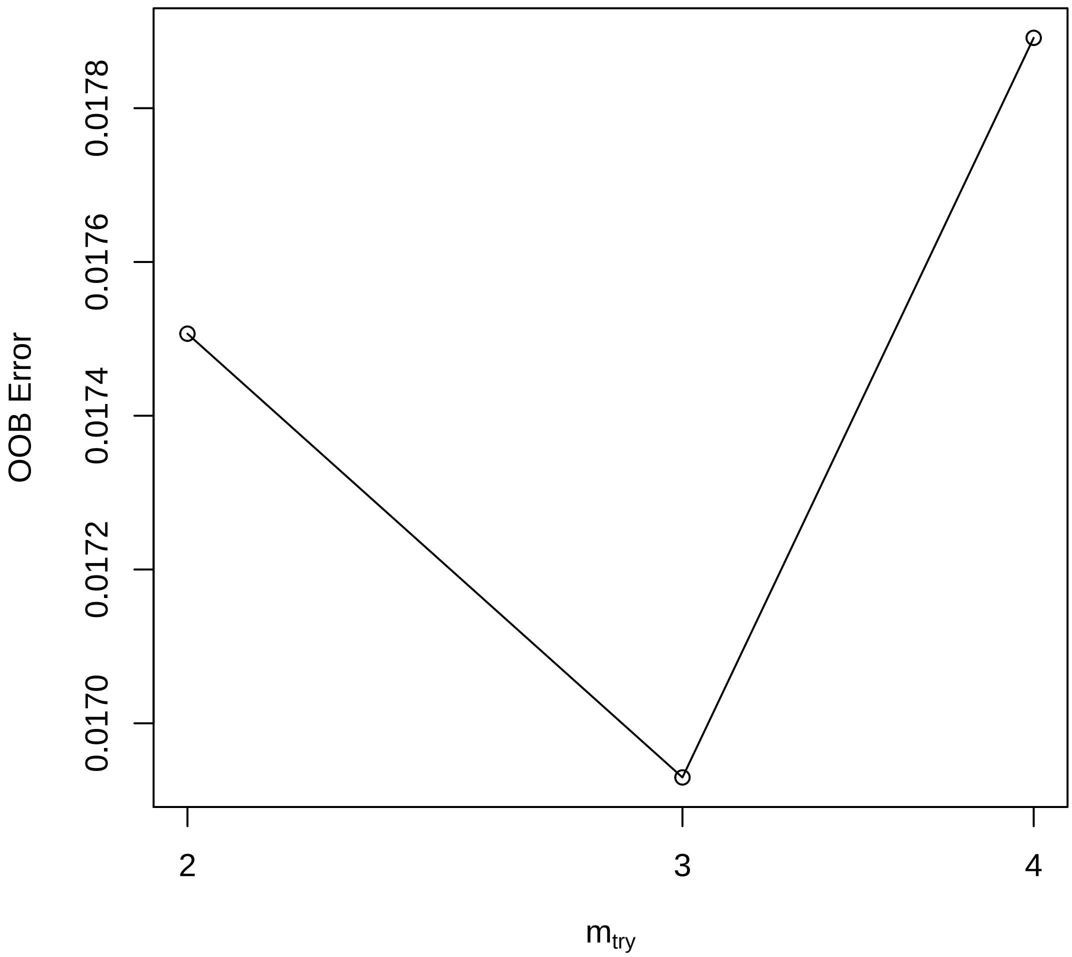



```
##                        white       red MeanDecreaseAccuracy MeanDecreaseGini
## fixed.acidity      16.236029 11.392739            17.130620        128.65467
## volatile.acidity   34.507548 25.790684            41.209756        372.80633
## citric.acid        12.016610  4.358919            12.633026         40.33570
## residual.sugar     18.756734 27.350086            30.436534        348.13763
```

```
## chlorides            28.334147 26.548532           34.270074         449.17942
## free.sulfur.dioxide   3.451608  7.740919            8.736965          96.83638
## total.sulfur.dioxide 30.351050 39.227384           45.747065         713.67576
## density              22.435608 27.678909           35.542487         228.51888
## pH                   11.714064 12.296837           15.851805          53.43231
## sulphates            18.706460 18.477426           25.335420         137.66892
## alcohol               6.114723  7.190922           10.779082          28.86811
```




```
## mtry = 2  OOB error = 1.75%
## Searching left ...
## Searching right ...
## mtry = 3     OOB error = 1.69%
## 0.03296703 0.01
## mtry = 4     OOB error = 1.79%
## -0.05681818 0.01
```




```
##       mtry   OOBError
## 2.OOB    2 0.01750673
## 3.OOB    3 0.01692959
## 4.OOB    4 0.01789150

## [1] 3
```

- **Brief summary**
  - In SVM, the best values of *cost* and *gamma* were "1" and "0.05", with the lowest error (0.0165).
  - In kNN, the best value of k was 5, with the greatest accuracy (0.8838).
  - In RF, the best value of *mtry* was 3, with the lowest OOBerror (0.0169).
  - Consistent with previous results, the variance important plot also revealed that volatile acid, chlorides and total sulfur dioxide were the top important variables to distinguish two wine types.

5. Model comparisons

After obtaining the well-tuned parameters, the final prediction models were then constructed using these new parameter settings so that we could further compare the prediction performance among three different models and their corresponding classification results. Hence, we used ***confusion matrix*** to visualize the classification results and ***accuracy*** to compare the performances of three models.

5.1 Confusion matrix

- SVM

	red	white
red	267	81
white	66	885

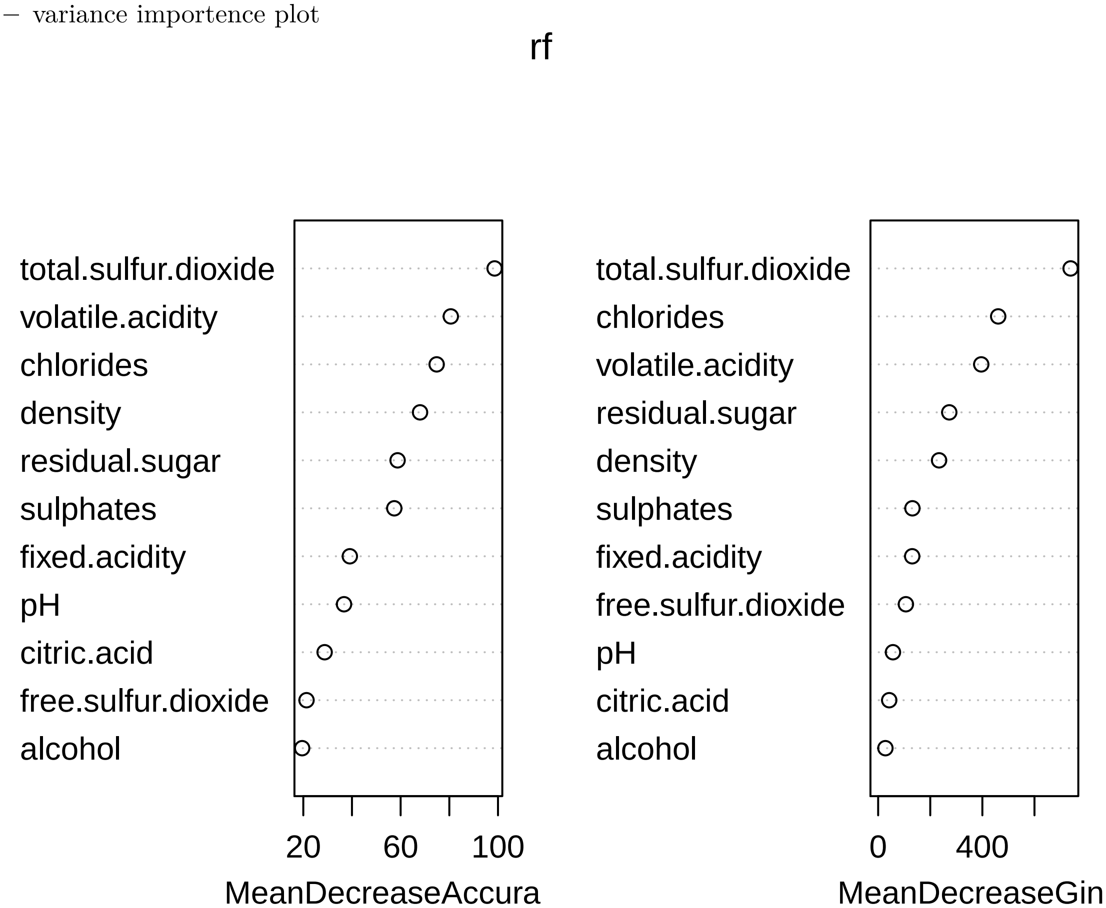
- kNN

	red	white
red	304	125
white	29	841

- Random forest
  - confusion matrix

	red	white
red	325	5
white	8	961

  - variance importence plot



5.2 Performance measurement

- **Brief summary**

	Accuracy
SVM	0.8868360
kNN	0.8814473
RF	0.9899923

- For the accuracy measures, the RF model achieved the highest results (98.99 %) followed by the SVM (88.68%) and kNN (88.14%).

6. Discussion

There are two main issues that need to be further discussed.

- Feature engineering and data standardization may help to improve the model performance.
- Ensemble learning approaches, such as bagging, boosting and stacking, can be further ultized to obtain better predictive performance.

Session Information

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.0.5 (2021-03-31)
##  os       macOS Big Sur 10.16
##  system   x86_64, darwin17.0
##  ui       X11
##  language (EN)
##  collate  zh_TW.UTF-8
##  ctype    zh_TW.UTF-8
##  tz       Asia/Taipei
##  date     2021-05-16
##
## - Packages -------------------------------------------------------------------
##  package      * version    date       lib source
##  abind          1.4-5      2016-07-21 [1] CRAN (R 4.0.2)
##  assertthat     0.2.1      2019-03-21 [1] CRAN (R 4.0.2)
##  backports      1.2.1      2020-12-09 [1] CRAN (R 4.0.2)
##  broom          0.7.6      2021-04-05 [1] CRAN (R 4.0.5)
##  car            3.0-10     2020-09-29 [1] CRAN (R 4.0.2)
##  carData        3.0-4      2020-05-22 [1] CRAN (R 4.0.2)
##  caret        * 6.0-86     2020-03-20 [1] CRAN (R 4.0.2)
##  cellranger     1.1.0      2016-07-27 [1] CRAN (R 4.0.2)
##  class        * 7.3-18     2021-01-24 [1] CRAN (R 4.0.5)
##  cli            2.4.0      2021-04-05 [1] CRAN (R 4.0.5)
##  codetools      0.2-18     2020-11-04 [1] CRAN (R 4.0.5)
##  colorspace     2.0-0      2020-11-11 [1] CRAN (R 4.0.2)
##  crayon         1.4.1      2021-02-08 [1] CRAN (R 4.0.2)
##  curl           4.3        2019-12-02 [1] CRAN (R 4.0.1)
##  data.table     1.14.0     2021-02-21 [1] CRAN (R 4.0.2)
##  DBI            1.1.1      2021-01-15 [1] CRAN (R 4.0.2)
##  digest         0.6.27     2020-10-24 [1] CRAN (R 4.0.2)
##  dplyr          1.0.5      2021-03-05 [1] CRAN (R 4.0.2)
##  e1071        * 1.7-6      2021-03-18 [1] CRAN (R 4.0.2)
##  ellipsis       0.3.1      2020-05-15 [1] CRAN (R 4.0.2)
##  evaluate       0.14       2019-05-28 [1] CRAN (R 4.0.1)
##  fansi          0.4.2      2021-01-15 [1] CRAN (R 4.0.2)
```

```
## farver         2.1.0      2021-02-28 [1] CRAN (R 4.0.2)
## forcats        0.5.1      2021-01-27 [1] CRAN (R 4.0.2)
## foreach        1.5.1      2020-10-15 [1] CRAN (R 4.0.2)
## foreign        0.8-81     2020-12-22 [1] CRAN (R 4.0.5)
## generics       0.1.0      2020-10-31 [1] CRAN (R 4.0.2)
## ggplot2      * 3.3.3      2020-12-30 [1] CRAN (R 4.0.2)
## ggpubr       * 0.4.0      2020-06-27 [1] CRAN (R 4.0.2)
## ggsignif       0.6.1      2021-02-23 [1] CRAN (R 4.0.2)
## glue           1.4.2      2020-08-27 [1] CRAN (R 4.0.2)
## gower          0.2.2      2020-06-23 [1] CRAN (R 4.0.2)
## gridExtra    * 2.3        2017-09-09 [1] CRAN (R 4.0.2)
## gtable         0.3.0      2019-03-25 [1] CRAN (R 4.0.2)
## haven          2.3.1      2020-06-01 [1] CRAN (R 4.0.2)
## hms            1.0.0      2021-01-13 [1] CRAN (R 4.0.2)
## htmltools      0.5.1.1    2021-01-22 [1] CRAN (R 4.0.2)
## httr           1.4.2      2020-07-20 [1] CRAN (R 4.0.2)
## ipred          0.9-11     2021-03-12 [1] CRAN (R 4.0.2)
## iterators      1.0.13     2020-10-15 [1] CRAN (R 4.0.2)
## kableExtra   * 1.3.4      2021-02-20 [1] CRAN (R 4.0.2)
## knitr        * 1.31       2021-01-27 [1] CRAN (R 4.0.2)
## labeling       0.4.2      2020-10-20 [1] CRAN (R 4.0.2)
## lattice      * 0.20-41    2020-04-02 [1] CRAN (R 4.0.5)
## lava           1.6.9      2021-03-11 [1] CRAN (R 4.0.2)
## lifecycle      1.0.0      2021-02-15 [1] CRAN (R 4.0.2)
## lubridate      1.7.10     2021-02-26 [1] CRAN (R 4.0.2)
## magrittr       2.0.1      2020-11-17 [1] CRAN (R 4.0.2)
## MASS           7.3-53.1   2021-02-12 [1] CRAN (R 4.0.5)
## Matrix         1.3-2      2021-01-06 [1] CRAN (R 4.0.5)
## ModelMetrics   1.2.2.2    2020-03-17 [1] CRAN (R 4.0.2)
## munsell        0.5.0      2018-06-12 [1] CRAN (R 4.0.2)
## nlme           3.1-152    2021-02-04 [1] CRAN (R 4.0.5)
## nnet           7.3-15     2021-01-24 [1] CRAN (R 4.0.5)
## openxlsx       4.2.3      2020-10-27 [1] CRAN (R 4.0.2)
## pillar         1.5.1      2021-03-05 [1] CRAN (R 4.0.2)
## pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 4.0.2)
## plyr           1.8.6      2020-03-03 [1] CRAN (R 4.0.2)
## pROC           1.17.0.1   2021-01-13 [1] CRAN (R 4.0.2)
## prodlim        2019.11.13 2019-11-17 [1] CRAN (R 4.0.2)
## proxy          0.4-25     2021-03-05 [1] CRAN (R 4.0.2)
## purrr          0.3.4      2020-04-17 [1] CRAN (R 4.0.2)
## R6             2.5.0      2020-10-28 [1] CRAN (R 4.0.2)
## randomForest * 4.6-14     2018-03-25 [1] CRAN (R 4.0.2)
## Rcpp           1.0.6      2021-01-15 [1] CRAN (R 4.0.2)
## readxl         1.3.1      2019-03-13 [1] CRAN (R 4.0.2)
## recipes        0.1.16     2021-04-16 [1] CRAN (R 4.0.2)
## reshape2       1.4.4      2020-04-09 [1] CRAN (R 4.0.2)
## rio            0.5.26     2021-03-01 [1] CRAN (R 4.0.2)
## rlang          0.4.10     2020-12-30 [1] CRAN (R 4.0.2)
## rmarkdown      2.7        2021-02-19 [1] CRAN (R 4.0.2)
## ROSE         * 0.0-3      2014-07-15 [1] CRAN (R 4.0.2)
## rpart          4.1-15     2019-04-12 [1] CRAN (R 4.0.5)
## rstatix        0.7.0      2021-02-13 [1] CRAN (R 4.0.2)
## rstudioapi     0.13       2020-11-12 [1] CRAN (R 4.0.2)
## rvest          1.0.0      2021-03-09 [1] CRAN (R 4.0.2)
```

```
##  scales         1.1.1      2020-05-11 [1] CRAN (R 4.0.2)
##  sessioninfo    1.1.1      2018-11-05 [1] CRAN (R 4.0.2)
##  stringi        1.5.3      2020-09-09 [1] CRAN (R 4.0.2)
##  stringr        1.4.0      2019-02-10 [1] CRAN (R 4.0.2)
##  survival       3.2-10     2021-03-16 [1] CRAN (R 4.0.5)
##  svglite        2.0.0      2021-02-20 [1] CRAN (R 4.0.2)
##  systemfonts    1.0.2      2021-05-11 [1] CRAN (R 4.0.2)
##  tibble         3.1.0      2021-02-25 [1] CRAN (R 4.0.2)
##  tidyr        * 1.1.3      2021-03-03 [1] CRAN (R 4.0.2)
##  tidyselect     1.1.0      2020-05-11 [1] CRAN (R 4.0.2)
##  timeDate       3043.102   2018-02-21 [1] CRAN (R 4.0.2)
##  utf8           1.2.1      2021-03-12 [1] CRAN (R 4.0.2)
##  vctrs          0.3.7      2021-03-29 [1] CRAN (R 4.0.2)
##  viridisLite    0.3.0      2018-02-01 [1] CRAN (R 4.0.1)
##  webshot        0.5.2      2019-11-22 [1] CRAN (R 4.0.2)
##  withr          2.4.1      2021-01-26 [1] CRAN (R 4.0.2)
##  xfun           0.22       2021-03-11 [1] CRAN (R 4.0.2)
##  xml2           1.3.2      2020-04-23 [1] CRAN (R 4.0.2)
##  yaml           2.2.1      2020-02-01 [1] CRAN (R 4.0.2)
##  zip            2.1.1      2020-08-27 [1] CRAN (R 4.0.2)
##
## [1] /Library/Frameworks/R.framework/Versions/4.0/Resources/library
```