School of Computer Science
The University of Adelaide

Artificial Intelligence
Assignment 2

Semester 1 2022
Due 11:59pm Saturday 7 May

---

1 Wine Quality Prediction with Decision Tree

Wine experts evaluate the quality of wine based on sensory data. We could also collect the features of wine from objective tests, thus the objective features could be used to predict the expert's judgement, which is the quality rating of the wine. This could be formed as a supervised learning problem with the objective features as the data features and wine quality rating as the data labels. In this assignment, we provide objective features obtained from physicochemical statistics for each white wine sample and its corresponding rating provided by wine experts. You are expect to implement **decision tree learning (DTL)**, and use the training set to train your decision tree, then provide wine quality prediction on the test set.

Wine quality rating is measured in the range of 0-9. In our dataset, we only keep the samples for quality ratings 5, 6 and 7. The 11 objective features are listed as follows [1]:

- f_acid: fixed acidity
- v_acid: volatile acidity
- c_acid: citric acid
- res_sugar: residual sugar
- chlorides: chlorides
- fs_dioxide: free sulfur dioxide
- ts_dioxide: total sulfur dioxide
- density: density
- pH: pH
- sulphates: sulphates
- alcohol: alcohol

Explanation of the Data.
`train`: The first 11 columns represent the 11 features and the 12th column is the wine quality. A sample is depicted as follows:

f_acid	v_acid	c_acid	res_sugar	chlorides	fs_dioxide	ts_dioxide	density	pH	sulphates	alcohol	quality
8.10	0.270	0.41	1.45	0.033	11.0	63.0	0.99080	2.99	0.56	12.0	5
8.60	0.230	0.40	4.20	0.035	17.0	109.0	0.99470	3.14	0.53	9.7	5
7.90	0.180	0.37	1.20	0.040	16.0	75.0	0.99200	3.18	0.63	10.8	5
8.30	0.420	0.62	19.25	0.040	41.0	172.0	1.00020	2.98	0.67	9.7	5
6.50	0.310	0.14	7.50	0.044	34.0	133.0	0.99550	3.22	0.50	9.5	5

`test`: Similar to `train`, but without the 12th colum as they are the values your model will predict. A sample is depicted as follows:

f_acid	v_acid	c_acid	res_sugar	chlorides	fs_dioxide	ts_dioxide	density	pH	sulphates	alcohol
7.0	0.360	0.14	11.60	0.043	35.0	228.0	0.99770	3.13	0.51	8.900000
6.3	0.270	0.18	7.70	0.048	45.0	186.0	0.99620	3.23	0.47	9.000000
7.2	0.290	0.20	7.70	0.046	51.0	174.0	0.99582	3.16	0.52	9.500000
7.1	0.140	0.35	1.40	0.039	24.0	128.0	0.99212	2.97	0.68	10.400000
7.6	0.480	0.28	10.40	0.049	57.0	205.0	0.99748	3.24	0.45	9.300000

1.1 Decision Tree Learning

From the given training data, our goal is to learn a function that can predict the wine quality rating of a wine sample, based on the objective features. In this assignment, the predictor function will be constructed as a decision tree. Since the attributes (objective features) are continuous valued, you shall apply the DTL algorithm for continuous data, as outlined in Algorithms 1 and 2. Once the tree is constructed, Algorithm 3 to predict the wine quality to a new wine sample.

Algorithm 1 DTL(*data*, *minleaf*)

Require: *data* in the form of N input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{N}, minleaf \geq 1$
if $(N \leq minleaf)$ or $(y_i = y_j$ for all $i, j)$ or $(\mathbf{x}_i = \mathbf{x}_j$ for all $i, j)$ **then**
  Create new leaf node n.
  if there is a unique mode (most frequent value) in $\{y_i\}_{i=1}^{N}$ **then**
    $n.label \leftarrow$ mode in $\{y_i\}_{i=1}^{N}$
  else
    $n.label \leftarrow$ `unknown`
  end if
  return n
end if
[attr, splitval] $\leftarrow$ ChooseSplit*(data)* $\Longrightarrow$ Algorithm 2
Create new node n.
$n.attr \leftarrow attr$
$n.splitval \leftarrow splitval$
$n.left \leftarrow$ DTL($data$ with $\mathbf{x}_i[attr] \leq splitval, minleaf$)
$n.right \leftarrow$ DTL($data$ with $\mathbf{x}_i[attr] > splitval, minleaf$)
return n

Algorithm 2 ChooseSplit($data$)

Require: $data$ in the form of N input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$.

```
bestgain ← 0
for each attr in data do
    Sort the array x_1[attr], x_2[attr], ..., x_N[attr].
    for i = 1, 2, ...N − 1 do
        splitval ← 0.5(x_i[attr] + x_{i+1}[attr])
        gain ← Information gain of (attr, splitval) // See lecture slides.
        if gain > bestgain then
            bestattr ← attr and bestsplitval ← splitval and bestgain ← gain
        end if
    end for
end for
return (bestattr, bestsplitval)
```

Algorithm 3 Predict_DTL($n, data$)

Require: Decision tree root node n, $data$ in the form of attribute values $\mathbf{x}$.

```
while n is not a leaf node do
    if x[n.attr] ≤ n.splitval then
        n ← n.left
    else
        n ← n.right
    end if
end while
return n.label
```

1.2 Deliverable

Write your decision tree learning program in Python 3.6.9 in a file called `winequality.py`. Your program must be able to run as follows:

```
$ python winequality.py [train] [test] [minleaf]
```

The inputs/options to the program are as follows:

- `[train]` specifies the path to a set of training data file.
- `[test]` specifies the path to a set of testing data file.
- `[minleaf]` is an integer greater than zero which specifies the second input parameter to the DTL algorithm (Algorithm 1).

Given the inputs, your program must learn a decision tree (following the prescribed algorithms) using the training data, then predict the quality rating of each of the wine sample in the testing data. Your program must then **print to standard output** (i.e., the command prompt) the list of predicted wine quality ratings, vertically based on the order in which the testing cases appear in `[test]`.

1.2.1 Python libraries

You are allowed to use the Python standard library to write your decision tree learning program (see `https://docs.python.org/3/library/` for the components that make up the Python v3.6.9 standard library). In addition to the standard library, you are allowed to use NumPy. Note that the marking program will not be able to run your program to completion if other third party libraries are used.

1.3 Submission

You must submit your program files on Gradescope. Instructions on accessing Gradescope and submitting assignments are provided at `https://help.gradescope.com/article/5d3ifaeqi4-student-canvas`. Please use the course code **X3ZJZE** to enrol into the course. **For undergraduates**, please submit your decision tree learning program (`winequality.py`) to **Assignment 2 - Undergraduates**. If there are any questions or issues with Gradescope, please contact Andrew via email at andrew.du@adelaide.edu.au.

1.4 Expected run time

Your program must be able to terminate within 300 seconds on the sample data given.

1.5 Debugging Suggestions

Step-by-step debugging by checking intermediate values/results will help you to identify the problems of your code. This function is enabled by most of the Python IDE. If not in your case, you could also print the intermediate values out. The values that are worth checking when debugging are (but not limited to):

- *bestsplitval*, *bestgain*, *bestattr* when splitting;
- *n.splitval*, *n.attr*, *n.label*, when creating nodes and prediction

You could use sample data or create data in the same format for debugging.

1.6 Assessment

I will compile and run your code on several test problems. If it passes all tests, you will get **15%** (undergrads) or **12%** (postgrads) of the overall course mark. **For undergraduates**, bonus marks of **3%** will be awarded if Section 2 is completed correctly.

There will be no further manual inspection/grading of your program to award marks on the basis of coding style, commenting or "amount" of code written.

1.7 Using other source code

You may not use other source code for this assignment. All submitted code must be your own work written from scratch. Only by writing the solution yourself will you fully understand the concept.

1.8 Due date and late submission policy

This assignment is due by **11:59pm Saturday 7 May**. If your submission is late, the maximum mark you can obtain will be reduced by 25% per day (or part thereof) past the due date or any extension you are granted.

Continues next page for postgraduate section.

2 Wine Quality Prediction with Random Forest

For postgraduate students, completing this section will give you the remaining **3%** of the assignment marks.

Random forest is an ensemble method which combines predictions of multiple decision trees. In this task, you will extend your knowledge learnt from decision tree learning to random forest learning (RFL). The process for a simplified RFL given N input-output pairs is:

(1) Randomly select a set of N samples via bootstrap sampling (see explianation later). This dataset is used for training a decision tree (i.e., the root node of the decision tree).

(2) Build a decision tree on the dataset from (1) and apply Algorithm 1.

(3) Repeat (1) and (2) until reaching the maximum number of trees.

This process is also shown in Algorithm 4. In random forest learning, a sample set is used to train a decision tree. That is to say, different trees in the forest could have different root data. For prediction, the random forest will choose the most voted label as its prediction.

For the wine quality prediction task, you shall apply Algorithm 4 for random forest learning and apply Algorithm 5 to predict the wine quality for a new wine sample.

Bootstrap sampling. It is a random sampling with replacement (i.e., replace the sampled data back to the original dataset). For example, if $N = 5$, so we have $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_5\}$ (omit y_i for simplicity). To perform bootstrap sampling, we randomly select one data point from the five, let's say $\mathbf{x}_3$, to be the first sample. Then we put $\mathbf{x}_3$ back to the dataset and draw another sample from the five data points. This time, the sample could also be $\mathbf{x}_3$ or another one. We repeat this process until we get N samples and form a sample set. In this way, the sample set could contain repeated data points. So the following cases are possible:

$\{\mathbf{x}_3, \mathbf{x}_3, \mathbf{x}_3, \mathbf{x}_3, \mathbf{x}_3\}$, when all the data samples are the same.
$\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_2, \mathbf{x}_5\}$, when all the data samples are different.
$\{\mathbf{x}_3, \mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1\}$, when some samples are repeated.
...

2.1 Deliverables

Write your random forest program in Python 3.6.9 in a file called `winequalityRFL.py`. Your program must be able to run as follows:

```
$ python winequalityRFL.py [train] [test] [minleaf] [n_trees] [random_seed]
```

The inputs/options to the program are as follows:

- `[train]` specifies the path to a set of training data file.

Algorithm 4 RFL(*data, minleaf, n_trees, rand_seed)*

Require: *data* in the form of N input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{N}, n\_trees > 1, minleaf \geq 1$.
1: $forest \leftarrow []$
2: *sample_indexes* $\leftarrow N * n\_trees$ random integers with value in $[0, N)$ generated by *rand_seed*
3: *count* $\leftarrow 0$
4: **for** *count* $< n\_trees$ **do**
5: *sampled_data* $\leftarrow N$ data pairs selected by N indexes from *sample_indexes* sequentially
6: n=DTL(*sampled_data*, *minleaf*) $\Longrightarrow$ Algorithm 1
7: $forest.append(n)$
8: **end for**
9: **return** $forest$

Algorithm 5 Predict_RFL($forest, data$)

Require: $forest$ is a list of tree roots, *data* in the form of attribute values $\mathbf{x}$.
1: *labels* $\leftarrow []$
2: **for** Each tree n in the $forest$ **do**
3: *label* $\leftarrow$ Predict_DTL $(n, data) \Longrightarrow$ Algorithm 3
4: $labels.append(n)$
5: **end for**
6: **return** the most voted label in *labels*

- `[test]` specifies the path to a set of testing data file.
- `[minleaf]` is an integer greater than zero which specifies the second input parameter to the RFL algorithm (Algorithm 4).
- `[n_trees]` is an integer greater than one which specifies the third input parameter to the RFL algorithm (Algorithm 4).
- `[random_seed]` is the seed value generate random values. Please use import random package, use random.seed(*rand_seed*) to set seed value and random.randint(0, M) to create a random value in [0, M], repeat random.randint to get enough random values.

Given the inputs, your program must learn a random forest (following the prescribed algorithms) using the training data, then predict the quality rating of each wine sample in the testing data. Your program must then **print to standard output** (i.e., the command prompt) the list of predicted wine quality ratings, vertically based on the order in which the testing cases appear in `[test]`.

Submit your program in the same way as the submission for Sec. 1. **For postgraduates**, please submit your learning programs (`winequality.py` and `winequalityRFL.py`) to **Assignment 2 - Postgraduates**. The due date, late submission policy and code reuse policy are also the same as in Sec. 1.

2.2 Expected run time

Your program must be able to terminate within 300 seconds on the sample data given.

2.3 Debugging Suggestions

In addition to Sec. 1.5, another value worth checking when debugging is (but not limited to):

- the $sample_indexes$ – by setting random seed, the indexes should be the same each time you run the code

2.4 Assessment

I will compile and run your code on a single test case. If it passes, you will get **3%** of the overall course mark.

$\sim\sim\sim$ The End $\sim\sim\sim$

References

[1] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems 47*, 4 (2009), 547–553.