

## 03\_ml\_01: ML1 (RF and GridSearchCV)

This homework is a classification task to identify whether a mushroom is edible or poisonous. This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981).

Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the credibility of a mushroom; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.

### Problems

Please complete class **MushroomClassifier** using the provided code template. The goal is to differentiate between edible and non-edible mushrooms. The details are as follows:

1. Load 'mushroom2020\_dataset.csv' data from the “Attachment” (note: this data set has been preliminarily prepared.).
2. Drop rows where the target (label) variable is missing.
3. Drop the following variables: 'id','gill-attachment', 'gill-spacing', 'gill-size','gill-color-rate', 'stalk-root', 'stalk-surface-above-ring', 'stalk-surface-below-ring', 'stalk-color-above-ring-rate','stalk-color-below-ring-rate','veil-color-rate','veil-type'
4. Examine the number of rows, the number of digits, and whether any are missing.
5. Fill missing values by adding the mean for numeric variables and the mode for nominal variables.
6. Convert the label variable e (edible) to 1 and p (poisonous) to 0 and check the quantity.  
class0: class1
7. Convert the nominal variable to numeric using a dummy code with drop\_first = True.
8. Split train/test with 20% test, stratify, and seed = 2020.
9. Create a Random Forest with GridSearch on training data with 5 CV with n\_jobs=-1.  
'criterion':['gini','entropy'] , 'max\_depth': [2,3] , 'min\_samples\_leaf':[2,5] ,  
'N\_estimators':[100] , 'random\_state': 2020
10. Predict the testing data set with classification\_report.

And return the output based on the question number:

- For Q1, following step 1, please returns an integer number of "na" are there in "gill-size" variables before doing the data prep.
- For Q2, following step 2-4, please return a tuple of rows of data and variables.
- For Q3, following step 5-6, please return quantity class0:class1.
- For Q4, following step 7-8, please return tuple of training and testing sets.
- For Q5, following step 9, please return best params after doing random forest grid search.
- For Q6, following step 10, return classification\_report.

Submission: **\*\* When submitting to the grader, submit ONLY libraries, class MushroomClassifier with your modified functions.\*\***

### Expected Results

Input	Output																														
print(hw.Q1())	121																														
print(hw.Q2())	(5764, 12)																														
print(hw.Q3())	label 0 3660 1 2104 Name: count, dtype: int64																														
print(hw.Q4())	((4611, 42), (1153, 42))																														
print(hw.Q5())	{'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 5, 'n_estimators': 100, 'random_state': 2020}																														
print(hw.Q6())	<table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>732</td> </tr> <tr> <td>1</td> <td>0.96</td> <td>0.97</td> <td>0.97</td> <td>421</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.98</td> <td>1153</td> </tr> <tr> <td>macro avg</td> <td>0.97</td> <td>0.98</td> <td>0.97</td> <td>1153</td> </tr> <tr> <td>weighted avg</td> <td>0.98</td> <td>0.98</td> <td>0.98</td> <td>1153</td> </tr> </tbody> </table>		precision	recall	f1-score	support	0	0.98	0.98	0.98	732	1	0.96	0.97	0.97	421	accuracy			0.98	1153	macro avg	0.97	0.98	0.97	1153	weighted avg	0.98	0.98	0.98	1153
	precision	recall	f1-score	support																											
0	0.98	0.98	0.98	732																											
1	0.96	0.97	0.97	421																											
accuracy			0.98	1153																											
macro avg	0.97	0.98	0.97	1153																											
weighted avg	0.98	0.98	0.98	1153																											

## Template codes

```

class MushroomClassifier:
    def __init__(self, data_path): # DO NOT modify this line
        self.data_path = data_path
        self.df = pd.read_csv(data_path)

    def Q1(self): # DO NOT modify this line
        """
        1. (From step 1) Before doing the data prep., how many "na" are there in
           "gill-size" variables?
        """
        # remove pass and replace with you code
        pass

    def Q2(self): # DO NOT modify this line
        """
        2. (From step 2-4) How many rows of data, how many variables?
        """
        # remove pass and replace with you code
        pass

    def Q3(self): # DO NOT modify this line
        """
        3. (From step 5-6) Answer the quantity class0:class1
        """
        # remove pass and replace with you code
        pass

    def Q4(self): # DO NOT modify this line
        """
        4. (From step 7-8) How much is each training and testing sets
        """
        # remove pass and replace with you code
        pass

    def Q5(self): # DO NOT modify this line
        """
        5. (From step 9) Best params after doing random forest grid search.
        """
        # remove pass and replace with you code
        pass

    def Q6(self): #Question6(self): # DO NOT modify this line
        """
        5. (From step 10) What is the value of macro f1 (2 digit) Predict the
           testing data set with classification_report, using scientific rounding
           (less than 0.5 dropped, more than 0.5 then increased)
        """
        # remove pass and replace with you code
        pass

def main(): # DO NOT modify this line
    hw = MushroomClassifier('mushroom2020_dataset.csv') #your csv path
    exec(input().strip()) # DO NOT modify this line

if __name__ == "__main__": # DO NOT modify this line
    main()

```