Pattern Recognition, Homework 3

- 309553018 林孟學

1. Part 1

(1) (5%) Please compute the Entropy and Gini Index of the given array by the formula on the slides.

```
Gini of data is  0.4628099173553719
Entropy of data is  0.9456603046006401
```

(2) (20%) Implement the Decision Tree algorithm (CART, Classification and Regression Trees) and train the model by the given arguments, and print the accuracy score on the test data.

```
def predict(self, X):
    '''
    input: X→ dataframe of 1 or more rows

    return: np array
    '''
    X = X[list(self.used_attribute)]
    # choose ONLY used attribute
    X = X.to_dict('records')
    # X → list of dictionaries
    pred = []
    for x in X:
        # x: dictionary
        Node = self.root
        while Node.predict == None:
            if float(x[Node.split_attr]) < Node.split_threshold:
                Node = Node.leftNode
            else:
                Node = Node.rightNode
        # come to leaf node
        pred += [Node.predict]
    return np.array(pred)
```

```
def accuracy(pred, real):
    '''
    pred, real: np array of shape (N)

    pred - real: 0→ same, +-1→ diff
    '''
    return (len(pred) - sum(abs(pred-real))) / len(pred)
```

(2.1) Using Criterion='gini' to train the model and show the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.

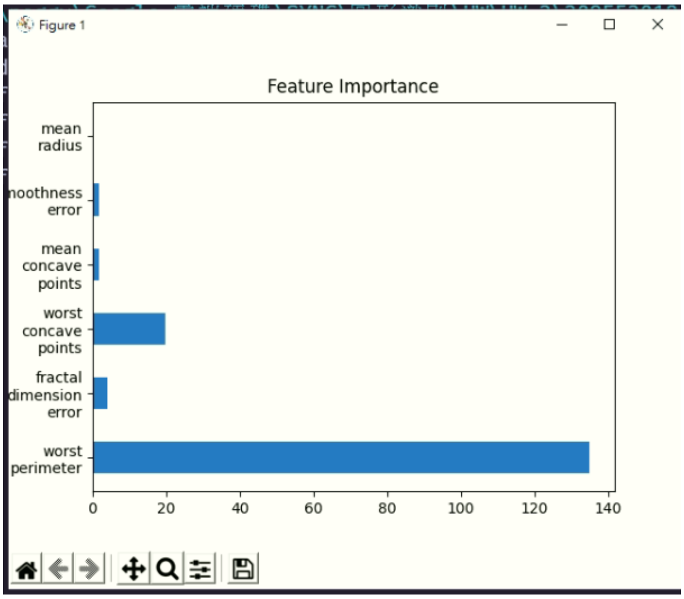
```
Accuracy clf_depth3:       0.916083916083916
Accuracy clf_depth10:      0.9090909090909091
```

(2.2) Using Max_depth=3 to train the model and show the accuracy score of test data by Criterion='gini' and Criterion='entropy', respectively.

```
Accuracy clf_depth3:       0.916083916083916
Accuracy clf_depth10:      0.9090909090909091
```

(3) (15%) Plot the feature importance of your Decision Tree model. You can use the model for Question 2.1, max_depth=10.

Here I use the formula in the reference.



(4) (20%) Implement the random forest algorithm by using the CART you just implemented for Question 2.

(4.1) Using Criterion='gini', Max_depth=None, Max_features=sqrt(n_features), Bootstrap=True to train the model and show the accuracy score of test data by n_estimators=10 and n_estimators=100, respectively.

```python
def predict(self, X):
    predictSum = np.zeros(len(X.index))
    for DT in self.decision_trees:
        predictSum += DT.predict(X)

    predictSum /= self.n_estimators

    pred = np.zeros(len(X.index), dtype=int)
    pred[predictSum < 0.5] = 0
    pred[predictSum >= 0.5] = 1

    return pred
```

```
Accuracy clf_10tree:     0.9300699300699301
Accuracy clf_100tree:    0.9370629370629371
```

(4.2) Using Criterion='gini', Max_depth=None, N_estimators=10, Bootstrap=True, to train the model and show the accuracy score of test data by Max_features=sqrt(n_features) and Max_features=n_features, respectively

```
Accuracy clf_random_features: 0.9440559440559441
Accuracy clf_all_features: 0.951048951048951
```

2. Part 2

(1)

(15%) By differentiating the error function below with respect to α_m,

$$E = e^{-\alpha_m/2} \sum_{n \in T_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in M_m} w_n^{(m)}$$

$$= (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

show that the parameters α_m in the AdaBoost algorithm are updated using $\alpha_m = ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$ in which ϵ_m is defined by $\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$.

$$\frac{\partial E}{\partial a_m} = \left(\frac{1}{2} e^{a_m/2} + \frac{1}{2} e^{-a_m/2}\right) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) + \frac{-1}{2} e^{-a_m/2} \sum_{n=1}^{N} w_n^{(m)} = 0$$

$$\frac{1}{2}\left(e^{a_m/2} + e^{-a_m/2}\right) \underbrace{\boxed{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) \Big/ \sum_{n=1}^{N} w_n^{(m)}}}_{\varepsilon_m} - \frac{1}{2} e^{-a_m/2} \sum_{n=1}^{N} w_n^{(m)} \Big/ \sum_{n=1}^{N} w_n^{(m)} = 0$$

$$\varepsilon_m e^{a_m/2} + \varepsilon_m e^{-a_m/2} - e^{-a_m/2} = 0$$

$$\varepsilon_m e^{a_m/2} = (1-\varepsilon_m) e^{-a_m/2}$$

$$\ln(\varepsilon_m) + \frac{a_m}{2} = \ln(1-\varepsilon_m) + \frac{-a_m}{2}$$

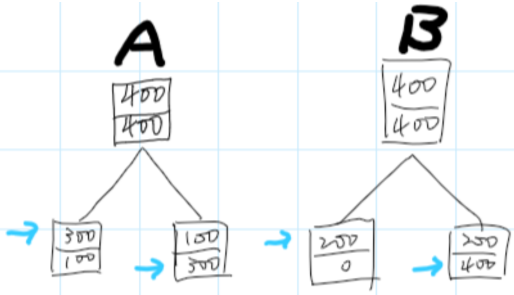
$$a_m = \ln(1-\varepsilon_m) - \ln(\varepsilon_m)$$

$$= \ln\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$$

(2)

(15%) Consider a data set comprising 400 data points from class C_1 and 400 data points from class C_2. Suppose that a tree model A splits these into (300, 100) assigned to the first leaf node (predicting C_1) and (100, 300) assigned to the second leaf node (predicting C_2), where (n, m) denotes that n points come from class C_1 and m points come from class C_2. Similarly, suppose that a second tree model B splits them into (200, 0) and (200, 400), respectively. Evaluate the misclassification rates for the two trees and hence show that they are equal. Similarly, evaluate the pruning criterion $C(T) = \sum_{\tau=1}^{|T|} Q_\tau(T) + \lambda|T|$ for the cross-entropy case $Q_\tau(T) = -\sum_{k=1}^{K} p_{\tau k} ln(p_{\tau k})$ for the two trees and show that tree B is lower than tree A. Leaf nodes are indexed by $\tau = 1, ..., |T|$, with leaf node τ represents a region R_τ, and $p_{\tau k}$ is the proportion of data points in region R_τ assigned to class k, where $k = 1, ..., K$.

Hint: The answer should contain λ which is the regularization parameter.



① **misclassification rate**

A: $(100+100)/800 = \frac{1}{4}$

B: $(\quad 200)/800 = \frac{1}{4}$

②

A $\quad P_{11} = \frac{300}{400} \quad P_{12} = \frac{100}{400}$

$P_{21} = \frac{100}{400} \quad P_{22} = \frac{300}{400}$

$Q_1(T_A) = -\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right)$

$Q_2(T_A) = -\left(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}\right)$

$C(T_A) = -2\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) + \lambda \times \overset{|T|}{2}$

$\doteqdot 0.488 + 2\lambda$

B

$P_{11} = \frac{200}{200}$ $P_{12} = \frac{0}{200}$

$P_{21} = \frac{200}{600}$ $P_{22} = \frac{400}{600}$

$Q_1(T_B) = -(1 \lg 1 + 0 \lg 0)$

$Q_2(T_B) = -(\frac{1}{3} \lg \frac{1}{3} + \frac{2}{3} \lg \frac{2}{3})$

$$C(T_B) = -(\frac{1}{3} \lg \frac{1}{3} + \frac{2}{3} \lg \frac{2}{3}) + \lambda \times 2$$

$$\fallingdotseq 0.159 + 2\lambda$$

$$C(T_A) > C(T_B)$$

$\fallingdotseq 0.488 + 2\lambda$ $\fallingdotseq 0.159 + 2\lambda$

(3)

(10%) Verify that if we minimize the sum-of-squares error between a set of training values $\{t_n\}_{n=1\sim N}$ (N is number of training data) and a single predictive value t, then the optimal solution for t is given by the mean of the $\{t_n\}_{n=1\sim N}$.

Find $\underset{t}{\operatorname{argmin}} \sum_{n=1}^{N} (t_n - t)^2$

$$\frac{d}{dt} \sum_{n=1}^{N} (t_n - t)^2 = 0$$

$$\sum_{n=1}^{N} 2(t_n - t) \times (-1) = 0$$

$$\sum_{n=1}^{N} (t_n - t) = 0$$

$$\sum_{n=1}^{N} t_n = \sum_{n=1}^{N} t = N \times t$$

$$t = \frac{\sum_{n=1}^{N} t_n}{N}$$