109550069_林宇柔

Machine Learning HW3

Part 1.

1. Gini Index and Entropy

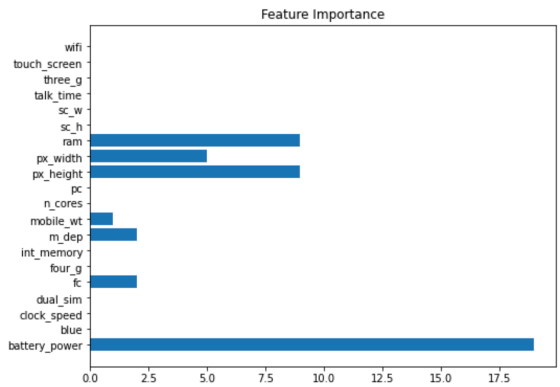
```
Gini of data is  0.4628099173553719
Entropy of data is  0.9456603046006401
```

2. Decision Tree

```
Decision Tree (criterion=gini, max_depth=3) accuracy: 0.92
Decision Tree (criterion=gini, max_depth=10) accuracy: 0.94
```

```
Decision Tree (criterion=gini, max_depth=3) accuracy: 0.92
Decision Tree (criterion=gini, max_depth=10) accuracy: 0.9333333333333333
```

3. Feature Importance



4. AdaBoost

```
AdaBoost (n_estimators=10) accuracy: 0.9433333333333334
AdaBoost (n_estimators=100) accuracy: 0.9766666666666667
```

5. Random Forest

```
Random Forest (n_estimators=10) accuracy: 0.71
Random Forest (n_estimators=100) accuracy: 0.9066666666666666
```

```
Random Forest (max_features = sqrt(n_features)) accuracy: 0.6966666666666667
Random Forest (max_features = n_features) accuracy: 0.9533333333333334
```

6. Train and Tune

```
Accuracy of tuned-model:  0.9766666666666667
```

Part 2.

1. Why does a decision tree have a tendency to overfit to the training set? Is it possible for a decision tree to reach a 100% accuracy in the training set? please explain. List and describe at least 3 strategies we can use to reduce the risk of overfitting of a decision tree.

Decision trees examine the training dataset thoroughly, if the maximum depth is not specified, they are able to find a possible split for every independent variable, so they are very data intensive, and thus it is possible for a decision tree to reach a 100% accuracy in the training set when overfitting occurs. To reduce the risk of overfitting, we can try cross-validation, training with more data, or removing features. Cross-validation splits the dataset into folds, and iteratively train the model on k-1 folds while using the remaining fold as the testing set, this allows to test the model on some completely unseen data. Training with more data is also a possible way since overfit might happen because the data size is too small and the model trains on the limited training data for several epochs. In addition, removing redundant features can lower the chance of overfitting because it makes the model less complex.

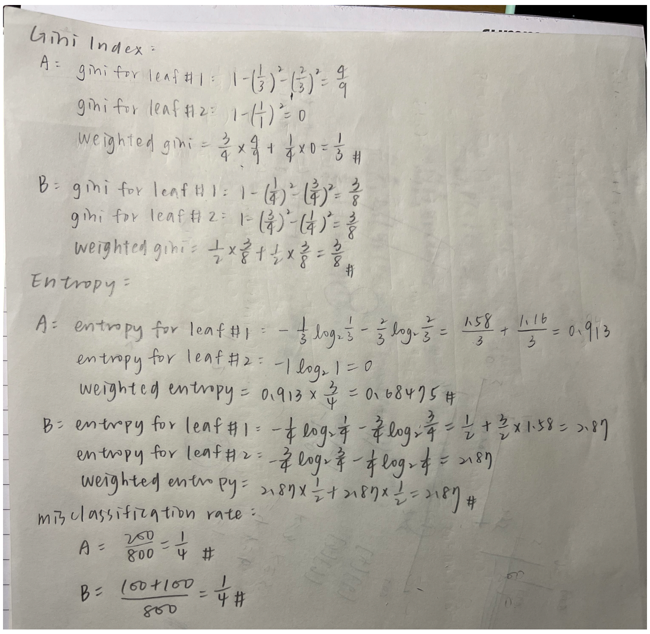
2. This part consists of three True/False questions. Answer True/False for each question and briefly explain your answer.

   (a) In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.
   False, the weights of the misclassified examples would be increased in order to emphasize.

   (b) In AdaBoost, weighted training error $\boldsymbol{\epsilon}_t$ of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t.
   False, it decreases since the updated weights try to minimize the exponential loss

   (c) AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.
   True, Adaboost achieves zero training error exponentially fast once the error is less than 0.5

3.



Gini Index:

A: gini for leaf #1: $1-(\frac{1}{3})^2-(\frac{2}{3})^2=\frac{4}{9}$

gini for leaf #2: $1-(\frac{1}{1})^2=0$

weighted gini $=\frac{3}{4}\times\frac{4}{9}+\frac{1}{4}\times 0=\frac{1}{3}$ #

B: gini for leaf #1: $1-(\frac{1}{4})^2-(\frac{3}{4})^2=\frac{3}{8}$

gini for leaf #2: $1-(\frac{3}{4})^2-(\frac{1}{4})^2=\frac{3}{8}$

weighted gini $=\frac{1}{2}\times\frac{3}{8}+\frac{1}{2}\times\frac{3}{8}=\frac{3}{8}$ #

Entropy:

A: entropy for leaf #1 $=-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}=\frac{1.58}{3}+\frac{1.16}{3}=0.913$

entropy for leaf #2 $=-1\log_2 1=0$

weighted entropy $=0.913\times\frac{3}{4}=0.68475$ #

B: entropy for leaf #1 $=-\frac{1}{4}\log_2\frac{1}{4}-\frac{3}{4}\log_2\frac{3}{4}=\frac{1}{2}+\frac{3}{2}\times 1.58=0.81$

entropy for leaf #2 $=-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4}=0.81$

weighted entropy $=0.81\times\frac{1}{2}+0.81\times\frac{1}{2}=0.81$ #

misclassification rate:

A $=\frac{200}{800}=\frac{1}{4}$ #

B $=\frac{100+100}{800}=\frac{1}{4}$ #