1. Desicision Tree:

Question A:

(i). Calculate entropy at each level of the decision tree.

we have $S_1 = \{No, Yes, Yes, Yes\}$ with $p_{s_1} = 0.75$.
which means we have an entropy:

$$L(S) = 4 \cdot 0.8113 = 3.2452$$

Depth 1:

①. ~~Sip~~ Based on package type:

Bagged: $S_1 = \{Yes, Yes\}$ with $p_{s_1} = 1$

Canned: $S_2 = \{No, Yes\}$ with $p_{s_2} = 0.5$

which means:

$$L(S) = L(S_1) + L(S_2) = -2 \cdot (1 \times \log_2 1 + 0.5 \log_2 0.5) = 2$$

②. Based on United Price > \$5:

Yes: $S_1 = \{No, Yes\}$ with $p_{s_1} = 0.5$

No: $S_2 = \{Yes, Yes\}$ with $p_{s_2} = 1$

which means:

$$L(S) = L(S_1) + L(S_2) = -2 \cdot (0.5 \log_2 0.5 + 1 \log_2 1) = 2$$

③ Based on Contains > 5 grams of fat:

Yes: $S_1 = \{No, Yes\}$ with $p_{s_1} = 0.5$

No: $S_2 = \{Yes, Yes\}$ with $p_{s_2} = 1$

which means:

$$L(S) = L(S_1) + L(S_2) = -2 (0.5 \log_2 0.5 + 1 \log_2 1) = 2$$

Since we have same entropy based on 3 different column, And we are using Information Gain as our spliting criterion, we can chose any of the 3. For me, I will choose ② United Price, and next step we only need to ~~ch~~ splitte S_1 since S_2 is already pure.

Depth 2:

①. Based on Package Type:

Bagged: $S_1 = \{Yes\}$. with $p_{s_1} = 1$

which we have:

$$L(S) = L(S_1) + L(S_2) = 0$$

②. Based on Contains > 5 grams of fat:

Yes: $S_1 = \{No\}$ with $p_{S_1} = 0$

No: $S_2 = \{Yes\}$ with $p_{S_2} = 1$

which we have:

$$L(S) = L(S_1) + L(S_2) = 0$$

(ii). Calculate the Information gain at each level of the decition tree.

① Splite 1:

$$\text{Information Gain} = L(S_{root}) - (L(S_{1Depth1}) + L(S_{2Depth1}))$$

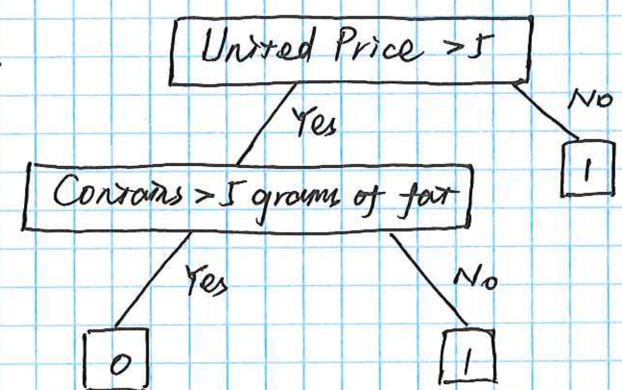
$$= 3.2452 - (1+1) = 1.2452$$

②. Splite 2:

$$\text{Information Gain} = L(S_{Depth1}) - (L(S_{1Depth2}) + L(S_{2Depth2}))$$

$$= 2 - 0 = 2.$$

(iii).



Step 1: $L(S) = 3.2452$

Step 2: $L(S) = 2$

Step 3: $L(S) = 0$

(iv) Much Same as (i)(ii)(iii).

Root:

we have $S_1 = \{No, Yes, Yes, Yes\}$ with $p_{S_1} = 0.75$

wich means we have an impurity:

$$L(S) = 4 \cdot 0.375 = 1.5$$

Depth 1:

we choose "United Price > 5" as the splitting criterion:

Yes: $S_1 = \{No, Yes\}$ with $P_{S_1} = 0.5$

No: $S_2 = \{Yes, Yes\}$ with $P_{S_2} = 1$

which means we have an impurity of:

$$L(S) = L(S_1) + L(S_2) = 2 \times 0.5 + 2 \times 0 = 1$$

Depth 2:

we choose "Contains > 5 grams of fat"

Yes: $S_1 = \{No\}$ with $P_{S_1} = 0$

No: $S_2 = \{Yes\}$ with $P_{S_2} = 1$

which means we have an impurity of:

$$L(S) = L(S_1) + L(S_2) = 1 \times 0 + 0 \times 0 = 0$$

For Information Gain:

① Splitte 1: $Gain = L(S_{root}) - (L(S_{1Depth1}) + L(S_{2Depth2}))$

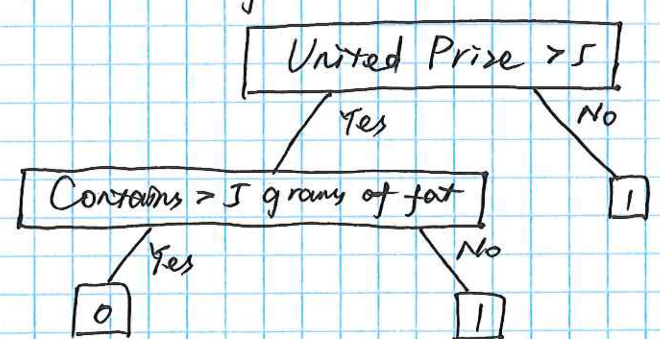
$$= 1.5 - 1.0 = 0.5$$

② Splitte 2:

$$Gain = L(S_{Depth1}) - (L(S_{1Depth2}) + L(S_{2Depth2}))$$

$$= 1.0 - 0.0 = 1.0$$

Tree Drawing:



Step 1: $L(S) = 1.5$

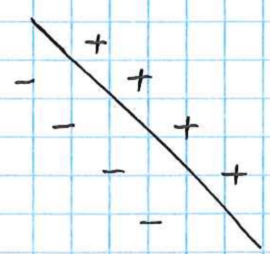
Step 2: $L(S) = 1.0$

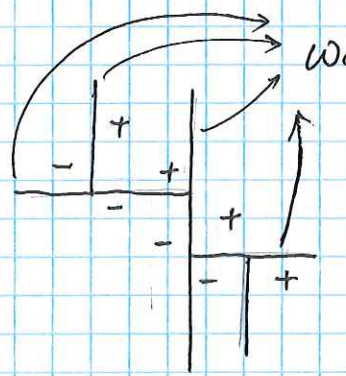
Step 3: $L(S) = 0$

Question B:

Compared to a Linear classifier, the decision tree is not always preferred for classification problems.

For Example:





Simple Linear Model
Can find the Classifier
eg: (SVM find max margin)

Question C:

(i) Using Gini-index as impurity measure. $L(s') = |s'|(1-p_{s'}^2-(1-p_{s'})^2)$

Root: we have $p_{s_1} = 0.5$ which means our impurity:

$$L(s) = 4 \cdot 0.5 = 2.0$$

Depth-1: ① splitting on $x=0$ gives us:

$x>0$: $S_1 = \{Neg, Pos\}$, $p_{s_1} = 0.5$

$x<0$: $S_2 = \{Pos, Neg\}$, $p_{s_2} = 0.5$

which means our impurity:

$$L(s) = L(s_1) + L(s_2) = 2 \cdot 0.5 + 2 \cdot 0.5 = 2.0$$

② splitting on $y=0$ gives us similar results with impurity also equals 2.0

Since our stopping condition is "no split of the root results in any reduction in impurity", then we do not split the root at all, our results turns as follows:

$\boxed{\text{root}}$

with classification error equals 0.5.

(ii) Using classification error as impurity measure:

Root: the classification error is 0.5,

we have our impurity is:

$$L(S) = 4 \times 0.5 = 2.0$$

Depth-1: ① Splitting on $x=0$ gives us:

$x>0$, $S_1 = \{Neg, Pos\}$ classification error 0.5

$x<0$, $S_2 = \{Pos, Neg\}$ classification error 0.5

$$L(S) = L(S_1) + L(S_2) = 2 \times 0.5 + 2 \times 0.5 = 2.0$$

② Splitting on $y=0$ gives us:

$y>0$, $S_1 = \{Pos, Neg\}$ ce = 0.5

$y<0$, $S_2 = \{Neg, Pos\}$ ce = 0.5

$$L(S) = L(S_1) + L(S_2) = 2 \times 0.5 + 2 \times 0.5 = 2.0$$

Same as (i), splitting gives ~~us~~ us no reduction in classification error and thus reachs our stopping condition, our resulting tree:

$\boxed{\text{root}}$

with classification error equals 0.5.

(iii) In order to achieve zero classification error, we need 99 thresholds in the worst case. That means, in this case, we must split each node in this tree in to its own leaf, and therefore we have 100 ~~leaf~~ leaves, which means 99 internal nodes (thresholds)

Question D:

The worst case complexity of the splitting number is

$$O(DN)$$

This is because given N data points we ~~a~~ can split them in N-1 possible positions and in each position, we can use one of the D features / attributes.

2. Overfitting Decision Trees:

Question A:

see "hw3_p2.html" for detail;

Question B:

Question C:

①. "Plot of Error vs. min_samples_leaf"

In this plot, early stopping can be represented as we move right along the x-axis, because when min_samples_leaf becomes larger, that means we have to use less ~~splitts~~ splits in our algorithm and thus stop earlier.

When min_samples_leaf in 0~5, which means we do not stop earlier, we see our train_error low but test_error high. This is overfitting; When min_samples_leaf in 10~15 we can see our train_error rises but test_error falls which indicates that early stopping helps preventing overfitting and improves generalization. However, when min_samples_leaf becomes too large, we can see that both test_error and train_error rise, which in this case, we find underfitting happens.

②. "Plot of Error vs. max_depths"

In this plot, early ~~stop~~ stopping can be represented as we move left along the x_axis, because a smaller max_depth means that our decision tree will have less levels and thus need not too much split and stop earlier.

When max_depth > 8, which means we do not stop early, we can observe that overfitting occurs, where train_error declines to nearly 0 and test error stays high. When there is early stopping (max_depth: 6~8) We can see train_error rises and test_error falls. We can conclude that early stopping helps preventing overfitting and improves generalization. But when we stop too early (max_depth < 4) both train_error and test_error are high which indicates underfitting happens.

3. The AdaBoost Algorithm.

Question A:

In this problem, we want to show:

$$E = \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f(x_i)) \geq \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(H(x_i) \neq y_i)$$

it suffices to show that for each pair of x_i, y_i,

$$\exp(-y_i f(x_i)) \geq \mathbb{1}(H(x_i) \neq y_i)$$

there are two situations:

(1) if y_i and $f(x_i)$ disagree in sign, then,

$$\mathbb{1}(H(x_i) \neq y_i) = 1$$

$$\exp(-y_i f(x_i)) \geq 1$$

therefore,

$$\exp(-y_i f(x_i)) \geq \mathbb{1}(H(x_i) \neq y_i)$$

(2) if y_i and $f(x_i)$ agree in sign, then,

$$\mathbb{1}(H(x_i) \neq y_i) = 0$$

$$\exp(-y_i f(x_i)) > 0$$

therefore,

$$\exp(-y_i f(x_i)) \geq \mathbb{1}(H(x_i) \neq y_i)$$

So overall, we have:

$$\exp(-y_i f(x_i)) \geq \mathbb{1}(H(x_i) \neq y_i)$$

Therefore,

$$\frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f(x_i)) \geq \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(H(x_i) \neq y_i)$$

Question B:

From the lecture note:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

So we can Write $D_t(i)$ as follows:

$$D_t(i) = \left(\prod_{j=1}^{t-1} \frac{D_j(i)\exp(-\alpha_j y_i h_j(x_i))}{Z_j}\right) \cdot D_1(t)$$

Then we can write:

$$Z_T = \sum_{i=1}^{m} \left(\prod_{j=1}^{T-1} \frac{\exp(-\alpha_j y_i h_j(x_i))}{Z_j}\right) D_1(i)\exp(-\alpha_T y_i h_T(x_i))$$

$$= \sum_{i=1}^{m} \left(\prod_{j=1}^{T-1} \frac{1}{Z_j}\right) \cdot \left(\prod_{j=1}^{T-1} \exp(-\alpha_j y_i h_j(x_i))\right) D_1(i)\exp(-\alpha_T y_i h_T(x_i))$$

$$= \sum_{i=1}^{m} \left(\prod_{j=1}^{T-1} \frac{1}{Z_j}\right)\left(\prod_{j=1}^{T} \exp(-\alpha_j y_i h_j(x_i))\right) D_1(i)$$

Therefore:

$$\left(\prod_{j=1}^{T-1} Z_j\right) \cdot Z_T = \sum_{i=1}^{m} \exp\left(-y_i \sum_{j=1}^{T} \alpha_j h_j(x_i)\right) D_1(i)$$

$$\prod_{j=1}^{T} Z_j = \sum_{i=1}^{m} \exp(-y_i f(x_i)) D_1(i)$$

Finally, we can use the fact that we initialize $D_1(i) = \frac{1}{m}$,

And We have

$$\prod_{j=1}^{T} Z_j = \sum_{i=1}^{m} \frac{1}{m} \exp(-y_i f(x_i))$$

$$= E$$

Question C:

We know that ε_t is the training set error of weak classifier h_t for weighted dataset:

$$\varepsilon_t = \sum_{i=1}^{m} D_t(i)\, \mathbb{1}(h_t(x_i) \neq y_i)$$

we consider a class of weak classifier that:

return +1 if h_t classifies x as positive,

return −1 if h_t classifies x as negative,

this class of classifiers the nomalizer Z_t can be represented as:

$$Z_t = (1-\varepsilon_t)\exp(-\alpha_t) + \varepsilon_t \exp(\alpha_t)$$

to prove this function:

①. when $h_t(x_i) \neq y_i$, we have $y_i h_t(x_i) = -1$, we have,

$$\varepsilon_t = \sum_{i=1}^{m} D_t(i) = 1$$

then, $$Z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

$$= \sum_{i=1}^{m} D_t(i)\exp(\alpha_t)$$

$$= \exp(\alpha_t)$$

②. when $h_t(x_i) = y_i$, we have $y_i h_t(x_i) = 1$, we have,

$$\varepsilon_t = 0$$

then, $$Z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

$$= \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t)$$

$$= \exp(-\alpha_t)$$

So we want to minimize Z_t with respect to α_t:

$$\frac{\partial Z_t}{\partial \alpha_t} = (1-\varepsilon_t)(-\alpha_t)\exp(-\alpha_t) + \varepsilon_t \alpha_t \exp(\alpha_t) = 0$$

$$\varepsilon_t \alpha_t \exp \alpha_t = \alpha_t (1-\varepsilon_t) \exp(-\alpha_t)$$

$$\exp 2\alpha_t = \frac{1-\varepsilon_t}{\varepsilon_t}$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

$$\Rightarrow \alpha_t^* = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$