Rishith Kyatham
CSE353
Professor : Yifan Sun

1a. I believe in this scenario, the assumption that this is i.i.d. is not reasonable as the features are not independent of each other. Having one feature such as a fever presence can make you more likely to have the other features. Therefore, deciding with i.i.d. is not reasonable in deciding whether a child has COVID or not.

1b. I believe in this scenario, the assumption that this is i.i.d. would not be reasonable even though these events are independent is that the distribution of somebody with a health condition would not be the same as that of a healthy person. Johnny and Debbie have different health conditions and the rest of the children are healthy so this would make this not be identically distributed.

1c. I believe in this scenario, the assumption that this is i.i.d. would not be reasonable because even though the changes of getting COVID are independent, they are not identically distributed as Chloe's child is in a different class than Amber and Brenda's children.

2.

2a) total # = 10+5+4+1 = 20

(red) $P_1 = \frac{10}{20} = 0.5$ (blue) $P_2 = \frac{5}{20} = 0.25$ (yellow) $P_3 = \frac{4}{20} = 0.20$ (black) $P_4 = \frac{1}{20} = 0.05$

$$E = -\sum_{i=1}^{N} P_i \log_2 P_i = -\left((0.5 \log_2 (0.5) + 0.25 \cdot \log_2(0.25) + 0.20 \cdot \log_2(0.20) + 0.05 \cdot \log_2(0.05)\right) =$$

(P_1 P_2 P_3 P_4)

$$1.68048 2 = \boxed{1.6805}$$

b) First lets find out all the probabilities

$P(\text{top}) = \frac{2}{3}$ $P(\text{red sock}) = \frac{1}{2}$ $P(\text{yellow sock}) = \frac{1}{5}$

$P(\text{bottom}) = \frac{1}{3}$ $P(\text{blue sock}) = \frac{1}{4}$ $P(\text{black sock}) = \frac{1}{20}$

Now, $P(\text{top, red sock}) = \frac{2}{3} \cdot 1 = \frac{2}{3}$ $P(\text{top, blue sock}) = \frac{2}{3} \cdot 0 = 0$

$P(\text{top, yellow sock}) = \frac{2}{3} \cdot 0 = 0$ $P(\text{top, black sock}) = \frac{2}{3} \cdot 0 = 0$

$P(\text{bottom, red sock}) = 0$ $P(\text{bottom, blue sock}) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$

$P(\text{bottom, yellow sock}) = \frac{1}{3} \cdot \frac{2}{5} = \frac{2}{15}$ $P(\text{bottom, black sock}) = \frac{1}{3} \cdot \frac{1}{20} = \frac{1}{30}$

$P(\text{red sock} \mid \text{top}) = 1$ $P(\text{blue sock} \mid \text{top}) = 0$

$P(\text{yellow} \mid \text{top}) = 0$ $P(\text{black sock} \mid \text{top}) = 0$

Now, we calculate

$$H(X|Y) = -\left(\frac{2}{3}\log_2(1) + \frac{1}{6}\log_2\left(\frac{1}{2}\right) + \frac{2}{15}\log_2\left(\frac{2}{5}\right) + \frac{1}{30}\log_2\left(\frac{1}{10}\right)\right) =$$

↑ everything not included come out to be 0

$$\boxed{0.45365}$$

c) $I(X;Y) = H(X) - H(X|Y) =$

$$1.6805 - 0.45365 = \boxed{1.22685}$$

3. Pseudocode (Wasn't able to figure out how to perfectly code it but figured out the logic and tried my best displaying it)

```
    maxVar = float("-inf")
    set1 = range(int(len(y)/2))
    set2 = range(int(len(y)/2), len(y))
    best_feat = 0
    splitval = 0.

    for c in range(0, len(x[0])):
        for r in np.unique(x):
            tempset1 = np.where(x[:,c] < r)
            tempset2 = np.where(x[:,c] >= r)

            weight1 = len(tempset1) / len(y)
            weight2 = len(tempset2) / len(y)

            informationGain = entropy(y) - (weight1*entropy(y[tempset1])) -
(weight2*entropy(y[tempset2]))

            if informationGain > maxVar:
                maxVar = max(maxVar, informationGain)
                set1 = tempset1
                set2 = tempset2
                best_feat = c
                splitval = r

    return best_feat, splitval, set1, set2
```

```
...   information gained in first step 0.330352313922730 46
```

One of the iterations while I was doing it gave me this, seems close.

On

```
        def visit_node(self, x):
        if self.is_leaf:
            return self.label
        """ Fill me in """
        if x[self.splitfeat] < splitval:
            return self.children[0].visit_node(x)
```

```
        else:
            return self.children[1].visit_node(x)
```

```
    def construct_node(self, sample_idx):
        node = Node(sample_idx, self.maxid + 1,  True)

        node.label = ss.mode(self.y[sample_idx])[0][0] # fill me in
```

At the end, I put down

```
tree.root.children = [left_child, right_child]
tree.leaves.extend(tree.root.children)
tree.print_tree()

for i in range(0, 24):
    treecurrLeaf = tree.leaves.pop(0)
    tempy_train = y_train[treecurrLeaf.sample_idx]
    tempx_train = X_train[treecurrLeaf.sample_idx,:]

    if purity(tempy_train) != 1:
        best_feat, splitval, set1, set2 = find_best_split(tempx_train,
tempy_train)
        left_child = tree.construct_node(set1)
        right_child = tree.construct_node(set2)
        treecurrLeaf.is_leaf = False
        treecurrLeaf.children.add_split_details(splitfeat = best_feat, splitval =
splitval)

        treecurrLeaf.children = [left_child, right_child]
        tree.leaves.extend(treecurrLeaf.children)
tree.print_tree()
print('twenty five train err:', tree.report_train_err())
print('twenty five test err:', get_test_err(tree))
```

I think this tree would overfit because the training accuracy would seem to be high and test accuracy is low.

Could not get (Report your train and test misclassification rate for 25 steps of training. ˆ (1 pt) Sketch out the resulting tree. (I recommend using a big piece of paper or a whiteboard + camera.)) parts properly due to my code not working correctly but the logic is right.

4. My Results :

```
[93]  ✓ 13.5s

...   prob. of "alice" 0.014548615047424706
      prob. of "queen" 0.002569625514869818
      prob. of "chapter" 0.0009069266523069947
```

```
prob. of "the alice" 0.0
prob. of "the queen" 0.039706780696395 85
prob. of "the chapter" 0.0
prob. of "the hatter" 0.031154551007941355
```

```
['alice', 'abide', 'voice', 'above', 'alive', 'twice', 'dunce', 'prize', 'smile', 'since']
```

Reporting Final Result :

```
Output exceeds the size limit. Open the full output data in a text editor
0.749 0.928
deep dwep deep
she hse she
this shit this
alice aleci alice
the eht the
theyll tleylh theyll
got xot got
pop pow pop
ran rwn ran
and azd and
for fou for
was aws was
do od of
such sucs such
she seh she
she seh she
cats cqts cats
this shit this
was wsa was
or ro to
such sucs such
itself etsilf itself
this tihs this
to ot it
```