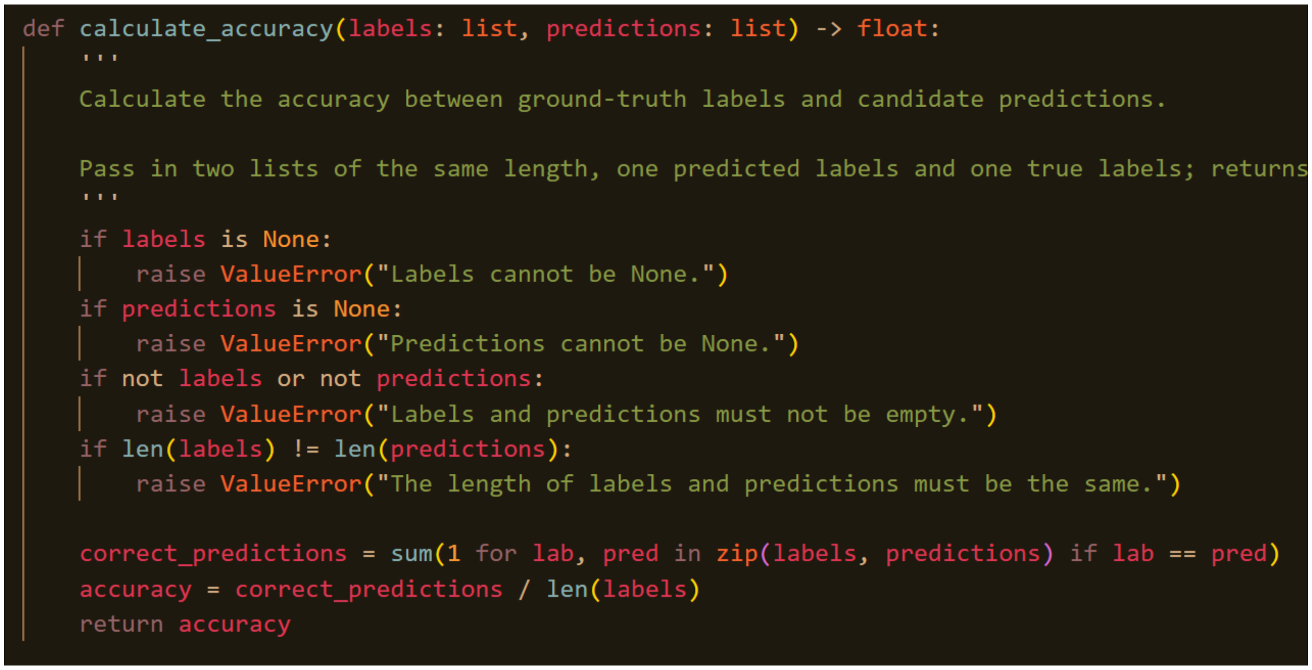
CS 6350/DS 4350: HW2-Q2 Decision Trees on the Nursery Dataset

Samir Abdelrahman

Yujin Song, Shubham Sanjay Sawant Fall 2025

Jackson Switzer - u1608753

4.1 Accuracy [6 points]

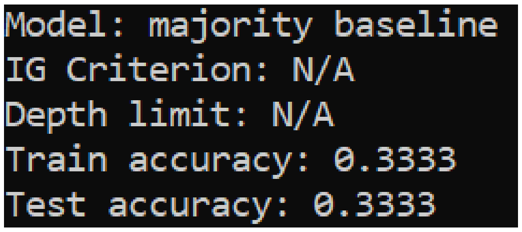


```
def calculate_accuracy(labels: list, predictions: list) -> float:
    '''
    Calculate the accuracy between ground-truth labels and candidate predictions.

    Pass in two lists of the same length, one predicted labels and one true labels; returns
    '''
    if labels is None:
        raise ValueError("Labels cannot be None.")
    if predictions is None:
        raise ValueError("Predictions cannot be None.")
    if not labels or not predictions:
        raise ValueError("Labels and predictions must not be empty.")
    if len(labels) != len(predictions):
        raise ValueError("The length of labels and predictions must be the same.")

    correct_predictions = sum(1 for lab, pred in zip(labels, predictions) if lab == pred)
    accuracy = correct_predictions / len(labels)
    return accuracy
```

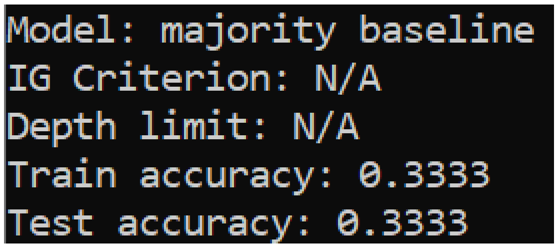
4.2 Majority Baseline Accuracy [8 points]



```
Model: majority baseline
IG Criterion: N/A
Depth limit: N/A
Train accuracy: 0.3333
Test accuracy: 0.3333
```

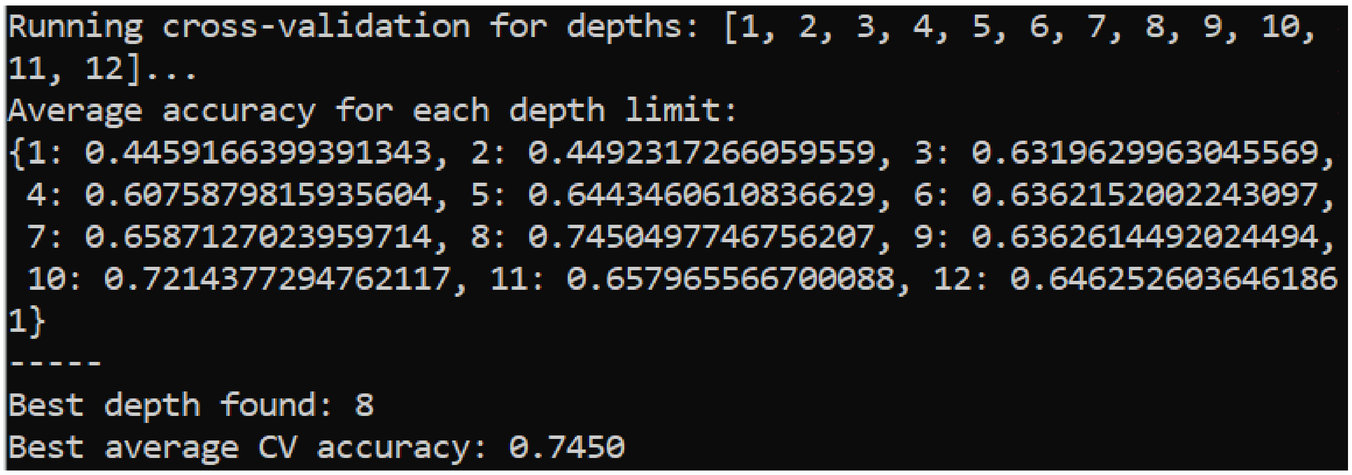
Accuracy might be a bad metric for measuring the quality of a model on this dataset because if the accuracy is very imbalanced, then the model can make good predictions by only predicting the most common label, without actually learning anything about the data. It would be better to use multiple metrics so we can see how well the model performs at predicting all classes correctly, such as precision and recall.

4.3 Simple Decision Tree [14 points]



```
Model: majority baseline
IG Criterion: N/A
Depth limit: N/A
Train accuracy: 0.3333
Test accuracy: 0.3333
```

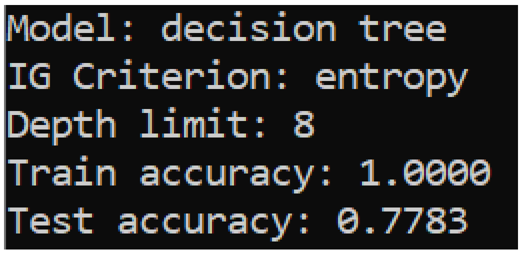
4.4 Decision Tree with Cross-Validation [20 points]



```
Running cross-validation for depths: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12]...
Average accuracy for each depth limit:
{1: 0.4459166399391343, 2: 0.4492317266059559, 3: 0.6319629963045569,
 4: 0.6075879815935604, 5: 0.6443460610836629, 6: 0.6362152002243097,
 7: 0.6587127023959714, 8: 0.7450497746756207, 9: 0.6362614492024494,
 10: 0.7214377294762117, 11: 0.657965566700088, 12: 0.646252603646186
1}
-----
Best depth found: 8
Best average CV accuracy: 0.7450
```

The best max depth I found was 8, with an accuracy of 74.50%
It makes sense that performance would plateau around a max depth of eight because there are eight features in the data, and all of them are categorical. That means splitting each value of a feature into its own sub-tree is probably going to be the best choice. You *could* split a feature into fewer branches than the number of values and then split on the same feature again later, but ultimately you'll still just end up with every possible combination of features from the training set at a leaf node (if the max depth is large enough). Any variation for max depths greater than eight must be because of the shuffling of the dataset before training or other non-determinism in tree-building algorithm.

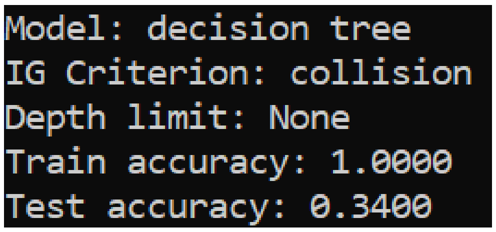
4.5 Decision Tree with Best Depth from CV [12 points]



```
Model: decision tree
IG Criterion: entropy
Depth limit: 8
Train accuracy: 1.0000
Test accuracy: 0.7783
```

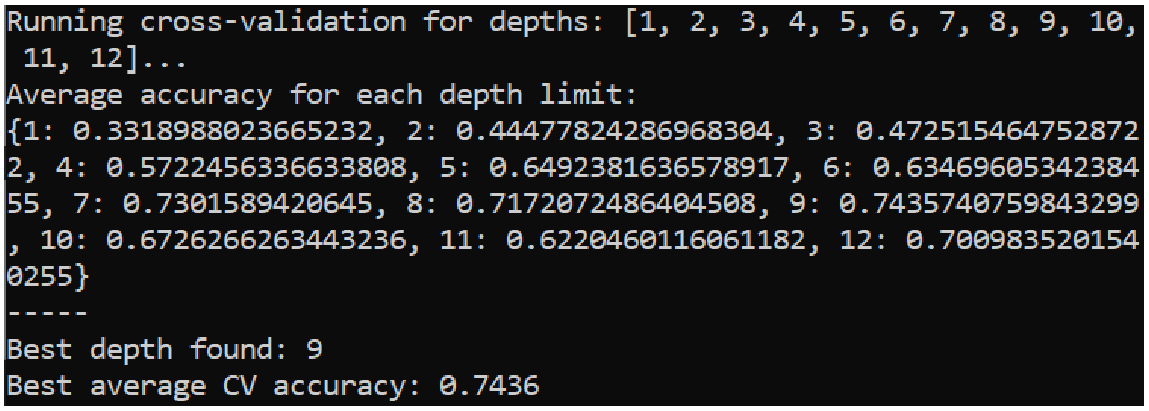
5 Bonus: Decision Trees with Collision Entropy [10 points]

5.1 Simple Decision Tree with Collision Entropy [3 points]



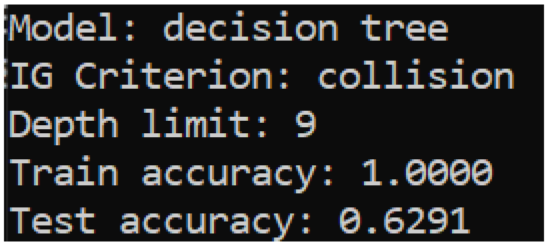
```
Model: decision tree
IG Criterion: collision
Depth limit: None
Train accuracy: 1.0000
Test accuracy: 0.3400
```

5.2 Decision Tree with Cross-Validation [3 points]



```
Running cross-validation for depths: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
 11, 12]...
Average accuracy for each depth limit:
{1: 0.3318988023665232, 2: 0.44477824286968304, 3: 0.472515464752872
2, 4: 0.5722456336633808, 5: 0.6492381636578917, 6: 0.63469605342384
55, 7: 0.7301589420645, 8: 0.7172072486404508, 9: 0.7435740759843299
, 10: 0.6726266263443236, 11: 0.6220460116061182, 12: 0.700983520154
0255}
-----
Best depth found: 9
Best average CV accuracy: 0.7436
```

5.3 Decision Tree with Best Depth from CV [4 points]



```
Model: decision tree
IG Criterion: collision
Depth limit: 9
Train accuracy: 1.0000
Test accuracy: 0.6291
```