Lab 1: Decision Trees

Machine Learning DD2421

Riccardo Fragale, Matteo Camellini

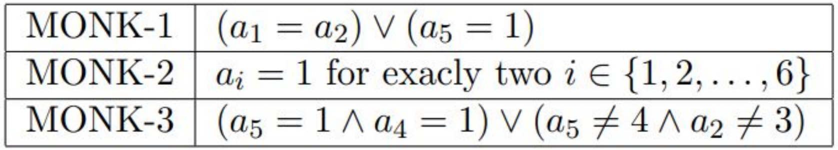
February 2026

Assignment 0

The primary challenge across all three datasets is the sparsity of training data relative to the total hypothesis space. For example, MONK-1 provides only 124 training examples out of 432 possible attribute combinations. Without a dedicated validation set or the ability to perform k-fold cross-validation effectively, measuring the model's true generalization capability during training is difficult. In particular:

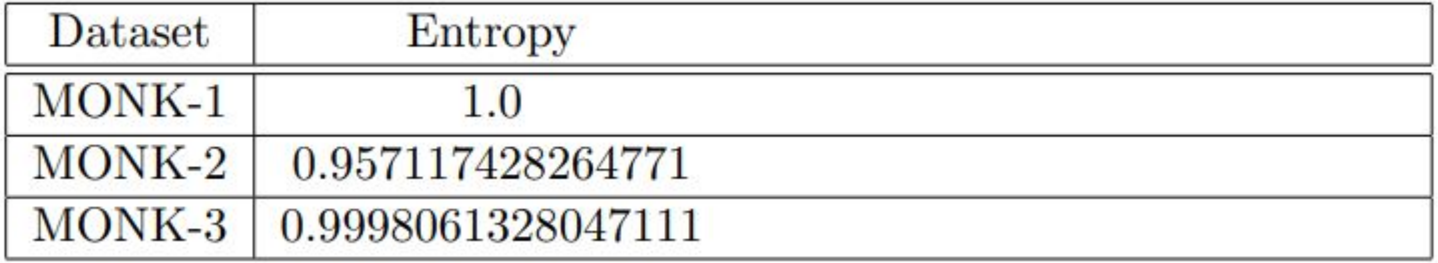
- **MONK-2 is the most difficult:** the underlying concept (exactly two attributes equal 1) involves a complex interaction between all six attributes.
- **MONK-3 is moderately difficult:** while the underlying concept is simpler than MONK-2, this dataset introduces 5% label noise in the training set. A standard decision tree (like ID3) attempts to classify every training point correctly, meaning it will likely memorize this noise. It also has the smallest training set (122 samples), further exacerbating the risk of learning false patterns.
- **MONK-1 is the easiest:** the concept is a standard logical form (Disjunctive Normal Form) relying only on attributes a1, a2, and a5. Decision trees are naturally suited to learning these types of explicit rules. As long as the training set contains enough examples to cover these specific attribute values, the tree should learn the concept efficiently.

Table 1: True concepts behind the MONK datasets



MONK-1	$(a_1 = a_2) \lor (a_5 = 1)$
MONK-2	$a_i = 1$ for exacly two $i \in \{1, 2, \ldots, 6\}$
MONK-3	$(a_5 = 1 \land a_4 = 1) \lor (a_5 \neq 4 \land a_2 \neq 3)$

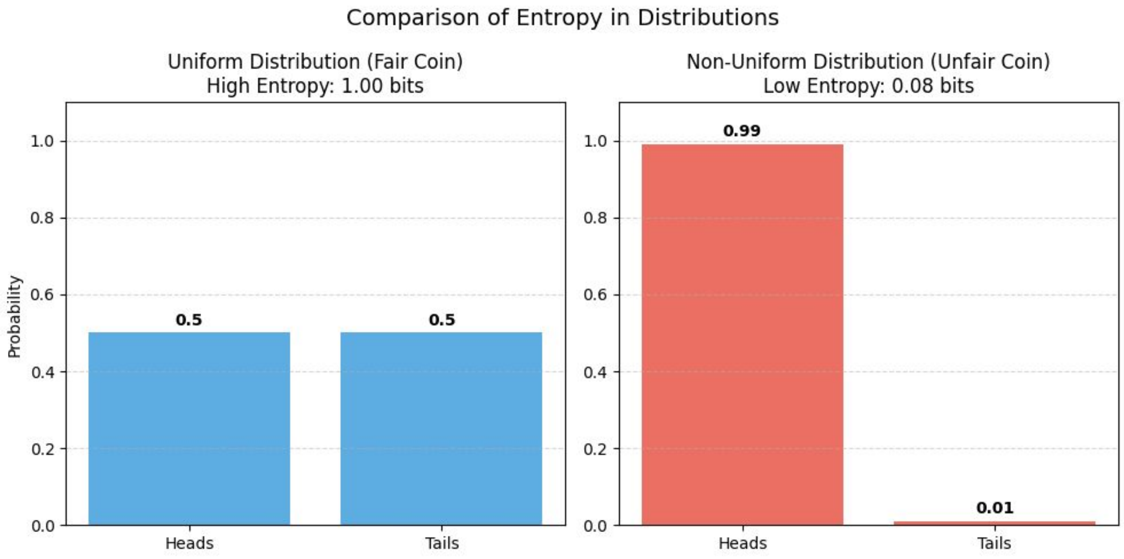
Assignment 1



Dataset	Entropy
MONK-1	1.0
MONK-2	0.957117428264771
MONK-3	0.9998061328047111

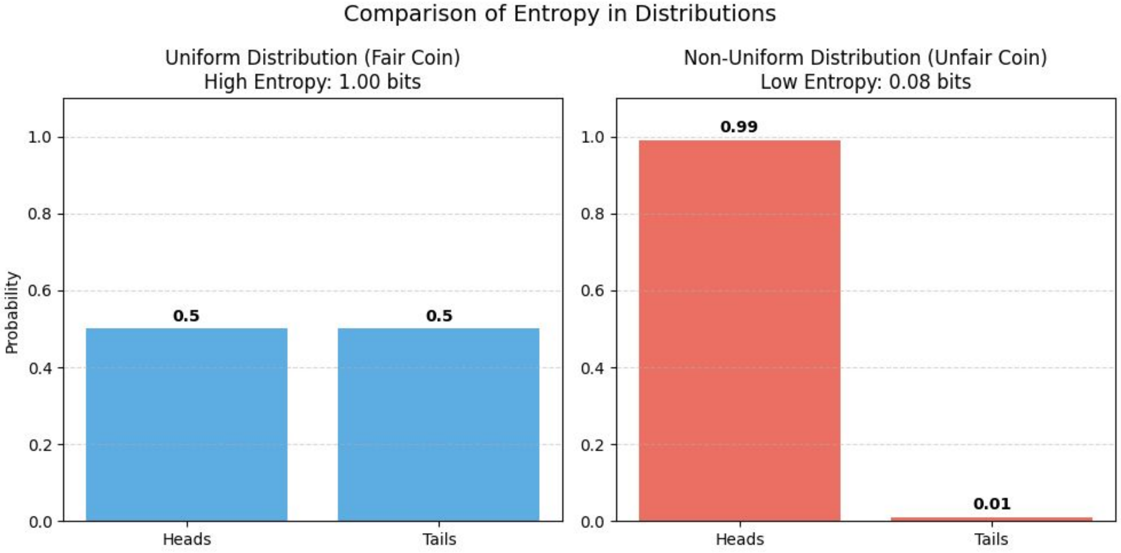
The entropy values for all three training datasets are extremely high (close to or exactly 1.0). This indicates that the classes are balanced in the datasets. Consequently, the initial uncertainty is maximized; without looking at the attributes, the algorithm has no way to predict the outcome better than a random coin flip.

Assignment 2



When we talk about a **uniform distribution**, we are describing a situation where every possible outcome has the same probability to appear in a single sample. This implies bigger unpredictability. A good example of it is the case of a fair coin flip where each of the 2 outcomes has 1/2 of probability. This is the maximum level of entropy possible for a set of outcomes as no outcome is more likely than any other.

Assignment 2 (cont.)

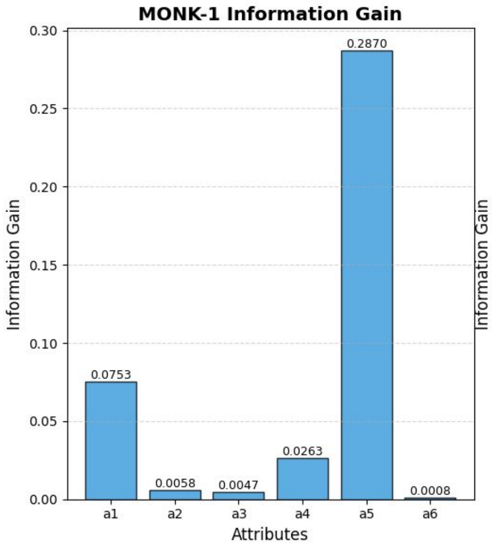


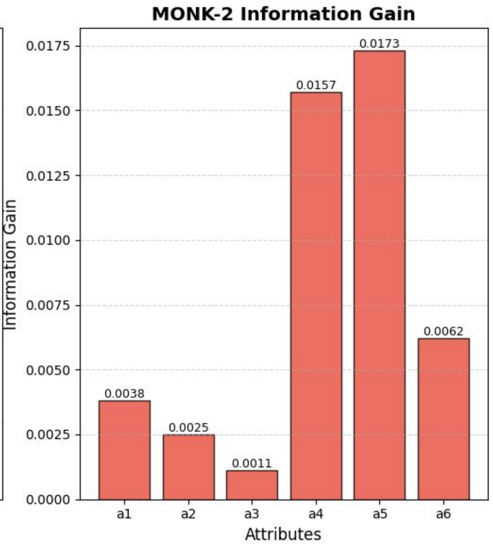
Then we have **non-uniform distributions** where some outcomes are more likely than others. In this case we have lower unpredictability and less surprise when certain outcomes appear. As a consequence the entropy is lower than a uniform distribution case. If the distribution is highly unbalanced towards a certain outcome the entropy might be very low. An example could be the case of a very unfair coin where the outcome HEAD appears 99% of the times.

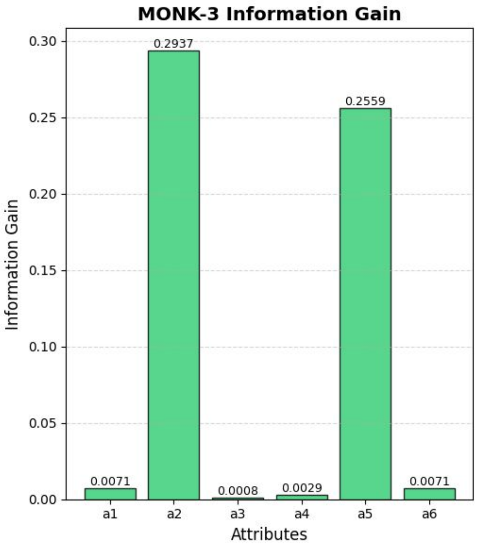
Assignment 3

The attribute that carries the highest information gain is the one to be selected for the splitting of the dataset in the most effective way.

- For the MONK1 dataset, attribute a5 carries an information gain of 0.2870 which is a lot more than any other attribute so it's the one to be chosen.

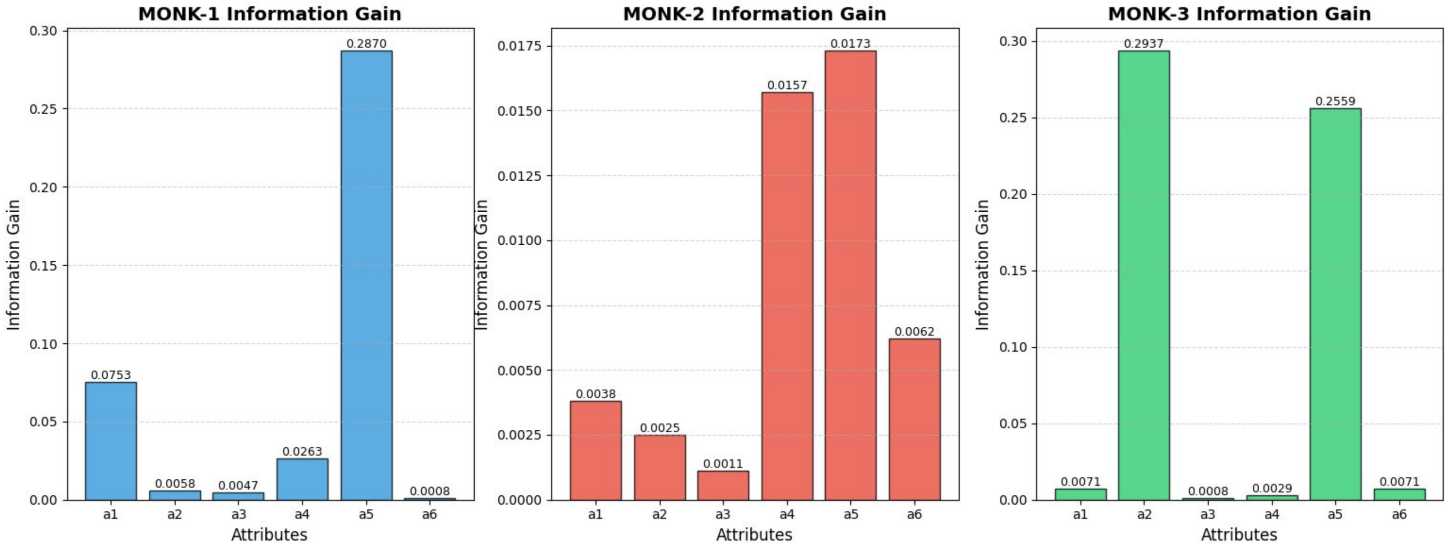






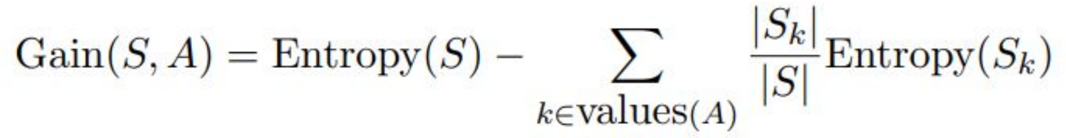
Assignment 3 (cont.)

- A similar approach leads us to select attribute a5 again for the MONK2 dataset even though here the difference is far less than in the previous case. As we individuated correctly in Assignment 0 this dataset is the harder to classify and we have a further proof here as no single attribute has a very big information gain when known.
- Finally, for the MONK3 dataset the one that has to be chosen to split the dataset is attribute a2 with an information gain of 0.2937. In this case we have a big gain but there is another relevant attribute which is again a5 but the gain is lower than a2 by 0.04.



Assignment 4

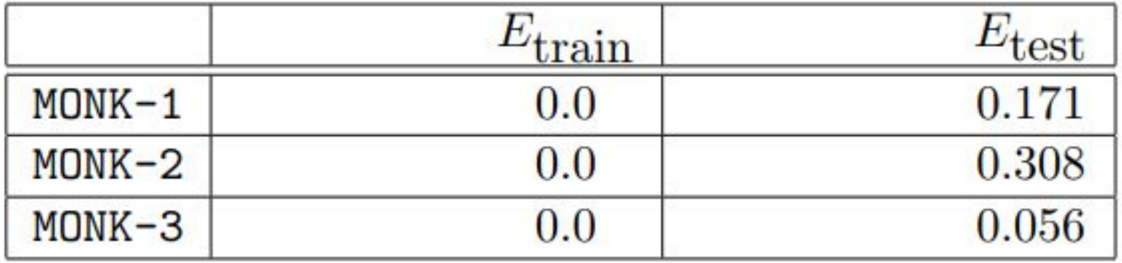
- Maximizing information gain is equivalent to minimizing the weighted entropy of the resulting subsets. We aim for subsets that are as "pure" as possible (Entropy → 0), meaning they contain examples of only one class.
- We minimize the weighted entropy of the resulting subsets since entropy measures uncertainty. By choosing the split that maximizes gain, we are choosing the attribute that most effectively reduces randomness, separating the data into predictable groups.
- This is a greedy approach and thus can be regarded as being a heuristic. The algorithm optimizes the decision locally at each node to get the best immediate result. While this usually produces a short, effective tree, it does not guarantee the absolute shortest tree possible (the global optimum), but serves as a highly efficient approximation.



$$\mathrm{Gain}(S, A) = \mathrm{Entropy}(S) - \sum_{k \in \mathrm{values}(A)} \frac{|S_k|}{|S|} \mathrm{Entropy}(S_k)$$

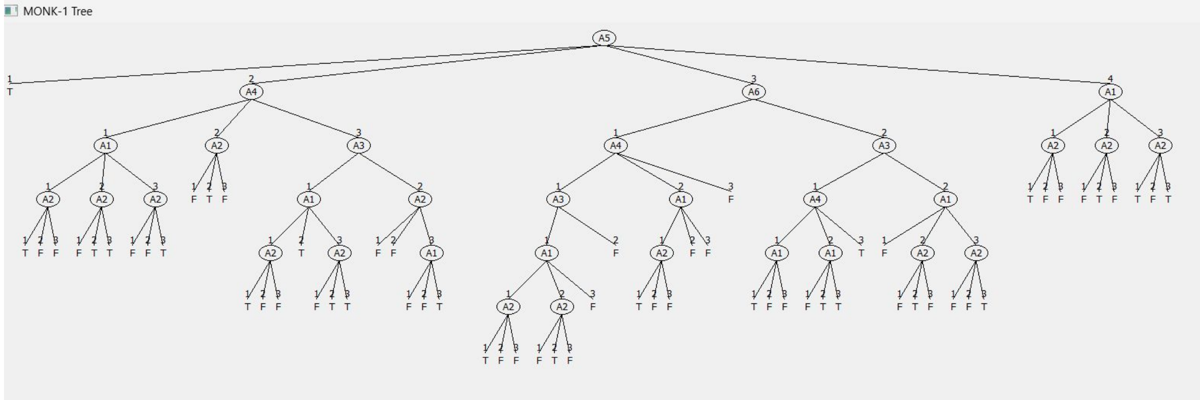
Assignment 5

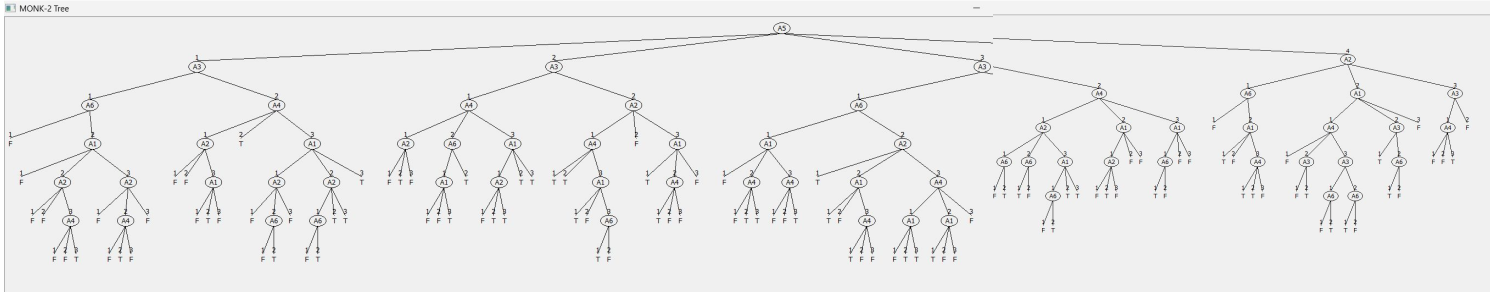
- The training error is 0.0 for all. This is because the algorithm splits nodes until every leaf is pure (single class), perfectly "memorizing" the training data. This leads to overfitting, that is, the tree captures specific noise or quirks rather than the general rule.
- **MONK-1**: moderate performance. The tree captured the main concept but overfitted, missing some generalizations.
- **MONK-2:** the greedy algorithm struggles with complex interactions (like exactly two true) and the low information gain across all attributes (from Assignment 3). It "brute forced" the solution, creating a massive tree that generalizes poorly.
- **MONK-3**: surprisingly good performance despite containing noise, likely due to the fact that the underlying rule is simple. The tree learned the concept well but also learned the noise. Pruning would reduce the error further.



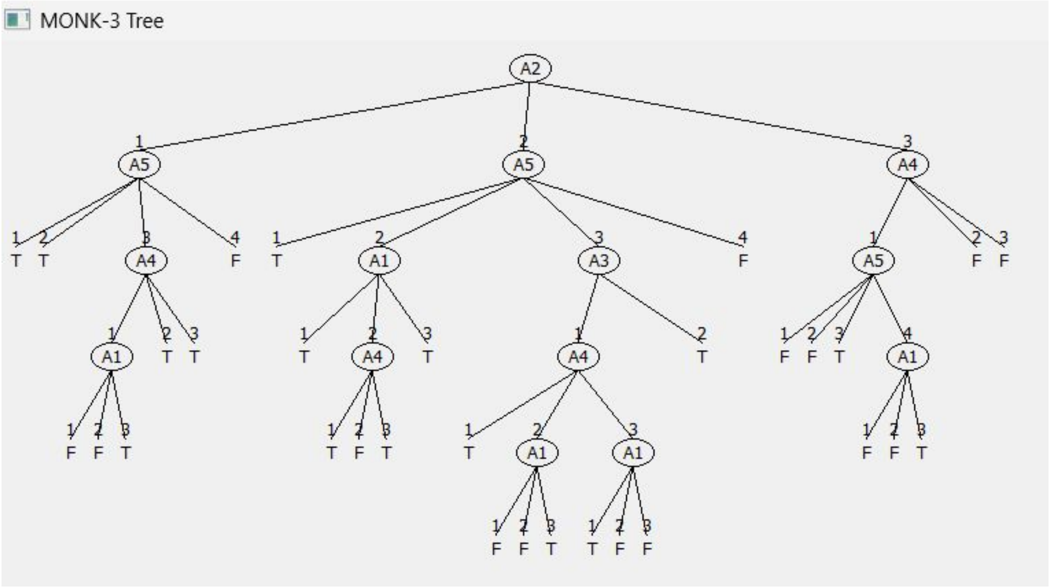
	E_{train}	E_{test}
MONK-1	0.0	0.171
MONK-2	0.0	0.308
MONK-3	0.0	0.056

Assignment 5 (cont.)





Assignment 5 (cont.)



Assignment 6

The idea behind the decision tree model is that the algorithm grows without any constraint trying to split the data until each leaf of the tree is clearly and undoubtedly classifiable. This process generally leads to a very low bias (since the model is able to distinguish with precision almost every case) but at the same time a pretty high variance; since it is very precise, changing just a few samples of the training dataset implies relevant changes in the entire structure of the tree.

Moreover, we might have overspecialization on the training dataset and so lower ability to generalize and to be really effective while predicting on new samples.

Since we want to improve our models towards better specialization we decide to remove branches that provide little predicting value (pruning a set of "irrelevant" nodes) accepting an increase in terms of bias as the model is a bit too simple to capture every single detail but in exchange for a significant reduction of the variance (slight changes in the training dataset won't impact as much as before).

Assignment 7

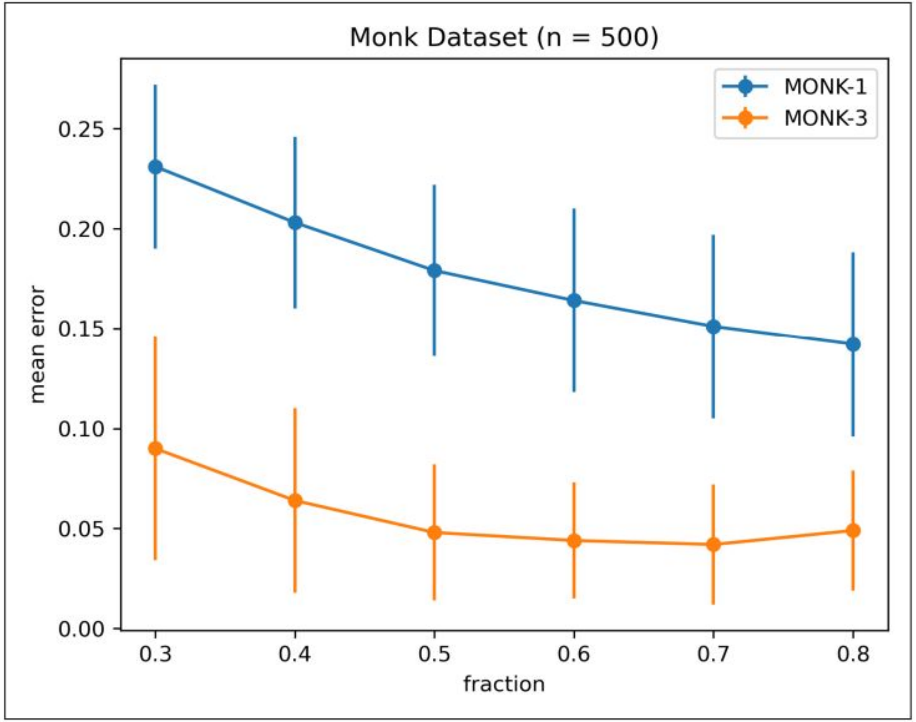


Figure 1: Test classification error as a function of the training/pruning fraction (MONK datasets, n = 500).

- For both datasets we observe that as the training fraction increases (0.3 → 0.8), error decreases for both datasets. This is because more training data produces a better base tree that learns the underlying concept more accurately before pruning begins.
- **MONK-1**: exhibits a significant, consistent error drop (~0.23 → ~0.14). The limiting factor is the data quantity. At low fractions (0.3), the tree misses parts of the concept. At high fractions (0.8), it learns the rule well, and the small validation set is sufficient to prune overfitted branches.

The pruning operation yields good results, as the error drops below 0.15 (compared to 0.171 for the full unpruned tree), proving that it successfully improves generalization.

Assignment 7 (cont.)

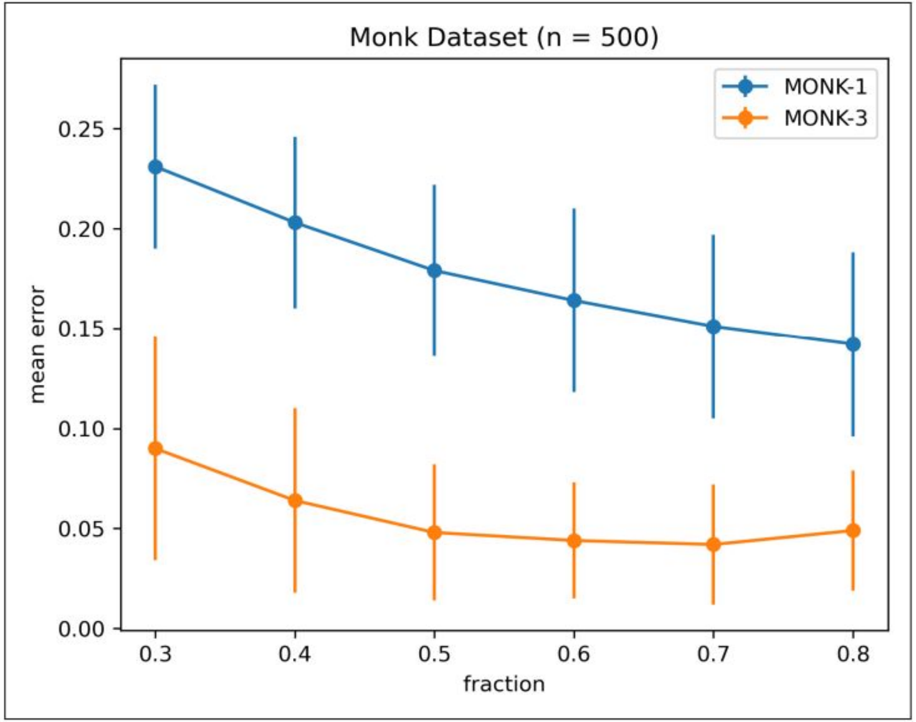


Figure 1: Test classification error as a function of the training/pruning fraction (MONK datasets, n = 500).

MONK-3: presents a flatter curve with a "sweet spot" at fraction 0.6–0.7 (error → 0.04). This dataset requires a balance between Training (to learn patterns) and Validation (to filter out the 5% noise).
Moreover, at 0.8 the performance plateaus/worsens because the validation set is too small to reliably distinguish noise from real rules.
Finally, the pruned error (0.04) beats the full tree error (0.056), confirming noise removal.

High Variance: tall error bars indicate performance is highly sensitive to the specific random split of data.

Thank you for listening