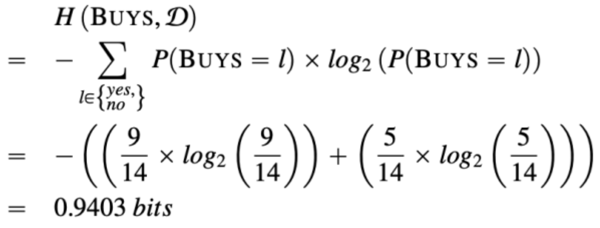
1. There are a range of factors that can contribute to this happening, but two important ones that we discuss in the chapter is the model overfitting to noise in the data, and sampling bias. It is possible for a model to be very accurate on one sample of data from a domain and not be accurate on another sample of data taken from the same domain. This is because the model may overfit to the sample of data it is trained on. What this means is that the model has memorized the training data so closely that it has modeled the noise in the sample of data it was trained on, and so it will likely make incorrect predictions on examples that are similar to the noisy examples in the dataset. Another problem that can lead to poor generalization is if the data used for training and testing the model suffers from sample bias. In this case, even if the model does not overfit the data, the model will not generalize well because it is trained on data that is not representative of the true distributions in the general population.

2.

The histogram for the different levels of the TACHYCARDIA feature are slightly different suggesting that there is a relationship between the DIA. B.P. and TACHYCARDIA features. It looks like patients with higher diastolic blood pressure values are more likely to suffer with tachycardia than those with lower blood pressure values. The histograms for the HEIGHT feature split according to the different target levels look very similar. This suggests that there is no strong relationship between the HEIGHT feature and the target feature, TACHYCARDIA.

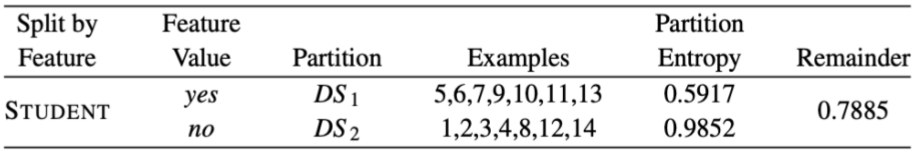
3.

To answer this question we need to calculate the information gain of the STUDENT feature at the root node and show that it is less than the information gain for AGE. To calculate information gain for the STUDENT feature we first calculate the entropy of the entire dataset D with respect to the target feature, BUYS



$$
\begin{aligned}
& H(\textsc{Buys}, \mathcal{D}) \\
= & -\sum_{l \in \left\{ \substack{yes, \\ no} \right\}} P(\textsc{Buys} = l) \times log_2\left(P(\textsc{Buys} = l)\right) \\
= & -\left(\left(\frac{9}{14} \times log_2\left(\frac{9}{14}\right)\right) + \left(\frac{5}{14} \times log_2\left(\frac{5}{14}\right)\right)\right) \\
= & \ 0.9403\ bits
\end{aligned}
$$

Next we calculate the remainder after the dataset is split based on the values of the STUDENT feature.



Split by Feature	Feature Value	Partition	Examples	Partition Entropy	Remainder
STUDENT	*yes*	DS_1	5,6,7,9,10,11,13	0.5917	0.7885
	no	DS_2	1,2,3,4,8,12,14	0.9852	

Finally, we calculate the information gain as the difference between the original entropy and the remainder:

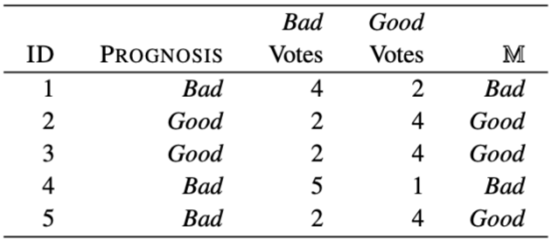
IG = 0.9403 - 0.7885 = 0.1518 bits

The information gain for STUDENT is 0.1518 which is less then the 0.247 for AGE, so STUDENT is not a better feature to use at the root node.

No. We can know this without performing the calculation, because each instance has a unique value for the ID feature. ID would not be a good feature at the root node of the tree (or in fact anywhere in the tree) because it actually contains no information, and the resulting decision tree would be massively overfitted to the training data. Information measures such as the gain ratio are designed to address this limitation in information gain.

4.

Ensembles based on bagging use simple majority voting as their aggregation mechanism. So for each instance we simply count the number of positive and negative votes to determine the output of the overall ensemble. The following table shows the vote counts for each instance in the test dataset and the target feature level that receives the most votes.



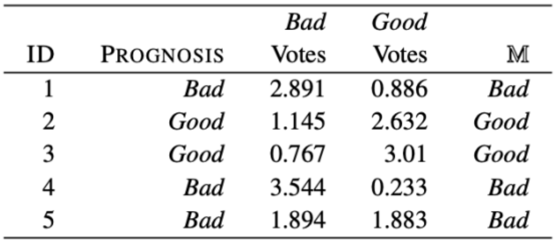
ID	PROGNOSIS	*Bad* Votes	*Good* Votes	$\mathbb{M}$
1	*Bad*	4	2	*Bad*
2	*Good*	2	4	*Good*
3	*Good*	2	4	*Good*
4	*Bad*	5	1	*Bad*
5	*Bad*	2	4	*Good*

The ensemble model made the correct prediction for 4 out of 5 instances in the test dataset and only made an incorrect prediction for one. Therefore the misclassification rate is 1/5 = 20%.

In the boosting case the votes are weighted by the confidence factor associated with each model. So we calculate a weighted vote for each target feature level, in this case Bad and Good, by adding together the confidence factors for the models that predict each target feature level.

For example, for the first instance in the test dataset, d1, the models that predict the Bad target level are M0, M1, M3, and M4. So the weighted vote for the Bad target level is 0.114 + 0.982 + 0.912 + 0.883 = 2.891. The models that predict the Good target level are M2 and M5. So the weighted vote for the Good target level is 0.653 + 0.233 = 0.886. The votes for Bad outweigh the votes for Good and so that is the overall model prediction in this case.

The votes for other test instances are calculated in the same way. The table below shows these weighted votes.



ID	Prognosis	*Bad* Votes	*Good* Votes	𝕄
1	*Bad*	2.891	0.886	*Bad*
2	*Good*	1.145	2.632	*Good*
3	*Good*	0.767	3.01	*Good*
4	*Bad*	3.544	0.233	*Bad*
5	*Bad*	1.894	1.883	*Bad*

d. The ensemble model made the correct prediction for all instances in the test dataset and so the misclassification rate is 0%.

5. a.- **Bagging** (Bootstrap Aggregating), where each tree is trained on a random sample of the dataset with replacement, and
   -**Subspace Sampling**, which involves considering only a randomly selected subset of features during tree induction.

   b. Random Forests reduce variance through **ensemble averaging/voting** and feature randomness. By training multiple decision trees on different subsets of data and features, they **diversify predictions**, mitigating the impact of outliers and reducing overfitting. Additionally, combining predictions from multiple trees through averaging further stabilizes the model's performance, leading to lower variance.

6. XGBoost: Employs gradient boosting with a more regularized model, mitigating overfitting. By leveraging gradient descent, it minimizes loss with each tree addition, ensuring robustness while enhancing predictive accuracy.
   Adaboost: Employes boosting method that sequentially corrects errors of the previous tree by giving more weight to misclassified instances.

7. a.The Plot exhibits overfitting of the model. As the tree grows deeper or depth increases, the training accuracy increases rapidly but the testing accuracy is decreasing which is a clear case of **overfitting**.
   b.The measure that can be taken here is **tree pruning**. Stop the growth of the tree beyond a depth of 5 to prevent overfitting.