CS577 - Machine Learning – Assignment 4

Alexandros Angelakis csdp1363, angelakis@csd.uoc.gr

Exercise 1 - Entropy and Decision Trees (Theoretical)

Our goal is to build the decision tree to classify as virus or not, based on the given train set in Table 1. In order to do that, we need to follow these steps:

1. Calculate the priors
2. Calculate the conditional probabilities
3. Calculate the entropy $H(\text{Virus})$
4. Calculate the conditional entropies
5. Calculate and select the feature with the highest Information Gain to split on.
6. Repeat from 1. until no further splits, with the new training set, after splitting.

	Body Length	**Bold letters**	**susp. adress**	**Virus**
s1	23	0	1	0
s2	18	1	0	0
s3	43	0	1	0
s4	68	0	0	1

Table 1: Train Set

	Body Length	**Bold letters**	**susp. adress**	**Virus**
s5	20	1	1	1
s6	25	1	0	0
s7	60	0	1	1
s8	35	0	0	0

Table 2: Test Set

1. Calculate the priors

$$P(\text{Virus} = 1) = \frac{1}{4}, \quad P(\text{Virus} = 0) = \frac{3}{4}$$
$$P(\text{Bold} = 1) = \frac{1}{4}, \quad P(\text{Bold} = 0) = \frac{3}{4}$$
$$P(\text{address} = 1) = \frac{1}{2}, \quad P(\text{address} = 0) = \frac{1}{2}$$

For continuous values, there are multiple ways to determine a threshold for discretization. I will use the mean of the feature, as it is the simplest method: mean(Body) = 38.5.

$$P(\text{Body} < 38) = \frac{1}{2}, \quad P(\text{Body} \geq 38) = \frac{1}{2}$$

2. Calculate the conditional probabilities

$$\begin{aligned}
P(\text{Virus=1}|\text{Body} < 38) = 0, \quad P(\text{Virus=0}|\text{Body} < 38) = 1 \\
P(\text{Virus=1}|\text{Body} \geq 38) = \frac{1}{2}, \quad P(\text{Virus=0}|\text{Body} \geq 38) = \frac{1}{2} \\
P(\text{Virus=1}|\text{Bold} = 0) = \frac{1}{3}, \quad P(\text{Virus=0}|\text{Bold} = 0) = \frac{2}{3} \\
P(\text{Virus=1}|\text{Bold} = 1) = 0, \quad P(\text{Virus=0}|\text{Bold} = 1) = 1 \\
P(\text{Virus=1}|\text{address} = 0) = 0, \quad P(\text{Virus=0}|\text{address} = 0) = 1 \\
P(\text{Virus=1}|\text{address} = 1) = \frac{1}{2}, \quad P(\text{Virus=0}|\text{address} = 1) = \frac{1}{2}
\end{aligned}$$

3. Calculate the entropy $H(\textbf{Virus})$

$$H(\text{Virus}) = -P(\text{Virus} = 1)log_2P(\text{Virus} = 1) - P(\text{Virus} = 0)log_2P(\text{Virus} = 0) = -\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4} \approx 0.8113$$

4. Calculate the conditional entropies

From now on, to simplify notation, I will represent "Virus" as "V," "Body" as "Bd," "Bold" as "Bl," and "Address" as "Ad."

$$\begin{aligned}
H(\text{V}|\text{Bd}) &= -P(\text{Bd} < 38)[P(\text{V=1}|\text{Bd} < 38)log_2P(\text{V=1}|\text{Bd} < 38) + P(\text{V}|\text{Bd} < 38)log_2P(\text{V=0}|\text{Bd} < 38)] \\
&\quad - P(\text{Bd} \geq 38)[P(\text{V=1}|\text{Bd} \geq 38)log_2P(\text{V=1}|\text{Bd} \geq 38) + P(\text{V=0}|\text{Bd} < 38)log_2P(\text{V=0}|\text{Bd} \geq 38)] \\
&= -\frac{1}{2}(0log_20 + 1log_21) - \frac{1}{2}(\frac{1}{2}log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2}) = -\frac{1}{2}log_2\frac{1}{2} = 0.5
\end{aligned}$$

$$\begin{aligned}
H(\text{V}|\text{Bl}) &= -P(\text{Bl} = 0)P(\text{V=1}|\text{Bl} = 0)log_2P(\text{V=1}|\text{Bl} = 0) + P(\text{V=0}|\text{Bl} = 0)log_2P(\text{V=0}|\text{Bl} = 0)] \\
&\quad - P(\text{Bl} = 1)[P(\text{V=1}|\text{Bl} = 1)log_2P(\text{V=1}|\text{Bl} = 1) + P(\text{V=0}|\text{Bl} = 1)log_2P(\text{V=0}|\text{Bl} = 1)] \\
&= -\frac{3}{4}(\frac{1}{3}log_2\frac{1}{3} + \frac{2}{3}log_2\frac{2}{3}) - \frac{1}{4}(0log_20 + 1log_21) = -\frac{1}{4}log_2\frac{1}{3} - \frac{1}{2}log_2\frac{2}{3} \approx 0.6887
\end{aligned}$$

$$\begin{aligned}
H(\text{V}|\text{Ad}) &= -P(\text{Ad} = 0)P(\text{V=1}|\text{Ad} = 0)log_2P(\text{V=1}|\text{Ad} = 0) + P(\text{V=0}|\text{Ad} = 0)log_2P(\text{V=0}|\text{Ad} = 0)] \\
&\quad - P(\text{Ad} = 1)[P(\text{V=1}|\text{Ad} = 1)log_2P(\text{V=1}|\text{Ad} = 1) + P(\text{V=0}|\text{Ad} = 1)log_2P(\text{V=0}|\text{Ad} = 1)] \\
&= -\frac{1}{2}(0log_20 + 1log_21) - \frac{1}{2}(\frac{1}{2}log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2}) = -\frac{1}{2}log_2\frac{1}{2} = 0.5
\end{aligned}$$

5. Calculate and select the feature with the highest information gain

We can see that there are two information gains with the same value. We can choose which feature to split on between "Body" and "Address." In this case, I will choose "Body."

$$IG(\text{Bd}) \approx 0.8113 - 0.5 \approx 0.3113$$
$$IG(\text{Bl}) \approx 0.8113 - 0.6887 \approx 0.1226$$
$$IG(\text{Ad}) \approx 0.8113 - 0.5 \approx 0.3113$$

Decision tree after the split in respect to "Body". When "Bd < 38", we know that "Virus = 0", no need to split further.

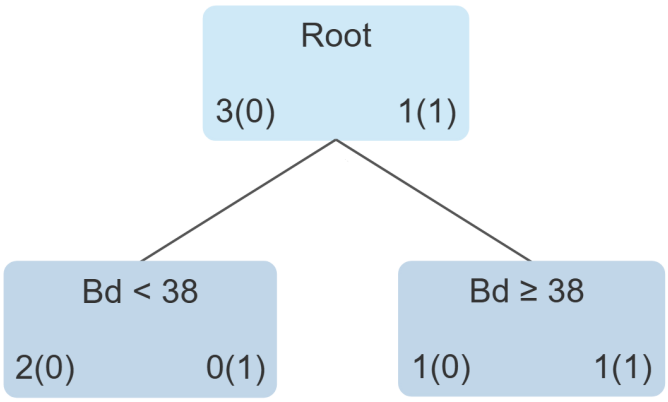

Figure 1: Split in respect to "Body Length"

6. Repeat from 1. with the new training set, after splitting

The new training set after splitting is:

	Bold Letters	**susp. adress**	**Virus**
s3	0	1	0
s4	0	0	1

We can see that the feature "Bold Letters" has the same value (0), for all the instances in the table. Since it does not vary, it can not split the dataset into meaningful subgroups. The entropy before and after splitting on this feature will remain the same, resulting in Information Gain = 0. Suspicious Address, on the other hand, has variation, which can be used to split the dataset into subsets. This will reduce the entropy of the system and provide some Information Gain. Since "Bold Letters" has no contribution to splitting, "Suspicious Address" becomes the natural next feature to split on.
So the final decision tree will look like this:

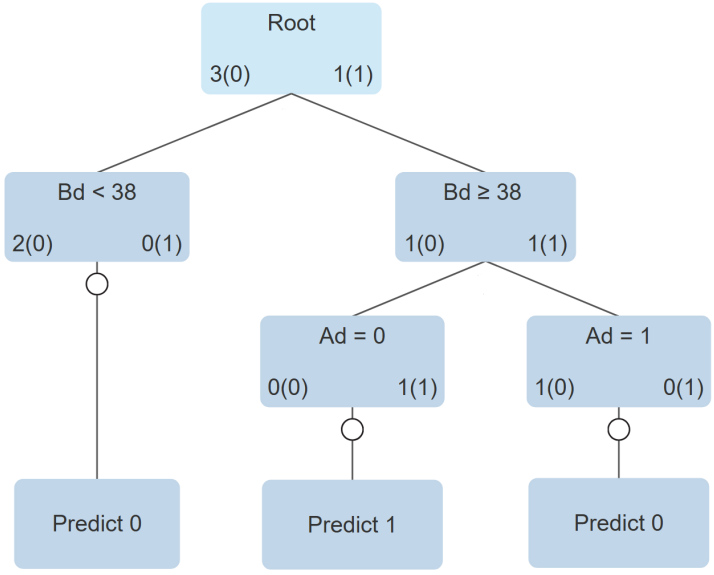

Figure 2: Split in respect to "Suspicious Address"

Classifying the test set samples and reporting the accuracy

In order to classify the test set samples, we need to pass the information from each sample through the decision tree we have just built. Then, we predict whether it is a virus or not. The table below shows the prediction for each sample in the test set:

	Predictions	**Ground Truth**
s5	0	1
s6	0	0
s7	0	1
s8	0	0

As we can see, the Decision Tree predicted each sample to belong to class 0. This results in an accuracy of 0.5, behaving like a random classifier.

Exercise 2 - Random Forest (Programming)

Part A

I implemented the function `TrainRF`, which trains n_trees Decision Trees. Each tree considers only $\sqrt{d}$ features, and the data are bootstrapped. The function returns a dictionary where each key is a unique identifier for a tree, and the corresponding value is the `DecisionTreeClassifier` object representing that tree.
The `PredictRF` function is a simple function that iterates through each tree in the forest and retrieves the predictions for each of them. It returns a matrix where each row corresponds to the predictions from an individual Decision Tree, and each column represents the predictions for a specific sample in the input data.
The default value for the `min_samples_leaf` variable is 1, and each tree was split using the `entropy` method.

Part B

In Figure 3, the histogram shows the accuracies of Decision Trees when the `min_samples_leaf` parameter is set to 1, meaning each leaf in the tree can have as few as one sample. The accuracy distribution is wider, with some Decision Trees achieving very high accuracy close to 0.90, but also some trees performing poorly (below 0.75). We can see that the mean accuracy of all Decision Trees is lower than that of the Random Forest, indicating that the Random Forest outperforms most individual trees. Additionally, we observe that almost all the Decision Trees have lower accuracies than the Random Forest, reflecting the overfitting nature of using very small leaf sizes.

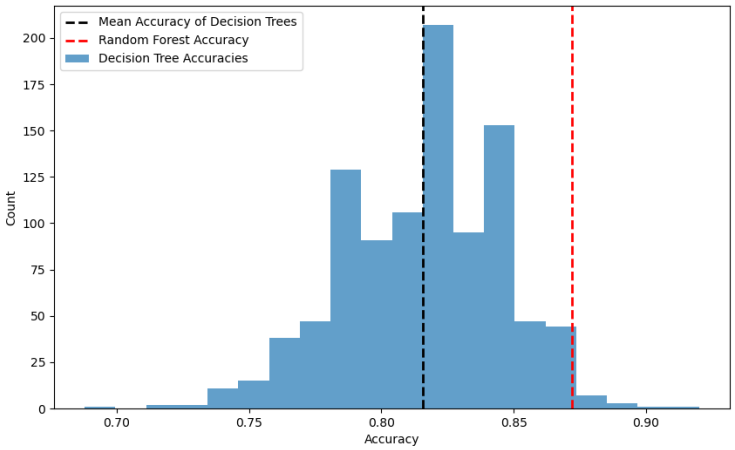

Figure 3: Histogram of each decision tree's accuracy, showing its mean and the random forest's accuracy (`min_samples_leaf = 1`).

In Figure 4, however, the histogram represents the accuracies of Decision Trees when the `min_samples_leaf` parameter is set to 10, meaning each leaf in the tree must have at least 10 samples. The accuracy distribution is narrower compared to the previous figure, with fewer extreme values and a tighter clustering around the mean. The mean accuracy of all Decision Trees is closer to that of the Random Forest, showing that increasing `min_samples_leaf` leads to more stable and less overfitted trees. It also reduces the variance in individual tree predictions.

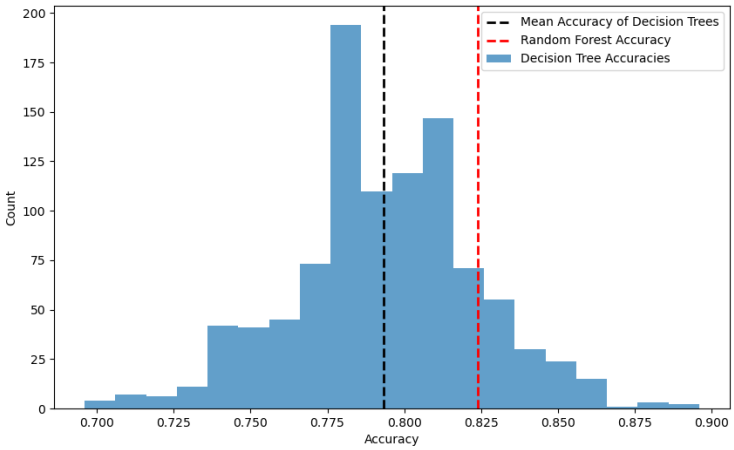

Figure 4: Histogram of each decision tree's accuracy, showing its mean and the random forest's accuracy (`min_samples_leaf = 10`)

Bonus

From Figure 5, we can see that the accuracy of the Random Forest is now very close to the mean accuracy of all the Decision Trees. This happens because the trees now use all the features, and the data are not bootstrapped.
Using the generated histogram, to calculate the probability that a single tree would achieve a similar or better accuracy than the Random Forest, we need to determine how many counts fall beyond the Random Forest accuracy and then divide that number by the total number of counts. From Figure 5, we can approximate the number of counts with better accuracy than the Random Forest to be around 490. The total number of decision trees is 1000. Therefore, the probability that a single tree would achieve a similar or better accuracy than the Random Forest is approximately:

$$\frac{490}{1000} = 49\%$$

This is similar to random. However, I might be way off with my calculations, as everything was estimated just by looking at the plot.

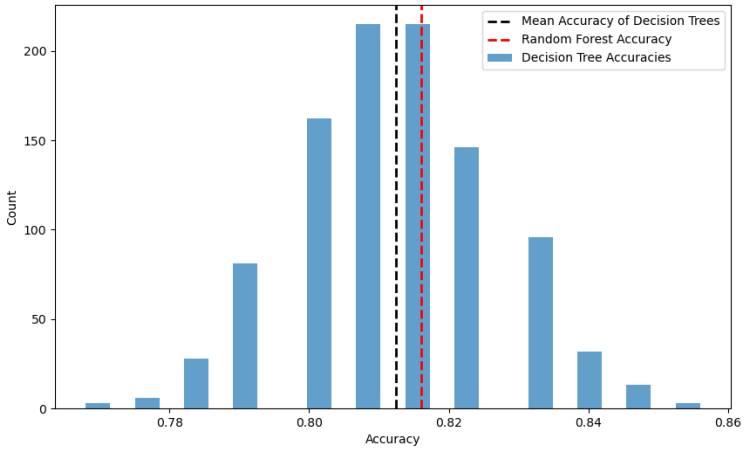

Figure 5: Histogram of each decision tree's accuracy, showing its mean and the random forest's accuracy (`min_samples_leaf = 1`, no bootstrap, all features used)