Question 1

Consider the training examples shown in Table 1 for a binary classification problem.

(a) Compute the Entropy for the overall collection of training examples.

Since the data set was split by 50% C0 and 50% C1, the entropy for the overall collection should be 1.

$$P(C0) = \frac{10}{20}, \quad P(C1) = \frac{10}{20}$$

Then,

$$\begin{aligned}\text{Entropy} &= -\left[\frac{10}{20} \times \log_2\left(\frac{10}{20}\right) + \frac{10}{20} \times \log_2\left(\frac{10}{20}\right)\right] \\ &= -(-0.5 \times 1 - 0.5 \times 1) \\ &= 1\end{aligned}$$

(b) Compute the Entropy for the Movie ID attribute.

$$\text{Entropy}_{\text{Movie ID}} = -1 \times \log_2 1 = 0.$$

(c) Compute the Entropy for the Format attribute.

Class	Count	Total
DVD	8	20
Online	12	

Then

$$\text{Entropy}_{\text{DVD}} = -\frac{6}{8} \times \log_2 \frac{6}{8} - \frac{2}{8} \times \log_2 \frac{2}{8} = 0.811$$

$$\text{Entropy}_{\text{Online}} = -\frac{4}{12} \times \log_2 \frac{4}{12} - \frac{8}{12} \times \log_2 \frac{8}{12} = 0.918$$

Therefor

$$\text{Entropy}_{\text{Format}} = 0.4 \times 0.811 + 0.6 \times 0.918 = 0.875$$

(d) Compute the Entropy for the Movie Category attribute using multiway split.

Easy to calculate

$$\text{Entropy}_{\text{Entertainment}} = -\frac{1}{4}\times \log_2\frac{1}{4} - \frac{3}{4}\times \log_2\frac{3}{4} = 0.811$$

$$\text{Entropy}_{\text{Comedy}} = -\frac{7}{8}\times \log_2\frac{7}{8} - \frac{1}{8}\times \log_2\frac{1}{8} = 0.544$$

$$\text{Entropy}_{\text{Documentaries}} = -\frac{2}{8}\times \log_2\frac{2}{8} - \frac{6}{8}\times \log_2\frac{6}{8} = 0.811$$

Therefore

$$\text{Entropy}_{\text{Movie Category}} = 0.2 \times 0.811 + 0.4 \times 0.511 + 0.4 \times 0.811 = 0.704$$

(e) Which of the three attributes has the lowest Entropy?

Movie ID has the lowest Entropy.

(f) Which of the three attributes will you use for splitting at the root node? Briefly explain your choice.

At the root node, the Movie Category should be used for splitting since it has the lowest Entropy value between Format and Movie Category. By using Movie Category, the gain was maximized. Even the Movie ID has the lowest value among all attributes, using this attribute cannot give any relevant gain regarding splitting.

Question 2

Consider the decision tree shown in Figure 1, and the corresponding training and test sets in Table 2 and 3 respectively.

(a) Estimate the generalization error rate of the tree using both the optimistic approach and the pessimistic approach. While computing the error with pessimistic approach, to account for model complexity, use a penalty value of 2 to each leaf node.

Penalty for each leaf node is 2.

$\text{Error(Optimistic)} = 0.$

$\text{Error(Pessimistic)} = \frac{0 + \{6 \times 2)}{15} = 0.8.$

(b) Compute the error rate of the tree on the test set shown in Table 3.

$$\text{Error} = \frac{4+3}{15} = 0.467$$

(c) Comment on the behavior of training and test set errors with respect to model complexity. Comment on the utility of incorporating model complexity in building a predictive model.

The optimistic generalization error on the training set increases when the decision tree grows while the training set error increases. This implicate the problem of overfitting.

The overfitting problem usually occurs when the model is too complicated. Therefore, corporate model complexity through evaluating pessimistic error and determine whether to add more nodes for the splitting model.

Question 3

Given the data sets shown in Figure 2, explain how the decision tree and k-nearest neighbor (K-NN) Classifiers would perform on these data sets.

(a) For the first case in the Figure 2.

Decision Tree: In this case, decision tree can generate satisfactory results since attributes can be distinguished from each other. Comparing two model by entropy gain, decision tree will have better performance.

K-NN: This model will not perform well since lots of noise attributes exist.

(b) For the second case in the Figure 2.

Decision Tree: Since the three classes are not distinguishable for decision tree. If the model try to capture the boundaries of classes, the tree could be large and complex and overfitting problem may occur.

K-NN: In this case, K-NN will work well since the algorithm was designed to solve such problems which contains distinguishable classes.

Question 4

Answer the following questions. Make sure to provide a brief explanation or an example to illustrate the answer.

(a) Are the rules mutually exclusive?

No, the rules are not mutually exclusive.

For example, the instance {Marital Status = Single, Home Owner = Yes} will trigger two rules in the rule sets.

(b) Is the rule set exhaustive?

No, the rule set is not exhaustive.

Instances contains condition {Marital Status = Divorced} may not trigger any rules.

(c) Is ordering needed for this set of rules?

Yes. The rule set is not mutually exclusive; therefore, one record can match two or more rules that give different predictions. {Martial Status = Single, Annual Income = Medium, Currently Employed = Yes} will trigger rule 2 and rule 5 which give different prediction results.

(d) Do you need a default class for the rule set?

Yes. The rule set is not mutually exclusive and exhaustive. To address those problem, a default rule is needed.

Question 5

Consider the problem of predicting whether a movie is popular given the following attributes: Format (DVD/Online), Movie Category (Comedy/Documentaries), Release Year, Number of world-class stars, Director, Language, Expense of Production and Length. If you had to choose between RIPPER and a k-nearest neighbor classifier, which would you prefer and why? Briefly explain why the other one may not work so well?

In this case, RIPPER is preferred. The attributes are diverse and have different relevance for the prediction targets as "popular" and "not popular". Therefore, by conducting RIPPER, it is possible to quantize the degree of relevance by calculating different parameters such as Gini Index and Entropy and even exclude the influence of irrelevant attributes (by discarding them).

However, the K-NN can be easily influenced by irrelevant attributes; thereafter, it may not perform well on this data set.