

## Predictive Models in Business Analytics – Assignment 1

Due: Thursday October 6, 2022 at 11:00am

<b>Group Number:</b>	<b><u>01</u></b>
----------------------	------------------

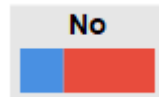
<b>Last Name</b> <i>(in alphabetical order)</i>	<b>First Name</b>
Jovanovic	Daniel
Sehgal	Rupin

### QUESTION 1 – WATER QUALITY [76 MARKS]

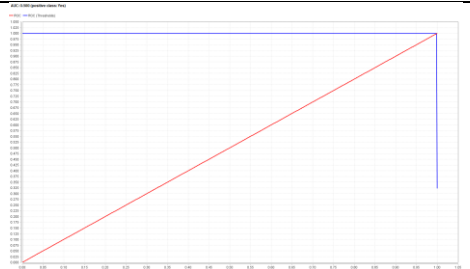

For all the questions below, use the gini index as splitting criterion and use the accuracy as the main performance criterion unless specified otherwise.

- a. (10 Marks) Split the data in 80% training data and 20% validation data with the use of shuffled sampling (use the default random seed of 1992). Use the Decision Tree operator to create the benchmark model (i.e., the Naïve Rule) in RapidMiner. Note that the default values for the parameters in the Decision Tree operator do not correspond to the benchmark model. Save your RapidMiner file as "A1\_Q1a\_Benchmark.rmp" in the processes folder of your Local Repository.

**Benchmark model (screenshot):**

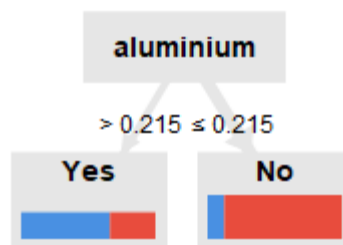


	Training performance			Validation Performance		
<b>confusion matrix</b>		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	1476	706	Pred. Unsafe	340	206
	Pred. Safe	0	0	Pred. Safe	0	0
<b>accuracy</b>	67.64%			62.27%		

<b>sensitivity (or recall)</b>	0.00%	0.00%
<b>specificity</b>	100.00%	100.00%
<b>precision (or TP rate)</b>	Unknown	Unknown
<b>f measure</b>	Unknown	Unknown
<b>lift</b>	Unknown	Unknown
<b>AUC</b>	0.500	0.500
<b>ROC curve (screenshot)</b>		

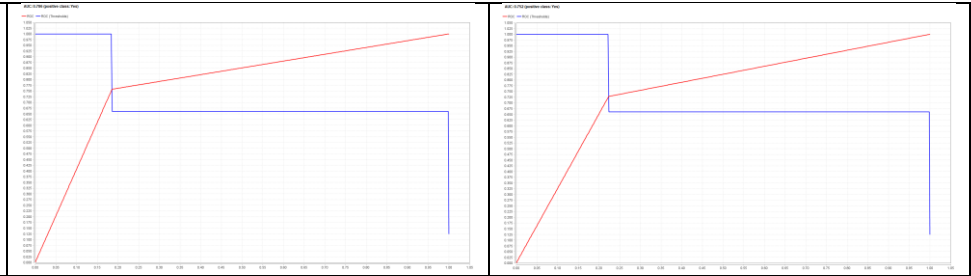
- b. (4 Marks) Save the file of Question 1a as "A1\_Q1b\_OneRule.rmp" in the processes folder of your Local Repository. Set the parameter values of the Decision Tree operator to create the One Rule model in RapidMiner.

**One Rule model (screenshot):**



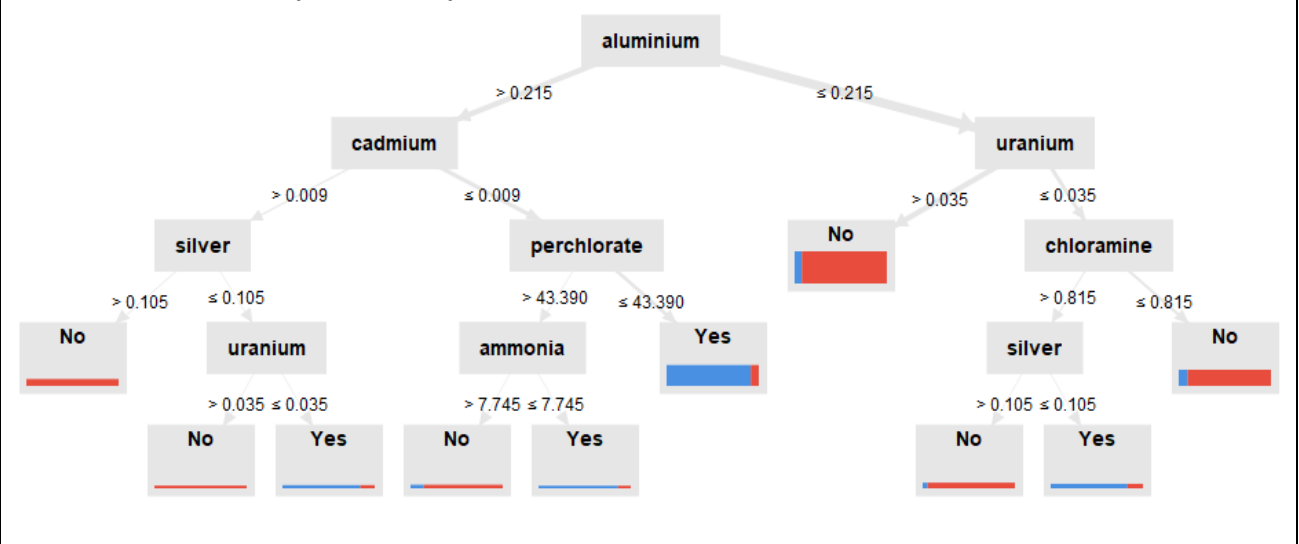
	Training performance	Validation Performance																		
<b>confusion matrix</b>	<table> <tr> <th></th><th>Unsafe</th><th>Safe</th></tr> <tr> <th>Pred. Unsafe</th><td>1202</td><td>171</td></tr> <tr> <th>Pred. Safe</th><td>274</td><td>535</td></tr> </table>		Unsafe	Safe	Pred. Unsafe	1202	171	Pred. Safe	274	535	<table> <tr> <th></th><th>Unsafe</th><th>Safe</th></tr> <tr> <th>Pred. Unsafe</th><td>264</td><td>56</td></tr> <tr> <th>Pred. Safe</th><td>76</td><td>150</td></tr> </table>		Unsafe	Safe	Pred. Unsafe	264	56	Pred. Safe	76	150
	Unsafe	Safe																		
Pred. Unsafe	1202	171																		
Pred. Safe	274	535																		
	Unsafe	Safe																		
Pred. Unsafe	264	56																		
Pred. Safe	76	150																		
<b>accuracy</b>	79.61%	75.82%																		
<b>sensitivity (or recall)</b>	75.78%	72.82%																		
<b>specificity</b>	81.44%	77.65%																		
<b>precision (or TP rate)</b>	66.13%	66.37%																		
<b>f measure</b>	70.63%	69.44%																		
<b>lift</b>	204.39%	175.92%																		
<b>AUC</b>	0.786	0.752																		

**ROC curve  
(screenshot)**

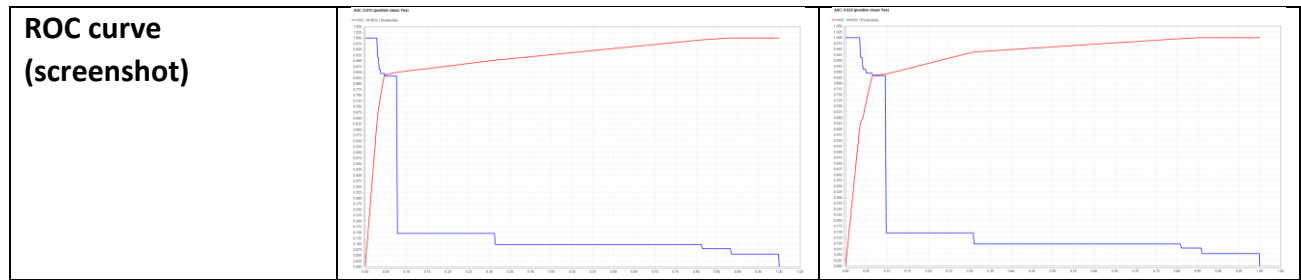


c. (4 Marks) Save the file of Question 1b as "A1\_Q1c\_DecisionTree.rmp" in the processes folder of your Local Repository. Update the Decision Tree operator to create a decision tree model in RapidMiner.

**Decision Tree model (screenshot):**

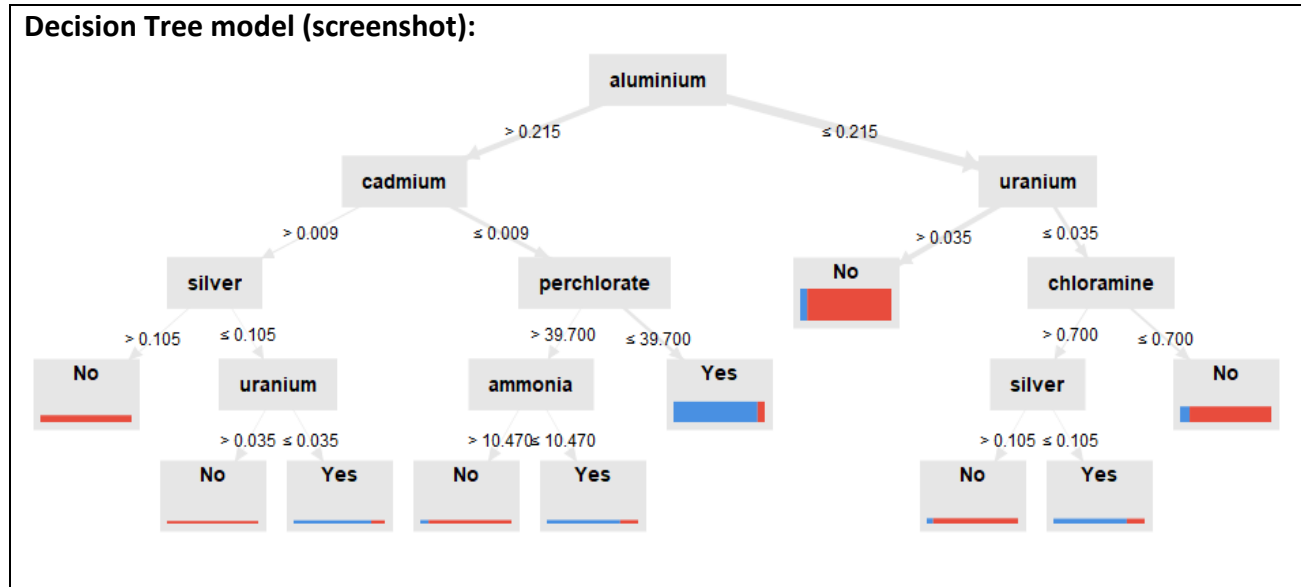


	Training performance			Validation Performance		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	1408	114	Pred. Unsafe	318	34
	Pred. Safe	68	592	Pred. Safe	22	172
accuracy	91.66%			89.74%		
sensitivity (or recall)	83.85%			83.50%		
specificity	95.39%			93.53%		
precision (or TP rate)	89.70%			88.66%		
f measure	86.68%			86.00%		
lift	277.22%			234.99%		
AUC	0.915			0.922		

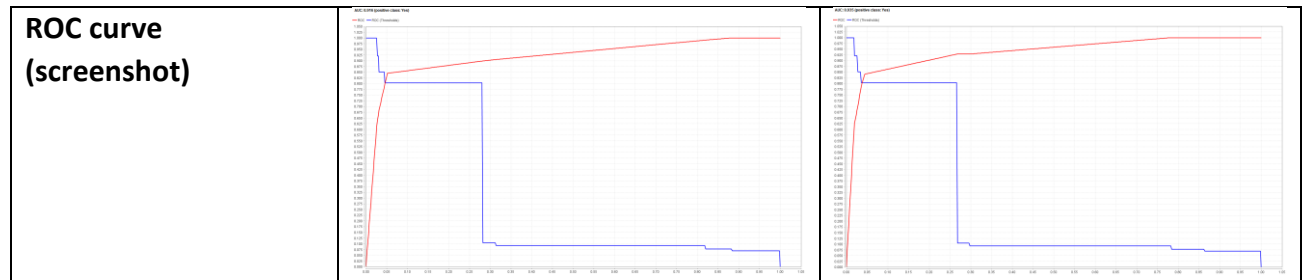


Besides shuffled sampling, there are also other alternatives to split the data in training and validation datasets. We will explore stratified sampling in the next couple of questions.

d. (2 Marks) Repeat questions 1c but with stratified sampling (still use the local random seed of 1992). You don't have to save the processes under a different file name as you only need to change one parameter value compared to the previous question.



	Training performance			Validation Performance		
confusion matrix						
		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	1378	113	Pred. Unsafe	347	29
	Pred. Safe	75	617	Pred. Safe	16	153
accuracy	91.39%			91.74%		
sensitivity (or recall)	84.52%			84.07%		
specificity	94.84%			95.59%		
precision (or TP rate)	89.16%			90.53%		
f measure	86.78%			87.18%		
lift	266.63%			271.10%		
AUC	0.916			0.935		



- e. (3 Marks) Compare the performance on the datasets between Q1c and Q1d. What is the impact of stratified sampling compared to shuffled sampling on the training and validation dataset? Motivate your answer.

The application of stratified sampling in question 1d resulted in greater performance across all of the relevant metrics for the validation dataset, which includes accuracy, sensitivity, specificity, precision, f-measure, lift and AUC. However, with regards to the training dataset, the stratified sampling only increased performance for the sensitivity, f-measure, and AUC metrics, while the accuracy, specificity, precision, and lift all experienced worse performance when compared to the model built with shuffled sampling.

- f. (2 Marks) Explain why the accuracy on the validation dataset is usually (slightly) better when stratified sampling is used compared to shuffled sampling?

Since the training dataset represents 80% of the original data, it is likely that each class within the sample is already appropriately represented when compared to the original data, which is why a similar accuracy percentage is observed with both shuffled and stratified sampling (as is seen in Questions 1c and 1d). However, because the validation dataset is much smaller, and only represents 20% of the original data, there is a smaller chance that each class in the sample will be adequately represented. Therefore, because stratified sampling ensures that each class is equally represented within the sample, this would explain why the accuracy is higher when stratified sampling is used for the validation dataset.

In other words, stratified sampling makes sure that each class in the data is equally represented within the generated sample. This is especially useful when dealing with a small subset of the original data, since it is likely that this subset does not fully represent each class appropriately. As a result, because the validation dataset is relatively small (i.e. only 20% of the original data), this suggests that stratified sampling will produce a higher accuracy when compared to shuffled sampling.

- g. (3 Marks) When constructing a test dataset, is it more appropriate to use shuffled sampling or stratified sampling? Motivate your answer.

We believe that stratified sampling is more appropriate to use when constructing a test dataset. The reason for this is due to the fact that a test dataset tends to be a smaller sample of the original data, and therefore, shuffled sampling may skew the data within the test set to the point where certain classes are either over- or under-represented. Conversely, since stratified sampling maintains data proportions with respect to the various classes in the original data, this ensures that the data within the test dataset accurately represents the data within the original set.

- h. (3 Marks) When there are less examples in the dataset, would there be more or less of a difference in the performance of your classification model on the validation dataset between shuffled sampling and stratified sampling? Motivate your answer.

If there are less examples in a given dataset, then there is a higher likelihood of obtaining skewed or non-representational data when taking a random sample of that dataset. Because a stratified sample is able to better represent the structure of the original data when compared to shuffled sampling, there would likely be more of a difference in the performance of a classification model applied to the validation dataset when there are less examples.

- i. (3 Marks) When the unbalance in your target attribute (in this question, the target attribute is “*is\_safe*”) is larger, would there be more or less of a difference in the performance of your classification model on the validation dataset between shuffled sampling and stratified sampling? Motivate your answer.

When there is a large unbalance of the target attribute, there would likely be more of a difference in the performance of a classification model applied to the validation dataset between shuffled sampling and stratified sampling. This is because a large imbalance with regard to the target attribute suggests that a certain class of that attribute is more heavily emphasized within that dataset, and since the validation set is relatively small, a random sample is likely to produce skewed results.

- j. (3 Marks) Other than shuffled sampling and stratified sampling to create a validation dataset, there can also be k-fold cross validation. Does it make sense to redo Question 1c with k-fold cross validation to create the subsets for the training and validation data sets when constructing the decision tree? Motivate your answer. Note that you do not have to implement k-fold cross validation in RapidMiner for this question yet.

We believe that it does make sense to redo question 1c with k-fold cross validation to create the subsets for the training and validation data sets when constructing the decision tree. Although it would require an increased amount of both computing power and time, using k-fold cross

validation will result in an enhanced degree of accuracy due to the additional evaluation of the model. Furthermore, this would provide a preventative measure against overfitting.

For the remaining questions, use 10-fold cross validation with shuffled sampling (and 1992 as the local random seed of 1992) on 80% of the total dataset, and use the remaining 20% as the test dataset (where you also use shuffled sampling with the local random seed of 1992 to generate the test dataset). You can use the same performance measures as before, but it is sufficient to apply the model on the validation and test datasets only (and NOT on the training dataset anymore).

- k. (10 Marks) Optimize the parameter values for the decision tree. Save the file of Question 1c as "A1\_Q1k\_Optimization.rmp" and include the optimization subtask. Be careful where to position the split data operator, the cross-validation operator and the optimization operator in your data mining process.

Q1k – Best Parameter Values	
splitting criterion	Information Gain
maximal depth	10
minimal leaf size	5
minimal size for split	20
number of prepruning alternatives	2

	Validation Performance			Test Performance		
<b>confusion matrix</b>		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	1383	81	Pred. Unsafe	319	23
	Pred. Safe	93	625	Pred. Safe	21	183
<b>accuracy</b>	92.03% (+/- 1.40%)			91.94%		
<b>sensitivity (or recall)</b>	88.18% (+/- 6.11%)			88.83%		
<b>specificity</b>	93.69% (+/- 1.78%)			93.82%		
<b>precision (or TP rate)</b>	86.98% (+/- 3.44%)			89.71%		
<b>f measure</b>	87.44% (+/- 3.37%)			89.27%		
<b>lift</b>	271.57% (+/- 27.18%)			237.76%		
<b>AUC</b>	0.955 (+/- 0.014)			0.966		

- l. (4 Marks) Repeat Question 1k but use stratified sampling for the 10-fold cross validation task (instead of shuffled sampling) – note that the test dataset is still generated with shuffled sampling. Similar to Question 1d, you don't have to save the processes under a different file name as you only need to change one parameter value compared to the previous question.

<b>Q1l – Best Parameter Values</b>
------------------------------------

splitting criterion	Information Gain
maximal depth	8
minimal leaf size	5
minimal size for split	20
number of prepruning alternatives	2

	Validation Performance	Test Performance																		
<b>confusion matrix</b>	<table> <tr> <td></td><td>Unsafe</td><td>Safe</td></tr> <tr> <td>Pred. Unsafe</td><td>1366</td><td>67</td></tr> <tr> <td>Pred. Safe</td><td>110</td><td>639</td></tr> </table>		Unsafe	Safe	Pred. Unsafe	1366	67	Pred. Safe	110	639	<table> <tr> <td></td><td>Unsafe</td><td>Safe</td></tr> <tr> <td>Pred. Unsafe</td><td>316</td><td>15</td></tr> <tr> <td>Pred. Safe</td><td>24</td><td>191</td></tr> </table>		Unsafe	Safe	Pred. Unsafe	316	15	Pred. Safe	24	191
	Unsafe	Safe																		
Pred. Unsafe	1366	67																		
Pred. Safe	110	639																		
	Unsafe	Safe																		
Pred. Unsafe	316	15																		
Pred. Safe	24	191																		
<b>accuracy</b>	91.89% (+/- 2.06%)	92.86%																		
<b>sensitivity (or recall)</b>	90.51% (+/- 4.29%)	92.72%																		
<b>specificity</b>	92.55% (+/- 2.45%)	92.94%																		
<b>precision (or TP rate)</b>	85.48% (+/- 4.10%)	88.84%																		
<b>f measure</b>	87.84% (+/- 3.04%)	90.74%																		
<b>lift</b>	264.22% (+/- 12.99%)	235.46%																		
<b>AUC</b>	0.956 (+/- 0.011)	0.975																		

**Compare the performance of the best model to the performance of the model you created in Question 1k:**

The accuracy for the test dataset was higher when stratified sampling was used in question 1l, while the accuracy for the validation dataset was slightly worse compared to the shuffled sampling used in question 1k. Furthermore, the sensitivity for the model in question 1l was higher for both the validation and test datasets, while the specificity was lower when compared to the model in question 1k. This suggests that the model built in question 1l is better able to avoid false negatives, but is more susceptible to false positives, when compared to the model from question 1k. The precision was higher across both data sets for the model in question 1k, while the f-measure was higher in the stratified sample model for the validation set and almost identical for the test set. In other words, this implies that the model in question 1l has a higher true positive rate, and has higher accuracy on the validation dataset as indicated by the f-measure. The lift of the model in question 1k is higher for both the validation and training set, and since lift is a measure of the effectiveness of our predictive model, it seems as though the shuffled sample model is more effective overall, from this perspective. Lastly, the AUC for the model in question 1l is slightly higher for both the validation and test datasets, which suggests that it has a better ability to distinguish between classes when compared to the model in question 1k.

Besides accuracy, let's consider some other performance measures to decide on the best classification model. In particular, there is a difference when you make a classification error.



Assume that it is six times costlier to incorrectly predict safe water (i.e., false positive) than it is to incorrectly predict unsafe water (i.e., false negative).

For the next questions, split the entire dataset in the following subsets: use 80% for the training and validation data and 20% for the test data. For the subset with 80% of the data, use 10-fold cross validation for the training and validation purposes. For both tasks to create training, validation and test datasets, use shuffled sampling with local random seed 1992 (similar to Question 1k).

- m. (5 Marks) Comment on the following performance measures if they are used for comparing the performance of classification models on the validation dataset under these (new) circumstances (i.e., when there is asymmetry in the misclassification cost): accuracy, AUC, sensitivity, specificity. Which of these performance measures is the most appropriate measure for this task? Motivate your answer.

For the given task, accuracy would not be the optimal metric to use since it doesn't fully consider the asymmetry of the misclassification cost present within the scenario. In general, the AUC is a measure which examines the ability of a model to distinguish between classes. In other words, the higher the AUC, the better the performance of the model at differentiating between positive and negative classes. Similar to accuracy, the AUC does not adequately address the higher cost associated with a false positive. Furthermore, sensitivity measures a model's true positive rate, which means that a model with a high sensitivity is able to avoid false negatives to a higher degree. However, since false negatives are six times less costly when compared to false positives, this performance measure is also suboptimal.

Therefore, within the context of this question, because a false positive is six times costlier than a false negative, the specificity is the most appropriate measure for this task because it specifically evaluates the model's ability to predict true negatives. In other words, because a model with high specificity suggests that it is able to correctly identify negative results, and subsequently has a greater ability to avoid false positives, this would be preferred within this scenario due to the higher cost of false positives.

- n. (4 Marks) Save the file of Question 1k as "A1\_Q1n\_AUC.rmp". Repeat Q1k but use the AUC as performance measure for deciding which classification model performs the best on the validation data set.

Q1n – Best Parameter Values	
splitting criterion	Information Gain
maximal depth	10
minimal leaf size	5
minimal size for split	30

number of prepruning alternatives	2
-----------------------------------	---

	Validation Performance			Test Performance		
confusion matrix						
		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	1386	85	Pred. Unsafe	320	22
	Pred. Safe	90	621	Pred. Safe	20	184
accuracy	91.98% (+/- 1.51%)			92.31%		
sensitivity (or recall)	87.69% (+/- 5.72%)			89.32%		
specificity	93.90% (+/- 2.08%)			94.12%		
precision (or TP rate)	87.33% (+/- 4.06%)			90.20%		
f measure	87.37% (+/- 3.27%)			89.76%		
lift	272.67% (+/- 28.21%)			239.06%		
AUC	0.957 (+/- 0.011)			0.966		

**Compare the performance of the best model to the performance of the model you created in Question 1k:**

After utilizing AUC as the performance measure, the model produced in question 1n outperformed the model from question 1k in almost all performance metrics with regard to the test dataset, except for AUC where both models matched each other. However, for the validation dataset, the model in question 1n performed comparatively worse with regard to accuracy, sensitivity, and f-measure. Subsequently, it performed slightly better than the model from question 1k when considering metrics such as specificity, precision, lift and AUC.

- o. (8 Marks) Save the file of Question 1k as "A1\_Q1o\_Threshold.rmp. Repeat Q1k but use a cut-off threshold value such that examples with propensity scores of 0.8 or higher for the important class are classified as "Yes" (or as being safe – recall that identifying whether the water sample is safe is considered the important class). Be careful where in the process to include the new cut-off values for the threshold for your training and validation process as well as for the test dataset (note that we have only included a threshold value to evaluate the performance in the test dataset during our lecture). Other than including an updated cut-off threshold value, also use the performance measure as suggested by you in Question 1m.

Q1o – Best Parameter Values	
splitting criterion	Information Gain
maximal depth	8
minimal leaf size	5
minimal size for split	20
number of prepruning alternatives	2

	Validation Performance	Test Performance
<b>confusion matrix</b>		
<b>accuracy</b>	90.24% (+/- 1.87%)	87.18%
<b>sensitivity (or recall)</b>	75.79% (+/- 5.22%)	70.87%
<b>specificity</b>	97.09% (+/- 1.47%)	97.06%
<b>precision (or TP rate)</b>	92.59% (+/- 3.69%)	93.59%
<b>f measure</b>	83.24% (+/- 3.53%)	80.66%
<b>lift</b>	289.36% (+/- 31.86%)	248.06%
<b>AUC</b>	0.957 (+/- 0.014)	0.975

**Compare the performance of the best model to the performance of the model you created in Question 1k:**

Upon the implementation of the cut-off threshold value, the model performed worse with respect to the accuracy, sensitivity, and f-measure metrics when compared to the model in question 1k for both the test and validation datasets. Conversely, the specificity, precision, lift, and AUC were all higher for the model in question 1o for both datasets.

- p. (8 Marks) Save the file of Question 1k as "A1\_Q1p\_Cost.rmp" and include an appropriate cost matrix that reflects the cost difference between false negatives and false positives (as mentioned above). Extend the data mining process such that the best classification model in terms of lowest misclassification cost is selected in your training and validation process.

<b>Q1p – Best Parameter Values</b>	
splitting criterion	Gini Index
maximal depth	4
minimal leaf size	20
minimal size for split	10
number of prepruning alternatives	5

	Validation Performance	Test Performance
<b>confusion matrix</b>		
<b>accuracy</b>	85.80% (+/- 2.05%)	83.70% ↓

<b>sensitivity (or recall)</b>	64.05% (+/- 7.35%)	62.62%
<b>specificity</b>	96.05% (+/- 2.74%)	96.47%
<b>precision (or TP rate)</b>	89.21% (+/- 5.95%)	91.49%
<b>f measure</b>	74.17% (+/- 4.72%)	74.35%
<b>lift</b>	278.91% (+/- 35.13%)	242.49%
<b>AUC</b>	0.832 (+/- 0.032)	0.810
<b>misclassification cost</b>	0.275 (+/- 0.102)	0.273

**Compare the performance of the best model to the performance of the model you created in Question 1o:**

After the inclusion of the cost matrix which reflects the cost difference between false negatives and false positives, the performance of the model in question 1p is worse across both the test and validation dataset with regard to all of the relevant performance measures, when compared to the model from question 1o.

## QUESTION 2 – MOBILE PRICING [24 MARKS]

- a. (10 Marks) Calculate the gain ratio and gini index for the following three-way split of a root node on the full dataset:  $RAM \leq 1106$ ,  $1106 < RAM \leq 3013.5$ ,  $RAM > 3013.5$ . Motivate your answer with calculations.

Gain ratio: Answer = 0.592

### Gain Ratio

- $RAM \leq 1106$

$$\begin{aligned} \text{Entropy (MP)} &= - \sum_{i=1}^n p_i \log(p_i) \\ &= - \left( \frac{402}{451} \log_2 \left( \frac{402}{451} \right) + \frac{49}{451} \log_2 \left( \frac{49}{451} \right) \right) \\ &= +0.496 \times \frac{451}{2000} \text{ (Normalizing)} = 0.112 \end{aligned}$$

- $1106 < RAM \leq 3013.5$

$$\begin{aligned} \text{Entropy (MP)} &= - \sum_{i=1}^n p_i \log(p_i) \\ \text{Entropy (MP)} &= - \left( \frac{98}{1029} \log_2 \left( \frac{98}{1029} \right) + \frac{451}{1029} \log_2 \left( \frac{451}{1029} \right) \right. \\ &\quad \left. + \frac{412}{1029} \log_2 \left( \frac{412}{1029} \right) + \frac{68}{1029} \log_2 \left( \frac{68}{1029} \right) \right) \\ &= +1.632 \times \frac{1029}{2000} = 0.840 \end{aligned}$$

- $RAM > 3013.5$

$$\begin{aligned} \text{Entropy (MP)} &= - \sum_{i=1}^n p_i \log(p_i) \\ &= - \left( \frac{88}{520} \log_2 \left( \frac{88}{520} \right) + \frac{432}{520} \log_2 \left( \frac{432}{520} \right) \right) \\ &= +0.656 \times \frac{520}{2000} = 0.171 \end{aligned}$$

$$\begin{aligned} \text{Entropy (Normalized)} &= 0.171 + 0.840 + 0.112 \\ &= 1.122 \end{aligned}$$

$$\begin{aligned}\text{Information Gain} &= \text{Entropy}(p, m) - \text{Entropy}(MP) \\ &= 2 - 1.122 \\ &= 0.878\end{aligned}$$

$$\begin{aligned}\text{Split Info} &= - \sum \frac{n_i}{n} \log \left( \frac{n_i}{n} \right) \\ &= - \left( \frac{451}{2000} \log \left( \frac{451}{2000} \right) + \frac{1029}{2000} \log \left( \frac{1029}{2000} \right) \right. \\ &\quad \left. + \frac{520}{2000} \log \left( \frac{520}{2000} \right) \right) \\ &= + 1.483\end{aligned}$$

$$\begin{aligned}\text{Gain Ratio} &= \frac{\text{Information gain}}{\text{Split Information}} \\ &= \frac{0.878}{1.483}\end{aligned}$$

$$\text{Gain Ratio} = 0.592$$

Gini index: Answer = 0.443

### Gini Index

○  $RAM \leq 1106$   
 $ICMP) = 1 - \sum_{i=1}^m p_i^2$

$$ICMP) = 1 - \left( \frac{402}{451}^2 + \frac{49}{451}^2 + \frac{0}{451}^2 + \frac{0}{451}^2 \right)$$

$$ICMP) = 1 - (0.806) \quad \leftarrow \text{To normalize}$$

$$ICMP) = 0.194 \times \frac{451}{2000} = 0.044$$

○  $1106 < RAM \leq 3013.5$

$$ICMP) = 1 - \sum_{i=1}^m p_i^2$$

$$ICMP) = 1 - \left( \frac{98}{1029}^2 + \frac{49}{1029}^2 + \frac{412}{1029}^2 + \frac{68}{1029}^2 \right)$$

$$ICMP) = 1 - 0.366 \quad \leftarrow \text{Normalizing}$$

$$ICMP) = 0.634 \times \frac{1029}{2000} = 0.326$$

○  $RAM > 3013.5$

$$ICMP) = 1 - \sum_{i=1}^m p_i^2$$

$$ICMP) = 1 - \left( \frac{88}{520}^2 + \frac{432}{520}^2 + \frac{0}{520}^2 + \frac{0}{520}^2 \right)$$

$$ICMP) = 1 - (0.719) \quad \leftarrow \text{Normalizing}$$

$$ICMP) = 0.281 \times \frac{520}{2000} = 0.073$$

$$\text{Gini Index (Normalized)} = 0.073 + 0.326 + 0.044$$
$$= 0.443$$

- b. (14 Marks) For this question, split the data using shuffled sampling with random local seed of 1992 such that 80% of the data is used for training and validation (use 10-fold cross validation with shuffled sampling) and 20% of the data is used for testing. In other words, split the dataset similar as you did in Questions 1k and beyond. Find the decision tree that is performing the best

(measured as the lowest average misclassification cost) on your validation dataset where you use the following cost matrix:

	True 0	True 1	True 2	True 3
Predicted 0	0	1	2	4
Predicted 1	1	0	1	2
Predicted 2	2	1	0	1
Predicted 3	4	2	1	0

Save the file as "A1\_Q2.rmp".

**Clearly explain your thought process as well as all steps that you have taken in RapidMiner to find the best performing decision tree.**

Initially, we started with as outlined in the question. We adjusted the performance measures, test cost matrix and the validation cost matrix to reflect the parameters provided in the question; we adjusted the class order definition (enumeration) accordingly as well.

From there we ran the model with the optimization parameters given in question 1k as a starting point. It returned a very low accuracy and high misclassification cost, proving that the optimization parameter was not setup in the correct manner. To improve on this, we looked at the optimized parameters the current model provided and made the appropriate adjustments to the parameter combinations for the optimization operator to explore. For example, if the maximal depth was capping out at 12, we would increase the range for the maximal depth within the optimization operator. This would apply inversely as well (i.e., minimal leaf size). We continued this process until we could no longer improve the performance of the model (i.e., could not reduce the misclassification costs any further).

Q2 – Best Parameter Values	
Criterion	Information Gain
Maximal Depth	10
Minimal leaf Size	5
Minimal Size for Split	12
The Number of Prepruning Alternatives	3

	Validation Performance				
confusion matrix		0	1	2	3
	Pred. 0	358	45	0	0
	Pred. 1	39	311	35	0
	Pred. 2	0	40	327	47
	Pred. 3	0	0	51	347



<b>accuracy</b>	83.94% (+/- 2.32%)
<b>misclassification cost</b>	0.161 (+/- 0.023)

	Test Performance				
<b>confusion matrix</b>		0	1	2	3
	Pred. 0	96	10	0	0
	Pred. 1	7	83	7	0
	Pred. 2	0	11	72	7
	Pred. 3	0	0	8	99
<b>accuracy</b>	87.50%				
<b>misclassification cost</b>	0.125				