HASKAYNE
SCHOOL OF BUSINESS

UNIVERSITY OF CALGARY
HASKAYNE SCHOOL OF BUSINESS

Predictive Models in Business Analytics – Assignment 2

Due: Wednesday October 26, 2022 at 2:00pm

Group Number:	**01**

Last Name *(in alphabetical order)*	**First Name**
Jovanovic	Daniel
Sehgal	Rupin

QUESTION 1 – WATER QUALITY [44 MARKS]

For this question, split the data three-ways where 60% is used for training, 20% for validation and 20% for test data based on shuffled sampling with local random seed 1992.

Let's 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). Therefore, we want to include the following cost matrix to calculate the misclassification cost:

	Actual Not Safe	Actual Safe
Predicted Not Safe	0	1
Predicted Safe	6	0

a. (7 Marks) Use the *Decision Tree* operator to create a classification model in RapidMiner, where the gini index is used as splitting criterion. For the decision tree model in RapidMiner, use the parameter values for the pruning and prepruning process as specified in the question

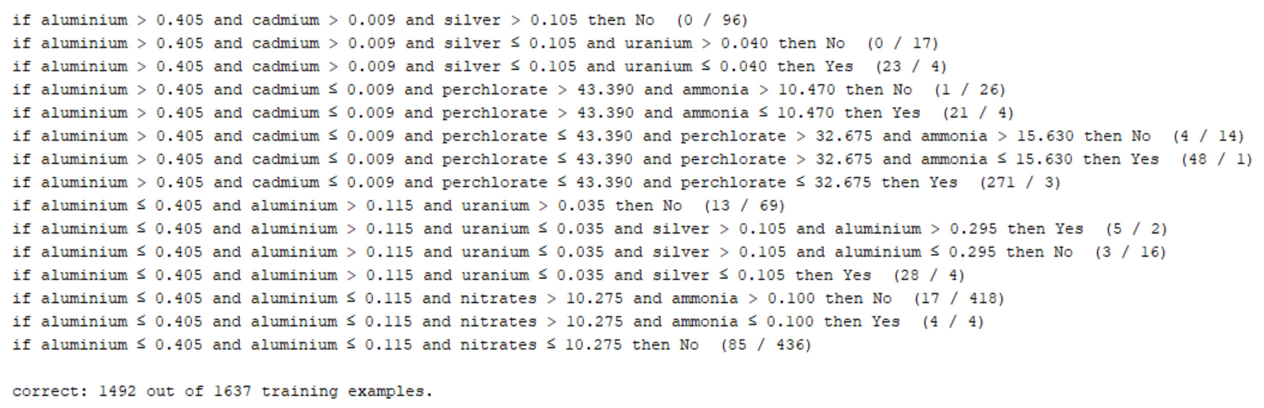
For the maximal depth, we will use the best performing value out of 6, 8 and 10. Other than optimizing over the maximal depth, we want to find the best cut-off threshold value for the important class to be classified as "Yes" (or the water to be classified as save) out of 0.02, 0.04, ..., 0.2. This corresponds to propensity scores for "Yes" of 0.98, 0.96, ..., 0.8, respectively, to be classified as "Yes". We use this definition all throughout the assignment. Save your RapidMiner file as "A2_Q1a_DecisionTree.rmp" in the processes folder of your Local Repository.

Q1a – Best Parameter Values	
maximal depth	6
cut-off threshold value	0.140

	Validation Performance			Test Performance		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	359	66	Pred. Unsafe	333	81
	Pred. Safe	3	117	Pred. Safe	7	125
accuracy	87.34%			83.88%		
sensitivity (or recall)	63.93%			60.68%		
specificity	99.17%			97.94%		
lift	290.37%			250.99%		
AUC	0.896			0.888		
misclassification cost	0.154			0.225		

b. (2 Marks) Save the file of Question 1a as "A2_Q1b_Tree_to_Rule.rmp" in the processes folder of your Local Repository. Update the process to create the induction rules from the decision tree with the parameter values that you found in Question 1a (i.e., no *Optimize Parameters* or *Cross Validation* is needed). Report the induction rules and verify that the performance for the test dataset is exactly the same.

Induction rules (screenshot):



```
if aluminium > 0.405 and cadmium > 0.009 and silver > 0.105 then No  (0 / 96)
if aluminium > 0.405 and cadmium > 0.009 and silver ≤ 0.105 and uranium > 0.040 then No  (0 / 17)
if aluminium > 0.405 and cadmium > 0.009 and silver ≤ 0.105 and uranium ≤ 0.040 then Yes  (23 / 4)
if aluminium > 0.405 and cadmium ≤ 0.009 and perchlorate > 43.390 and ammonia > 10.470 then No  (1 / 26)
if aluminium > 0.405 and cadmium ≤ 0.009 and perchlorate > 43.390 and ammonia ≤ 10.470 then Yes  (21 / 4)
if aluminium > 0.405 and cadmium ≤ 0.009 and perchlorate ≤ 43.390 and perchlorate > 32.675 and ammonia > 15.630 then No  (4 / 14)
if aluminium > 0.405 and cadmium ≤ 0.009 and perchlorate ≤ 43.390 and perchlorate > 32.675 and ammonia ≤ 15.630 then Yes  (48 / 1)
if aluminium > 0.405 and cadmium ≤ 0.009 and perchlorate ≤ 43.390 and perchlorate ≤ 32.675 then Yes  (271 / 3)
if aluminium ≤ 0.405 and aluminium > 0.115 and uranium > 0.035 then No  (13 / 69)
if aluminium ≤ 0.405 and aluminium > 0.115 and uranium ≤ 0.035 and silver > 0.105 and aluminium > 0.295 then Yes  (5 / 2)
if aluminium ≤ 0.405 and aluminium > 0.115 and uranium ≤ 0.035 and silver > 0.105 and aluminium ≤ 0.295 then No  (3 / 16)
if aluminium ≤ 0.405 and aluminium > 0.115 and uranium ≤ 0.035 and silver ≤ 0.105 then Yes  (28 / 4)
if aluminium ≤ 0.405 and aluminium ≤ 0.115 and nitrates > 10.275 and ammonia > 0.100 then No  (17 / 418)
if aluminium ≤ 0.405 and aluminium ≤ 0.115 and nitrates > 10.275 and ammonia ≤ 0.100 then Yes  (4 / 4)
if aluminium ≤ 0.405 and aluminium ≤ 0.115 and nitrates ≤ 10.275 then No  (85 / 436)

correct: 1492 out of 1637 training examples.
```

c. (4 Marks) Save the file of Question 1a as "A2_Q1c_Rule_Induction.rmp" in the processes folder of your Local Repository. Update the *Decision Tree* operator to the *Rule Induction* operator and optimize the parameter values as specified in the question to minimize the misclassification cost on the validation dataset.

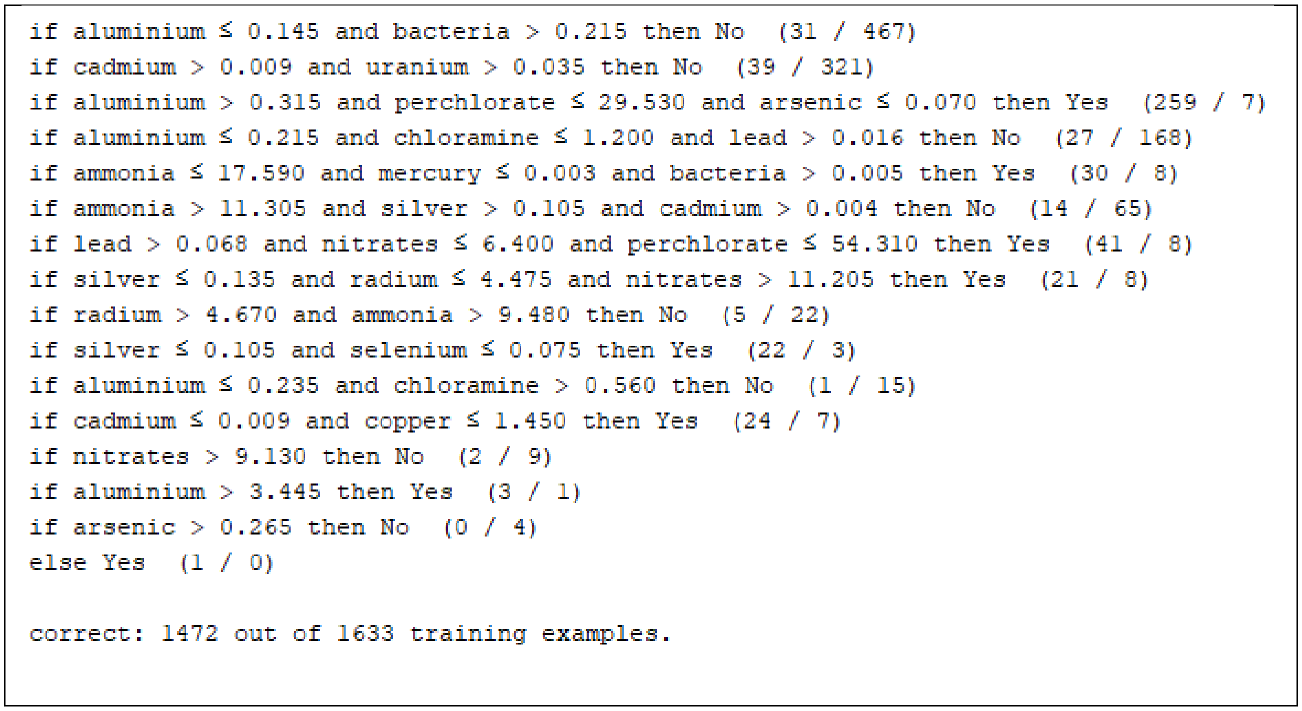
Q1c – Best Parameter Values	
sample ratio	0.450
pureness	0.930
minimal prune benefit	0.800
cut-off threshold value	0.060

	Validation Performance			**Test Performance**		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	362	92	Pred. Unsafe	335	110
	Pred. Safe	0	91	Pred. Safe	5	96
accuracy	83.12%			78.94%		
sensitivity (or recall)	49.73%			46.60%		
specificity	100.00%			98.53%		
lift	297.81%			251.93%		
AUC	0.894			0.855		
misclassification cost	0.169			0.256		

Compare the performance of the direct and indirect approach to generate induction rules for classification:

The direct approach utilized in question 1c performed worse compared to the indirect approach used in question 1b with regard to both accuracy and sensitivity, and had only a slightly lower AUC for the validation set. Conversely, the direct approach performed better in relation to the specificity and lift metrics, but it also generated a higher misclassification cost. Similarly, the direct approach performed worse when comparing the accuracy, sensitivity, and AUC on the test set, while it once again performed slightly better for the specificity and lift metrics, and had a higher misclassification cost.

Induction rules (screenshot):



```
if aluminium ≤ 0.145 and bacteria > 0.215 then No  (31 / 467)
if cadmium > 0.009 and uranium > 0.035 then No  (39 / 321)
if aluminium > 0.315 and perchlorate ≤ 29.530 and arsenic ≤ 0.070 then Yes  (259 / 7)
if aluminium ≤ 0.215 and chloramine ≤ 1.200 and lead > 0.016 then No  (27 / 168)
if ammonia ≤ 17.590 and mercury ≤ 0.003 and bacteria > 0.005 then Yes  (30 / 8)
if ammonia > 11.305 and silver > 0.105 and cadmium > 0.004 then No  (14 / 65)
if lead > 0.068 and nitrates ≤ 6.400 and perchlorate ≤ 54.310 then Yes  (41 / 8)
if silver ≤ 0.135 and radium ≤ 4.475 and nitrates > 11.205 then Yes  (21 / 8)
if radium > 4.670 and ammonia > 9.480 then No  (5 / 22)
if silver ≤ 0.105 and selenium ≤ 0.075 then Yes  (22 / 3)
if aluminium ≤ 0.235 and chloramine > 0.560 then No  (1 / 15)
if cadmium ≤ 0.009 and copper ≤ 1.450 then Yes  (24 / 7)
if nitrates > 9.130 then No  (2 / 9)
if aluminium > 3.445 then Yes  (3 / 1)
if arsenic > 0.265 then No  (0 / 4)
else Yes  (1 / 0)

correct: 1472 out of 1633 training examples.
```

Compare the induction rules from this question to those generated in Question 1b:
The induction rules generated in question 1b have 15 separate "if-then" rules, with a rule accuracy ratio of 0.9114 (1492/1637). Conversely, the induction rules derived from question 1c have 16 "if-then" rules including the "else" term at the very end, along with a rule accuracy ratio of 0.9014 (1472/1633). As a result, although the induction rules from question 1c have one more "if-then" rule when compared to the induction rules generated in question 1b, it has a slightly lower accuracy ratio. Furthermore, the induction rules from question 1b all start with assessing whether aluminum is greater than or less/equal to 0.405, and have other apparent consistencies with regard to the rules surrounding the assessment of cadmium, uranium, silver, perchlorate and ammonia. On the other hand, the induction rules from question 1c appear to be significantly less uniform, and include more attributes.

d. (3 Marks) Save the file of Question 1a as "A2_Q1d_NaiveBayes.rmp" in the processes folder of your Local Repository. Update the *Decision Tree* operator to the *Naïve Bayes* operator and find the cut-off threshold value that minimizes the misclassification cost on the validation dataset. Explore the following possible cut-off threshold values: 0.01, 0.02, …, 0.1.

Q1d – Best Parameter Values	
cut-off threshold value	0.01

	Validation Performance			**Test Performance**		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	352	136	Pred. Unsafe	336	160
	Pred. Safe	10	47	Pred. Safe	4	46

accuracy	73.21%	69.96%
sensitivity (or recall)	25.68%	22.33%
specificity	97.24%	98.82%
lift	245.57%	243.84%
AUC	0.802	0.809
misclassification cost	0.360	0.337

Compare the performance of the Naïve Bayes classification model to the performance of the decision tree model and the induction rules that you created in Question 1a and Question 1c, respectively:
With regard to the validation dataset, the Naïve Bayes classification model performed worse across all of the relevant metrics, and naturally had a higher misclassification cost, when compared to the model created in question 1a. For the test dataset, the Naïve Bayes model also had a higher misclassification cost and performed worse for most of the metrics including accuracy, sensitivity, lift, and AUC, however it performance better with regard to specificity.

When comparing the Naïve Bayes model to the model built in question 1c, the Naïve Bayes once again performed worse across all of the performance metrics and had a higher misclassification cost for the validation set. However, for the test set, although the Naïve Bayes had a higher specificity, it performed worse in all of the other metrics and had a higher misclassification cost.

e. (4 Marks) Save the file of Question 1a as "A2_Q1e_k-NN.rmp" in the processes folder of your Local Repository. Create a new classification "model" by applying the k-Nearest Neighbor technique in RapidMiner, where we want to find the parameter values that minimize the misclassification cost of the validation dataset as specified in the question. Make sure to include weighted votes, but don't use the Normalize operator in the process yet.

Q1e – Best Parameter Values	
numerical measure	Camberra Distance
k	18
cut-off threshold value	0.320

	Validation Performance			**Test Performance**		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	360	109	Pred. Unsafe	332	126
	Pred. Safe	2	74	Pred. Safe	8	80
accuracy	79.63%			75.46%		
sensitivity (or recall)	40.44%			38.83%		

specificity	99.45%	97.65%
lift	289.98%	240.95%
AUC	0.906	0.878
misclassification cost	0.222	0.319

Compare the performance of the best k-NN model to the performance of the best performing classification model that you created so far in this assignment:
The best performing classification model so far in this assignment has been the decision tree model from question 1a. The k-NN model developed in question 1e had a lower accuracy, sensitivity and lift for the validation set compared to the decision tree model, however it outperformed in terms of specificity and AUC, but had a higher overall misclassification cost. For the test set, the k-NN model performed worse across all of the performance metrics including accuracy, sensitivity, specificity, lift, and AUC, while also producing a higher misclassification cost. These results suggest that the best performing classification model is still the decision tree model from question 1a.

In the next two questions, I want you to think about how to use the *Normalize* operator.

f. (8 Marks) Comment on each of the four normalization approaches where to include the *Normalize* operator in the data mining process (as discussed in the question). In particular, identify which of the four approaches is methodologically correct. Motivate your answer. Also discuss why each of the three other strategies is not correct (i.e., discuss the mistake you would make if that approach would be used).

The first approach is invalid as it taints both the validation and test sets. Normalizing the entire data set and then splitting the results introduces the training parameters into the test set, which should be left untouched as it is for testing our model and would provide the most realistic results.

The second approach also manipulates the test data by normalizing which as mentioned should not occur as it wouldn't provide realistic results.

The third approach is also incorrect as it contaminates the validation dataset with normalization, instead the validation dataset should be trained using the normalization that occurred on the training dataset instead of being normalized on its own.

The best approach is the fourth approach is optimal as it omits any parameter transfers (info leaks) from the training set to the validation and test sets. Therefore, it avoids any bias to be in the validation or test set. All the other approaches introduce bias into either the validation or test set. If one were to use the other approaches you would assume higher performance measures that most likely would not hold up in a real-life scenario.

g. (8 Marks) Save the file of Question 1e as "A2_Q1g_k-NN_normalized.rmp" in the processes folder of your Local Repository. Next, implement the normalization approach that you selected in Question 1f such that it correctly includes the normalization. For this question, use the Z-transformation for the normalization.

Q1g – Best Parameter Values	
numerical measure	Manhattan Distance
k	18
cut-off threshold value	0.280

	Validation Performance			**Test Performance**		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	361	117	Pred. Unsafe	332	131
	Pred. Safe	1	66	Pred. Safe	8	75
accuracy	78.35%			74.54%		
sensitivity (or recall)	36.07%			36.41%		
specificity	99.72%			97.65%		
lift	293.37%			239.50%		
AUC	0.919			0.916		
misclassification cost	0.226			0.328		

Compare the performance of the best model to the performance of the k-NN model you created in Question 1e:
For the validation set, the model created in question 1g had a higher specificity, lift, and AUC when compared to the k-NN model developed in question 1e, however, it displayed a lower accuracy and sensitivity which likely contributed to the higher misclassification cost that was observed as well. As for the test set, the model from question 1g exhibited a lower accuracy, sensitivity, and lift, while showing a higher AUC and Misclassification cost compared to the model from question 1e. Furthermore, the model from question 1g and 1e each had the exact same specificity for the test dataset.

h. (4 Marks) Save the file of Question 1g as "A2_Q1h_SVM.rmp" in the processes folder of your Local Repository. Create a new classification model by replacing the *k-NN* operator with the *Support Vector Machine* operator in RapidMiner, where we want to optimize over the parameter values specified in the question such that the best model minimizes the misclassification cost over the validation dataset.

Q1h – Best Parameter Values	
Kernel type	Anova

C	0.100
cut-off threshold value	0.060

	Validation Performance			**Test Performance**		
confusion matrix		Unsafe	Safe		Unsafe	Safe
	Pred. Unsafe	355	85	Pred. Unsafe	335	87
	Pred. Safe	7	98	Pred. Safe	5	119
accuracy	83.12%			83.15%		
sensitivity (or recall)	53.55%			57.77%		
specificity	98.07%			98.53%		
lift	277.96%			254.36%		
AUC	0.944			0.954		
misclassification cost	0.233			0.214		

Compare the performance of the best model to the performance of the (best) classification models that you found in the previous questions:
When it is compared to the decision tree model built in question 1a, the SVM model performed worse with regard to accuracy, sensitivity , specificity, and lift for the validation set. Additionally, the SVM model had a higher AUC value, but also generated a higher misclassification cost. For the test set however, the SVM model displayed a higher specificity, lift, and AUC, while also resulting in a lower accuracy, sensitivity, and misclassification cost. As a result, it would appear that the SVM model produced lower misclassification costs for the test set, however the decision tree model had lower costs with regard to the validation set. Despite these mixed results, we believe that the decision tree model from question 1a is the better overall model due to its relatively superior performance across most of the relevant performance metrics.

However, when compared to the k-NN model from question 1e, the SVM model had a higher accuracy, sensitivity, AUC, and misclassification cost, while it also displayed a lower specificity and lift for the validation set. As for the test set, the SVM model outperformed the model from question 1e across all of the relevant metrics and also generated a lower misclassification cost. Despite the fact that the SVM model had a higher misclassification cost for the validation dataset, we believe that it generally performed better than the k-NN model from question 1e due to its superior performance across the majority of performance metrics.

i. (4 Marks) Save the file of Question 1a as "A2_Q1i_NeuralNet.rmp" in the processes folder of your Local Repository. Create a new classification model by replacing the *Decision Tree* operator with the *Neural Network* operator in RapidMiner, where one hidden layer with 12 nodes is included and we want to optimize over the parameter values as specified in the question such that the misclassification cost on the validation set is minimized. Make sure to include the *Shuffle* option as one of the parameter options, and use the local random seed of 1992. Also include the *Decay* option, such that your learning rate becomes smaller over time. Remember that neural networks also require normalization. However, the normalization process is included as parameter option within the Neural Network operator of RapidMiner. Therefore, no separate *Normalize* operators are needed.

Q1i – Best Parameter Values	
training cycles	275
learning rate	0.100
momentum	0.800
cut-off threshold value	0.130

	Validation Performance	**Test Performance**
confusion matrix		

Validation Performance

	Unsafe	Safe
Pred. Unsafe	360	107
Pred. Safe	2	76

Test Performance

	Unsafe	Safe
Pred. Unsafe	333	121
Pred. Safe	7	85

	Validation Performance	**Test Performance**
accuracy	80.00%	76.56%
sensitivity (or recall)	41.53%	41.26%
specificity	99.45%	97.94%
lift	290.18%	244.88%
AUC	0.888	0.902
misclassification cost	0.218	0.299

Compare the performance of the best model to the performance of the (best) classification models that you found in the previous questions:
When comparing the neural net model from question 1i to the decision tree model from question 1a for the validation set, the neural net model only outperformed with regard to the specificity metric, while it had a lower value for accuracy sensitivity, lift, and AUC, along with a higher misclassification cost. As for the test set, the neural net model had lower accuracy, sensitivity, and lift values, while it had a higher AUC and misclassification cost, whereas the specificity was the exact same compared to the decision tree model. Given the lower misclassification cost and better overall performance, the decision tree model appears to be superior.

Compared to the k-NN model from question 1e, the neural net model displayed better performance for the accuracy, sensitivity, and lift metrics for the validation dataset. Furthermore, despite a lower AUC value, the neural net model had the exact same specificity as the k-NN model from question 1e and had a slightly lower misclassification cost. With regards to the test set, the neural net model performed better across all of the given performance metrics and also had a lower misclassification cost. As a result, the neural net model is superior compared to the k-NN model from question 1e due to its lower misclassification costs and better performance across most of the relevant metrics.

For the validation set, the neural net model showed greater performance compared to the SVM model from question 1h for the specificity and lift metrics. However, the neural net model performed worst in regards to accuracy, sensitivity, and AUC, but also had a lower misclassification cost. For the test set, the neural net model performed worse across all of the performance metrics, and also had a higher misclassification cost when compared to the SVM model. Despite these slightly mixed results, the SVM model showed greater overall performance compared to the neural net model, especially with regard to the test dataset.

QUESTION 2 – ENSEMBLE [26 MARKS]

This question will use the same dataset as Question 1. Instead of single prediction models, we will create and compare ensemble models in this question.

For this question, split the data such that 80% of the dataset is used for training and the remaining 20% is used for validation based on shuffled sampling with local random seed 1992. Note that there is no test sub-dataset as we will not be using the Optimize Parameters (Grid) operator.

For the performance evaluation of the ensemble model, let's use a cut-off threshold value for the important class of 0.15 (i.e., propensity scores of 0.85 or higher for "Yes" will be classified as "Yes"; otherwise the instance is classified as "No"). Furthermore, let's still assume the same asymmetrical cost for misclassifications as specified in Question 1.

a. (8 Marks) For our first ensemble model, we will use the majority vote over three prediction techniques: Decision Tree, Naïve Bayes and Neural Network (with the parameter values specified in the question). Save your RapidMiner file as "A2_Q2a.rmp" in the processes folder of your Local Repository.

	Validation Performance
confusion matrix	
accuracy	85.16%
sensitivity (or recall)	63.59%
specificity	98.24%

	Unsafe	Safe
Pred. Unsafe	334	75
Pred. Safe	6	131

lift	253.44%
AUC	0.881
misclassification cost	0.203

b. (5 Marks) Other than using different types of prediction algorithms, we can also use the same prediction algorithm multiple times. For this question, we want to create 30 different decision trees (based on the parameter values as specified in Question 2a) where we use a different 65% of the training data for constructing such decision trees (don't forget to use the local random seed value of 1992). The final prediction should be a majority vote over these 30 classifications from each of the decision trees.

Save the file of Question 1a as "A2_Q2b.rmp" in the process folder of your Local Repository. Update the file such that you include this type of ensemble model.

	Validation Performance		
confusion matrix		Unsafe	Safe
	Pred. Unsafe	331	48
	Pred. Safe	9	158
accuracy	89.56%		
sensitivity (or recall)	76.70%		
specificity	97.35%		
lift	250.76%		
AUC	0.906		
misclassification cost	0.187		

c. (5 Marks) In our next ensemble model, we want to create 30 decision trees again (similar to our previous question). However, this time we want to use a random subset of only 65% of the attributes to include in our training process to create each decision tree instead of a random subset of only 65% of the training instances. Don't forget to use the local random seed value of 1992 again. For the attributes selected, we want to make sure to include the best split points for creating branches of our decision tree (i.e., no random split points). The final prediction should be a majority vote over these 30 classifications from each of the decision trees.

Save the file of Question 2b as "A2_Q2c.rmp" in the process folder of your Local Repository. Update the file such that you include this type of ensemble model.

	Validation Performance		
confusion matrix		Unsafe	Safe
	Pred. Unsafe	336	76
	Pred. Safe	4	130

accuracy	85.35%
sensitivity (or recall)	63.11%
specificity	98.82%
lift	257.14%
AUC	0.969
misclassification cost	0.183

d. (5 Marks) Next, we want to create one final decision tree based on 30 different decision trees (based on the parameter values specified in Question 2a) that are constructed in a sequential manner where we update the weights of the training examples based on the prediction and the error rate of the previous decision tree.

Save the file of Question 2b as "A2_Q2d.rmp" in the process folder of your Local Repository. Update the file such that you include this type of ensemble model.

	Validation Performance
confusion matrix	<table><tr><td></td><td>Unsafe</td><td>Safe</td></tr><tr><td>Pred. Unsafe</td><td>332</td><td>55</td></tr><tr><td>Pred. Safe</td><td>8</td><td>151</td></tr></table>
accuracy	88.46%
sensitivity (or recall)	73.30%
specificity	97.65%
lift	251.71%
AUC	0.958
misclassification cost	0.189

e. (3 Marks) Compare the performance of the different ensemble models that you explored in this question. Mention specifically which performance measures and which values you are using for the comparison.

In order to compare the different ensemble models we will primarily be using the accuracy and misclassification cost performance measures. Misclassification cost will be the main metric assessed in our comparison because of its importance within the context of deciding whether the water is safe or not to drink, while accuracy will be used as the supporting comparison metric to further establish which ensemble performed the best.

The ensemble models created in question 2a, question 2b, question 2c, and question 2d each had a misclassification cost of 0.203, 0.187, 0.183, and 0.189, respectively. As can be seen, the model developed in question 2c had the lowest misclassification cost, and therefore performed the best according to this measure. Conversely, the model created in question 2a had the highest misclassification cost, and as a result performed the worse.

Furthermore, the ensemble models from question 2a, question 2b, question 2c, and question 2d each had an accuracy value of 85.16%, 89.56%, 85.35%, and 88.46%, respectively. As is shown, the model built in question 2b performed the best with regard to this metric with an accuracy of 89.56%, while the ensemble model from question 2a performed the worst.

In conclusion, we believe that the best performing model is the ensemble model developed in question 2c, mainly driven by its low misclassification cost. Additionally, although the model from question 2c did not have the highest accuracy, it was relatively close the accuracy value generated by the model from question 2b, which was the best performing model in this category.

QUESTION 3 – SOME CALCULATIONS [30 MARKS]

Consider the same example as from class, where we have 24 training instances that indicate whether a household is an owner or non-owner of a riding lawn mower. However, in this question, there are three predictor variables (see question for the data)

THIS QUESTION DOES NOT REQUIRE RAPID MINER.

a. (6 Marks) Use Naïve Bayes to determine the classification for the following example, where you use a cut-off threshold value of 0.5 to convert the propensity score to an actual classification:

Income	Lot Size	Gated	Ownership
65.2	19.6	No	???

Motivate your answer with calculations where you clearly show your intermediate and final probabilities (not an Excel file).

Q3A) P(Owner) $= \frac{12}{24} = 0.5$

$\left.\begin{array}{l} \mu_{Income} = 79.4750 \quad \delta_{Income} = 18.778814 \\ \mu_{Lsize} = 20.2667 \quad \delta_{Lsize} = 2.020501 \end{array}\right\}$ Owner

$$P(\text{Income} = 65.2 \mid \text{Class} = \text{owner}) = \frac{1}{\sqrt{2\pi\mu}} \cdot e^{(x-\mu)^2/2\delta^2} = \frac{1}{\sqrt{2\pi(79.475)}} \cdot e^{(65.2-79.475)^2/2(18.7788)^2}$$

$$= 0.059742644$$

$$P(\text{Lot size} = 19.6 \mid \text{Class} = \text{owner}) = \frac{1}{\sqrt{2\pi\mu}} \cdot e^{(x-\mu)^2/2\delta^2} = \frac{1}{\sqrt{2\pi(20.26667)}} \cdot e^{\frac{(19.6-20.26667)^2}{2(0.020501)^2}}$$

$$= 0.093575$$

$$P(\text{Gated} = \text{No} \mid \text{Class} = \text{owner}) = \frac{6}{12} = 0.5$$

$$P(\text{owner} \mid X) = (0.5 \times 0.5 \times 0.093575 \times 0.059742644) / P(x)$$
$$= 0.001397566 / P(x)$$

P(Non owner) $= \frac{12}{24} = 0.5$

$\mu_{Income} = 57.4 \quad \delta_{income} = 14.167055$

$\mu_{Lsize} = 17.6\overline{3} \quad \delta_{Lsize} = 2.112875$

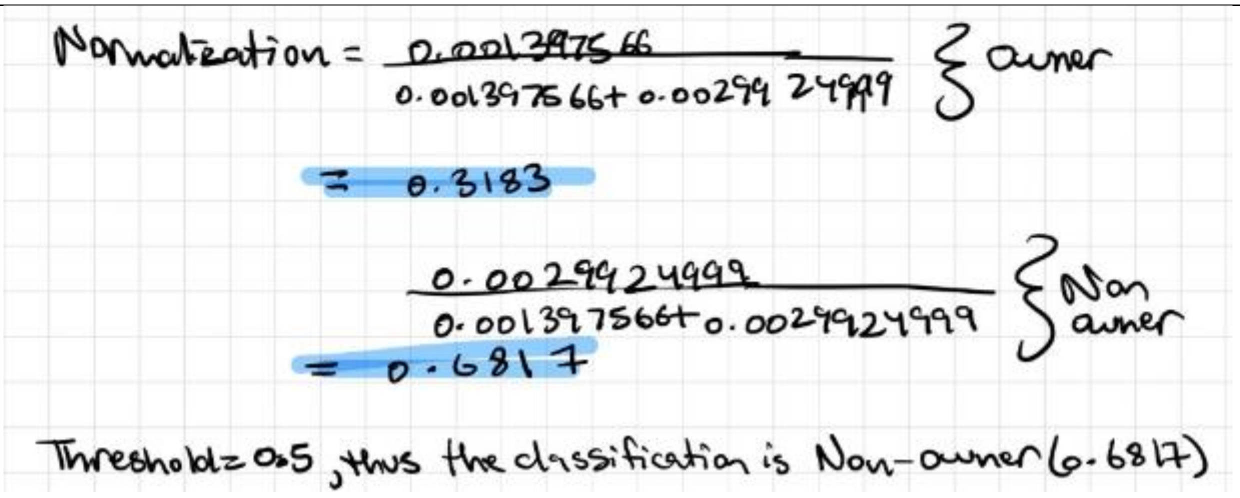
$$P(\text{Income} = 65.2 \mid \text{Class} = \text{non owner}) = \frac{1}{\sqrt{2\pi\mu}} \cdot e^{(x-\mu)^2/2\delta^2}$$
$$= \frac{1}{\sqrt{2\pi\, 57.4}} \cdot e^{(65.2-57.4)^2/2(14.167055)^2}$$
$$= 0.0612742683$$

$$P(\text{Lot size} = 19.6 \mid \text{Class} = \text{owner}) = \frac{1}{\sqrt{2\pi\mu}} \cdot e^{(x-\mu)^2/2\delta^2}$$
$$= \frac{1}{\sqrt{2\pi(17.6\overline{3})}} \cdot e^{(19.6-17.6\overline{3})^2/2(2.112875)^2}$$
$$= 0.1465133371$$

$$P(\text{Gated} = \text{Yes} \mid \text{Non owner}) = \frac{8}{12} = 0.\overline{6}$$

$$P(\text{Non owner} \mid x) = (0.5 \times 0.\overline{6} \times 0.146513371 \times 0.0612742683) / P(x)$$

$$= 0.0029924999 / P(x)$$



$$\text{Normalization} = \frac{0.00139766}{0.00139766 + 0.0029924999} \left\{ \text{owner} \right.$$

$$= 0.3183$$

$$\frac{0.0029924999}{0.00139766 + 0.0029924999} \left\{ \text{Non owner} \right.$$

$$= 0.6817$$

Threshold = 0.5, thus the classification is Non-owner (0.6817)

b. (13 Marks) Use the k-NN classification technique with k = 5 and a non-weighted majority vote to determine your final classification for the same example specified in Question 3a. Only use the two numerical predictor variables for this classification, and don't forget to normalize these two attributes first with the use of a Z Transformation. For the distance measure, use the four distance measures specified in the question.

Motivate your answer with calculations but do so in Excel since you have to calculate the distance between the new example and each of the 24 examples from the training dataset (for each distance measure). Upload your Excel file when submitting the assignment on D2L.

Instance	Euclidean distance	Manhattan distance	Camberra distance	Cosine similarity
1	0.5597	0.7569	12.4454	1.5107
2	1.5432	2.1787	3.3350	2.9181
3	0.8239	0.8438	0.6642	0.3817
4	0.5284	0.6811	0.8436	0.1173
5	1.9815	2.7487	2.1772	1.0039
6	2.2744	2.4332	1.6130	2.0704
7	2.3139	2.9860	4.0354	2.3906
8	1.4561	2.0423	2.2649	1.0207
9	0.2530	0.3567	1.6559	0.6141
10	1.4889	1.8987	1.7836	1.5687
11	1.2213	1.7058	1.3355	0.0632
12	0.8151	0.9630	1.9297	1.5212
13	0.4951	0.4951	2.9474	1.4401
14	0.7979	1.1207	1.1370	0.2551
15	0.9886	1.0086	2.2400	2.3434
16	1.1593	1.4409	1.1536	0.5843
17	1.2572	1.7735	4.3825	2.7348
18	1.1540	1.6320	3.5690	1.5419

19	1.5112	1.7756	2.0377	2.2336
20	0.4958	0.5346	12.1410	2.0951
21	1.5954	2.2171	2.4175	1.8017
22	1.6598	1.9563	2.4326	1.0568
23	2.4152	3.0236	1.9891	2.1852
24	1.9798	2.0879	1.6250	2.4337

	Euclidean distance	Manhattan distance	Camberra distance	Cosine similarity
classification	Owner	Owner	Owner	Owner

c. (6 Marks) If you were to use the Mahalanobis distance for the classification, how would you classify the example from Question 3a? Similar to Question 3b, only include the two numerical predictor attributes. However, you shouldn't normalize the data for this question. Motivate your answer with calculations in Excel. Upload your Excel file when submitting the assignment on D2L. In your report, include the covariance matrix, the inverse of the covariance matrix, and the vector-matrix multiplications as also reported in the lecture notes.

Covariance matrix:

Covariance Matrix (owner)

	Income	Lot size
Income	352.6439	-11.8182
Lot size	-11.8182	4.0824

Covariance Matrix (non-owner)

	Income	Lot size
Income	200.7055	-2.5891
Lot size	-2.5891	4.4642

Inverse of covariance matrix:

Inverse Matrix (owner)

	Income	Lot size
Income	0.0031	0.0091
Lot size	0.0091	0.2713

Inverse Matrix (non-owner)

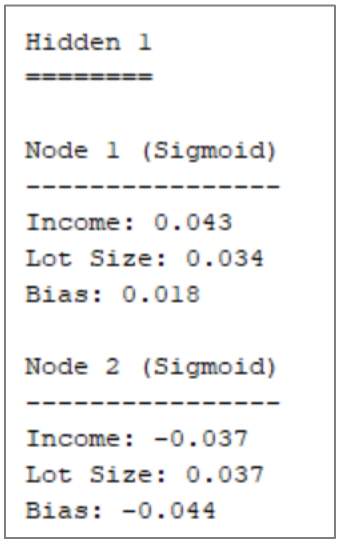
	Income	Lot size
Income	0.0050	0.0029
Lot size	0.0029	0.2257

Vector-matrix multiplications:

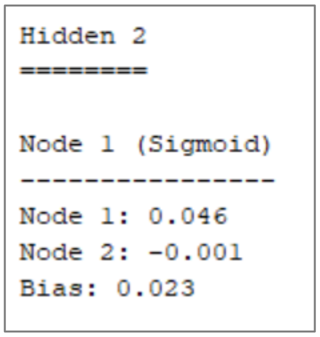
$$\bullet\ \text{Owner} = \sqrt{(-14.2750 \quad -0.6667) \cdot \begin{pmatrix} 0.0031 & 0.0091 \\ 0.0091 & 0.2713 \end{pmatrix} \cdot \begin{pmatrix} -14.2750 \\ -0.6667 \end{pmatrix}} = 0.9662$$

$$\bullet\ \text{Non-Owner} = \sqrt{(7.8000 \quad 1.9667) \cdot \begin{pmatrix} 0.0050 & 0.0029 \\ 0.0029 & 0.2257 \end{pmatrix} \cdot \begin{pmatrix} 7.8000 \\ 1.9667 \end{pmatrix}} = 1.1259$$

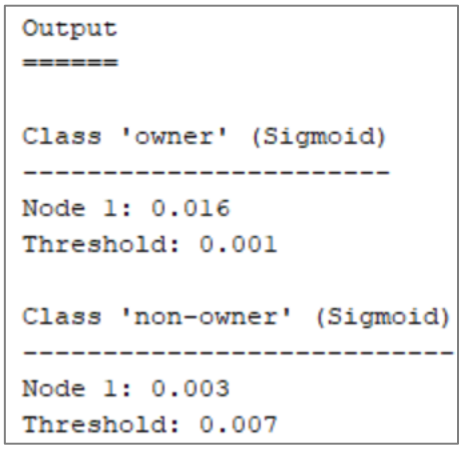
d. (5 Marks) Assume that you create a neural network model with this training set, where the network structure consists of 2 hidden layers (one layer with 2 nodes and the second layer with 1 node). Similar to Question 3b and 3c, only the two numerical predictor attributes are included. After normalizing these two variables with the Range Transformation, let the model coefficients (i.e., the weights and bias) for the model be given by RapidMiner as:



```
Hidden 1
========

Node 1 (Sigmoid)
----------------
Income: 0.043
Lot Size: 0.034
Bias: 0.018

Node 2 (Sigmoid)
----------------
Income: -0.037
Lot Size: 0.037
Bias: -0.044
```



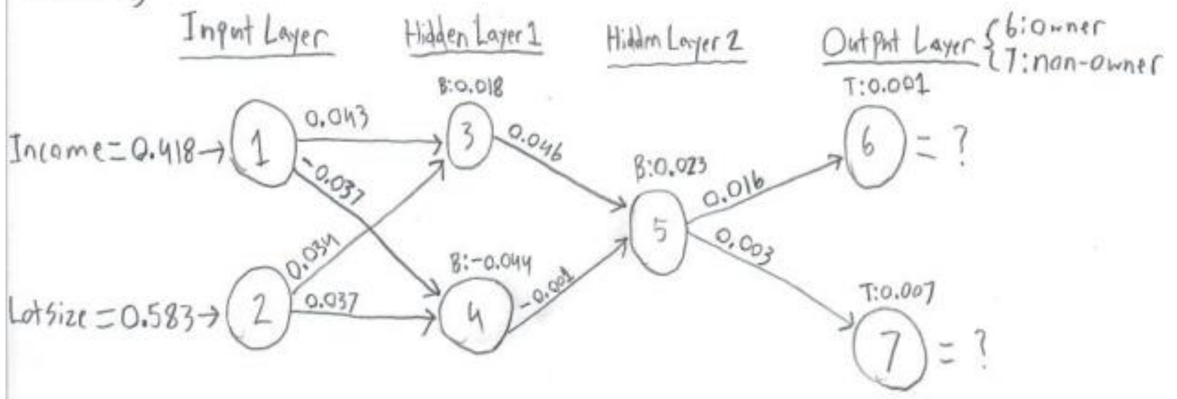
```
Hidden 2
========

Node 1 (Sigmoid)
----------------
Node 1: 0.046
Node 2: -0.001
Bias: 0.023
```



```
Output
======

Class 'owner' (Sigmoid)
-----------------------
Node 1: 0.016
Threshold: 0.001

Class 'non-owner' (Sigmoid)
---------------------------
Node 1: 0.003
Threshold: 0.007
```

Apply this model to the test example from Question 3a, where the values of the income and lot size of this example become 0.418 and 0.583 after applying the Range Transformation. What would be the classification of this example, motivate your answer with calculations (not an Excel file)?

Q3. d)



Logistic (Sigmoid): $g(Z) = 1/(1 + \exp(-Z))$ | $\text{Output}_j = g\left(\theta_j + \sum_{i=1}^{p} W_{ij} \cdot X\right.$

- Hidden Layer 1: Node 1 $= \dfrac{1}{1 + e^{-[0.018 + (0.043 \cdot 0.418) + (0.034 \cdot 0.583)]}} = 0.5139$
- Hidden Layer 1: Node 2 $= \dfrac{1}{1 + e^{-[-0.044 + (-0.037 \cdot 0.418) + (0.037 \cdot 0.583)]}} = 0.4905$
- Hidden Layer 2: Node 1 $= \dfrac{1}{1 + e^{-[0.023 + (0.046 \cdot 0.5139) + (-0.001 \cdot 0.4905)]}} = 0.5115$
- Output Layer: Owner $= \dfrac{1}{1 + e^{-[0.001 + (0.016 \cdot 0.5115)}} = 0.5023$
- Output Layer: Non-Owner $= \dfrac{1}{1 + e^{-[0.007 + (0.003 \cdot 0.5115)}} = 0.5021$

- $0.5023 > 0.5021$

$\therefore$ Owner > Non-owner

$\therefore$ $\boxed{\text{Classification} = \text{Owner}}$ ← Answer