CSC3730: Homework 2

Submission details: Submit the deliverables to the corresponding dropbox at Moodle site with the specified file names.

Problem 1: [20 pt] K nearest neighbor classifier (KNN)

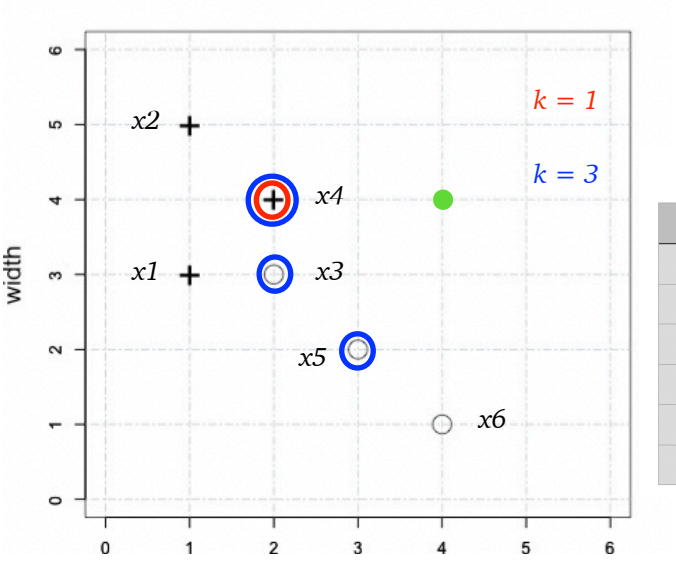
Consider K-NN using Euclidean distance on the following data set (each 2D point belongs to one of two classes: o and +).

a. [4 pt] For a new sample with length 4 and width 4, what is the predicted label using 1-NN and 3-NN, respectively?

o	+
$At\ k = 1, \sqrt{(4-4)^2 + (1-4)^2} = 3$	$\sqrt{(2-4)^2 + (4-4)^2} = 2$
$\sqrt{(3-4)^2 + (2-4)^2} = \sqrt{5} \approx 2.23$	$\sqrt{(1-4)^2 + (3-4)^2} = \sqrt{10} \approx 3.16$
$\sqrt{(4-2)^2 + (4-3)^2} = \sqrt{5} \approx 2.23$	$\sqrt{(1-4)^2 + (5-4)^2} = \sqrt{10} \approx 3.16$

Answer: $At\ k = 1,$ The predicted label will be the plus. The plus that is closer using k = 1 K–NN compared to using the circle. This would give a decision boundary that was similar to a logarithmic function.

Answer: $At\ k = 3,$ The predicted label will be the circle. While the plus is closer by 0.23, there are two circles that are almost as close. Considering that there are more of the circle, the predicted point should be a circle. This would give a decision boundary that is more linear, dividing the two regions of plus and circle.



Distance to Points

x	y	Point	x1	x2	x3	x4	x5	x6
1	3	x1	0.00	2.00	1.00	1.41	2.24	3.61
1	5	x2	2.00	0.00	2.24	1.41	3.61	5.00
2	3	x3	1.00	2.24	0.00	1.00	1.41	2.83
2	4	x4	1.41	1.41	1.00	0.00	2.24	3.61
3	2	x5	2.24	3.61	1.41	2.24	0.00	1.41
4	1	x6	3.61	5.00	2.83	3.61	1.41	0.00

b. [12 pt] Given the 6 samples in the figure, what is Leave-one-out (LOO) cross validation error for 1-NN and 3-NN, respectively? In each fold, test one of the six samples using the others as training data. Please show the error in each fold and then show the cross validation error averaged over six folds.

Fold: 1 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x2	+		
x3	○		
x4	+		
x5	○	✓	

Fold: 2 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x2	+		
x3	○		
x4	+		
x6	○	✓	

Fold: 3 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○	✘	
x2	+		
x3	○		
x5	○		
x6	○		

Fold: 1 Validation

Points		Prediction	Error
x6		○	0

Fold: 2 Validation

Points		Prediction	Error
x5		○	0

Fold: 3 Validation

Points		Prediction	Error
x4		○	1

Fold: 4 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○	✘	
x2	+		
x4	+		
x5	○		
x6	○		

Fold: 5 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x3	○		
x4	+	✓	
x5	○		
x6	○		

Fold: 6 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x2	+		
x3	○	✘	
x4	+		
x5	○		
x6	○		

Fold: 4 Validation

Points		Prediction	Error
x3		+	1

Fold: 5 Validation

Points		Prediction	Error
x2		+	0

Fold: 6 Validation

Points		Prediction	Error
x1		○	0

Fold 1 Error: 3 errors on 6 folds. $3 \div 6 = \frac{1}{2}$ *or* $50\,\%$

Fold: 1 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x2	+		
x3	○		
x4	+		
x5	○	✓	

Fold: 2 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x2	+		
x3	○		
x4	+		
x6	○	✓	

Fold: 3 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○	✘	
x2	+		
x3	○		
x5	○		
x6	○		

Fold: 1 Validation

Points		Prediction	Error
x6		○	0

Fold: 2 Validation

Points		Prediction	Error
x5		○	0

Fold: 3 Validation

Points		Prediction	Error
x4		○	1

Fold: 4 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○	✘	
x2	+		
x4	+		
x5	○		
x6	○		

Fold: 5 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x1	○		
x3	○		
x4	+	✓	
x5	○		
x6	○		

Fold: 6 Train

k = 1		True =0 False = 1	
Points	Type (Truth)	Nearest Neighbor	Error
x2	+		
x3	○	✘	
x4	+		
x5	○		
x6	○		

Fold: 4 Validation

Points		Prediction	Error
x3		+	1

Fold: 5 Validation

Points		Prediction	Error
x2		+	0

Fold: 6 Validation

Points		Prediction	Error
x1		○	0

Fold 3 Error: 1 errors on 6 folds. $1 \div 6 = \frac{1}{6}$ *or* $16.7\,\%$

c. [4 pt] Through cross validation, which K is better for making predictions based on the training data?

Answer: Cross validation $k = 3$ would be the better choice for making predictions based on the training data due to the reduced error of 16.7%.

Problem 2: [10 pt] Decision Tree

We would like to predict TaxFraud (cheat or not) using the input attributes including Refund and Marital status with the following data. Build a simple decision tree with only one layer using information gain as a way of deciding which of the two attributes to use. You will need to compute information gain (use log in base 2). Print out the solution with calculation steps. No programming needed.

No.	Refund	Marital	Cheat
1	Yes	Married	No
2	No	Single	No
3	No	Single	Yes
4	No	Married	No
5	No	Married	Yes

:

Before Splitting: $Entropy(Parent) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.9710$

$Entropy(Refund = Yes) = -0\log_2(0) - 1\log_2(1) = 0$

$Entropy(Refund = No) = -\frac{2}{4}\log_2(\frac{2}{4}) - \frac{2}{4}\log_2(\frac{2}{4}) = 1$

$Split\ on\ Refund = \frac{2}{5} \times 0 + \frac{3}{5} \times 1 = 0.6 \quad Gain = 0.9710 - 0.6 = 0.3710$

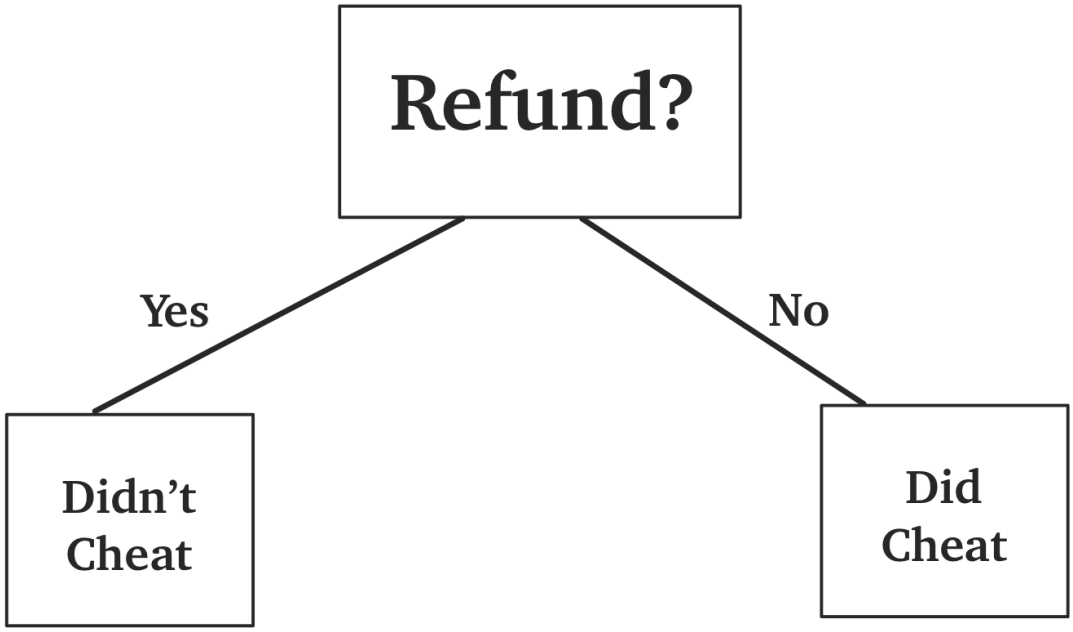
$Entropy(Married = Yes) = \frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) = 0.9183$

$Entropy(Married = No) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{1}\log_2(\frac{2}{4}) = 1$

$Split\ on\ Marriage = \frac{2}{5} \times 0.9183 + \frac{3}{5} \times 1 = 0.9673$

$Gain = 0.9710 - 0.9673 = 0.0037$

Due to the higher gain, the split should be done on whether there was a refund received.


Refund?
Yes
No
Didn't Cheat
Did Cheat

Problem 3: [20 pt] Classifiers using WEKA

In this task you will use several tools from WEKA to analyze additional multiclass data. Download and install WEKA http://www.cs.waikato.ac.nz/ml/weka/ Read the instruction. https:/www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka. pdf. The dataset hw2_pen_data.arff examines several features extracted from a penbased handwriting study of 10992 instances. Each observation has 16 features extracted from writing a single handwritten digit and the known labels (class 0,1,2,3,4,5,6,7,8,9). We want you to classify the results based on the 10 classes in the file.

Load the data into the WEKA Explorer (or double click the .arff file if you installed WEKA). Click the classify tab at the top. Under classifier you can select the different classifiers that WEKA offers. You can adjust the input parameters to many of the models by clicking on the text to the right of the Choose button in the Classifier section. You should choose **10-fold Cross validation** as the test option.

Report the overall prediction accuracy of each of the following methods:

1. KNN: Under the classifiers>Lazy select IBk.

2. Decision Tree: Under the classifiers>trees select J48.

3. Logistic Regression: Under the classifiers>functions folder select Logistic. Write your answers to the following questions:

   a. [11pt] What is the prediction accuracy for each of the three models? Which of the 3 models performed the best (i.e., highest accuracy)?

KNN	99.3632%
Decision Tree	96.5611%
Logistic Regression:	95.5513%

b. [9pt] The predication accuracy may vary if you change the values of certain model parameters. For each classifier, briefly explain what parameters you change, how prediction accuracy changes, and why you think it improved prediction accuracy.

Decision Tree with 40% Train	94.4049%
Decision Tree with 60% Train	95.7471%
Decision Tree with 80% Train	96.4058%
Decision Tree with 88% Train	97.4223%
Decision Tree with 95% Train	97.4545%
Decision Tree with 99.3% Train	100%
Decision Tree with 25 Fold CV	96.7431%
Decision Tree with 42 Fold CV	96.7522%
Decision Tree with 70 Fold CV	96.7886%
Decision Tree with 100 Fold CV	96.7704%
Decision Tree with 200 Fold CV	96.7704%
Decision Tree with 10500 Fold CV	96.7522%
Logistic Regression with 40% Train	95.4359%
Logistic Regression with 60% Train	95.929%
Logistic Regression with 80% Train	95.9509%
Logistic Regression with 88% Train	96.4367%
Logistic Regression with 95% Train	95.4545%
Logistic Regression with 25 Fold CV	95.4545%
Logistic Regression with 42 Fold CV	95.4785%
Logistic Regression with 70 Fold CV	95.4694%
Logistic Regression with 100 Fold CV	95.524%
Logistic Regression with 200 Fold CV	95.5149%
KNN with 40% Train	99.0144%
KNN with 60% Train	99.1813%
KNN with 80% Train	99.3631%
KNN with 88% Train	99.3935%
KNN with 95% Train	99.0909%
KNN with 96% Train	98.8636%
KNN with 25 Fold CV	99.3723%
KNN with 42 Fold CV	99.3541%
KNN with 70 Fold CV	99.3723%
KNN with 100 Fold CV	99.3814%
KNN with 200 Fold CV	99.3814%
KNN with 2000 Fold CV	99.3723%

For each model I worked to perform similar tests so that there would be consistency among the comparisons. Changes to this consistency was with the KNN model I performed a 2000 fold CV, with the Decision Tree I performed a 10500 fold CV, and with the percentage of the data set that was used to train the numbers may vary to represent the optimized percentage for model accuracy.

With the Decision tree I found that the optimized training percentage was with 99.3%. With it being a decision tree it has questions and outcomes developed in the model from the data. This was able to give 100% accuracy. With the CV folds though the highest amount that was attained with accuracy was at 70 folds and 98.7886%. I found that the decision tree was quick, not the quickest, but enough that I would attempt a 10500 fold CV.

With the Logistic Regression I found that an 88% training set would produce a 96.4367% accuracy and a 100 fold CV would produce an 95.524% accuracy. Linear Regression was the slowest validation of the three models.

The KNN model was also one of the quickest of the models to validate. The KNN model had optimal validation at 88% training set and at 100 to 200 folds CV. Going in between the 100 to 200 folds, I found that the outcomes of accuracy were similar to each other.

By increasing the folds this allowed the models to improve the predictive training across the model. As the percentage of training set went up, the prediction value improved as well across the models. While the optimum for the Decision Tree was at 99.3% training, which is the equivalent of reading the entire book of data, the Logistic Regression and the KNN models were optimized at 88% of the training data used. This shows that for the Logistic Regression there is a point that is “good enough” in the training that produces the highest accuracy. By using more training the model was able to have better predictions.

Deliverables: A text file named “HW2.pdf” containing the answers of the questions 1 to 3.