# COMP 4450 – Introduction to Data Mining

# Homework# 3

# Possible Points: 30

The assignment is due at the beginning of class on **February 27, 2019**. You will need to type your answers using a word processor. You may not draw any picture by hand. This is an individual assignment and you MUST turn in your own work. The only person you may consult with is your instructor.

1. **(5 pts)** Consider the following data: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
2. Smooth the data by bin means, using a bin depth of 3. Show your steps.
3. Use min-max normalization to transform the value 35 onto the range [0.0,1.0].
4. Use z-score normalization to transform the value 35.
5. Use normalization by decimal scaling to transform the value 35.
6. Which of the above three normalization methods you would prefer to use for the given data? Justify your answer.
7. **(5 pts)** Consider a two-category classification task with the following training data where the instances are described by two real-valued attributes:

|  |  |  |
| --- | --- | --- |
| A1 | A2 | Class |
| 1 | 5 | A |
| 2 | 6 | A |
| 2 | 7 | B |
| 3 | 7 | A |
| 3 | 8 | B |
| 4 | 8 | A |
| 5 | 9 | A |
| 5 | 1 | B |
| 6 | 2 | B |
| 7 | 2 | A |
| 7 | 3 | B |
| 8 | 3 | A |
| 8 | 4 | B |
| 9 | 5 | B |

1. **(2 pts)** Classify the instance (6,1) using a 3-nearest neighbor algorithm. Use the Euclidean distance. Show your work.
2. **(3 pts)** Use Weka to find a value of *k* that minimizes leave-one-out cross-validation error. Show the cross-validation errors for each value of *k*. Vary *k* between 1 and 10. You will need to create a Weka formatted input file (.arff) with the given training data.
3. **(16 pts)** Consider the following credit worthiness dataset where the last two columns contain two possible class assignments.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instance# | Debt | Income | Married | Credit1 | Credit2 |
| X1 | low | Low | yes | good | good |
| X2 | low | medium | yes | good | good |
| X3 | low | High | yes | good | bad |
| X4 | medium | Low | no | bad | bad |
| X5 | medium | medium | yes | good | good |
| X6 | medium | medium | no | bad | bad |
| X7 | medium | High | no | good | good |
| X8 | high | Low | no | bad | good |
| X9 | high | Low | yes | bad | good |
| X10 | high | medium | yes | bad | bad |
| X11 | high | medium | no | bad | bad |
| X12 | high | High | no | bad | bad |

1. **(2 pts)** Assume that *Credit*1 is the true class assignment and *Credit*2 is the calculated class assignment. Compute the true positive, false positive, false negative, true negative, and accuracy.
2. **(8 pts)** Construct a complete decision tree (ID3) using *Credit*1 as the class assignment. Show all calculations.
3. **(2 pts)** Using the DT you generated in (b), determine the credit worthiness of the following individuals:

An unmarried person.

A married person who has low debt and high income.

A person who has high debt and low income.

A unmarried person who has medium debt and medium income.

A person who has medium income.

A person who has low debt and medium income.

1. **(4 pts)** Use Weka to construct a decision tree using *Credit*1 as the class assignment. Copy and paste the tree from the Weka output window and draw it in the form of a tree. Compare this decision tree with the one you generated in (b). There are several tree classification algorithms available in Weka -- use the ID3 algorithm. Note that you will need to create a Weka formatted input file (.arff).
2. **(4 pts)** Consider the following contact lenses dataset where the last two columns contain two possible class assignments. Assume that *Lenses*1 is the true class assignment and *Lenses*2 is the calculated class assignment. Construct the confusion matrix.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Age | Prescription | Astigmatism | TearProdRate | Lenses1 | Lenses2 |
| young | myope | no | reduced | none | none |
| young | myope | no | normal | soft | soft |
| young | myope | yes | reduced | none | soft |
| young | hypermetrope | no | reduced | none | none |
| young | hypermetrope | no | normal | soft | none |
| young | hypermetrope | yes | reduced | none | none |
| pre-presbyopic | myope | no | reduced | none | none |
| pre-presbyopic | myope | yes | reduced | none | soft |
| pre-presbyopic | myope | yes | normal | hard | soft |
| pre-presbyopic | hypermetrope | no | reduced | none | none |
| pre-presbyopic | hypermetrope | yes | reduced | none | none |
| pre-presbyopic | hypermetrope | yes | normal | none | none |
| presbyopic | myope | no | reduced | none | none |
| presbyopic | myope | no | normal | none | hard |
| presbyopic | myope | yes | reduced | none | none |
| presbyopic | myope | yes | normal | hard | soft |
| presbyopic | hypermetrope | no | reduced | none | none |
| presbyopic | hypermetrope | no | normal | soft | hard |
| presbyopic | hypermetrope | yes | reduced | none | none |
| presbyopic | hypermetrope | yes | normal | none | none |