**CSCI 6951 Data Science and Machine Learning**

**Decision Trees**

For this assignment we will work with the GoSki data set that contains 4 nominal attributes: snow, weather, season and physical condition, one class go skiing and 14 instances. The goal is to use the entropy and information gain measures as well as the Weka algorithms to build decision trees models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **snow** | **weather** | **season** | **physical condition** | **go skiing** |
| sticky | foggy | low | rested | **no** |
| fresh | sunny | low | injured | **no** |
| fresh | sunny | low | rested | **yes** |
| fresh | sunny | high | rested | **yes** |
| fresh | sunny | mid | rested | **yes** |
| frosted | windy | high | tired | **no** |
| sticky | sunny | low | rested | **yes** |
| frosted | foggy | mid | rested | **no** |
| fresh | windy | low | rested | **yes** |
| fresh | windy | low | rested | **yes** |
| fresh | foggy | low | rested | **yes** |
| fresh | foggy | low | rested | **yes** |
| sticky | sunny | mid | rested | **yes** |
| frosted | foggy | low | injured | **no** |

**Question 1.** In order to build a decision tree based on the data above, first we have to select one attribute for the root node by calculating the information gain for each possible attribute, as shown in the lecture. Please hand in the resulting tree including all the calculation steps.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| snow | | | weather | | | season | | |
|  | yes | no |  | yes | no |  | yes | no |
| fresh | 7 | 1 | sunny | 5 | 1 | low | 6 | 3 |
| sticky | 2 | 1 | foggy | 2 | 3 | mid | 2 | 1 |
| frosted | 0 | 3 | windy | 2 | 1 | high | 1 | 1 |
| fresh | 7/9 | 1/5 | sunny | 5/9 | 1/5 | low | 6/9 | 3/5 |
| sticky | 2/9 | 1/5 | foggy | 2/9 | 3/5 | mid | 2/9 | 1/5 |
| frosted | 0/9 | 3/5 | windy | 2/9 | 1/5 | high | 1/9 | 1/5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| physical condition | | | go | |
|  | yes | no | yes | no |
| rested | 9 | 2 | 9 | 5 |
| tired | 0 | 1 |  |  |
| injured | 0 | 2 |  |  |
| rested | 9/9 | 2/5 | 9/14 | 5/14 |
| tired | 0/9 | 1/5 |  |  |
| injured | 0/9 | 2/5 |  |  |

* Snow
  + Fresh
    - Info([7, 1]) = entropy(7/8,1/8)= -(7/8)log(7/8)-(1/8)log(1/8)=0.163
  + Sticky
    - Info([2, 1]) = entropy(2/3,1/3)=-(2/3)log(2/3)-(1/3)log(1/3)=0.276
  + Frosted
    - Info([0, 3]) = entropy(0/3,3/3)=0
  + Info([7, 1], [2, 1], [0, 3]) = (8/14)\*0.163+(3/14)\*0.276=0.152
* Weather
  + Sunny
    - Info([5, 1]) = entropy(5/6,1/6)=-(5/6)log(5/6)-(1/6)log(1/6)=0.196
  + Foggy
    - Info([2, 3]) = entropy(2/5,3/5)=-(2/5)log(2/5)-(3/5)log(3/5)=0.292
  + Windy
    - Info([2, 1]) = entropy(2/3,1/3)=-(2/3)log(2/3)-(1/3)log(1/3)=0.276
  + Info([5, 1], [2, 3], [2, 1]) = (6/14)\*0.196+(5/14)\*0.292+(3/14)\*0.276=0.247
* Season
  + Low
    - Info([6, 3]) = entropy(6/9,3/9)=-(6/9)log(6/9)-(3/9)log(3/9)=0.276
  + Mid
    - Info([2, 1]) = entropy(2/3,1/3)=-(2/3)log(2/3)-(1/3)log(1/3)=0.276
  + High
    - Info([1, 1]) = entropy(½,½)=-(½)log(½)-(½)log(½)=0.301
  + Info([6, 3], [2, 1], [1, 1]) = -(9/14)\*0.276-(3/14)\*0.276-(2/14)\*0.301=0.280
* Physical Condition
  + Rested
    - Info([9, 2]) = entropy(9/11,2/11)=-(9/11)log(9/11)-(2/11)log(2/11)=0.206
  + Tired
    - Info([0, 1]) = entropy(0/1,1/1)=0
  + Injured
    - Info([0, 2]) = entropy(0/2,2/2)=0
  + Info([9, 2], [0, 1], [0, 2]) = (11/14)\*0.206=0.162
* Go
  + Info([9,5 ]) = entropy(9/14,5/14)=-(9/14)log(9/14)-(5/14)log(5/14)=0.283
* Gain(snow) = 1-0.152=0.848
* Gain(weather) = 1-0.247=0.753
* Gain(season) = 1-0.280=0.72
* Gain(physical condition) = 1-0.162=0.838

**The attribute selected as the splitting attribute at the root of the tree is: ?????**

Snow

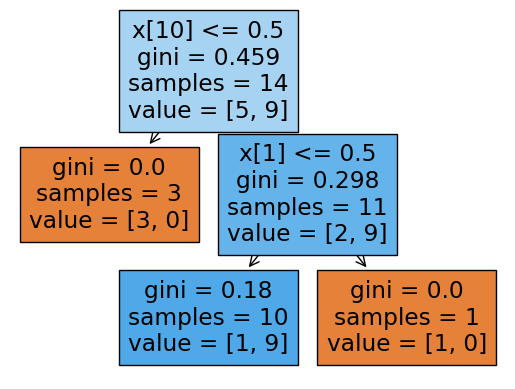
**Question 2.** Draw the tree stump for the goski data. Specify the number of “yes” and “no” in each leaf node.

**Question 3.** Using a Jupyter Notebook and sklearn run majority voting algorithm, KNN (k=3), decision tree algorithm and Naïve Bayes using the training set as the test option and record Correctly Classified Instances, Incorrectly Classified Instances, Accuracy%, Kappa Statistics, a, b (from the confusion matrix), Time taken to build model, Number of leaf nodes. Which algorithm performs better

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Correctly CI | Incorrectly  CI | Accuracy% | Kappa Statistics | a | b | c | d | Time | Number of leaves |
| MV | 4 | 0 | 100.0 | 1.0 | 2 | 0 | 0 | 2 | 0.011 | 5 |
| KNN | 4 | 1 | 75 | 0.5 | 1 | 1 | 0 | 2 | 0.005 |  |
| DT | 4 | 0 | 100.0 | 1.0 | 2 | 0 | 0 | 2 | 0.002 | 5 |
| NB | 4 | 0 | 100.0 | 1.0 | 2 | 0 | 0 | 2 | 0.008 |  |

Decision Tree performs better

**Question 4.** Plot the decision tree generated by the CART algorithm.



**Question 5.** Classify the following test instances using first the majority voting algorithm, KNN (k=3), decision tree algorithm and Naïve Bayes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **snow** | **weather** | **season** | **physical condition** | **go skiing** | **go skiing- MV** |
| sticky | foggy | low | rested | **no** | **no** |
| fresh | sunny | low | injured | **no** | **yes** |
| frosted | sunny | low | rested | **yes** | **yes** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **snow** | **weather** | **season** | **physical condition** | **go skiing** | **go skiing – KNN** |
| sticky | foggy | low | rested | **no** | **yes** |
| fresh | sunny | low | injured | **no** | **yes** |
| fresh | sunny | low | rested | **yes** | **yes** |