COMP 462 – MACHINE LEARNİNG

ID3 ALGORTIHM

&

DECISION TREE

Doğa Poyraz TAHAN

041503044

Introduction

Goal:

Automatically deciding on categorical result when inputs are categorical

Answer:

```
OUTLOOK=overcast-> OUTPUT = yes
OUTLOOK=sunny->HUMIDITY=high-> OUTPUT = no
OUTLOOK=sunny->HUMIDITY=normal-> OUTPUT = yes
OUTLOOK=rain->WIND=strong-> OUTPUT = no
OUTLOOK=rain->WIND=weak-> OUTPUT = yes
```

Console output gives the order of decision tree's possibility lines.

One should read the solution in the console in these steps:

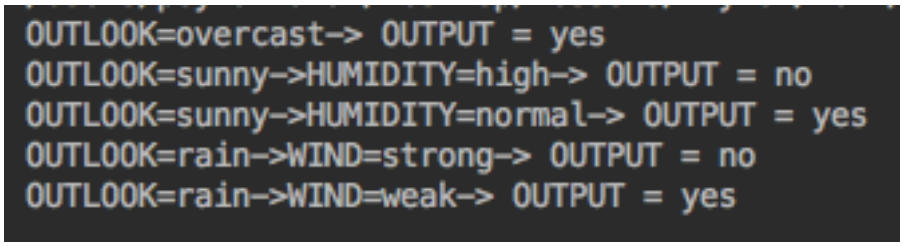
1. Look at i th feature name
2. Select the rows that have the same attributes
3. Go to step 1 untill you see a output.

How to run the code?

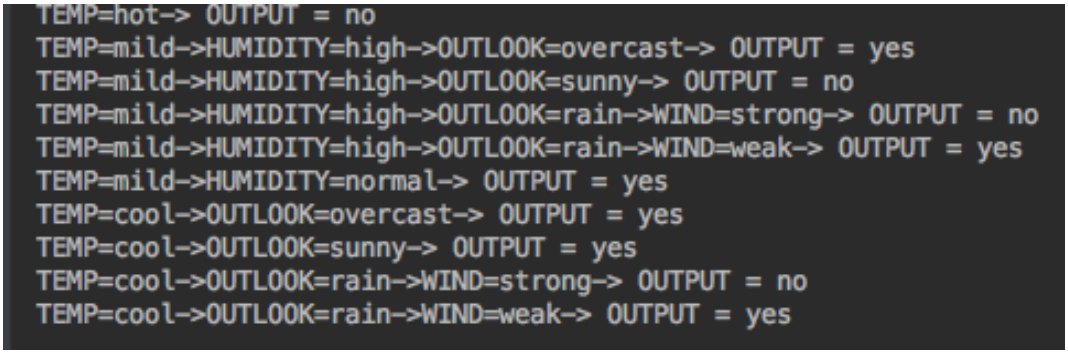
- Open the folder with the src in a pyhton ide
- Change the data set according to your wish
  - How to change is in the comments
- Run with the start button
- Output will be on the Console
- Screen will be created as well.

Solutions

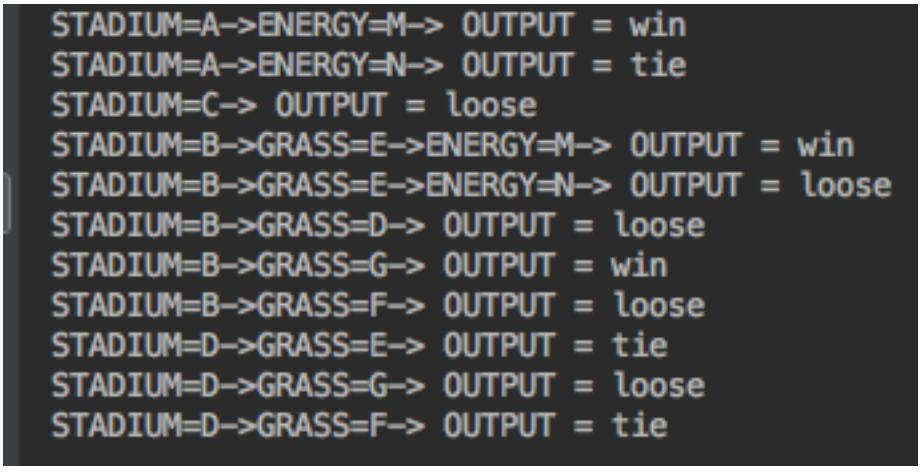
Output For Data Set 1:



Output For Data Set 2:

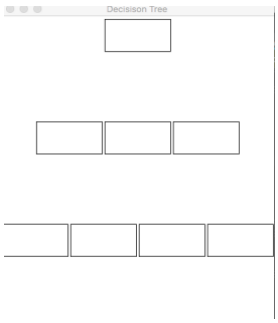


Output For Data Set 3:

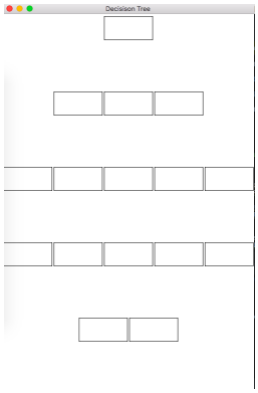


Graph Of Decision Tree

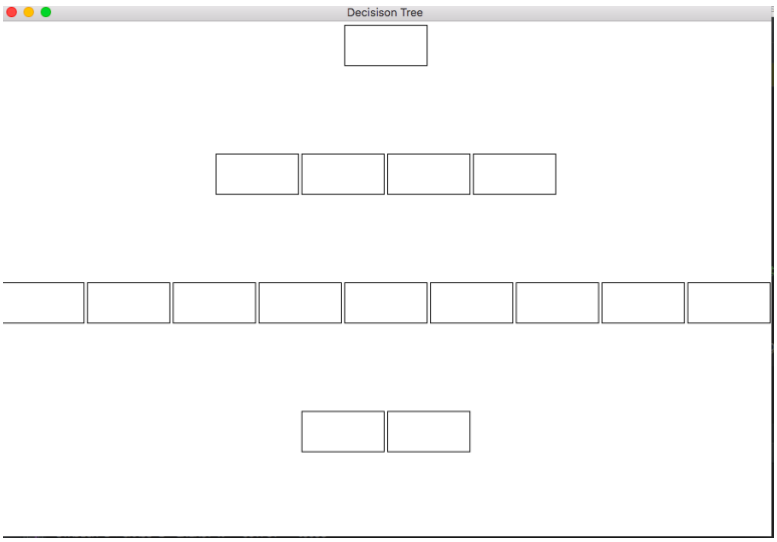
Data Set 1:



Data Set 2:



Data Set 3:



In this project, we broadly aimed at achieving two objectives.

1. First, it is requested that we create a decision mechanism by using id3 Algorithm:

   ID3 Algorithm: [1]

   - Calculate the entropy of every attribute using the data set
   - Split the set into subsets using the attribute for which the resulting entropy (after splitting) is minimum (or, equivalently, information gain is maximum)
   - Decide tree node containing that attribute
   - Recurses on subsets using remaining attributes.

   For this used the these functions:

   - constructDecisionTree(data, labels)

     Mathematical Formulas

     - gain(parentEnt, entropyList, rowCount)
     - entropy(table)

     Utility functions

     - majority(classList)
     - getAttributes(feature, dataSetTable)
     - getDataSetTable(path)
     - split(data, axis, val)
     - getBestFeature(data)

2. Second, it is requested that we output a result that is visiaul as posibble
   I. Display Decision steps for each leaf

      For this used the these functions:

      - displayTreePathWay(node=None)

   II. Graphically show your Decision Tree

      For this used the these functions:

      -

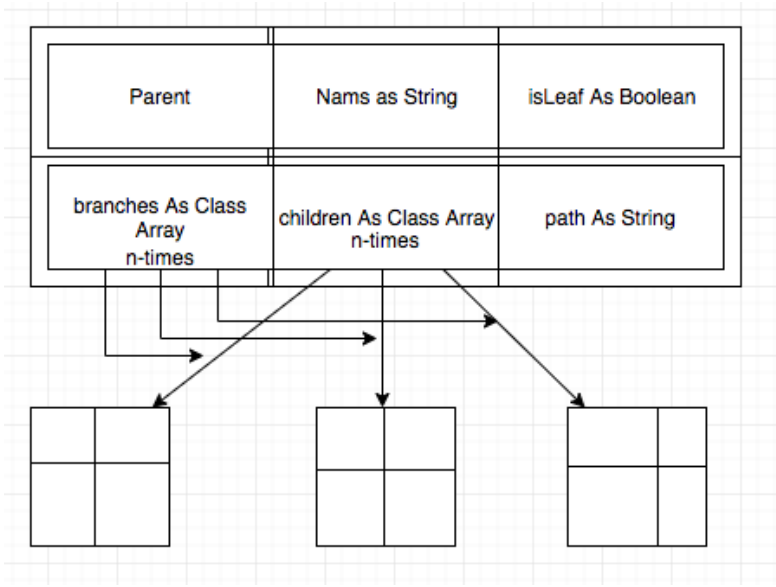
   In addition, testing our modules by applying them to at least three examples. Measure the running time of each algorithm and discuss whether it is consistent with the theoretical analysis of their complexity. Algorithm takes the most time with string comparison. Yet it was fast with the dataset we have tested.

Description and Implementation of Algorithms

1. Node Class

a.) Tree representation:

– A linked data structure in which each node is an object



b.) Node representation:

– name
– isLeaf
– branches
– children
– path

1.1) Implementations of ID3 Algorithms

1.1. a) Entropy Calculation:

— $-\sum_i Pi * log2Pi$

1.1.b) Information Gain Calculation:

— entropy of parent class - weigted average of entropy of client classes

1.1.c) Find Max Info. Gain:

— The feature which provide highest information gain will be choosen

1.1.d) Subset Your Example Data:

— Select rows with the branch names

— Take out the feature that you have divide on

1.1.e) Recurse with the Smaller data:

— Finish recursion when only one example left

—Finish recursion when one feature is left

—Finish recursion only one type of output is left

BIBLOGRAFY

- **[1]:** Grzymala-Busse, Jerzy W. "Selected Algorithms of Machine Learning from Examples." *Fundamenta Informaticae* 18, (1993): 193–207.