

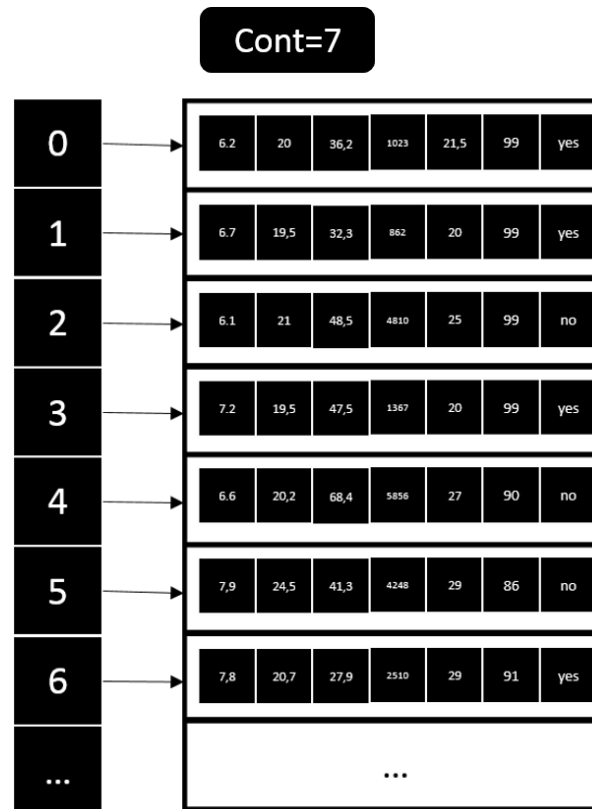
DETECTION OF THE COFFEE LEAF RUST USING DECISION TREES

Andrés Grimaldos Echavarría

Alejandro Salazar Arango

Medellín, October 29

Data Structure Designed



Graph 1: Python dictionary (hash table) of data. Each one has ph, soil temperature, soil moisture, illuminance,, env. temperature, env. humidity and label

Data Structure Operations

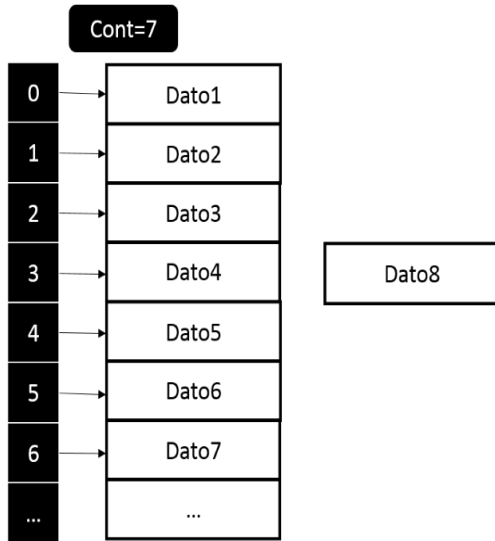


Figure 1: adding

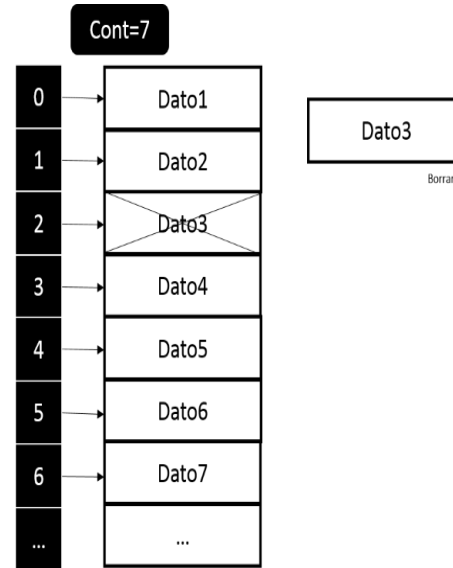
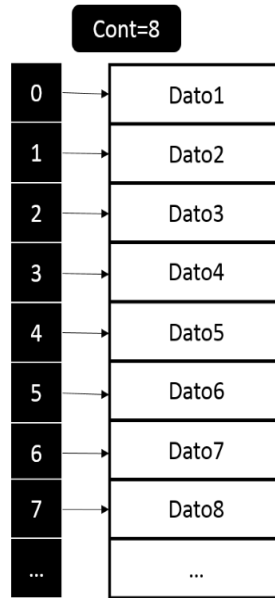
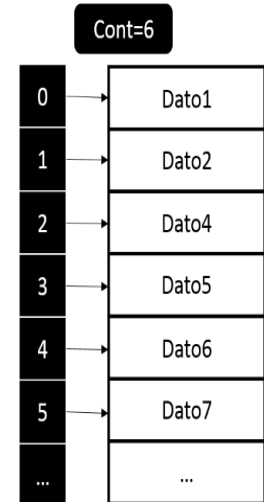


Figure 2: deleting



Data Structure Operations

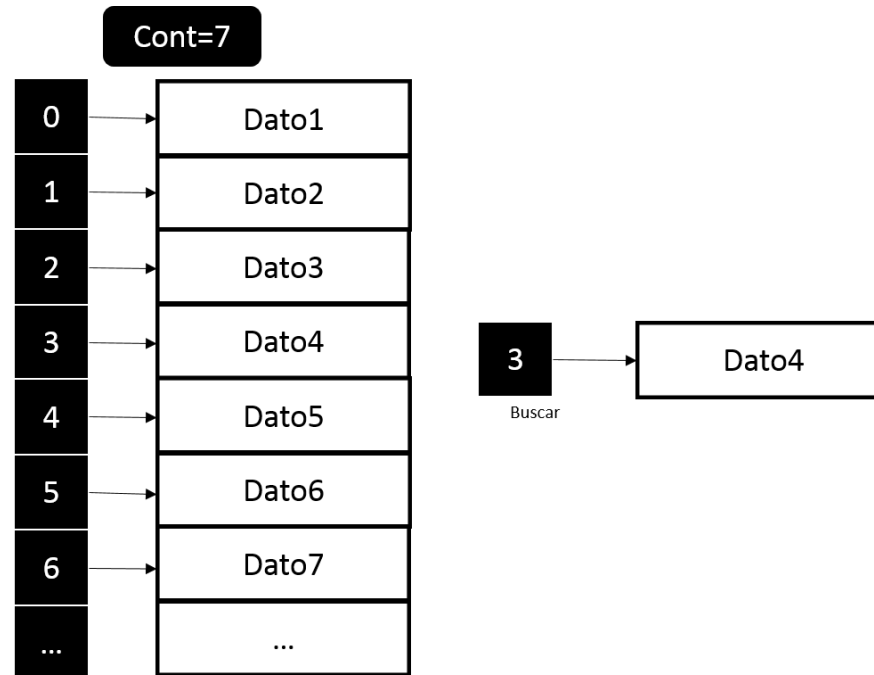


Figure 3: searching

Decision tree

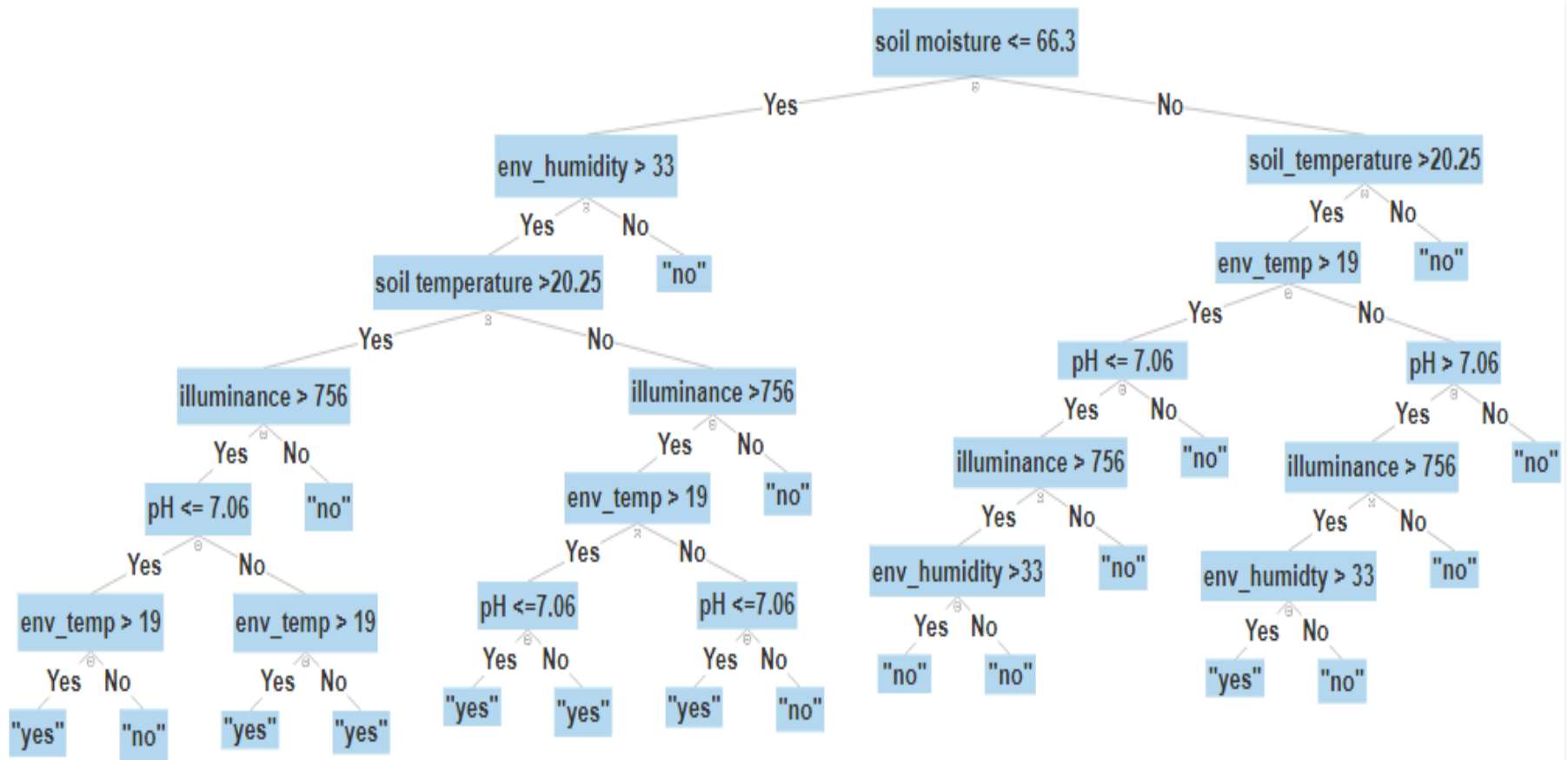


Figure 4: decision tree

Design Criteria of the Data Structure

- Complexity of the operations.
- Easy to use.
- Allows to create a decision tree for predictions.
- Easy to understand and to implement.

Time and Memory Consumption

	Dataset1 (data_set_balanced.csv)
Data Structure storage	0.284019 MB

Table 1: Memory usage

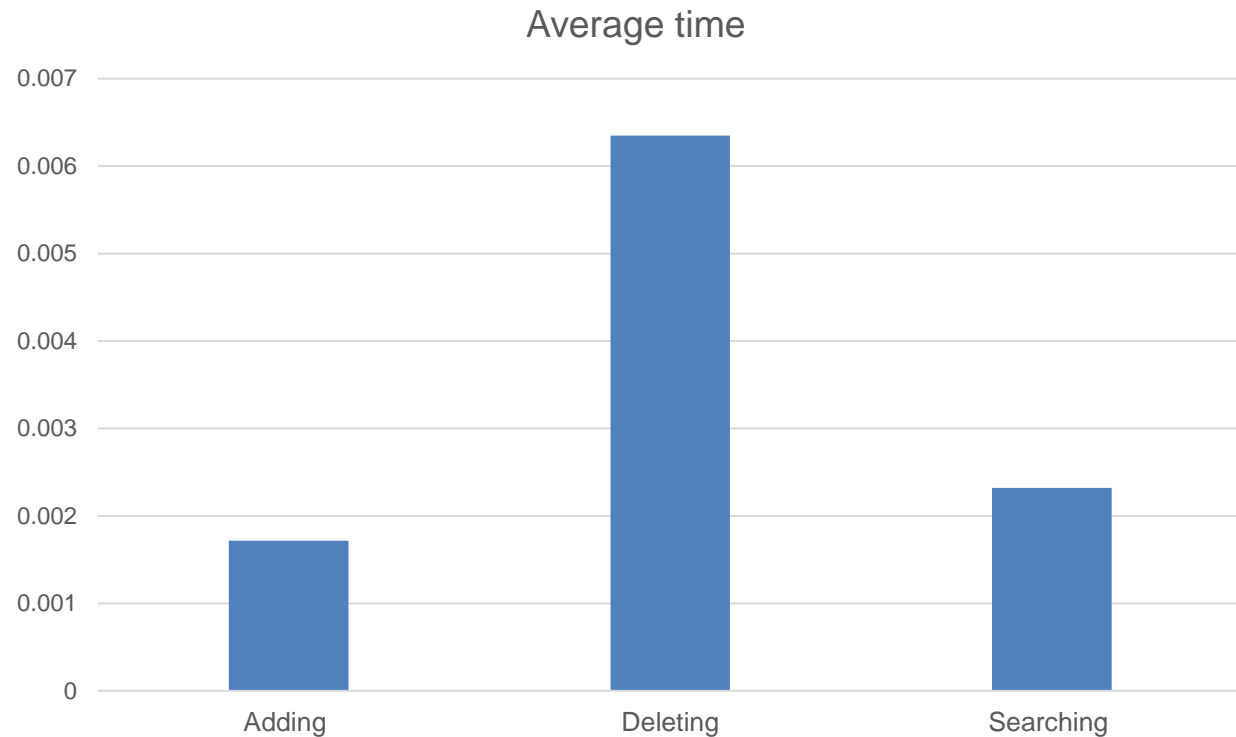
	Best time	Worst Time	Average Time
CART	12.8859 9 s	14.18105 s	13.0192 s

Table 2: CART algorithm times

Time and Memory Consumption

Operation	Average Time
Adding	0,00171649 ms
Searching	0,00231917 ms
Deleting	0,00634765625 ms

Table 3: operations times



Graph 2: operations times

Implementation

```
main.py  saved
32     return self.tb[value]
33     def begin(self,st):
34         datos = pd.read_csv(st)
35         n = 0
36         while True:
37             try:
38                 ph = datos.at[n,'ph']
39                 soilT = datos.at[n,'soil_temperature']
40                 soilM = datos.at[n,'soil_moisture']
41                 light = datos.at[n,'illuminance']
42                 envT = datos.at[n,'env_temperature']
43                 envH = datos.at[n,'env_humidity']
44                 label = datos.at[n,'label']
45                 dt=node(ph,soilT,soilM,light,envT,envH,label)
46                 self.insert(dt)
47                 n = n + 1
48             except:
49                 break
50         self.model=chef.fit(datos,{"algorithm":"CART"})
51
52     dtFram=table()
53     dtFram.begin("data_set_balanced.csv")
54     print(dtFram.buscar(0).ph)
55     print(dtFram.ct)
56     dtFram.delete(425)
57     print(dtFram.ct)
58     d = node(7.41,20, 22.1, 1609, 23, 80, "")
59     dtFram.insert(d)
60     print(dtFram.ct)
61
```

```
https://AchingSourBoard.andresgrimaldos.repl.run
CART tree is going to be built...
Accuracy: 72.14912280701755 % on 456 instances
finished in 13.063640832901001 seconds
6.44
456
455
yes
456
➤
```

Figure 4: Implementation example