Coffee and rust: Detection and prevention for improving exportation quality

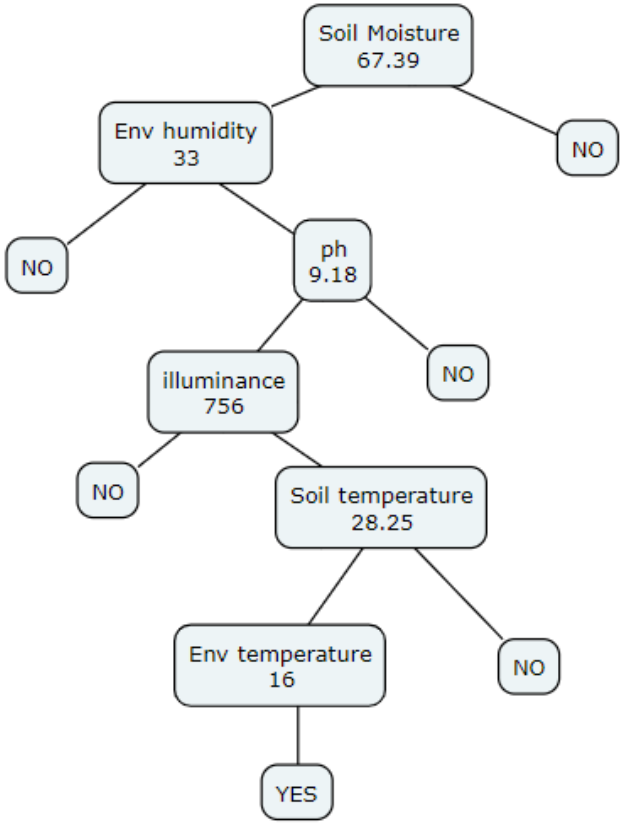
Isabella Quintero Villegas
Sofia Vega Lopez

Medellín, October 29th 2019

Designed Data Structure

ph	soil_temperature	soil_moisture	illuminance	env_temperature	env_humidity	label
6.44	21.0	65.22	1431.0	19.0	99.0	yes
6.23	27.0	19.2	1204.0	36.0	42.0	yes
7.53	24.5	48.55	3303.0	26.0	87.0	yes
7.33	24.5	32.97	5437.0	26.0	79.0	yes
7.07	22.25	49.28	3270.0	25.0	99.0	yes
6.9	27.0	50.36	2154.0	30.0	43.0	yes
6.49	20.75	52.9	1429.0	19.0	99.0	yes
6.52	21.5	52.9	5005.0	28.0	92.0	yes
6.51	25.5	21.01	4872.0	35.0	51.0	yes
6.54	26.25	23.55	1275.0	27.0	85.0	yes

Graphic 1: The data is uploaded to the program as a matrix, in which rows are number of data and columns are each characteristic



Graphic 2: Decision tree built after training

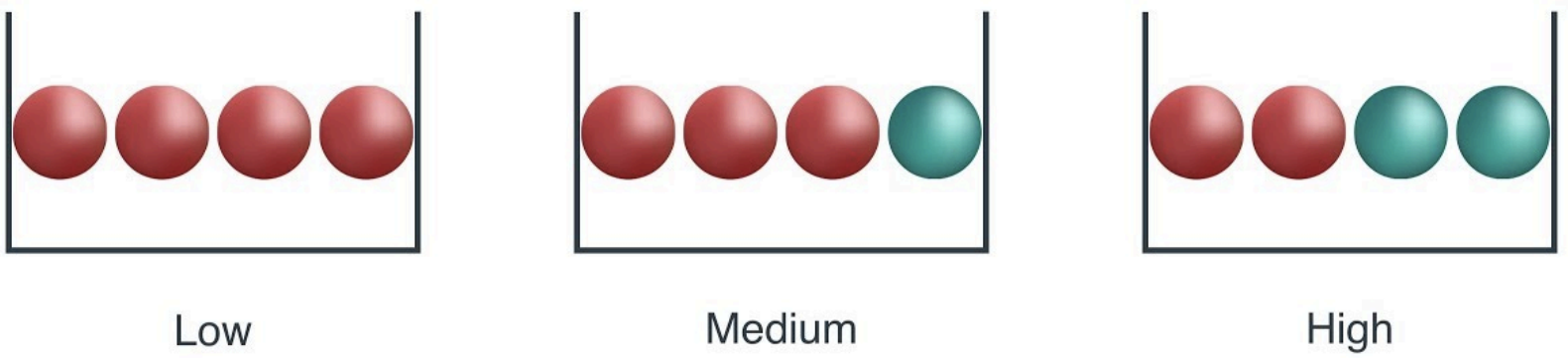
```
1  def findDecision(obj): #obj[0]: ph, ol
2      if obj[2]<=67.39:
3          if obj[5]>33:
4              if obj[0]<=9.18:
5                  if obj[3]>756:
6                      if obj[1]<=28.25:
7                          if obj[4]>16:
8                              return 'yes'
9                      elif obj[1]>28.25:
10                         return 'no'
11                 elif obj[3]<=756:
12                     return 'no'
13             elif obj[0]>9.18:
14                 return 'no'
15         elif obj[5]<=33:
16             return 'no'
17     elif obj[2]>67.39:
18         return 'no'
19
```

Graphic 3: Base for building the tree

Data Structure Operations

calculateEntropy

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

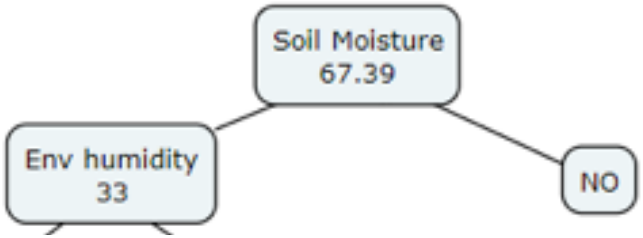


findDecision

- ***Information gain***

$$\text{Information gain} = \text{entropy (parent)} - [\text{weightes average}] * \text{entropy (children)}$$

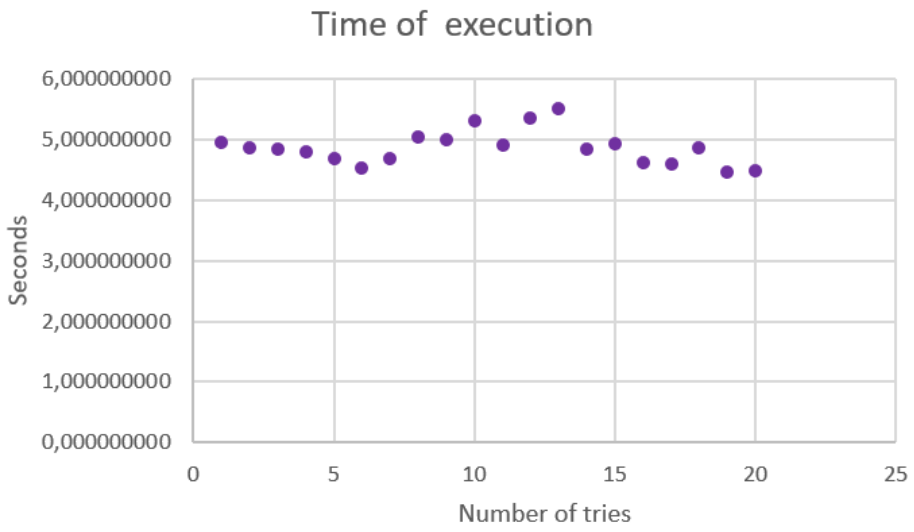
- ***Split information***



Design Criteria of the Data Structure

- Better data organization
- Higher Accuracy
- Easier Access
- Consistent Results

Time and Memory Consumption



Memory Used

Memory consumption	**140,9MB**

This is subject to the amount of features in the data

Complexity

Operation	Complexity
calculateEntropy	$O(n * m)$
findDecision	$O(n^2 * m^3)$
BuildDecisionTree	$O(n^3 * m^3)$

Table to report complexity analysis

- n is the number of columns
- m is the number of values in each columns

Implementation

```
import Chefboost as chef
import pandas as pd

archivo = input("INSERT FILE NAMED FOLLOWED BY .CSV:\n")
# READ THE DATA SET FROM THE CSV FILE
df = pd.read_csv(str(archivo))
df.columns = ['ph', 'soil_temperature', 'soil_moisture', 'illuminance', 'env_temperature','env_humidity','Decisior
# print(df.head(10)) #UNCOMMENT IF WANT FIRST 10 ROWS PRINTED OUT


config = {'algorithm':'C4.5'} # CONFIGURE THE ALGORITH. CHOOSE BETWEEN ID3, C4.5, CART, Regression
model = chef.fit(df.copy(), config) #CREATE THE DECISION TREE BASED OF THE CONFIGURATION ABOVE
config = {'enableRandomForest': True, 'num_of_trees':5}
model = chef.fit(df, config)

resultados = pd.DataFrame(columns = ["Real", "Predicción"]) #CREATE AN EMPTY PANDAS DATAFRAME
 SAVE ALL REAL VS ESTIMATED VALUES IN THE ABOVE DATAFRAME
for i in range(300):
    l = []
    feature = df.iloc[i]
    prediction = chef.predict(model, feature)
    l.append(prediction)
    resultados.loc[i] = l
    print(l)

ASK THE USER FOR A NEW RECORD
nuevo = input("INSERT NEW RECORD AS A LIST:\n")
feature = eval(nuevo)
prediction = chef.predict(model, feature)
print(prediction)
```

Report in arXiv

I. Quintero-Villegas, S. Vega-Lopez, and M. Toro. Coffee And Rust. Detection And Prevention For Improving Exportation Quality. ArXiv e-prints, Oct. 2019. Available at: https://arxiv.org/submit/2915423