**DESIGN ACTIVITY**

*IMAS 21/22*

**

**Authors**: *Sergi Albiach, Anna Garriga,*

*Benet Manzanares and Ramon Mateo*

Index

[**Introduction**](#_ukbrde2pi7en) **2**

[**Agent Description**](#_sie9f2yguaap) **3**

[User Agent](#_w7mtpaftw9y4) 3

[Coordinator Agent](#_13dce0ylfpbx) 4

[Classifier Agent](#_1v372sphjv4a) 5

[**System Architecture**](#_vee1xfp2sk4y) **6**

[**Bibliography**](#_tpvwowscbggk) **8**

# Introduction

The goal of this project is to develop an agent-based decision support system. In particular, we aim to develop a system that is able to solve a classification problem in a collaborative way. For this particular case, we will use the dataset AUDIT [1], from the UCI Machine Learning Repository [2], that contains 767 instances with 25 variables defining each sample.

At the end, the final goal of this collaborative classifier is to predict whether a firm is fraudulent, basing the decision on some risk factors.

Each of the M classifiers will make their decision using the C4.5 [3] decision tree algorithm implementation from the WEKA 3.8.5 library [4][5], called J48 [6]. Each of these classifiers is representing a different firm with only 6 attributes, representing that each of them was able to collect a different subset of the variables. Moreover, they will use only 300 samples to train the classifier, where 25% of them will be used for validation. Finally, 50 instances from the dataset will be reserved to test the complete collaborative classifier. Every test call will be made using a subset of 15 samples with 20 of the 25 attributes.

Then, the final classifier will take into account the predictions of each of the agents and, by means of some coordination mechanism, it will have to output a final prediction.

# Agent Description

## User Agent

* **Role/s:** Based on the configuration given in a XML file, it is in charge of loading the dataset, splitting the data into training and test sets, starting the training and managing classification queries from the user. Its communication with other agents is limited to the coordination agent.
* **Properties:**
  + Social ability: It communicates with the coordinator agent in order to train or use/evaluate the classification system.
  + Rationality: All its actions will be done with a clear goal. For instance, the training process.
* **Number of agents:**  1

| **MESSAGE\_NAME** | **DESCRIPTION** |
| --- | --- |
| train\_agents(DATA, COORD\_AGENT) | Send the training data to the coordinator agent to train the classification system. |
| test\_agents(DATA, COORD\_AGENT) | Send the test data to the coordinator agent in order to obtain the accuracy of the model. |

| **FUNCTION\_NAME** | **DESCRIPTION** |
| --- | --- |
| (TRAIN, TEST) = prepare\_data(DATASET) | Splits the loaded data into test and training sets |

## Coordinator Agent

* **Role/s:** It is responsible for splitting the training dataset between the agents and selecting the attributes that each of them will analyze. It is also in charge of receiving the output classifications from the classifier agents and aggregating them using the normalized accuracies from the training step as weights [7]. This result will be sent to the user agent.
* **Properties:** 
  + Social ability: It will receive queries from the user agent and answer them. Additionally, it will communicate with all the classifiers.
  + Rationality: All its communications and processes will be done in order to achieve the desired goals.
  + Reasoning capabilities: It will be able to aggregate the results of all the classifiers based on their results during the training.
* **Number of agents:** 1

| **MESSAGE\_NAME** | **DESCRIPTION** |
| --- | --- |
| end(PROCESS\_NAME,VALUE,USER\_AGENT) | Sends to the user agent the notification of the ending of a process with a return value. |
| train(DATA, CLASS\_AGENT) | Sends the data to a classification agent to start the training process |
| predict(DATA, CLASS\_AGENT) | Sends the data to a classification agent to start the testing process |

| **FUNCTION\_NAME** | **DESCRIPTION** |
| --- | --- |
| (ATTRIB\_DISTRIBUTION) = assign\_attributes(N\_ATTRIBUTES, N\_CLASSIFIERS, ATTRIBUTES) | Assigns randomly the attributes to the classifiers and returns a dictionary with the name of the attributes for each classifier. |
| (DATA\_ASSIGNMENT) = split\_data(DATA, ATTRIB\_DISTRIBUTION) | Returns a dictionary where every agent has associated its subset of the data based on the attribute distribution |
| (FINAL\_ACCURACY) =  compute\_accuracy(VALIDATION\_ACCURACY, AGENTS\_OUTPUT) | Computes the final accuracy given the classification of the agents on the test set and its accuracy on the validation step |

## Classifier Agent

* **Role/s:** It receives a part of the training data and creates a Decision Tree based on it. It also receives test cases and sends the output classification result to the coordinator agent.
* **Properties:** 
  + Social ability: It receives orders from the coordinator agent and returns the corresponding outputs.
  + Rationality: Only works to achieve its goals. For example, it will not perform a classification if it is not necessary.
  + Learning: It will use the J48 [6] decision tree algorithm, learning from the training data.
  + Reasoning capabilities: It will extrapolate the knowledge from the training data to its inferences during evaluation.
* **Number of agents:** 10

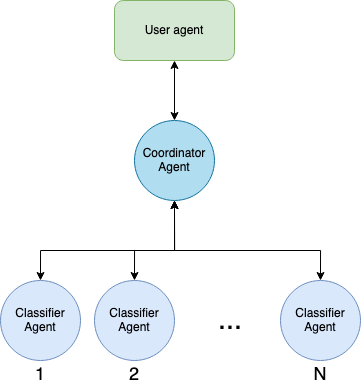
**MESSAGE\_NAME**

| **MESSAGE\_NAME** | **DESCRIPTION** |
| --- | --- |
| end\_training(PROCESS\_NAME,ACC,COORD\_AGENT) | Sends to the coordinator agent the notification of the ending of the training process with its accuracy. |
| end\_prediction(PROCESS\_NAME,CLASSES,COORD\_AGENT) | Sends to the coordinator agent the notification of the ending of the prediction process with the resulting output classes. |

| **FUNCTION\_NAME** | **DESCRIPTION** |
| --- | --- |
| (ACCURACY) =  train\_tree(DATA) | Creates the decision tree and returns the accuracy of the training step. |
| (PREDICTIONS) =  predict(DATA) | Classifies the input data and returns the classification results |

# System Architecture

Our proposed System Architecture consists of a hierarchy with the User Agent at the top, the Coordinator Agent just below receiving and answering User Agent communications, and Classifier Agents at the bottom receiving and answering Coordinator Agent’s communications (Fig. 1).



**Fig. 1**: general diagram of the system architecture

To do these connections, the system performs four steps:

The first step consists of the training process (Fig. 2). From the 767 instances of the dataset, the User Agent selects 717 and sends them to the Coordinator Agent. Then the coordinator agent distributes them among the Classifier Agents, giving 300 instances and 6 attributes to each Classifier and making sure to allocate all the instances and attributes. With all 300 instances, each Agent selects 225 and does the training, after this it uses the other 75 to test and gets a value of accuracy.

The second step consists of returning accuracy (Fig. 3). Each Classifier Agent returns the accuracy value and a message saying that they have ended the training to the Coordinator. When the Coordinator has received all the messages it sends another message to the User notifying it.

| **Fig. 2**: diagram of step 1: training data | **Fig. 3**: diagram of step 2: returns of the training |
| --- | --- |

The third step consists in the shipping of the testing data (Fig. 4). The agent sends 15 of the remaining 50 instances for the testing, but with only 20 of their 25 attributes to the Coordinator. The Coordinator checks which classifiers have all 6 attributes among the 20 received and sends them the testing data with the 6 attributes. The Classifiers test the received instances.

The fourth step consists in the return of the results to the User Agent (Fig. 5). Each Classifier Agent who had testing data returns the classification to the Coordinator. Then the Coordinator makes a final decision taking into account all the classifications received and the accuracy of each Classifier. Finally, the Coordinator sends the final classification to the User Agent.

| **Fig. 4**: diagram of step 3: shipping testing data | **Fig. 5**: diagram of step 4: final classification |
| --- | --- |

The kind of architecture would be hybrid between reactive and deliberative: On one side, Classifier Agents are simple and when they receive the order to train or test they do so, but they do not do anything by themselves. However, on the other side, the Coordinator Agent has an explicit model of the world and the decisions it makes are via logical reasoning.

# Bibliography

[1] AUDIT dataset. Retrieved on 6/10/2021 from <https://archive.ics.uci.edu/ml/datasets/Audit+Data>

[2] UCI Machine Learning Repository. Retrieved on 6/10/2021 from <https://archive.ics.uci.edu/ml/index.php>

[3] C4.5 decision tree algorithm. Retrieved on 6/10/2021 from <https://en.wikipedia.org/wiki/C4.5_algorithm>

[4] Waikato Environment for Knowledge Analysis (WEKA) library. Retrieved on 6/10/2021 from <https://www.cs.waikato.ac.nz/~ml/weka/index.html>

[5] WEKA 3.8.5. Retrieved on 6/10/2021 from <https://mvnrepository.com/artifact/nz.ac.waikato.cms.weka/weka-stable/3.8.5>

[6] J48 decision tree implementation of C4.5. Retrieved on 15/10/2021 from <https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html>

[7] Ordered weighted averaging aggregation operator (OWA). Retrieved on 9/10/2021 from <https://en.wikipedia.org/wiki/Ordered_weighted_averaging_aggregation_operator>