

UNIVERSITAT DE LLEIDA



GRAU EN ENGINYERIA INFORMÀTICA  
INTEL·LIGÈNCIA ARTIFICIAL

---

## Tercera pràctica d'Intel·ligència Artificial

---

*Autors:*

Jaume Giralt Barbé  
Jordi Onrubia Palacios

*Professor:*

Carlos Jose Ansotegui Gil  
Josep Pon Farreny

20 de gener de 2017

## Índex

<b>1</b>	<b>Contingut</b>	<b>2</b>
1.1	Contingut Principal . . . . .	2
1.2	Contingut Secundari . . . . .	2
<b>2</b>	<b>Decisions de disseny</b>	<b>3</b>
2.1	Tree Predict . . . . .	3
2.2	Bayesian Learning . . . . .	3
2.3	K-Means Cluster . . . . .	3
<b>3</b>	<b>Avaluació Experimental</b>	<b>3</b>

## Índex de taules

1	Percentatge del training set i resultat del comportament del test . . . . .	4
---	---	---

## Índex de figures

1	Gràfica algoritme A* Nodes/Temps . . . . .	4
---	--	---

# 1 Contingut

## 1.1 Contingut Principal

El contingut principal es centre en els següents fitxers:

- **Tree Predict:** Aquest és el fitxer principal de la pràctica, el que ens ha donat més feina per fer la realització. Són mètodes transparents de classificar observacions que després d'entrenar-lo amb dades amb característiques similars ho ordena fent servir sentències *if then* formant un **arbre**.
- **Bayesian Learning:** Aquest fitxer està implementat un altre mètode de aprenentatge supervisat. Amb aquest mètode podem classificar si una paraula/frase pertany a una categoria o a una altra després d'entrenar-lo. Un exemple del seu funcionament a la vida real seria el Anti-SPAM del correu electrònic.
- **K-Means Cluster:** L'algorisme K-means és un mètode d'agrupament que té com a objectiu la partició d'un conjunt  $n$  observacions en  $k$  grups en el qual cada observació pertany al grup més proper a la mitjana. És un mètode d'aprenentatge no supervisat.

## 1.2 Contingut Secundari

A més a més de els fitxers esmentats anteriorment, també hem afegit un joc de proves per realitzar l'avaluació experimental. També hem afegit scripts utilitzats per extreure les dades i per crear els gràfics amb l'eina GNU-Plot.

## 2 Decisions de disseny

### 2.1 Tree Predict

En aquest document és on hem hagut d'implementar més funcions per la pràctica. En primer lloc, hem implementat les funcions **buildTree** i **buildTreeIterative**, que són les funcions encarregades de muntar el arbre a partir de les dades d'entrada. La funció **buildtree** rep d'entrada un conjunt de dades i les divideix en columnes segons la impuresa ja que a més decrement, millor són els subconjunts. Aquesta impuresa ve donada al utilitzar la funció amb les funcions de l'index de Gini o l'Entropia. El resultat d'aquesta funció és un arbre amb les seves branques verdader i fals on es mostra el resultat de les preguntes realitzades. En la versió iterativa del algoritme hem aplicat els coneixements adquirits en l'assignatura de Algorítmica i Complexitat on ens explicaven que un dels mètodes per passar una funció recursiva a iterativa era la utilització de piles per simular les crides recursives. Es per això que hem implementat la nostra pròpia pila per fer aquesta funció. Una altra funció que hem hagut d'implementar és la funció **classifier** que consisteix en donat unes característiques i un arbre, ens dona l'atribut que volem cercar. Aquesta funció la farem servir amb la funció de **testPerformance** que donat dos sets de característiques, entrenem un arbre amb el *trainingSet* i per cada conjunt de característiques del *testSet* cridem a l'anterior funció esmentada i comprovem si ens dona l'argument correcte. Amb aquesta funció i amb funcions auxiliars és com hem pogut realitzar l'avaluació experimental.

### 2.2 Bayesian Learning

### 2.3 K-Means Cluster

## 3 Avaluació Experimental

Percentatge TrainSet	Percentatge Good
0.1	19.04
0.2	37.5
0.3	44.89
0.4	35.71
0.5	40.0
0.6	53.57
0.7	47.61
0.8	57.14
0.9	57.14

Taula 1: Taula per al test de les variables de tipus categoria

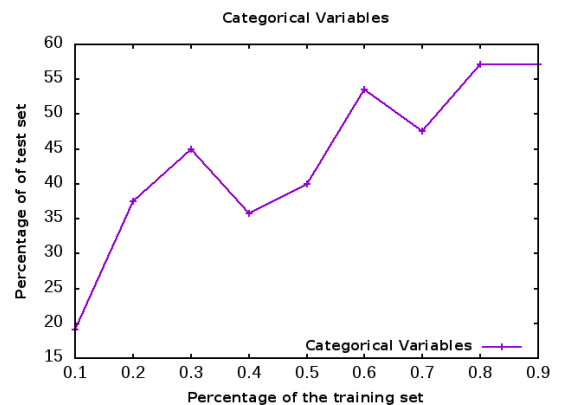


Figura 1: Gràfica per al test de les variables de tipus categoria