



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

**MÒDUL PEDAGÒGIC EN UN SISTEMA TUTOR
INTEL·LIGENT PER A PREDIR L'EVOLUCIÓ
DE L'ALUMNAT**

Jordi Rodríguez Queralto

Directora: Dra. Maria Salamó Llorente
Realitzat a: Departament de Matemàtica
Aplicada i Anàlisi. UB

Barcelona, Gener de 2015

Agraïments

Voldria donar les gràcies a la meva tutora d'aquest Treball de Fi de Grau, Maria Salamó Llorente, no només he tingut l'ocasió de fer aquest projecte gràcies a ella, sinó que ha estat constantment donant suport en tot moment, constantment guiant-me i involucrant's-hi tot el que ha estat possible, sinó que ha demostrat tenir una paciència inigualable. Sense la seva ajuda, el projecte de fi de grau no hagués estat una experiència tant positiva i didàctica.

Abstract

Intelligent Tutoring Systems are systems based on Artificial Intelligence that can analyze models of knowledge to get to understand the learner's state. Its function is trying to help him or her through emulating the role of a human tutor.

In this project we'll implement the Pedagogical Module of an Intelligent Tutoring System. Specifically, we'll implement a prediction system that tries to estimate the grades students will have at the end of the academic year. For this we'll incorporate methods of Machine Learning. Machine Learning is a discipline from the field of Mathematics and Computer Science that studies algorithms to predict future outcomes and act accordingly, and, if applicable, implement an automated behaviour that tries to aim for the best possible result. Specifically, the algorithms we'll use are called classifiers, a kind of Supervised Learning algorithm. We'll implement a library with different kinds of classifiers, test their functionality using Ten-Fold Cross-Validation and simulate its usage with different datasets that come from the UCI repository and from 4 subjects that are taught in grades at the University of Barcelona, to analyze their accuracy and efficiency.

Furthermore, this library will need to be usable by teachers. Using it, they will be able to input their history from previous years and their current course to get an estimate of that year's results. By doing this, they will be able to detect a deviation from previous courses and work on getting it back on track. The implementation will have the options to either go for the most theoretically precise classifier or use the pre-set option for a fast execution. The most precise option will be determined by testing every single classifier using the train set, while the pre-set classifier will be chosen based on this project's tests and the algorithm's consistency. These functions will work through a user interface, which will be used by teachers from different fields of the University of Barcelona. This means the interface will need to be user-friendly and easy to learn.

Índex

1	Introducció	1
1.1	Motivació personal i marc del projecte	1
1.1.1	Definició Intelligent Tutoring System (ITS)	1
1.1.2	Mòduls Intelligent Tutoring System	2
1.2	Contexte d'aplicació	3
1.3	Objectius del projecte	3
1.4	Organització de la memòria	4
2	Visió del problema	5
2.1	Definició del problema	5
2.2	Tecnologies a usar	6
2.2.1	Weka	7
2.2.2	Llibreria CBR-KB	8
2.2.3	Java Swing	10
3	Anàlisi i disseny	11
3.1	Requisits	11
3.1.1	Requisits funcionals	11
3.1.2	Requisits no funcionals	12
3.1.3	Casos d'ús de la interfície	12
3.2	Estructura de codi emprada	16
4	Experimentació i anàlisi de resultats	18
4.1	Detalls dels experiments	18
4.2	Experiment 1: Anàlisi de resultats amb bases de dades del repositori UCI . .	19
4.3	Experiment 2: Anàlisi de resultats amb bases de dades de les assignatures . .	23
4.4	Experiment 3: Anàlisi de resultats usant varis cursos de les assignatures . . .	27
4.5	Experiment 4: Anàlisi de resultats reduint el conjunt de proves de les assignatures	30
5	Estudi econòmic	34
5.1	Diagrama de Gantt	34
5.2	Roadmap	36
5.3	Cost del projecte	37
6	Conclusions i línies futures	38
	Annex: Manual d'usuari de la interfície	ii

Índex de figures

1.1	Diagrama dels mòduls que formen un Sistema de Tutor Intel·ligent	2
2.1	Diagrama de fluxe simplificat del sistema de predicció	5
2.2	Cicle de CBR	8
2.3	Principals paquets del codi font que implementa el CBR	9
3.1	Arquitectura Model-Vista-Controlador	12
3.2	Diagrama de classes	16
4.1	Resultats dels 5 millors classificadors per a la base de dades Iris, 150 instàncies a classificar	20
4.2	Resultats dels 5 millors classificadors per a la base de dades Glass, 214 instàncies a classificar	20
4.3	Resultats dels 5 millors classificadors per a la base de dades Vehicle, 846 instàncies a classificar	21
4.4	Resultats dels 5 millors classificadors per a la base de dades Ionosphere, 351 instàncies a classificar	21
4.5	Resultats dels 5 millors classificadors per a la base de dades Sonar, 208 instàncies a classificar	22
4.6	Comparació del percentatge d'encerts més alt per a cada base de dades (classificació normal), amb el percentatge d'encerts del CBR.	23
4.7	Resultats de realitzar Ten-Fold Cross-Validation a les dades de Disseny de Software, any 2013-2014, 83 instàncies a classificar	24
4.8	Precision i Recall corresponent a la classificació binària de Disseny de Software, any 2013-2014	24
4.9	Resultats de realitzar Ten-Fold Cross-Validation a les dades de Fonaments de Tecnologia, any 2012-2013, 71 instàncies a classificar	25
4.10	Precision i Recall corresponent a la classificació binària de Fonaments de Tecnologia, any 2012-2013	25
4.11	Resultats de realitzar Ten-Fold Cross-Validation a les dades de Programació 1, any 2013-2014, 90 instàncies a classificar	26
4.12	Precision i Recall corresponent a la classificació binària de Programació 1, any 2013-2014	27
4.13	Resultats de realitzar Ten-Fold Cross-Validation a les dades de TIC, any 2013-2014, 71 instàncies a classificar	28
4.14	Precision i Recall corresponent a la classificació binària de TIC, any 2013-2014	28
4.15	Percentatge d'encerts intentant predir el resultat de la base de dades de DSW any 2013-2014 utilitzant la base de dades de DSW any 2012-2013 com a entrenament	29

4.16	Percentatge d'encerts intentant predir el resultat de la base de dades de FDT any 2012-2013 utilitzant la base de dades de FDT any 2011-2012 com a entrenament	30
4.17	Percentatge d'encerts intentant predir el resultat de la base de dades de P1 any 2013-2014 utilitzant la base de dades de P1 any 2012-2013 com a entrenament	31
4.18	Percentatge d'encerts intentant predir el resultat de la base de dades de DSW any 2013-2014 amb menys dades visibles utilitzant la base de dades de DSW any 2012-2013 com a entrenament	32
4.19	Percentatge d'encerts intentant predir el resultat de la base de dades de FDT any 2012-2013 amb menys dades visibles utilitzant la base de dades de FDT any 2011-2012 com a entrenament	33
4.20	Percentatge d'encerts intentant predir el resultat de la base de dades de P1 any 2013-2014 amb menys dades visibles utilitzant la base de dades de P1 any 2012-2013 com a entrenament	33
5.1	Diagrama de Gantt	35

Índex de taules

2.1	Taula amb l'evolució durant el curs acadèmic de tres alumnes escollits aleatòriament. DSW, curs 2013-2014	6
3.1	Cas d'ús 1: Carregar dades històriques	13
3.2	Cas d'ús 2: Carregar dades actuals	13
3.3	Cas d'ús 3: Eliminar totes les dades històriques	13
3.4	Cas d'ús 4: Eliminar totes les dades actuals	14
3.5	Cas d'ús 5: Visualitzar fitxer de dades de l'històric	14
3.6	Cas d'ús 6: Visualitzar fitxer de dades actual	14
3.7	Cas d'ús 7: Realitzar execució ràpida amb el classificador recomanat	15
3.8	Cas d'ús 8: Realitzar execució exhaustiva cridant a la bateria de proves	15
3.9	Cas d'ús 9: Obrir resultats de la predicció	15
4.1	Informació bàsica sobre les bases de dades utilitzades del repositori UCI	19
4.2	Matriu de confusió de l'algorisme MLPClassifier amb la base de dades Iris	22
5.1	Cost de les diferents parts del projecte	37

Capítol 1

Introducció

1.1 Motivació personal i marc del projecte

Sempre he tingut bastant d'interès en el món de la intel·ligència artificial, em resulta curiós tot el que es refereix a la cerca de patrons i la creació de respostes automatitzades. Estudiar a la Universitat de Barcelona i entendre el funcionament dels algorismes només ha accentuat aquest interès, veient més la quantitat d'usos que poden arribar a tenir i lo pràctics que són.

Donat aquest interès, vaig veure una molt bona oportunitat en aquest projecte, en el qual he pogut participar gràcies a la Dra. Maria Salamó.

Aquest projecte forma part del Projecte d'innovació docent per implementar un **Intelligent Tutoring System** a la Universitat de Barcelona, en el qual entraré en més detall en els següents apartats.

1.1.1 Definició Intelligent Tutoring System (ITS)

Un **Intelligent Tutoring System**, o Sistema de Tutor Intel·ligent és un sistema informàtic que aprèn del comportament dels alumnes mitjançant tècniques d'Intel·ligència Artificial i proposa continguts en funció de la demanada o necessitats individuals de l'alumnat.

Aquest aprenentatge automàtic realitzat es basa en l'anàlisi de dades sobre els alumnes. Per tant, per tal de comprendre el funcionament del Sistema de Tutor Intel·ligent començarem definint la informació amb que treballa.

Aquesta és:

1. **Coneixement previ** proporcionat pel professorat sobre la matèria sent ensenyada; estratègies d'aprenentatge, possibles errors i problemes de l'alumnat.
2. **Experiència passada** que el sistema d'aprenentatge ha adquirit a través d'interactuar amb l'alumnat, sobre quins errors fan els alumnes, quants exemples fan falta per tal que n'aprenquin, i què obliden els alumnes.
3. **Preferències** sobre la importància de cada tema, el nivell de realització que es busca de l'alumne.
4. **Observacions** dels resultats dels estudiants i observacions del resultat de la seva interacció amb el Sistema de Tutor Intel·ligent.

Treballant amb aquesta informació, l'ITS ens ha de retornar canvis que s'haurien de fer per millorar l'evolució de l'alumnat (trobar els temes de l'assignatura que els alumnes solen portar més fluixos, modificacions a la planificació de l'assignatura i a les proves).

La introducció i gestió d'algunes d'aquestes dades són tasques que normalment duu a terme el professor de forma manual. Mitjançant la implementació d'un Sistema de Tutor Intel·ligent s'aconseguiria realitzar un ensenyament personalitzat, que té en compte les necessitats individuals de cada alumne.

Addicionalment, un Sistema de Tutor Intel·ligent pot ser útil també als professors, permetent al professorat saber com estan assolint els objectius d'aprenentatge els seus alumnes i saber així quins punts necessiten millores.

1.1.2 Mòduls Intelligent Tutoring System

Prèviament s'ha definit Sistema de Tutor Intel·ligent com una sola entitat, però en realitat es tracta de diversos mòduls amb funcions específiques i connectats entre sí, que poden ser entitats independents sempre i quan es comuniquin entre elles.

Aquestes entitats o mòduls són:

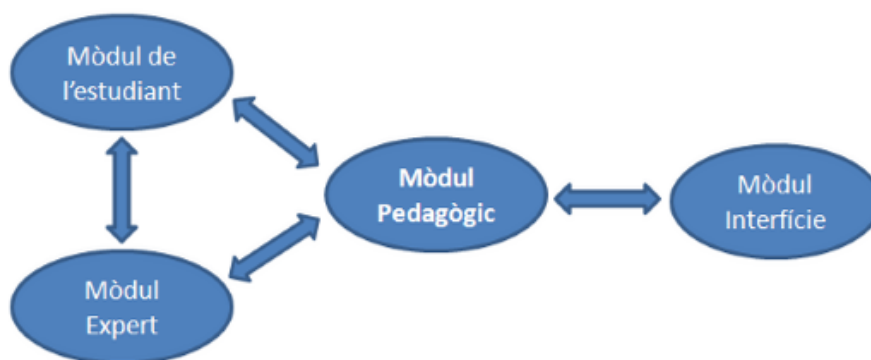


Figura 1.1: Diagrama dels mòduls que formen un Sistema de Tutor Intel·ligent

- **Mòdul de l'estudiant:** s'encarrega de modelar les característiques individuals de l'estudiant, entre elles el coneixement individual del domini. Definit per Robles el 1993: [4]:

“El model de l'estudiant reflecteix quant sap l'estudiant sobre el domini, així com les experiències cognitives i d'aprendre, a partir dels quals es pot fer diagnòstic”

- **Mòdul expert o model de coneixement del domini:** recull tot el coneixement específic de l'assignatura. Aquest mòdul ha de contenir tota la informació necessària per evaluar una prova d'un alumne, i establir una comunicació directe entre l'alumne i el sistema.
- **Mòdul pedagògic o mòdul tutor:** és el nucli del sistema, utilitza la informació proporcionada pels tres altres mòduls per tal de fer l'anàlisi que realitzaria un tutor de manera individual de la informació disponible d'un alumne, i crear un conjunt de suggerències per millorar la qualitat de l'ensenyament a l'alumne. En resum, el que està fent es triar el que considera com a la millor acció per a l'alumne concret, en referència als exàmens, temari o les accions que sigui que se li ha proporcionat com a possibilitats.
- **Mòdul de la interfície d'usuari:** és la interfície que veurà l'estudiant a l'hora d'utilitzar el sistema. Aquesta ha de tenir tota l'estructura interna del sistema amagada a l'usuari, i permetre obtenir resultats de manera simple i concisa. Segons Robles [4]:

“La interfície permet a l’usuari interactuar amb el sistema. Tres tipus d’usuaris poden ser distingits: l’estudiant, l’instructor i el Desenvolupador del sistema”

Això significa que, mentre que l’ús de l’orientació està disponible per a qualsevol usuari, hi ha d’haver també funcions específiques per a supervisar el funcionament del sistema, restringides per a ús exclusiu del desenvolupador i professors o tutors implicats.

Com es pot veure, en un ITS hi ha molts mòduls involucrats i que depenen entre ells. Aquest projecte correspon a la primera fase d’implementació d’un ITS, i s’ha considerat que el mòdul més important, i el qual està més íntimament relacionat amb els altres és el Mòdul Pedagògic. És per això que aquest projecte es centrarà en el Mòdul Pedagògic com a primer pas del sistema complet.

1.2 Contexte d’aplicació

El Mòdul Pedagògic està pensat per ser utilitzat per a tots els ensenyaments, a tots els cursos, però sobretot és necessari als primers cursos de Grau en què els alumnes han de realitzar una gran quantitat de treballs, normalment curts però nombrosos. Tant alumnes com professors necessiten tenir una eina que els ajudi a comprendre l’evolució durant el curs i que fagi una predicció sobre el percentatge d’assimilació dels continguts de l’assignatura durant el quadrimestre. En qualsevol cas, en aquest projecte es farà una avaluació en 4 assignatures de 3 graus diferents:

- Programació 1, del Grau d’Enginyeria Informàtica de la Universitat de Barcelona
- Disseny de Software, del Grau d’Enginyeria Informàtica de la Universitat de Barcelona
- Fonaments de Tecnologia, del Grau d’Informació i Documentació de la Universitat de Barcelona
- TIC, del Grau de Lingüística de la Universitat de Barcelona

1.3 Objectius del projecte

L’objectiu principal d’aquest projecte és definir, dins del Mòdul Pedagògic d’un Sistema de Tutor Intel·ligent, un sistema de predicció i desenvolupar una eina inicialment pensada per al professorat que permeti predir abans de que acabi el curs si un alumne aconseguirà superar una assignatura. D’aquesta manera, el professor pot prendre la iniciativa i intentar rectificar la marxa de l’estudiant, reforçant els seus punts febles.

Encara que la fita d’aquest projecte docent és centrar-se en el sistema de predicció del Mòdul Pedagògic i en l’eina per al professorat, és important remarcar que aquest Mòdul Pedagògic consisteix en la primera peça d’un Sistema de Tutor Intel·ligent que es desenvoluparà en diferents fases. Un cop completades totes les fases del Sistema de Tutor Intel·ligent, tant l’alumnat com el professorat podran utilitzar-lo en la dinàmica de l’aprenentatge. L’alumnat podrà rebre el *feedback* des del Mòdul Pedagògic ja que serà el mecanisme que tindrà el Sistema Tutor per a indicar automàticament als alumnes la seva evolució i per proposar-los exercicis en funció del resultat de la predicció en cadascuna de les assignatures.

El Mòdul Pedagògic està pensat per poder-se fer servir en qualsevol assignatura de Grau de la Universitat de Barcelona, independentment de la titulació, però sobretot pot ser de gran utilitat per als primers cursos de Grau en què els alumnes han de realitzar gran nombre de proves parcials i avaluacions de temes puntuas. Per exemple, en els primers cursos del

Grau d'Enginyeria Informàtica, els alumnes realitzen una gran quantitat de petits programes i proves parcials

Dintre del Sistema de Tutor Intel·ligent, aquest treball de fi de grau s'enfoca en la implementació de la part del Mòdul Pedagògic, a més a més de fer una interfície perquè en fagi ús el professorat. El mòdul pedagògic haurà d'extreure les dades de l'alumnat en un format amigable pel professorat, analitzar l'històric de dades dels alumnes de cursos acadèmics anteriors, i predir l'evolució dels alumnes del curs actual.

Fer la predicció comportarà l'ús de tècniques d'Intel·ligència Artificial, en concret es farà ús de sistemes classificadors. S'implementarà una llibreria amb un conjunt de classificadors, s'estudiaran els seus resultats, i es crearà una interfície adient per fer-ne ús.

En concret, els objectius específics d'aquest projecte són:

- Implementar una llibreria que incorpori diferents tècniques de predicció (classificadors)
- Analitzar la llibreria i extreure conclusions dels resultats obtinguts en un conjunt d'assignatures que serviran com a prova pilot de la llibreria implementada
- Facilitar l'ús de la llibreria amb una interfície d'usuari amb tasques ben definides i clares

1.4 Organització de la memòria

La memòria es divideix de la següent manera:

- En el capítol 2 s'entra en les especificacions sobre el problema que es busca resoldre amb aquest projecte i quines eines s'utilitzen per a fer-ho.
- En el capítol 3 s'explica el disseny en quant a estructura de codi, i el disseny de la interfície. Inclou un diagrama de classes i els casos d'ús de l'usuari per a la interfície.
- En el capítol 4 es realitza l'anàlisi de la llibreria de sistemes classificadors utilitzada amb bases de dades del repositori UCI i amb bases de dades de les assignatures.
- En el capítol 5 es fa referència a l'estudi econòmic, el valor que tindria aquest projecte si es tractés d'una aplicació oberta al mercat. Inclou el diagrama de Gantt del projecte i un roadmap de les diferents versions que s'han implementat durant el transcurs d'aquest.
- En el capítol 6 s'extreuen les conclusions resultants d'aquest projecte i s'expliquen les línies futures.
- En l'annex es disposa d'un manual d'usuari, amb els requisits previs a complir per tal que la llibreria funcioni correctament i amb els passos que realitzarà a l'hora d'utilitzar la interfície d'usuari.

Capítol 2

Visió del problema

2.1 Definició del problema

El Mòdul Pedagògic es planteja com un sistema de predicció, concretament, la situació que pretén resoldre és la següent: tenim una quantitat de notes limitades del curs actual i volem, a partir d'aquestes, predir la nota final (rang) d'un estudiant del curs més possible. És a dir, tenim una tasca de classificació, en que comparem les dades acadèmiques d'anys anteriors amb les dades incompletes del curs actual per tal de predir el resultat final.

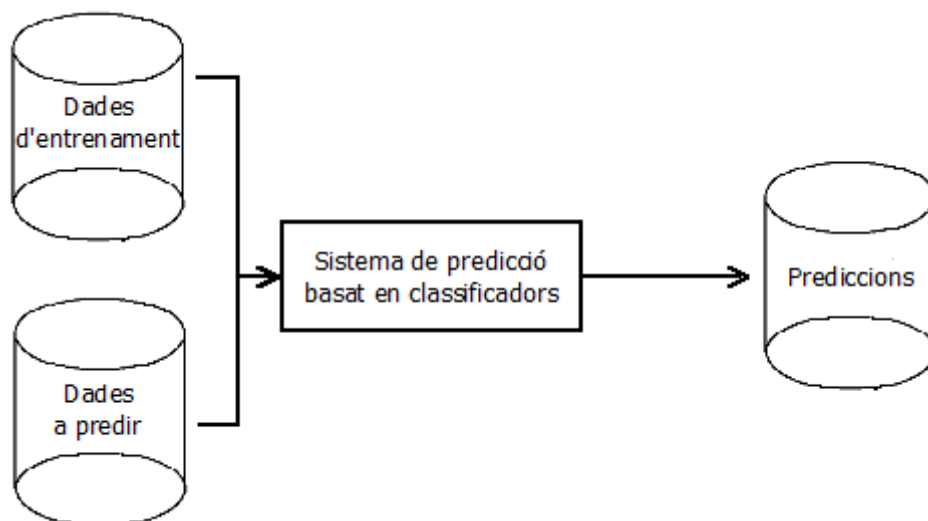


Figura 2.1: Diagrama de fluxe simplificat del sistema de predicció

Tal i com es pot veure a la Figura 2.1, el sistema de predicció parteix d'un conjunt d'entrenament, que en aquest projecte es correspon a les dades d'un curs acadèmic d'una assignatura. Aquestes dades contenen un conjunt d'alumnes i per cadascun d'ells, el conjunt de notes de les proves, exàmens i exercicis lliurats i la nota final de l'assignatura

El sistema s'alimenta després de les dades a predir. Aquestes dades són algunes de les proves, parcials o exercicis que els alumnes han estat realitzant al llarg de l'actual curs acadèmic. Amb tot això, el sistema classificador prediu com a sortida la nota final de cadascun dels alumnes.

La predicció es farà, inicialment, de cara a estimar la nota definitiva de l'alumnat, utilitzant la informació de que es disposa fins a un moment donat del curs.

Distingirem dos etiquetatges diferents per a fer la predicció:

1. **Binària:** Aprovat o Suspens
2. **Per rang:**
 - Suspens: notes inferiors a 5.
 - Aprovat: notes superiors a 5 però inferiors a 6.
 - Bé: notes superiors a 6 però inferiors a 7.
 - Notable: notes superiors a 7 però inferiors a 9.
 - Excel·lent: notes superiors a 9.

P1	P2	P3	E1	E2	E3	Parcial1	Parcial2	NotaFinal
8	5	8	9	8	6	5,5	3,3	3,8
10	8,25	9,5	9	9	9	4,25	7,1	7,7
5	5	6	8	10	2	7,25	5,8	5,99

Taula 2.1: Taula amb l'evolució durant el curs acadèmic de tres alumnes escollits aleatòriament. DSW, curs 2013-2014

A la Taula 2.1 veiem un exemple del conjunt de dades de cada alumne que se li introdueix al nostre Mòdul Pedagògic per a cada assignatura. La idea és que només tenim aquestes dades al complet per a l'històric dels cursos anteriors, mentres que del curs acadèmic actual tenim algunes de les columnes (per exemple: P1, P2, E1, E2, Parcial1), i hem d'intentar aproximar la nota final obtinguda per cada alumne.

La quantitat de columnes (quantitat de pràctiques, d'exercicis i de parcials) depèn de l'assignatura però com a condició necessària, la quantitat de columnes i el seu format haurà de ser igual entre els diferents arxius de dades d'entrenament per al correcte funcionament del sistema de predicció. De cara a la introducció del fitxer de dades a predir, però, tindrem una quantitat de columnes desconeguda, hipotèticament menor que en els fitxers d'entrenament, però sempre superior a 0.

2.2 Tecnologies a usar

Tant la llibreria com la interfície seran utilitzades per diferents usuaris, de manera que com a llenguatge de programació s'ha escollit el **Java**, que té la gran avantatge de ser multiplataforma. Addicionalment, les llibreries d'aprenentatge automàtic es poden connectar fàcilment utilitzant Java. La seva descarrega es realitza a la web oficial de Java ¹.

Les llibreries per a la implementació dels classificadors són:

1. **CBR-KB:** llibreria implementada i desenvolupada a la Universitat de Barcelona. S'ha escollit degut al seu alt percentatge de predicció, i degut a les possibilitats de modificar els algorismes que ofereix internament.
2. **Weka:** llibreria standard d'aprenentatge artificial que incorpora un gran ventall d'algorismes de classificació, clusterització, i amb paquets per a la normalització de dades i extracció de diferents criteris per mesurar la qualitat dels algorismes a l'hora de fer prediccions.

¹<https://www.java.com/es/download/>

Per a la interfície s'ha utilitzat (**Java Swing**), degut a la facilitat per la facilitat d'implementació que suposa tot i tenir un alt grau de personalització possible, i a la senzillesa per introduir-lo en una estructura Model-View-Controller, que s'explicarà més detalladament en l'apartat 3.1.2.

2.2.1 Weka

Weka, acrònim de Waikato Environment for Knowledge Analysis, és una col·lecció d'algorismes d'Aprenentatge Automàtic (*Machine Learning*) desenvolupat per la Universitat de Waikato. Es pot obtenir de la seva pàgina web oficial ^{II}.

La seva API conté eines de pre-processament de dades, algorismes d'aprenentatge i mètodes d'evaluació, interfícies d'usuari i visualització de dades i un entorn de treball per comparar algorismes.

Per fer ús d'aquesta llibreria es necessita una base de dades en format **ARFF** (Attribute Relation File Format). Weka proporciona els mètodes necessaris per fer la transició del format **CSV** (Coma-Separated Value) al format ARFF.

La llibreria es divideix en els següents paquets centrals, més els subpaquets que parteixen de la seva estructura:

- **Weka.associations:** Conté les *interfaces* ^{III} que utilitzaran els algorismes de regles d'associació.
- **Weka.attributeSelection:** Conté les diferents classes que realitzen la selecció d'atributs ^{IV}.
- **Weka.classifiers:** Paquet principal a utilitzar per aquest projecte. Conté els classificadors, tots els quals implementen la *interface* `weka.classifiers.Classifier`. A més, conté els mètodes primitius de tots els classificadors que permeten realitzar l'entrenament, la classificació d'instàncies d'una bases de dades no etiquetada, o la realització de proves amb dades conegudes. Dintre dels classificadors disponibles, s'utilitzaran tots aquells que puguin treballar amb dades numèriques etiquetades.
- **Weka.clusterers:** Conté les *interfaces* utilitzades per la clusterització, mètode d'Aprenentatge No Supervisat que, per tant, no hem utilitzat en aquest projecte.
- **Weka.core:** Conté totes les classes que formen l'estructura principal de la llibreria Weka i les utilitats per realitzar la conversió de text a dades de la nostra llibreria.
- **Weka.datagenerators:** Genera els resultats de les proves o classificacions realitzades.
- **Weka.estimators:** Conté les interfaces pels estimadors de probabilitats. No utilitzat per aquest projecte.
- **Weka.experiment:** Compta les funcionalitats de l'*experimenter* de la interfície gràfica de Weka.
- **Weka.filters:** Aquest *package* Conté les classes que transformen bases de dades afegint o eliminant atributs, refent les mostres de la base de dades, eliminant exemples i més. Aquest *package* permet també el pre-processament de dades, funció molt important en l'Aprenentatge Automàtic.
- **Weka.gui:** Implementa els mètodes d'interfície d'usuari. Conté el mètode *main* que crida l'executable de Weka.

^{II}<http://www.cs.waikato.ac.nz/ml/weka/downloading.html>

^{III}Classe que simbolitza un objecte "bàsic" del qual n'hi ha diferents tipus. Especifica mètodes i propietats que haurien d'implementar la família d'objectes que la implementen.

^{IV}En aprenentatge automàtic i estadística, la selecció d'atributs és el procés de seleccionar un subconjunt de característiques pertinents per utilitzar en la construcció del model.

2.2.2 Llibreria CBR-KB

Llibreria que implementa un algorisme de **CBR** standard, l'ACBR (Adaptative Case-Based Reasoning) i diversos algorismes per al manteniment de la memòria de casos. Aquesta llibreria prové d'un projecte anterior [2], d'Irene Martí Diéguez.

La idea fonamental de l'algorisme de CBR és definir les experiències passades com a Casos, i un cop trobem un cas nou desconegut realitzar els passos del cicle CBR, que està dividit en les següents fases:

1. **Recuperar** (*Retrieve*) el cas o els casos més similars,
2. **Reutilitzar** (*Reuse*) la informació i el coneixement d'aquest cas per resoldre el nou problema,
3. **Revisar** (*Revise*) la solució obtinguda, i
4. **Retenir** (*Retain*) el nou cas per ser utilitzat en la resolució de problemes futurs.

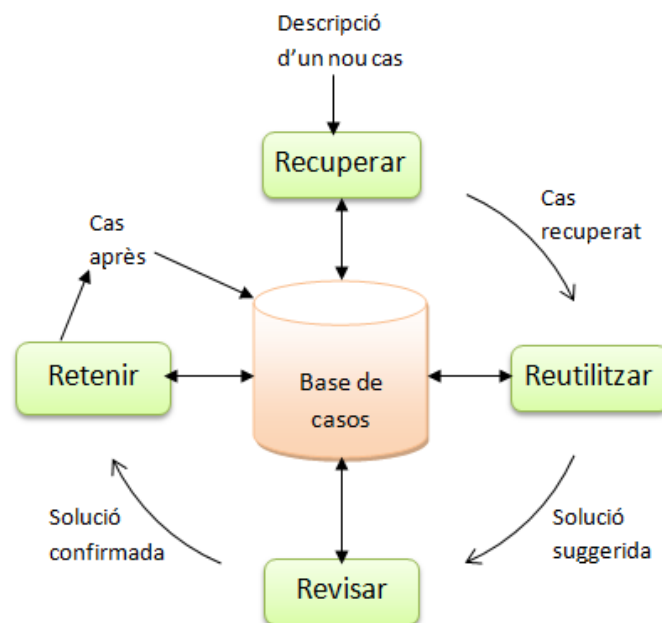


Figura 2.2: Funcionament del cicle de CBR

El codi consta de les diferents parts necessàries per implementar l'algorisme CBR, a més de la variant ACBR (Adaptative Case-Based Reasoning, Raonament Adaptatiu Basat en Casos), que no s'ha utilitzat per l'aplicació del Mòdul Pedagògic però es podria utilitzar.

El funcionament de la llibreria CBR s'ha adaptat i es realitza fent crides similars al mètode Main del codi original. Aquest funciona de la següent manera.

1. Inicialització de la base de casos (*CaseBase*) amb les dades d'entrenament.
2. Recuperació de la configuració que rebrà el CBR, que detalla les funcions internes a usar.
3. Execució del cicle CBR amb les dades a predir.

La llibreria o codi original consta de les següents classes:

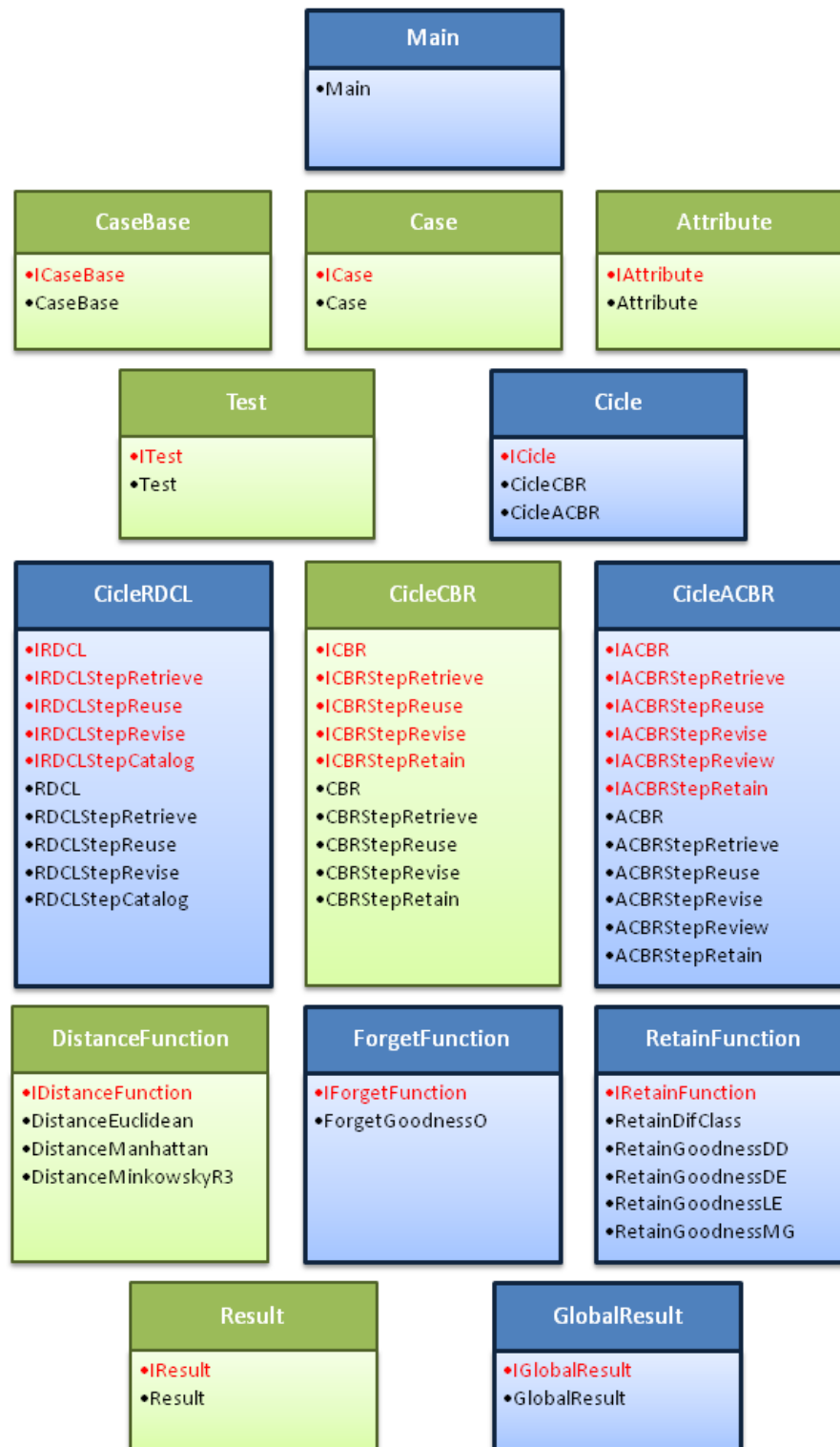


Figura 2.3: Dins dels paquets, les interfícies estan escrites en vermell i les classes en negre.

2.2.3 Java Swing

Java Swing és una llibreria publicada per Oracle com a part del llenguatge Java que permet als programadors crear interfícies gràfiques d'usuari. Forma part del **JFC**, o Java Foundation Classes, on succeeix l'anterior interfície per Java, **AWT** (*Abstract Window Toolkit*). Swing incorpora una millor aparença respecte AWT a més de la capacitat de personalitzar-la.

L'IDE ^V **NetBeans** utilitzada té l'avantatge de poder gestionar la interfície de manera gràfica, en comptes d'haver de fer-ho completament programàticament.

^VIntegrated Development Environment, entorn dissenyat per a facilitar el desenvolupament del codi d'un programa

Capítol 3

Anàlisi i disseny

En aquest capítol es tindran en compte els requisits que haurà de complir l'aplicació final, composta per les dues parts (llibreria i interfície pel professorat).

En primer lloc, es mostrarà l'estructura que ha seguit el codi implementat i es passarà, en segon lloc, a analitzar el funcionament de la interfície d'usuari.

3.1 Requisites

En el desenvolupament del projecte s'han definit un conjunt de requisits. En aquesta secció es veurà en detall els requisits funcionals, no funcionals i els diagrames de casos d'ús de la interfície pel professorat.

3.1.1 Requisites funcionals

En primer lloc, definiré els requisits funcionals de l'aplicació desenvolupada per al professorat. Aquests són:

- L'aplicació ha de rebre d'entrada:
 - Un o més fitxers excel amb les dades **completes** de les notes en els exercicis, exàmens i/o pràctiques dels alumnes d'un o més cursos anteriors de l'assignatura.
 - Un únic fitxer amb dades **incompletes** dels alumnes de l'any actual.

I retornarà de sortida el mateix fitxer d'alumnes de l'any actual amb el resultat donat.

- La interfície ha de donar l'opció a l'usuari d'utilitzar tots els classificadors de la llibreria o fer una classificació ràpida amb una selecció predefinida dels millors classificadors.
- La interfície ha de permetre a l'usuari visualitzar els fitxers de dades carregats i obrir-los.

En quant a la llibreria:

- Treballa amb el format ARFF (Attribute-Relation File Format, format específic de la llibreria Weka, i que aprofita a més la llibreria del CBR.
- Ha de ser capaç de convertir arxius del format XLS o CSV a dades del programa.
- Ha de ser capaç de tornar dades del programa a format XLS per tal de mostrar els resultats obtinguts.

- Ha de poder tant fer l'execució de proves (*Ten-Fold Cross-Validation*) com realitzar una classificació o predicció.
- Ha de poder donar a escollir si es volen utilitzar múltiples algorismes o un de sol.
- Ha de poder donar a escollir quin algorisme es vol utilitzar.

3.1.2 Requisits no funcionals

Aquí anem a veure els requisits no funcionals de la llibreria i l'aplicació per al professorat. Així tenim:

- Tant la llibreria com la aplicació per al professorat han de ser compatibles amb qualsevol Sistema Operatiu. És per això que s'ha escollit Java en el desenvolupament.
- El codi ha de ser estructurat i entenedor, i la interfície ha de ser senzilla d'usar.

Seguirà l'arquitectura **Model-Vista-Controlador**, patró simple però globalment usat en el desenvolupament de programes amb multitud de llenguatges de programació.

Això significa que es crearan les classes tenint en ment la divisió de funcions en tres paquets:

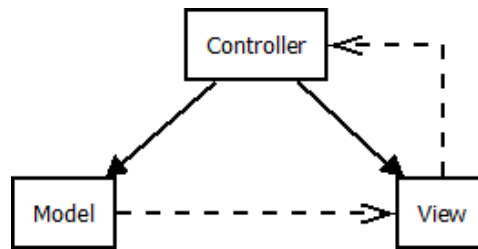


Figura 3.1: Arquitectura Model-Vista-Controlador

Model: Implementa la lògica del programa. És el responsable de recuperar les dades i convertir-les en conceptes rellevants per a l'aplicació. Això inclou processament, validació, associació i totes les altres funcions que manipulin les dades.

Controller: Proveeix una representació de les dades modelades. És la responsable de proporcionar a l'usuari una interfície amb la que treballar, amb la informació que aquest pot visualitzar.

View: Manipula les peticions de l'usuari. És la connexió que té amb la capa controlador de l'aplicació, i, per tant, l'encarregat de gestionar el fluxe de dades entre el funcionament intern de l'aplicació i l'usuari.

3.1.3 Casos d'ús de la interfície

Aquí explicarem els casos d'ús que realitzaran el professorat com a usuaris.

Els usuaris podran realitzar una quantitat de tasques limitades, que són:

- Carregar els dos tipus de dades.
- Esborrar completament tot el conjunt de dades d'un dels dos tipus.
- Obrir un dels arxius que ha carregat prèviament.

- Realitzar els dos tipus d'execucions.

Així doncs, aquestes tasques es divideixen en els següents casos d'ús:

1. Carregar dades històriques

Descripció	L'usuari vol carregar un o varis fitxers de dades com a històric
Actors	Usuari
Flux bàsic	1. L'actor selecciona el botó o menú item de carregar dades històriques 2. El sistema obre el selector d'arxius. 3. L'actor selecciona el o els fitxer a carregar
Flux alternatiu	L'actor decideix que no vol carregar dades i cancela l'operació
Postcondicions	El sistema ha carregat la informació del fitxer i mostra el fitxer entre la llista

Taula 3.1: Cas d'ús 1: Carregar dades històriques

UC 2. Carregar dades actuals

Descripció	L'usuari vol carregar un únic fitxer com a dades actuals
Actors	Usuari
Flux bàsic	1. L'actor selecciona el botó o menú item de carregar dades actuals 2. El sistema obre el selector d'arxius. 3. L'actor selecciona el fitxer a carregar
Flux alternatiu	L'actor decideix que no vol carregar dades i cancela l'operació
Postcondicions	El programa ha carregat la informació del fitxer i mostra el nom del fitxer i la quantitat d'instàncies que conté

Taula 3.2: Cas d'ús 2: Carregar dades actuals

UC 3. Eliminar totes les dades històriques

Descripció	L'usuari vol esborrar la informació carregada com a històric
Actors	Usuari
Precondicions	Hi ha fitxers de dades carregats
Flux bàsic	1. L'actor selecciona la pestanya d'històric de dades 2. El sistema mostra els botons i la llista de fitxers de dades històric. 3. L'actor selecciona el fitxer de dades a borrar i selecciona el botó d'esborrar dades 4. El sistema confirma l'esborrat de dades
Postcondicions	El programa no té cap instància de dades històriques carregat

Taula 3.3: Cas d'ús 3: Eliminar totes les dades històriques

UC 4. Eliminar totes les dades actuals

Descripció	L'usuari vol esborrar la informació carregada com a dades actuals
Actors	Usuari
Precondicions	Hi ha un fitxer de dades carregat
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor selecciona la pestanya de dades actuals 2. El sistema mostra el fitxer de dades actuals amb els botons 3. L'actor selecciona el botó d'esborrar dades 4. El sistema confirma l'esborrat de dades
Postcondicions	El programa no té cap instància de dades actuals carregat

Taula 3.4: Cas d'ús 4: Eliminar totes les dades actuals

UC 5. Visualitzar fitxer de dades de l'històric

Descripció	L'usuari vol visualitzar un fitxer de dades històric carregat
Actors	Usuari
Precondicions	Hi ha fitxers de dades carregats
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor selecciona la pestanya d'històric de dades 2. El sistema mostra el fitxer de dades actuals amb els botons 3. L'actor selecciona el nom del fitxer de dades a obrir 4. L'actor selecciona el botó d'obrir fitxer de dades
Postcondicions	S'ha obert el fitxer de dades amb el programa predeterminat de l'ordinador de l'usuari

Taula 3.5: Cas d'ús 5: Visualitzar fitxer de dades de l'històric

UC 6. Visualitzar fitxer de dades actual

Descripció	L'usuari vol visualitzar el fitxer de dades actuals carregat
Actors	Usuari
Precondicions	Hi ha un fitxer de dades carregat
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor selecciona la pestanya de dades actuals 2. L'actor selecciona el botó d'obrir fitxer de dades actual
Postcondicions	S'ha obert el fitxer de dades amb el programa predeterminat de l'ordinador de l'usuari

Taula 3.6: Cas d'ús 6: Visualitzar fitxer de dades actual

UC 7. Realitzar execució ràpida amb el classificador recomanat

Descripció	L'usuari vol utilitzar el classificador recomanat per a fer una predicció
Actors	Usuari
Precondicions	Hi ha un fitxer de dades actual carregat i un o més fitxers de dades històriques
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció d'execució ràpida 2. L'actor li dona al botó d'executar 3. El sistema passa a realitzar la predicció amb el CBR Classifier sobre les dades a predir
Postcondicions	El programa ha obtingut un resultat, que mostra automàticament si l'usuari així ho desitja

Taula 3.7: Cas d'ús 7: Realitzar execució ràpida amb el classificador recomanat

UC 8. Realitzar execució exhaustiva cridant a la bateria de proves

Descripció	L'usuari vol cridar la bateria de proves per fer una recomanació amb el teòricament millor classificador
Actors	Usuari
Precondicions	Hi ha un fitxer de dades actual carregat i un o més fitxers de dades històriques
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor selecciona l'opció d'execució exhaustiva 2. L'actor li dona al botó d'executar 3. El sistema passa a realitzar ten-fold cross-validation amb el conjunt d'entrenament, i posteriorment la predicció amb el millor classificador
Postcondicions	El programa ha obtingut un resultat, que mostra automàticament si l'usuari així ho desitja

Taula 3.8: Cas d'ús 8: Realitzar execució exhaustiva cridant a la bateria de proves

UC 9. Obrir resultats de la predicció

Descripció	L'usuari vol visualitzar els resultats obtinguts amb l'execució prèviament realitzada
Actors	Usuari
Precondicions	S'ha realitzat prèviament una execució simple o exhaustiva
Flux bàsic	<ol style="list-style-type: none"> 1. L'actor li dona al botó d'Obrir Resultats 2. El sistema passarà a obrir amb el programa predeterminat per fitxers Excel els resultats
Postcondicions	S'ha obert el fitxer de dades amb les prediccions afegides amb el programa predeterminat de l'ordinador de l'usuari

Taula 3.9: Cas d'ús 9: Obrir resultats de la predicció

3.2 Estructura de codi emprada

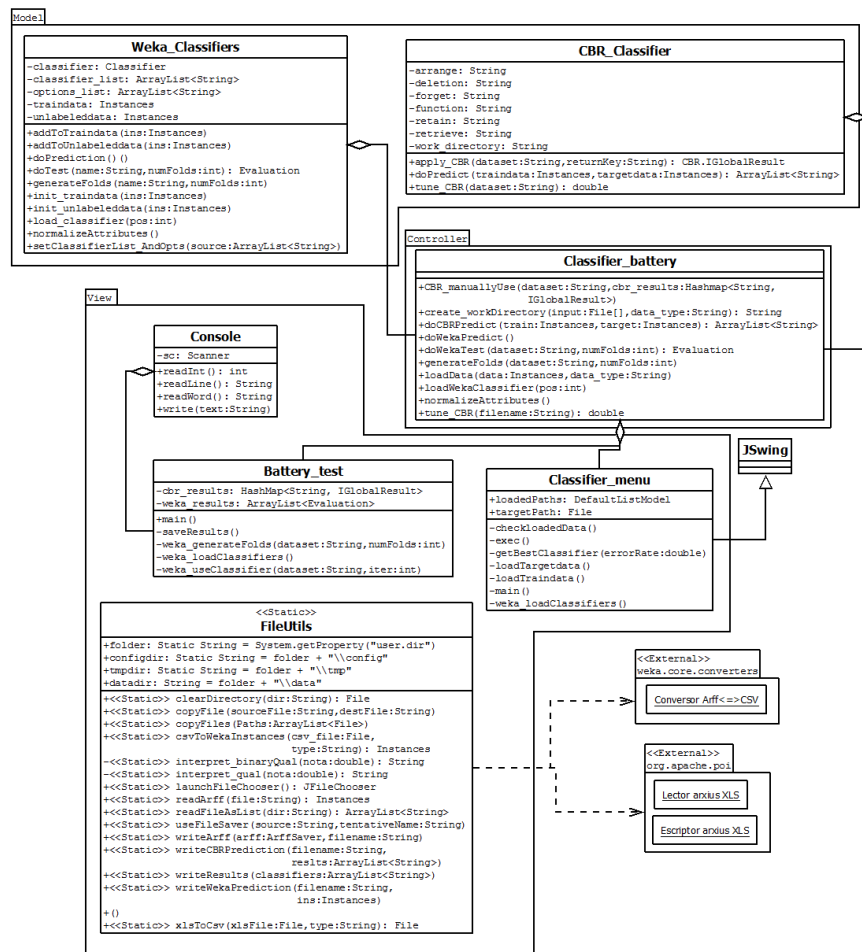


Figura 3.2: Diagrama de classes de l'aplicació

La classe FileUtils és l'encarregada de realitzar conversions de les dades d'origen a dades del programa, utilitzant les classes contenides en Weka.core.converters amb el suport del lector de fitxers Excel de la llibreria Apache POI. Aquesta conversió pot ser:

- XLS → CSV → Weka Instances
- CSV → Weka Instances
- ARFF → Weka Instances

Un cop la classe FileUtils ens ha traduït els fitxers a dades del programa, hem de valorar què farà el programa amb aquestes dades.

El codi es divideix en tres *packages* de classes:

- *Model*
 - Classe **CBR_Classifier**: Classe implementa el funcionament de la llibreria CBR.
 - Classe **Weka_Classifiers**: Classe que implementa el funcionament de la llibreria Weka
- *Controller*

- Classe **Classifier_battery**: Classe que realitza l'encapsulament de les crides realitzades a les llibreries dels sistemes classificadors de weka i a la llibreria de l'algorisme CBR.
- *View*
 - Classe **Battery_test**: Classe que implementa el funcionament de la bateria de proves per a l'usuari.
 - Classe **Classifier_menu**: Classe que implementa el funcionament de la interfície pel professorat.
 - Classe **Console**: Classe que implementa l'ús de la consola del programa, tant d'entrada com de sortida.
 - Classe **FileUtils**: Classe estàtica que conté qualsevol interacció amb fitxers del programa.

Com a punt final, és important remarcar que tant la classe **Battery_test** com la classe **Classifier_menu** contenen mètodes **main**. En concret, **Battery_test** llença una bateria de proves de tots els classificadors amb una única base de dades i guarda el percentatge d'encerts, mentres que **Classifier_menu** és l'execució del programa que utilitzarà el professorat. Aquest **Classifier_menu** agafa una quantitat indefinida de fitxers d'entrenament i un únic fitxer de dades a predir per a realitzar la classificació.

Ambdós mètodes **main** funcionen utilitzant la classe **Classifier_battery**, que Conté les crides a l'algorisme CBR (**CBR_Classifier**) i a la classe que gestiona el funcionament de les crides a la llibreria de classificadors de Weka, **Weka_Classifiers**.

Dintre d'aquesta llibreria s'han utilitzat totes les crides que permet fer el programa per a fitxers amb dades **númeriques etiquetades**, amb la configuració que utilitza per defecte Weka.

Per fer la classificació, en el cas del conjunt d'entrenament la llibreria pot rebre un nombre de fitxers indefinits. Aquests fitxers en un cas ideal haurien de tenir exactament la mateixa quantitat informació entre ells per cada alumne, ja que d'altra manera provoca soroll en la predicció. Això es deu a que la llibreria de Weka defineix com a valors que falten aquells atributs que no es troben en unes instàncies però sí en altres. El tractament dels valors que falten és diferent segons l'algorisme, de manera que les conseqüències són diferents en cada cas.

El conjunt a predir es troba en un únic fitxer, de manera que no hi hauria d'haver problema a l'hora de carregar-los com una sola base de dades.

Addicionalment, a l'hora de realitzar la predicció hem de normalitzar el conjunt d'entrenament i el de predicció, ja que tenen una quantitat diferent de dades, o la classificació no seria necessària. Això s'ha resolt eliminant tots aquells atributs que no tinguin en comú els dos conjunts durant el curs de l'execució. Un cop acabada es retornen les dades a l'estat inicial per si es vol realitzar una execució diferent.

Capítol 4

Experimentació i anàlisi de resultats

4.1 Detalls dels experiments

En aquest capítol s'avaluarà la llibreria del sistema de predicció del Mòdul Pedagògic realitzant 4 experiments. A continuació es detalla en què consisteix cadascun dels experiments i, posteriorment, les següents seccions mostraran els resultats obtinguts.

- **Experiment 1:** Utilitzant bases de dades del repositori UCI ^I, es faran *ten-fold cross-validation* ^{II}, o proves creuades per a posar a prova la llibreria de classificadors. Es mostraran els resultats dels 5 classificadors més precisos per a cada base de dades, amb el seu percentatge d'encerts. Per a les bases de dades amb quantitat de classes binària es mostrarà a més la precisió i sensibilitat de cadascuna de les dues classes.
- **Experiment 2:** Es tornaran a realitzar *ten-fold cross-validation* però aquest cop amb conjunts de dades de les assignatures. Aquestes són:
 - Disseny de Software (DSW)
 - Fonaments de Tecnologia (FDT)
 - Programació 1 (P1)
 - TIC

D'entre els diferents cursos disponibles s'ha escollit realitzar les proves sobre l'últim curs disponible, ja que és aquest sobre el qual es realitzaran les prediccions dels següents apartats.

Es distingiran dos tipus de classificacions: binària, i per rang.

- **Experiment 3:** Es provarà l'eficàcia del sistema de predicció a l'hora d'intentar predir l'evolució de l'alumnat. Per a això s'agafarà les bases de dades de les assignatures, utilitzant com a dades a predir l'últim curs disponible i com a conjunt d'entrenament el penúltim curs disponible.

^I<http://archive.ics.uci.edu/ml/>

^{II}Proves consistents en: 1. Partir les dades en 10 sets de tamany $n/10$. 2. Entrenar amb 9 datasets i agafar com a mostra de prova 1. 3. Repetir 10 times i agafar una precisió mitjana

- **Experiment 4:** Un cop més, s'aplicarà el sistema de predicció a l'hora de predir l'evolució de l'alumnat. Aquest cop, però, s'ha reduït la quantitat de notes (columnes de la Taula 2.1) per a simular un cas més probable.

4.2 Experiment 1: Anàlisi de resultats amb bases de dades del repositori UCI

Les primeres proves realitzades per provar l'eficàcia de la llibreria s'han realitzat utilitzant les bases de dades del repositori **UCI** (University of California at Irvine). D'entre els *datasets* dades disponibles, s'han escollit les següents:

Nom dataset	Descripció
Iris	Base de dades petita i simple d'utilitzar que sol ser utilitzada per provar reconeixement de patrons. Conté 3 classes de 50 instàncies cada una, on cada classe es refereix a un tipus de planta de la família Iris. Una de les classes és fàcilment distingible de les altres dues, mentres que aquestes dues són més difícils de distingir entre elles.
Glass	Base de dades el estudi de la qual va ser motivada per investigacions criminològiques (identificar correctament el vidre trobat a l'escena del crim pot servir d'evidència). Conté 6 tipus de vidre, definits en termes del contingut del seu òxid, i per tant el procés de classificació serà reconèixer aquest òxid i el tipus de vidre a que es correspon
Vehicle	Característiques extretes de siluetes de cotxes per una extensió de la HIPS (Hierarchical Image Processing System). Corresponent a dades d'imatges extretes de cotxes, pintats en negre i amb una càmera i els cotxes en una posició fixada. Inclou diferents marques de cotxes, les quals s'han de distingir a partir de les característiques proporcionades.
Ionosphere	Informació d'un radar capturat per 16 antenes d'alta freqüència. Els objectius foren els electrons lliures trobats a la ionosfera. El procés de classificació pretén distingir les "bones captures i les "males captures. Es consideren bones captures aquelles que mostren evidència d'algun tipus d'estructura en la ionosfera, mentres que es consideren males captures les que no ho fan.
Sonar	Base de dades de classificació de senyals de sonar. El procés de classificació ha de distingir si la senyal ha topat amb una roca o una mina, utilitzant la informació obtinguda de diferents senyals, que augmenten seqüencialment la seva freqüència. Aquesta informació és l'energia dins d'una banda de freqüència particular, integrada al llarg d'un període de temps concret.

Taula 4.1: Informació bàsica sobre les bases de dades utilitzades del repositori UCI

S'han escollit aquests datasets per provar els classificadors amb bases de dades variades (nombre d'instàncies diferent, quantitat de classes i atributs dissímils, i temàtica del dataset aïllada, el que significa que les dades tindran patrons totalment diferents).

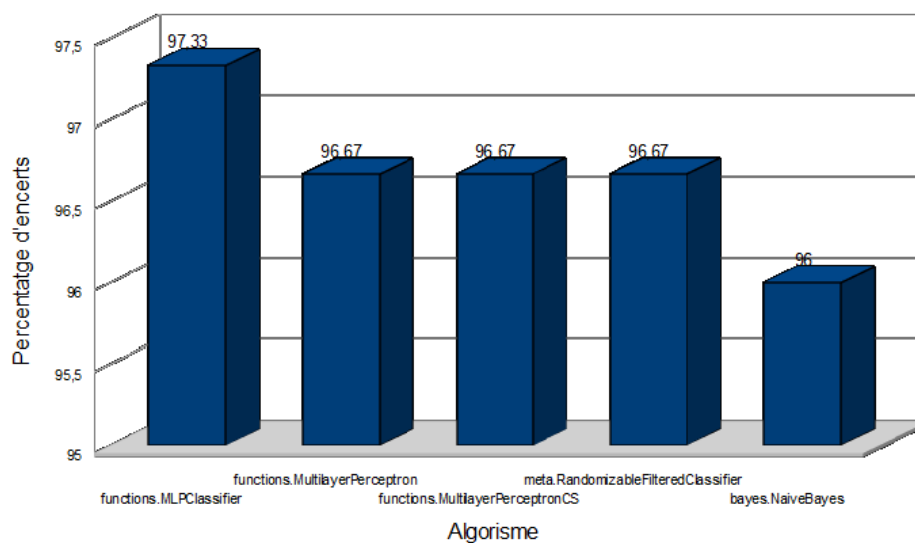


Figura 4.1: Resultats dels 5 millors classificadors per a la base de dades Iris, 150 instàncies a classificar

A la Figura 4.1 es pot veure que els millors classificadors de la llibreria per al dataset Iris han estat: MLPClassifier, MultilayerPerceptron, MultilayerPerceptronCS, RandomizableFilteredClassifier i NaiveBayes. El millor obté un 97.33% com a percentatge d'encerts mentre que el cinquè millor ha obtingut un 96%.

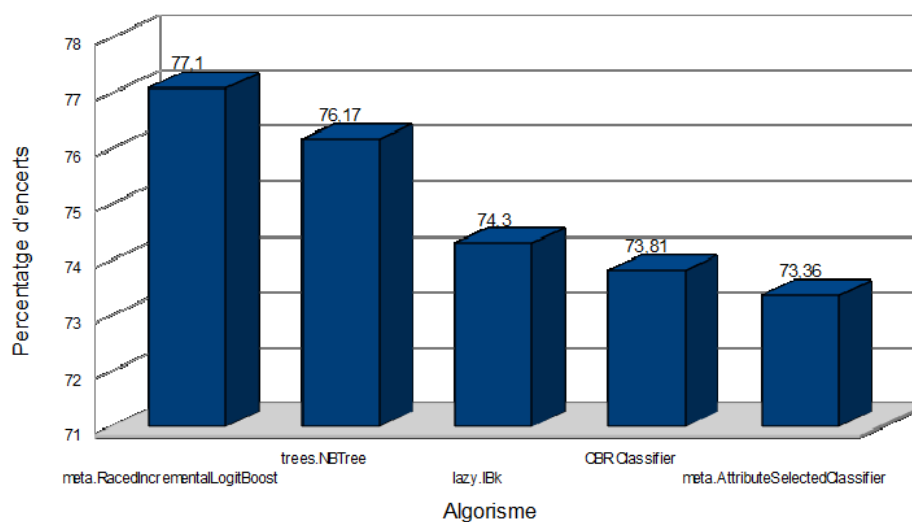


Figura 4.2: Resultats dels 5 millors classificadors per a la base de dades Glass, 214 instàncies a classificar

Podem veure a la Figura 4.2 quins han estat els 5 millors classificadors d'entre la llibreria, obtinguts en la classificació del dataset glass, resultats completament diferents que els obtinguts amb les proves de l'Iris. Aquests classificadors són: RacedIncrementalBoost, NBTree, IBk, CBR Classifier i AttributeSelectedClassifier. El millor classificador ha obtingut un 77,1% d'encerts mentre que el cinquè millor tenia un 73.36% de percentge d'encerts.

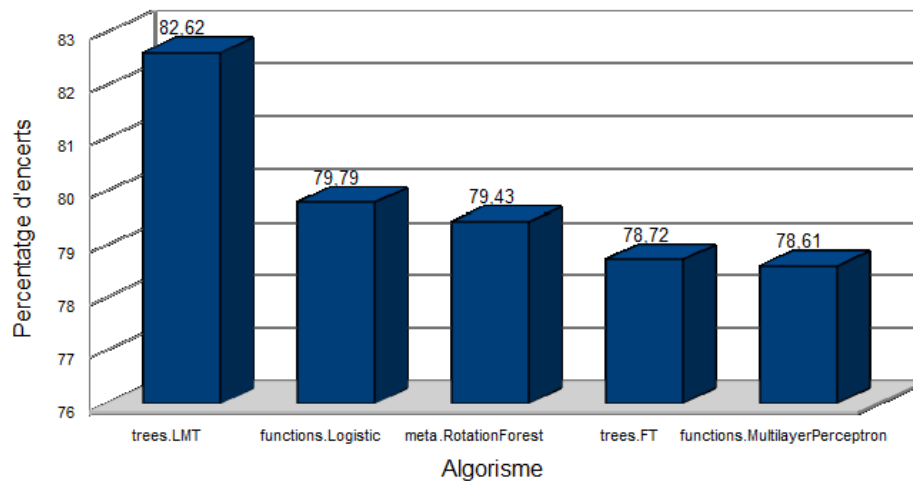


Figura 4.3: Resultats dels 5 millors classificadors per a la base de dades Vehicle, 846 instàncies a classificar

A la Figura 4.3 veiem els 5 millors classificadors dels de la llibreria pel dataset Vehicle, classificadors que són gairebé tots diferents que en els demés datasets anteriors. Aquests classificadors són: LMT, Logistic, RotationForest, FT i MultilayerPerceptron. Únicament el MultilayerPerceptron coincideix, prèviament el segon millor classificador per al dataset Iris. El millor classificador té un 82.62% d'encerts i el cinquè millor té un 78.61%.

Una altra mesura que s'ha utilitzat per valorar les bases de dades amb quantitat binària de classes és la precisió (precision) i la sensibilitat (recall) de cadascuna de les dues classes.

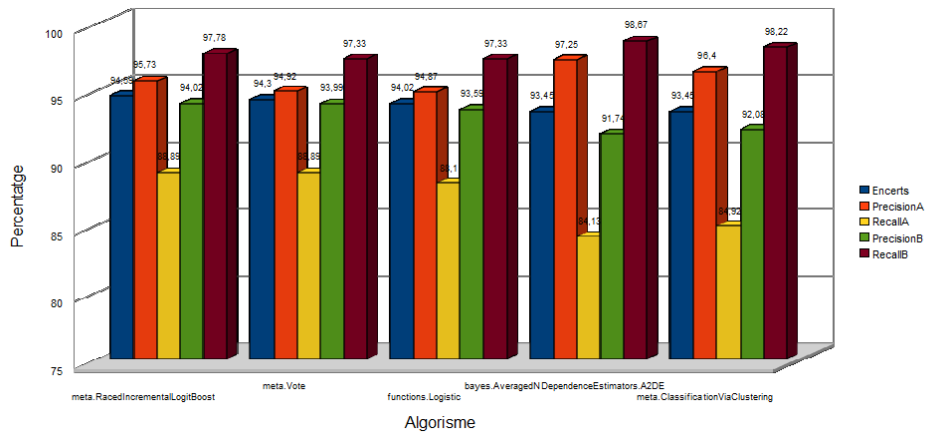


Figura 4.4: Resultats dels 5 millors classificadors per a la base de dades Ionosphere, 351 instàncies a classificar

La Figura mostra 4.4 els 5 classificadors de la llibreria amb més alt percentatge d'encerts per al dataset Ionosphere: RacedIncrementalBoost, Vote, Logistic, A2DE, ClassificationViaClustering. D'entre aquests trobem dues coincidències amb els datasets anteriors: RacedIncrementalBoost, obtingut prèviament com a millor classificador del dataset Glass, i Logistic, el segon millor de Vehicle. El percentatge d'encerts més alt és 94,59% i el cinquè millor és 93,45%, diferència mínima. Un altre característica curiosa a comentar és que els classificadors semblen ser molt més sensibles a la classe B (corresponent a "bones lectures") que a la classe A ("males lectures").

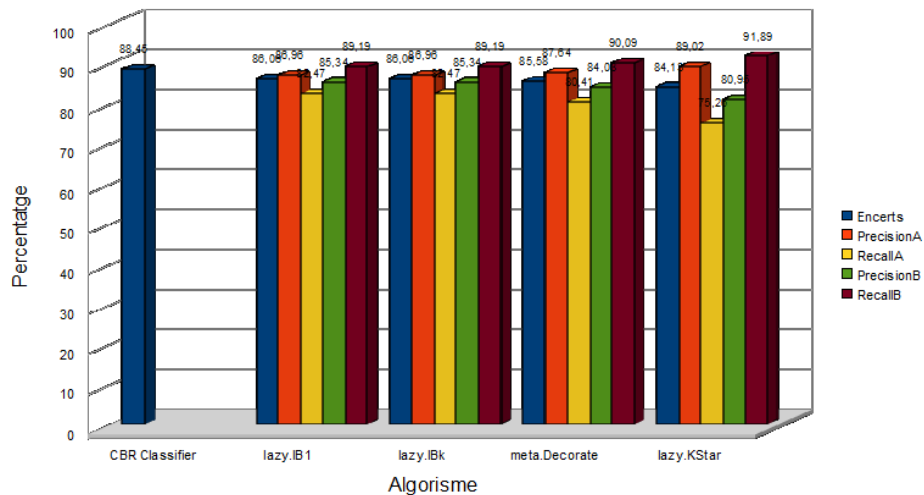


Figura 4.5: Resultats dels 5 millors classificadors per a la base de dades Sonar, 208 instàncies a classificar

Comentar primer de tot que en la Figura 4.5 es mostra diferent el CBR Classifier dels demás algorismes ja que la llibreria actual no em permetia extreure la matriu de confusió, necessària per calcular la precisió i la sensibilitat. Un cop dit això, en aquesta mateixa Figura 4.5 podem veure els 5 classificadors de la llibreria amb un percentatge d'encert més alt per al dataset Sonar. Aquests són: CBR Classifier, IB1, IBk, Decorate i KStar. L'única coincidència amb els datasets anteriors és el CBR, quart millor del dataset Glass. El percentatge d'encerts més alt obtingut és 88,46% i el cinquè millor és el 84,13%. En aquest cas veiem que les precisions i sensibilitats estan més equilibrades que en el dataset Ionosphere, encara que com més baixa el percentatge d'encerts, més augmenta la sensibilitat de la classe B ("Mina") en comparació a la classe A ("Pedra"), portant-nos a pensar que l'increment d'errors pot ser degut a que li treu massa importància a etiquetar com a classes A.

En quant al conjunt de proves en total, s'han trobat els millors resultats amb la base de dades Iris, com s'indica a la descripció del conjunt de dades. Com podem comprovar amb la següent matriu de confusió

50	0	0
0	47	3
0	1	49

Taula 4.2: Matriu de confusió de l'algorisme MLPClassifier amb la base de dades Iris

Els classificadors no solen equivocar-se en la primera classe, el que ens facilita un 33% mínim d'encerts i ens deixa amb dues úniques classes que classificar.

En les bases de dades amb més atributs i classes, Glass i Vehicle, continuem tenint un percentatge d'encerts superior al 75%, mentre que amb les bases de dades d'Ionosphere i Sonar ens trobem al voltant del rang del 85-90% d'encerts.

Finalment, es fa una comparació del CBR Classifier versus els millors classificadors de Weka per a cada conjunt de dades del repositori UCI.

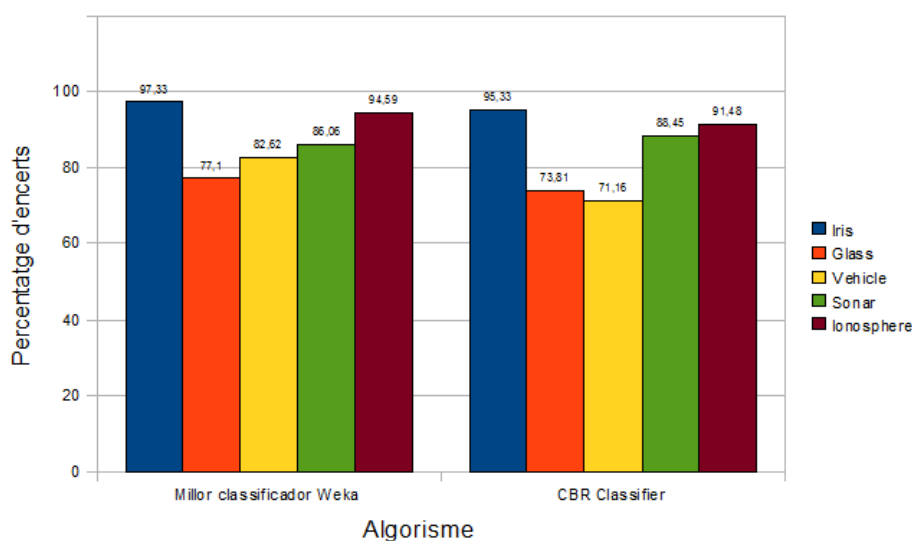


Figura 4.6: Comparació del percentatge d'encerts més alt per a cada base de dades (classificació normal), amb el percentatge d'encerts del CBR.

Com podem veure, el CBR no es queda enrere en cap de les classificacions, amb una diferència menor del 5% en 4 de les 5 bases de dades.

És important tenir en compte aquesta comparació ja que ens fa veure que el CBR fa classificacions consistents, a diferència d'altres algorismes que poden perdre precisió depenent de la base de dades. És per això que l'hem escollit com a classificador predefinit per a les proves dels dos últims apartats.

4.3 Experiment 2: Anàlisi de resultats amb bases de dades de les assignatures

El segon experiment s'ha realitzat fent *ten-fold cross-validation* les notes de que disposem per a cada assignatura.

En el següents apartats es treballarà mirant de predir les notes de l'últim curs de cada assignatura, de manera que realitzarem el *ten-fold cross-validation* també sobre aquest mateix conjunt. S'ha escollit realitzar les proves sobre l'últim curs acadèmic ja que és el que, hipotèticament, tindria un format de notes més semblant a les trobades en el curs actual.

Es distingiran dues classificacions diferents amb que provar els classificadors:

1. Classificació per rang amb les notes acadèmiques del curs
2. Classificació binària (aprovat/suspens)

En un principi, la segona classificació hauria de ser més simple, ja que hi ha menys lloc a error (menys classes possibles per escollir, més instàncies que estudiar com a classe aprovats), mentres que la primera pot donar complicacions depenent de l'assignatura.

Per mostrear els resultats s'han agafat els 5 classificadors amb percentatge d'encert més alts de la classificació per rang junt als seus resultats al fer la classificació binària. Addicionalment, per a cada assignatura s'ha inclòs la precisió i la sensibilitat de les dues classes (aprovat/suspens) obtingudes en la classificació binària.

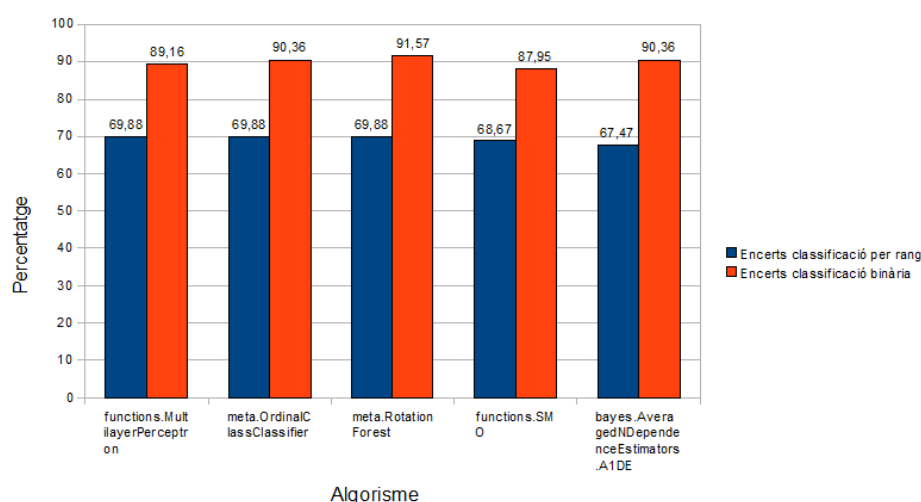


Figura 4.7: Resultats de realitzar Ten-Fold Cross-Validation a les dades de Disseny de Software, any 2013-2014, 83 instàncies a classificar

Com podem observar a la Figura 4.7, la classificació binària per DSW ha estat bastant satisfactòria mentre que la classificació per rang ha tingut uns resultats relativament baixos. Els millors classificadors per a la classificació per rang han estat: MultilayerPerceptron, OrdinalClassClassifier, RotationForest, SMO i A1DE. El percentatge d'encerts més alt ha estat per a la classificació per rang un 69,88% i per a la classificació binària un 91,57% (dintre dels 5 millors classificadors per a la classificació per rang), i el cinquè percentatge més alt ha estat 67,47% per a la classificació per rang i 87,95% per a la classificació binària.

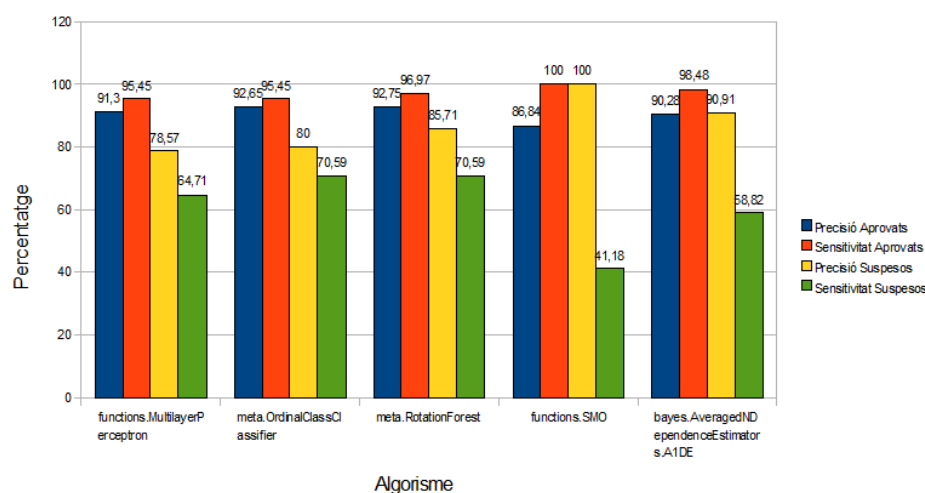


Figura 4.8: Precision i Recall corresponent a la classificació binària de Disseny de Software, any 2013-2014

A la Figura 4.8 podem observar que la sensibilitat per a suspesos és molt baixa (entre un 41,18% i un 70,59%), el que significa que la font dels errors en la classificació binària és per haver identificat alumnes suspesos com a aprovats. En el treball futur es volen analitzar diferents propostes per augmentar la sensibilitat dels algorismes de classificació.

A la Figura 4.9 podem observar els 5 millors classificadors per al curs acadèmic 2012-2013

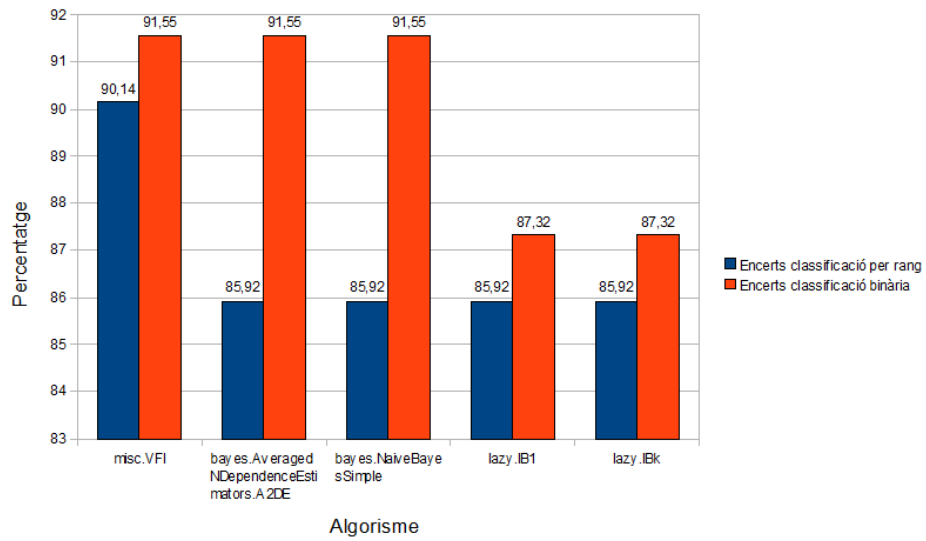


Figura 4.9: Resultats de realitzar Ten-Fold Cross-Validation a les dades de Fonaments de Tecnologia, any 2012-2013, 71 instàncies a classificar

de FDT. Aquests són: VFI, A2DE, NaiveBayesSimple, IB1, IBK. El percentatge d'encerts més alt per a la classificació per rang és del 90,14%, mentre que el cinquè més alt és del 85,92%. D'entre aquests classificadors, el millor executant la classificació binària ha obtingut un 91,55%, i el pitjor un 87,32%.

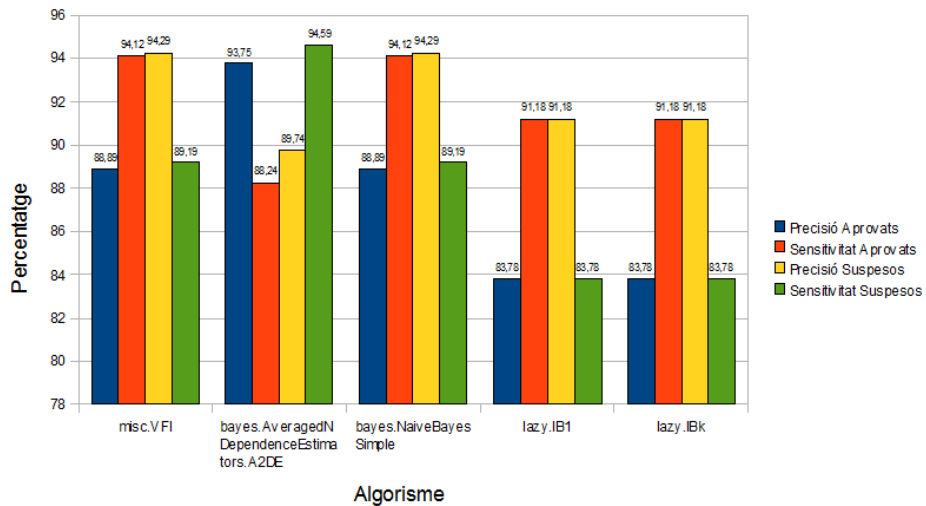


Figura 4.10: Precision i Recall corresponent a la classificació binària de Fonaments de Tecnologia, any 2012-2013

A la Figura 4.10 podem observar que la precisió i sensitivitat de la classificació binària d'FDT és força equilibrada, encara estan però lleugerament decantada cap al cantó de la falta de sensitivitat dels suspesos.

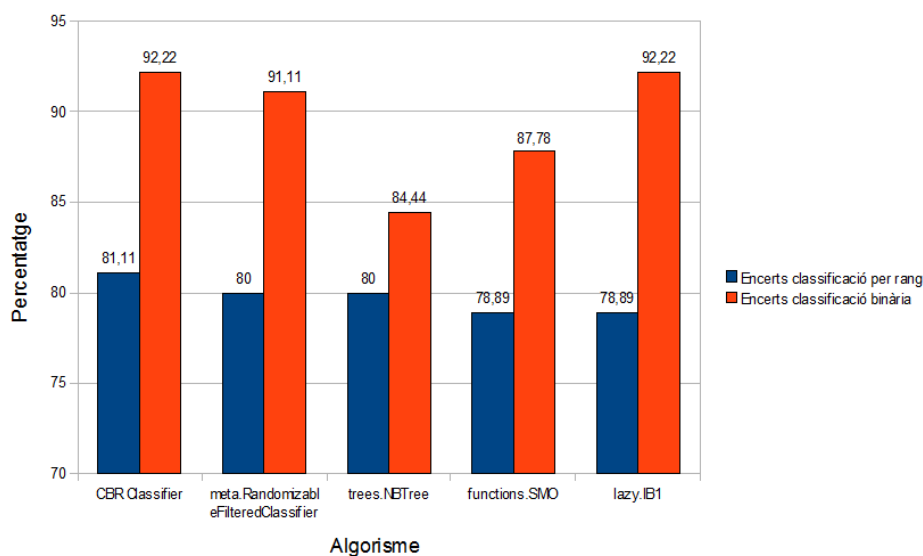


Figura 4.11: Resultats de realitzar Ten-Fold Cross-Validation a les dades de Programació 1, any 2013-2014, 90 instàncies a classificar

Com podem observar a la Figura 4.11, els classificadors amb més alt percentatge d'encerts per al curs acadèmic 2013-2014 de P1 han estat: CBR Classifier, RandomizableFilteredClassifier, NBTree, SMO i IB1, amb uns percentatges d'encerts molt semblants tots, tots compresos entre un 81,11% i un 78,89% per a la classificació per rang i entre un 92,22% i un 84,44% per a la classificació binària. L'única coincidència dels classificadors amb els millors classificadors d'assignatures anteriors és l'IB1, quart millor classificador de FDT.

A la Figura 4.12 podem veure una diferenciació amb les classificacions binàries de les assignatures anteriors: en aquest cas, la sensibilitat és menor pels aprovats que pels suspesos. Malgrat tot, la diferència no és prou gran com per suposar complicacions, i la gràfica mostra una classificació bastant equilibrada.

Als resultats mostrats a la Figura 4.13 es veu que els millors classificadors per a classificar per rang el curs 2013-2014 de TIC són A2DE, A1DE, HoeffdingTree, BayesNet i NaiveBayes amb una diferència mínima d'entre 77,27% i 74,24% d'encerts. A la classificació binària també veiem poca diferència entre aquests classificadors, tenint el millor un 90,91% d'encerts i el pitjor d'entre ells 86,36%.

A la Figura 4.14 veiem un patró completament oposat a les dues primeres assignatures. En aquest cas, la sensibilitat és més baixa en tots els classificadors per als aprovats, mentre que la classe de suspesos no tenen menys de 90% de sensibilitat.

Analitzant totes les gràfiques juntes observem que fer una classificació binària és bastant simple en la majoria dels casos. Les dificultats es solen trobar en aquelles notes que es troben en la vora entre l'aprovat i el suspès, i en aquells casos extrems d'alumnes que han tingut una caiguda en picat de les seves notes durant el curs acadèmic.

En canvi, a la classificació per rang tenim bastanta més dificultat. Això pot ser degut a que hi ha moltes més vores que entre classes (5, 6, 7 i 9) i que a més a més cada classe té menys instàncies de les que aprendre.

Dintre de totes les assignatures a analitzar s'ha tingut especialment dificultat amb la base de dades de DSW. En aquesta assignatura, no s'ha aconseguit fer una classificació per rang superior al 69,88%. Per altra banda, a l'assignatura d'FDT s'ha aconseguit arribar a un considerablement alt 90,14% d'encerts amb la classificació de rang.

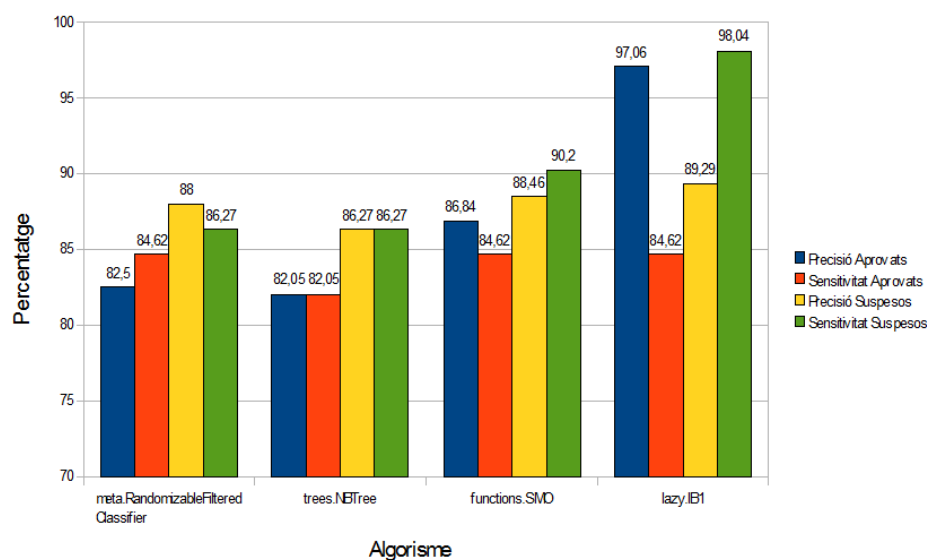


Figura 4.12: Precisió i Recall corresponent a la classificació binària de Programació 1, any 2013-2014

En quant a la classificació binària, cap base de dades ha tingut especialment problemes a l'hora de realitzar la prova. El mínim obtingut en els classificadors analitzats ha estat 84,44% en l'assignatura de programació 1, obtingut pel classificador NBTree.

4.4 Experiment 3: Anàlisi de resultats usant varis cursos de les assignatures

En aquest apartat s'han realitzat proves del sistema de predicció implementat. S'han utilitzat el conjunt de dades de cada assignatura amb varis cursos disponibles (DSW, FDT i P1), agafant el penúltim curs acadèmic com a conjunt d'entrenament i l'últim curs acadèmic com a conjunt de dades a predir. La llibreria retorna amb aquestes dades el fitxer d'origen amb una nova columna amb la predicció de la nota final.

S'han aplicat els dos tipus de classificació explicats prèviament (per rang i binària), inicialment definits a l'apartat 2.1, per a fer les prediccions. Per a cadascuna d'aquestes prediccions s'han fet dues execucions, aquestes són:

- La primera execució utilitza el sistema de predicció de la llibreria CBR amb els paràmetres definits per defecte per Irene Martí Diéguez.
- La segona execució realitza proves amb el conjunt de dades d'entrenament amb tota la bateria de classificadors de Weka i tots els possibles paràmetres del CBR, i es queda amb el que ha donat un percentatge d'encerts més alt per a realitzar la predicció.

Així doncs, passant als resultats gràfics, extraïem:

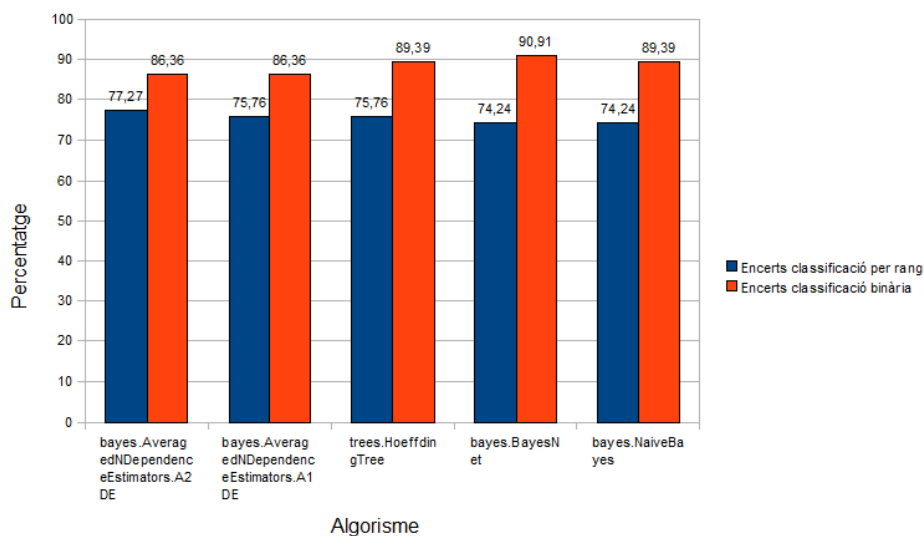


Figura 4.13: Resultats de realitzar Ten-Fold Cross-Validation a les dades de TIC, any 2013-2014, 71 instàncies a classificar

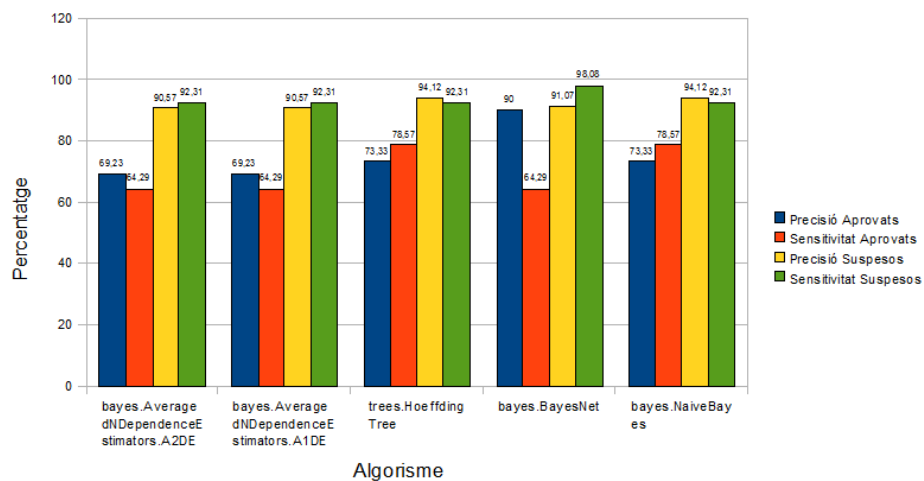


Figura 4.14: Precisió i Recall corresponent a la classificació binària de TIC, any 2013-2014

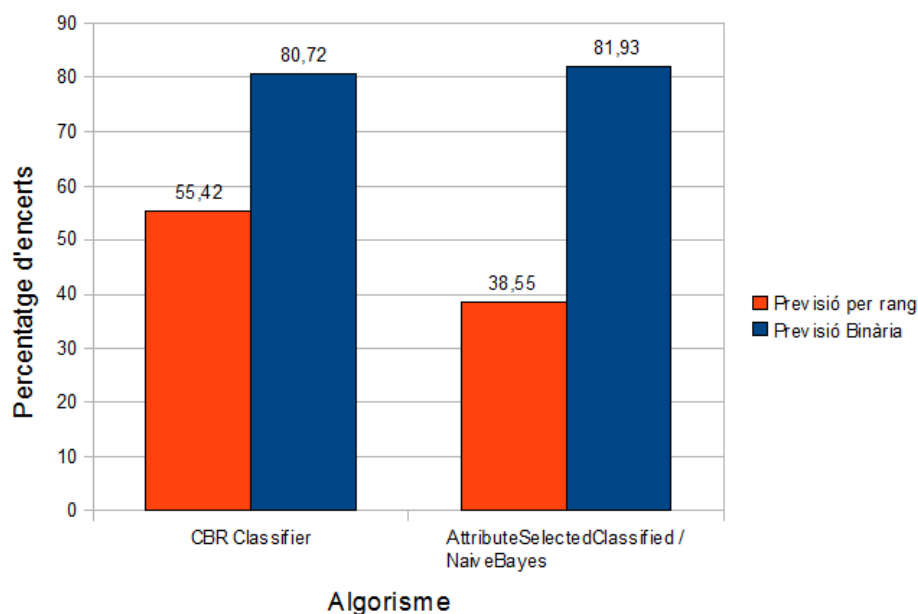


Figura 4.15: Percentatge d'encerts intentant predir el resultat de la base de dades de DSW any 2013-2014 utilitzant la base de dades de DSW any 2012-2013 com a entrenament

De les dades de DSW, s'han utilitzat tant d'entrenament com de dades a predir les columnes P1, P2, P3, E1, E2, E3, Parcial1. Les dades estan completes en el conjunt a predir, de manera que no hauria de ser massa complicat pel classificador trobar l'etiquetatge correcte.

La Figura 4.15 ens mostra però que no ha estat així per a la classificació per rang. El classificador que havia obtingut un millor resultat en les *ten-fold cross-validation* automàtiques de la interfície (AttributeSelectedClassifier) ha obtingut un 38,55% d'encerts, i el CBR ha obtingut un 55,42% d'encerts. La classificació binària ha estat més simple, obtenint un 81,93% d'encerts amb el CBR i un 80,72% d'encerts amb el NaiveBayes.

De les dades de FDT, s'han utilitzat tant d'entrenament com de dades a predir les columnes P1, P2, P3, Parcial1 i Parcial2.

En aquest cas, com podem veure a la Figura 4.17, tot i tenir menys dades hem tingut uns resultats considerablement millors que en l'assignatura de DSW. La classificació de rang ha obtingut un 71,83% d'encerts per al CBR i un 70,42% d'encerts per a l'execució exhaustiva, corresponent a SimpleLogistic. L'execució de la classificació binària ha obtingut un 87,32% d'encerts i un 77,46% d'encerts respectivament, amb el RotationForest com a millor classificador de Weka.

De les dades de P1, s'han utilitzat tant d'entrenament com de dades a predir les columnes E1, E2, E3, E4, Parcial1, Parcial2.

En la Figura 4.17 veiem com en aquest cas el CBR ha obtingut pocs encerts (45,56%) a l'hora de realitzar la classificació per rang, però l'execució exhaustiva, corresponent al A1DE, ha obtingut un considerable 74,44%. L'execució exhaustiva ha estat més precisa també per a la classificació binària, resultant en un 90%, xifra per sobre del 78,89% del CBR.

Observant totes les gràfiques, el que podem extreure és que tenir més quantitat de dades no té per què significar que la predicció obtindrà un alt percentatge d'encerts, al menys amb un conjunt de dades d'entrenament limitat. El fet que en l'assignatura de DSW es produeixin tants errors tot i ser la que conté més columnes ens dona a pensar que algunes de les columnes no són prou rellevants de cara a la nota final, cosa que el sistema de predicció no arriba a

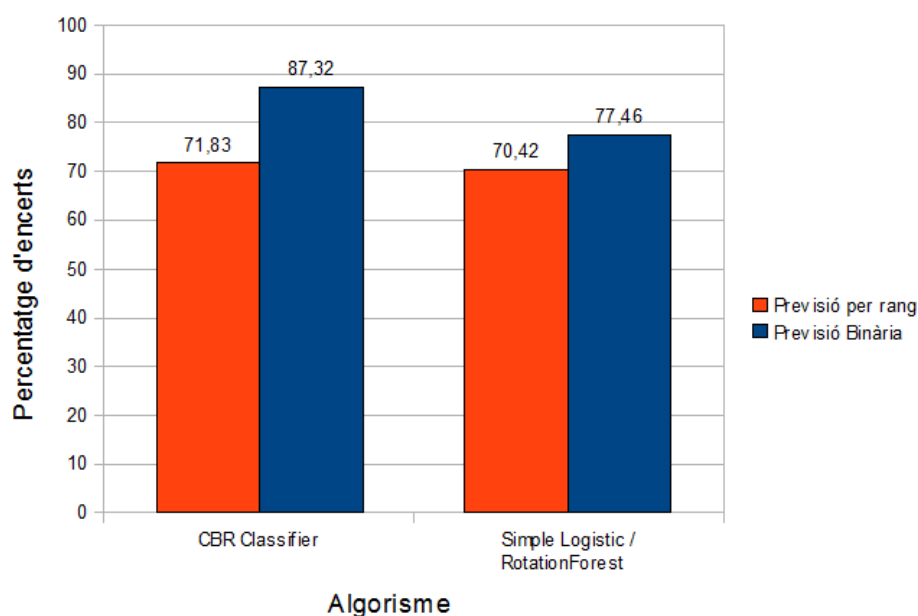


Figura 4.16: Percentatge d'encerts intentant predir el resultat de la base de dades de FDT any 2012-2013 utilitzant la base de dades de FDT any 2011-2012 com a entrenament

veure amb les dades limitades de què disposa.

4.5 Experiment 4: Anàlisi de resultats reduint el conjunt de proves de les assignatures

En aquest apartat s'han realitzat les segones proves del sistema de predicció implementat. Aquestes segueixen l'idea de simular l'ús de la llibreria que realitzarà en el professorat amb la interfície. Per això, s'han utilitzat el conjunt de dades de cada assignatura amb varis cursos disponibles (DSW, FDT i P1), agafant el penúltim curs acadèmic com a conjunt d'entrenament i l'últim curs acadèmic com a conjunt de dades a predir, eliminant columnes per a simular el curs inacabat. S'ha eliminat una columna de les dades a predir de cada apartat disponible (exercicis, problemes, nota dels parcials). La llibreria retorna amb aquestes dades el fitxer d'origen amb una nova columna amb la predicció de la nota final.

És important remarcar que eliminant columnes de les dades a predir, també estarem traient-les del conjunt de dades a predir, ja que el pas previ a la predicció és normalitzar els conjunts de dades per deixar únicament les columnes en comú

S'han aplicat els mateixos dos tipus de classificació explicats prèviament (per rang i binària), inicialment definits a l'apartat 2.1, per a fer les prediccions. Per a cadascuna d'aquestes prediccions s'han fet dues execucions, aquestes són:

- La primera execució utilitza el sistema de predicció de la llibreria CBR amb els paràmetres definits per defecte per Irene Martí Diéguez.
- La segona execució realitza proves amb el conjunt de dades d'entrenament amb tota la bateria de classificadors de Weka i tots els possibles paràmetres del CBR, i es queda amb el que ha donat un percentatge d'encerts més alt per a realitzar la predicció.

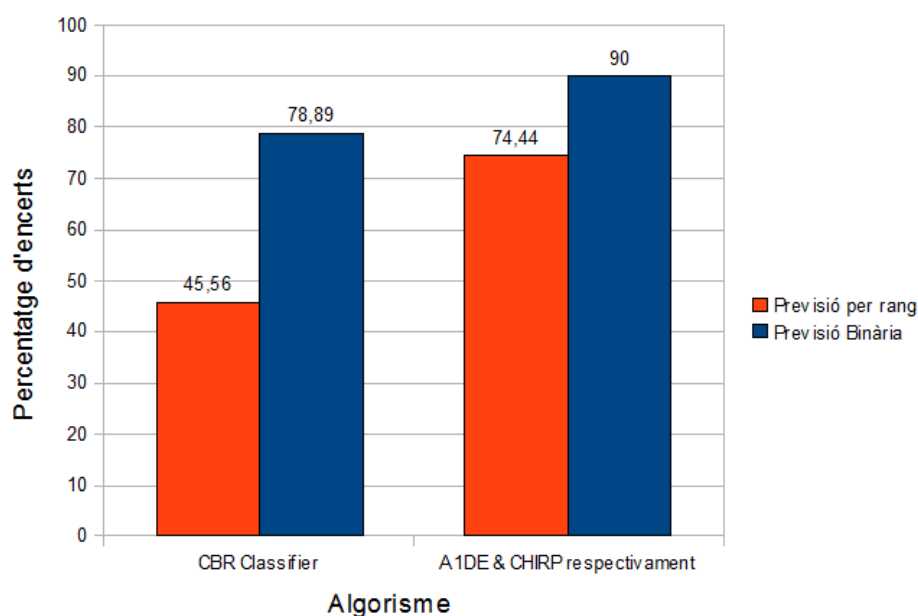


Figura 4.17: Percentatge d'encerts intentant predir el resultat de la base de dades de P1 any 2013-2014 utilitzant la base de dades de P1 any 2012-2013 com a entrenament

Els resultats que hem obtingut són els següents:

De les dades de DSW s'han utilitzat les columnes P1, P2, E1, E2, Parcial1.

A la Figura 4.18 podem observar que continuem sense obtenir resultats poc precisos per a la predicció del curs 2013-2014 de DSW. S'ha obtingut un 51,81% d'encerts per a la predicció per rang del CBR i un 40,96% amb l'MLPClassifier. A la classificació binària s'ha obtingut un 83,13% i un 87,95% d'encerts respectivament, sent l'FT el millor classificador de Weka per aquesta classificació.

És important destacar que, tot i tenir menys dades, ambdues classificacions binàries i la segona classificació per rang han donat totes més encerts que en les proves de l'Experiment 3, de l'apartat 4.4. Això ens dona a pensar que una de les columnes eliminades (P3 i E3) no té rellevància de cara a la nota final.

De les dades de FDT, s'han utilitzat tant d'entrenament com de dades a predir les columnes E1, E2, E3, Parcial1.

A la Figura 4.19 hem pogut observar que el millor classificador de Weka ha estat el ClassificationViaRegression, amb 71,83% d'encerts en la classificació binària en comparació al 73,24% del CBR. Per a la classificació binària, el millor classificador de Weka ha estat el DataNearBalancedND, amb un 90,14% d'encerts, i el CBR ha obtingut un 81,69% d'encerts.

En comparació als resultats obtinguts a l'apartat 4.4, podem veure que la classificació per rang ha obtingut una millora encara que mínima (inferior del 2% en ambdós casos). Per a la classificació binària, però, s'ha trobat una millora considerable, de 6% i 13% d'encerts respectivament.

De les dades de P1, s'han utilitzat tant d'entrenament com de dades a predir les columnes E1, E2, E3, Parcial1.

A la Figura 4.20 podem observar com BayesNet ha estat el millor classificador de Weka per ambdues classificacions, amb un 63,33% d'encerts per a la classificació per rang i un 77,78% d'encerts per a la classificació binària. En comparació, el CBR ha resultat un 54,44% en la predicció per rang i un 76,67% en la predicció binària. Si comparem amb l'apartat 4.4,

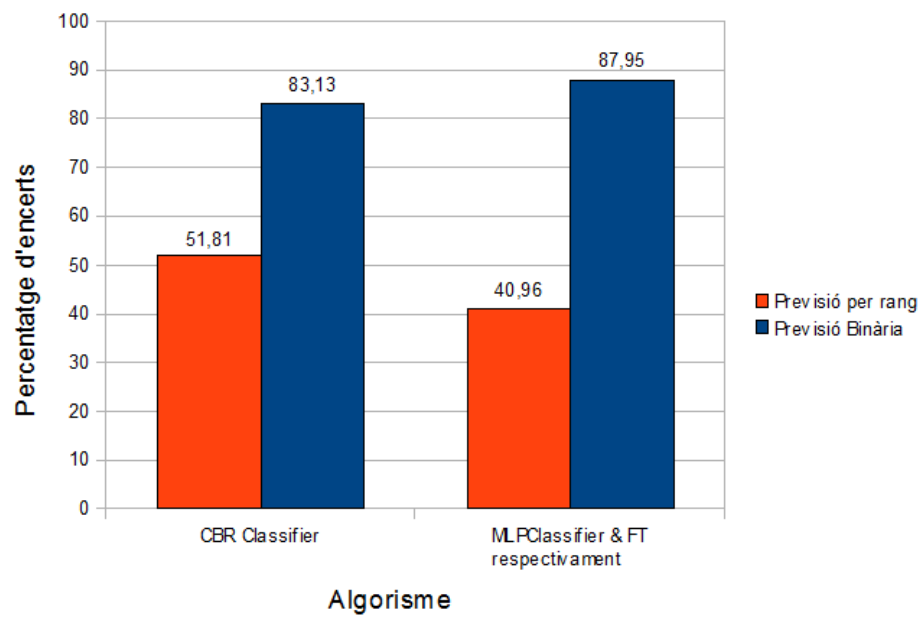


Figura 4.18: Percentatge d'encerts intentant predir el resultat de la base de dades de DSW any 2013-2014 amb menys dades visibles utilitzant la base de dades de DSW any 2012-2013 com a entrenament

podem veure que únicament la predicció per rang del CBR ha donat resultats millors, ha diferència dels casos que hem trobat amb les bases de dades de les demás assignatures.

Observant totes les gràfiques podem extreure la hipòtesis que, per a la majoria d'assignatures les últimes notes que s'obtenen tenen menys valor que les notes inicials del curs. Aquest cas no sembla donar-se únicament en l'assignatura de P1, al menys en els casos limitats de què disposem.

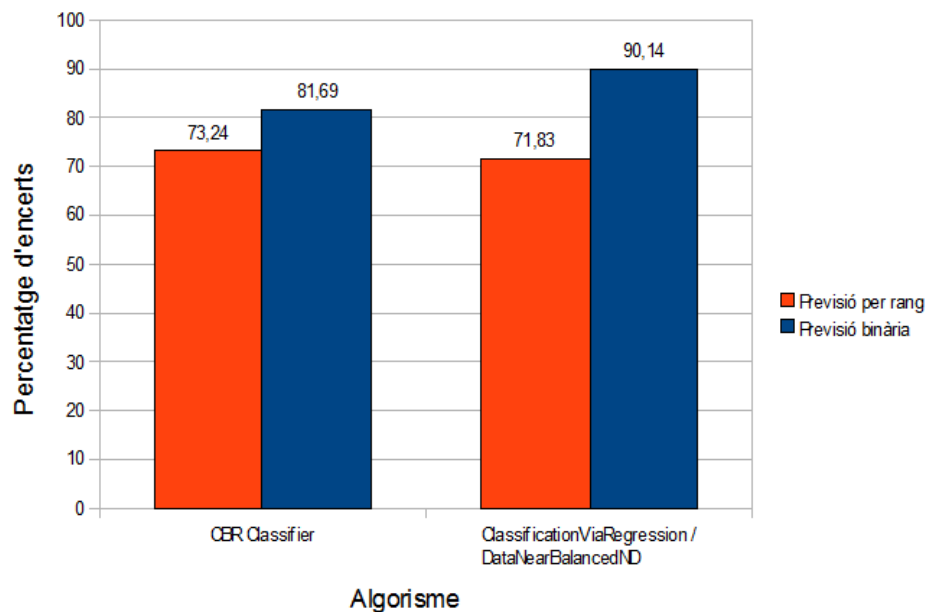


Figura 4.19: Percentatge d'errors intentant predir el resultat de la base de dades de FDT any 2012-2013 amb menys dades visibles utilitzant la base de dades de FDT any 2011-2012 com a entrenament

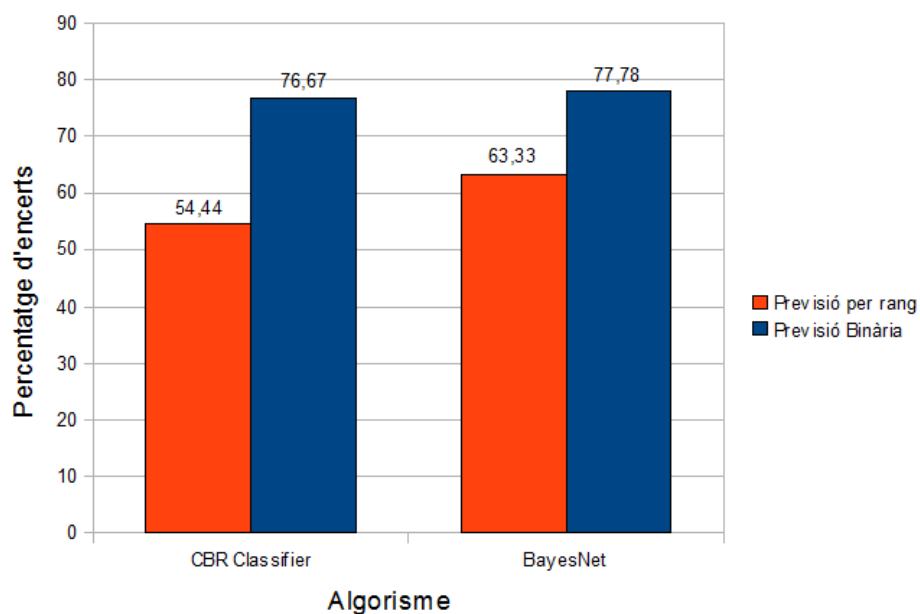


Figura 4.20: Percentatge d'errors intentant predir el resultat de la base de dades de P1 any 2013-2014 amb menys dades visibles utilitzant la base de dades de P1 any 2012-2013 com a entrenament

Capítol 5

Estudi econòmic

5.1 Diagrama de Gantt

En aquesta secció detallem el diagrama de Gantt. Dintre del transcurs d'aquest projecte, bona part del primer mes ha estat dedicada a l'aprenentatge de les eines a utilitzar, en concret de les tècniques de la llibreria Weka, en que s'han buscat com funcionen i s'ha provat la seva interfície d'usuari per habitar-me al format de treball que té. Seguidament s'ha passat al desenvolupament, dividit en implementacions, realitzades en períodes normalment d'entre 2 setmanes i 1 mes, detallades en l'apartat 5.2.

Enmig d'aquest desenvolupament a més se'ns ha proveït la segona llibreria de classificadors d'aquest projecte: la llibreria CBR-KB. Així doncs, he hagut d'aprendre també com funciona, a més d'adaptar el codi a les necessitats de la llibreria a implementar.

A més, se'ns ha demanat aplicar les funcions de la llibreria a la interfície pel professorat. S'havia fet una primera versió bàsica d'una sola setmana per a la interfície per tal de rebre *feedback*, i a partir d'aquest s'havia passat a desenvolupar la interfície final del projecte.

L'últim mes del projecte ha estat dedicat al redactat de la memòria, amb algunes petites modificacions de desenvolupament realitzades entremig que no han diferenciat prou per suposar una versió més de l'aplicació.

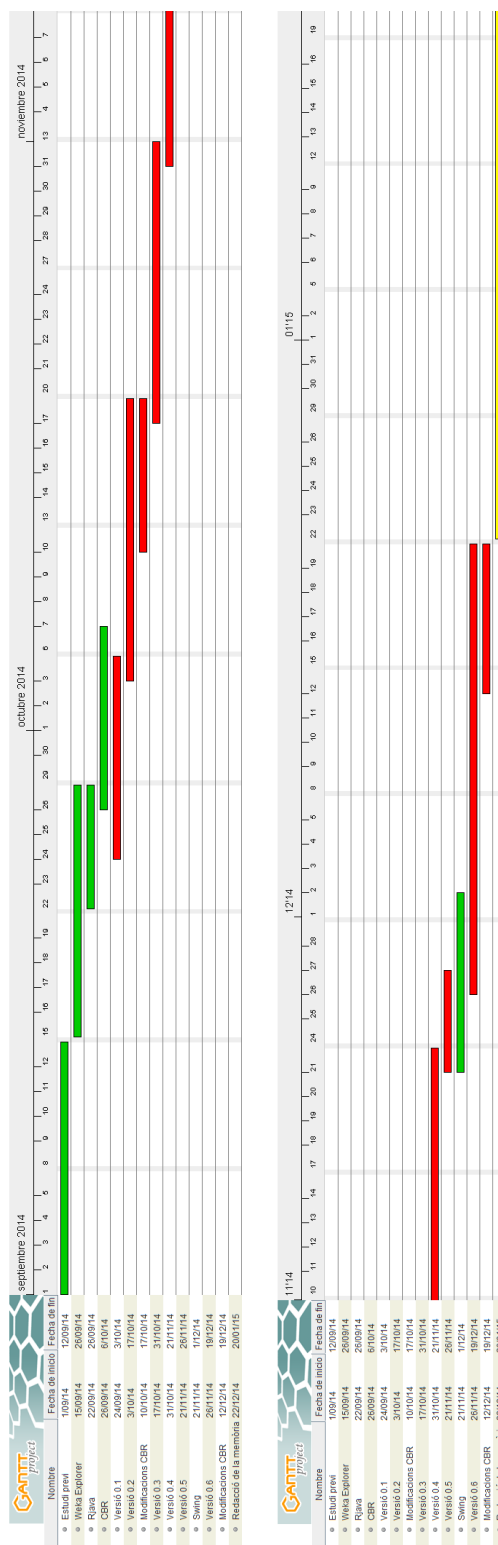


Figura 5.1: Diagrama de Gantt

5.2 Roadmap

Aquest seria el camí que s'ha fet al llarg del desenvolupament de la llibreria i de la interfície per al professorat, dividit en subobjectius (tal i com s'especifica al diagrama de Gantt de l'apartat 5.1) per tal de mantenir un control del progrés del projecte.

- *Versió 0.1*
Data de finalització: 3 d'octubre de 2014
Condicions de finalització:
 - Es carreguen classificadors de la llibreria Weka.
 - Es pot fer test de la base de dades Iris del repositori UCI.
 - Les interaccions de moment estan *hard-coded*.

- *Versió 0.2*
Data de finalització: 17 d'octubre de 2014
Condicions de finalització:
 - Es guarden tots els classificadors a utilitzar amb els seus paràmetres.
 - Es pot llençar una bateria de classificadors per realitzar proves.
 - Es guarden el percentatge d'encerts per a cada classificador.

- *Versió 0.3*
Data de finalització: 31 d'octubre de 2014
Condicions de finalització:
 - La llibreria del CBR està integrada
 - Es pot interactuar amb el fluxe del programa mitjançant consola.
 - La generació de *folds* va controlada pel codi propi en comptes de fer ús de les proves creuades de Weka.
 - Tota la bateria de classificadors de Weka i el CBR utilitzant els mateixos folds per evitar que l'aleatorietat afecti els resultats
 - Es guarda la matriu de confusió de cada classificació.

- *Versió 0.4*
Data de finalització: 21 de novembre de 2014
Condicions de finalització:
 - Es carreguen els arxius amb els que actua mitjançant un *FileChooser*.
 - Es poden llegir dades de les bases de dades d'assignatures.
 - Arreglat un bug amb la generació de *folds*.

- *Versió 0.5*
Data de finalització: 26 de novembre de 2014
Condicions de finalització:
 - Implementada una primera interfície pel professorat.
 - Es pot realitzar una classificació de dades del curs acadèmic actual.

- *Versió 0.6*
Data de finalització: 19 de desembre de 2014
Condicions de finalització:
 - Es normalitzen els atributs entre les dades del curs actual i les dades de cursos anteriors eliminant aquelles que no tenen en comú.
 - Es guarda la classificació en una còpia del fitxer d'origen de dades a classificar amb una columna extra pels resultats.

- Reimplementada la interfície per complir amb les condicions especificades amb la reunió.

5.3 Cost del projecte

Tasca	Temps dedicat (hores)	Preu (€/ h)	Inversió (€)
Estudi i anàlisi del projecte	20	22	440
Aprenentatge CBR	15	22	330
Integració CBR	40	24	960
Implementació de la llibreria	240	26	6240
Implementació de la interfície	100	26	2600
Memòria i estudi dels resultats	80	24	1920
TOTAL	495	25,23	12490

Taula 5.1: Cost de les diferents parts del projecte

La principal càrrega ha suposat la implementació de la llibreria, ja que ha suposat diferents tasques pesades, com són:

- L'adaptació dels convertidors disponibles per a les dades que utilitzem.
- La cerca i implementació dels classificadors de Weka que serveixen per al nostre tipus de dades.
- El funcionament de les proves, que inclou una generació de folds personalitzada.
- El funcionament de les prediccions, adaptat als nostres tipus de dades, i amb la possibilitat de realitzar una bateria de proves per decidir quin classificador ens convé.
- L'extracció dels resultats en un format amigable per l'usuari.
- La integració de la llibreria del projecte aïllat del CBR-KB.

Aquesta integració del CBR ha suposat una sèrie de problemes, ja que estava implementat amb finalitats estadístiques. Moltes característiques han hagut de ser polides (direccions *hard-coded*, paràmetres no utilitzats) i s'han hagut d'afegir noves funcions per a la predicció, ja que estava pensant únicament per a funcionar amb *ten-fold cross-validation*.

La implementació de la interfície ha estat una tasca que, tot i no ser complicada, ha suposat una càrrega debut a la seva naturalesa iterativa.

Tot això ha suposat un cost força elevat degut a la mescla de coneixements necessaris per a la seva implementació, a més de la complexitat de treballar en els entorns proporcionats.

Capítol 6

Conclusions i línies futures

Aquest projecte buscava implementar les bases del Mòdul Pedagògic i fer-lo usable pel professorat i s'ha aconseguit. La llibreria ha complert amb unes especificacions marcades inicialment, excepte alguns detalls que s'afegiran en la continuació d'aquest projecte. La interfície pel professorat, en canvi, s'ha definit sobre la marxa i per tant, s'ha tingut més flexibilitat per provar metodologies per complir amb els requisits demanats.

A partir d'aquest projecte, es passarà a acabar de polir tant la llibreria com la interfície per millorar el seu funcionament. En concret, dintre de la llibreria es buscarà afegir:

- La variant ACBR (*Adaptative Case-Based Reasoning*) de la llibreria CBR.
- La possibilitat de realitzar *feature selection*, mètodes que decideixen quines proves, exercicis o parcials són els més importants per tal de predeir l'evolució de l'alumnat en cadascuna de les assignatures. El seu ús ens permetrà controlar els casos més complicats que hem trobat amb els experiments i millorar els resultats globals.
- Provar paràmetres més concrets per a cada classificador de la llibreria utilitzada

A més de poder incorporar possiblement més llibreries integrables en codi Java, com podria ser R (utilitzable amb Rjava). També seria convenient realitzar més proves per continuar estudiant com funciona el sistema de predicció de notes.

En quant a la interfície, es buscaran afegir una sèrie de millores:

- Afegir missatges d'avís a l'usuari quan s'utilitzen conjunts de dades amb diferent nombre de columnes.
- Incorporar més formats de fitxers de dades.
- Podria lligar-se l'aplicació a la Web per un ús més còmode pel professorat.
- Afegir una opció d'ús avançat pels usuaris més experts.

La importància que ha tingut aquest projecte és que, a més de ser útil per sí mateix, serveix com a nucli per a les iteracions posteriors del Mòdul Pedagògic i de base o punt de partida per implementar les demés parts del Sistema de Tutor Intel·ligent.

Realitzar aquest projecte, sent conscient que serà utilitzat pel professorat i que servirà per a un projecte d'innovació molt més gran, ha acabat sent una feina molt gratificant. En algun moment he sentit la càrrega que suposava haver-lo de completar i la por a no voler traicionar les expectatives, però he de dir que, després de tot, me n'alegro molt d'haver tingut l'oportunitat de participar en aquest projecte.

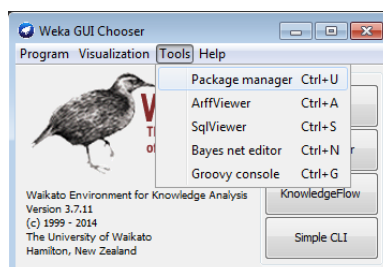
Bibliografia

- [1] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); *The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.*
- [2] Irene Martí Diéguez *Plataforma de Raonament basat en casos per a la gestió del coneixement.* Projecte de Fi de Grau, 2014
- [3] Leon Bottou*, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, L. D. Jackel *Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition*
- [4] Robles, *Asesores inteligentes para apoyar el proceso de enseñanza de lenguajes de programación.* Tesis de grado. ITSM. 1993.
- [5] Java. <https://www.java.com/es/download/>
- [6] Java Wiki. <http://java.wikia.com/wiki/>
- [7] Netbeans. <https://netbeans.org/>
- [8] Web oficial de Weka. <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>
- [9] Repositori UCI. <http://archive.ics.uci.edu/ml/>
- [10] Viquipèdia, Internet. <http://www.wikipedia.org/>

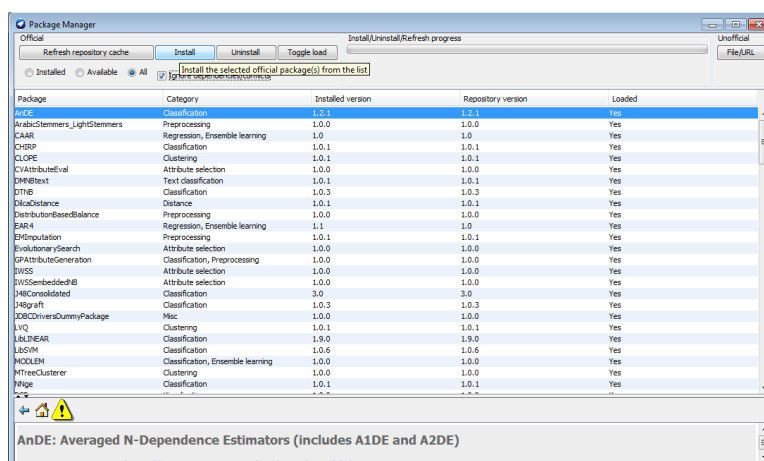
Annex: Manual d'usuari de la interfície

Com a requisit inicial, per utilitzar la llibreria de classificadors de Weka s'ha de descarregar l'aplicació de Weka <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>, segons la versió del vostre sistema operatiu.

Un cop tinguem la llibreria instal·lada, hem d'accedir al package manager.



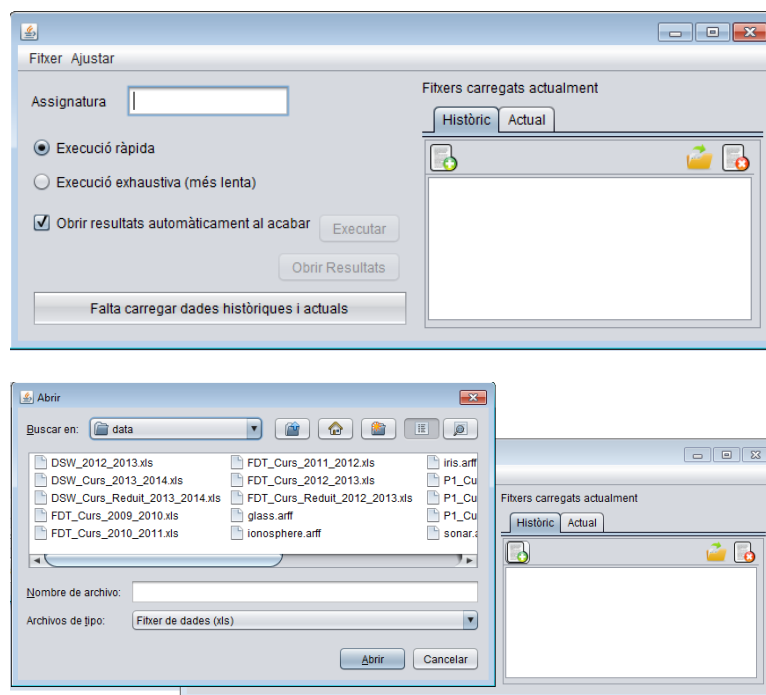
En aquest, instal·lem tots els paquets oficials disponibles, si volem estalviar-nos complicacions. D'altra manera, podem anar un a un escollint els paquets utilitzats en el programa. La llista de classificadors es troba a la direcció **config/Parameterized_classifiers.txt** del projecte.



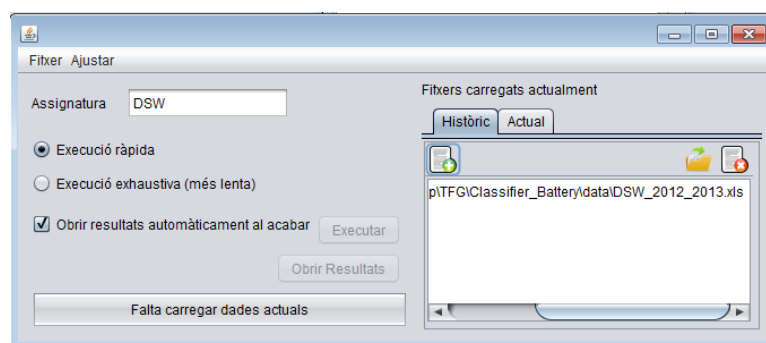
Un cop instal·lada la llibreria de Weka, podem passar a fer ús de l'aplicació.

Primerament podem carregar els fitxers de l'històric de notes de l'alumnat o bé les dades actuals a predir. En aquesta execució carreguem primer l'històric.

L'històric es pot carregar pel menú o pel primer botó començant per l'esquerra que trobem a la pestanya de l'històric. Ambdós obren un selector d'arxius, amb el qual seleccionarem els fitxers que afegir al conjunt d'entrenament.



Podrem observar que tal i com passa a la següent imatge, el fitxer s'afegeix a la llista de l'històric.



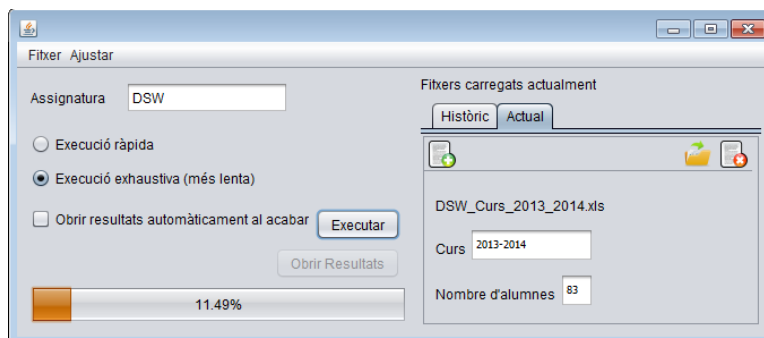
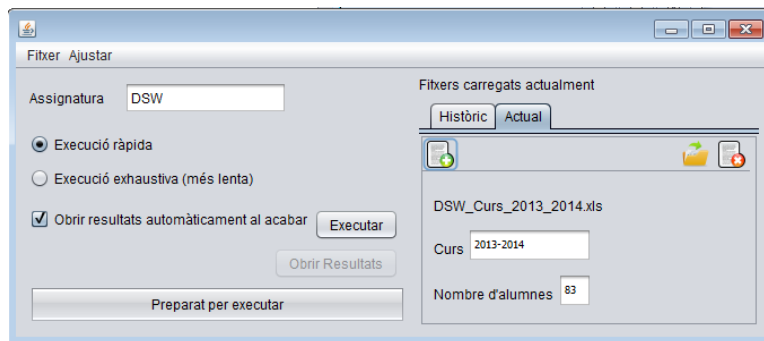
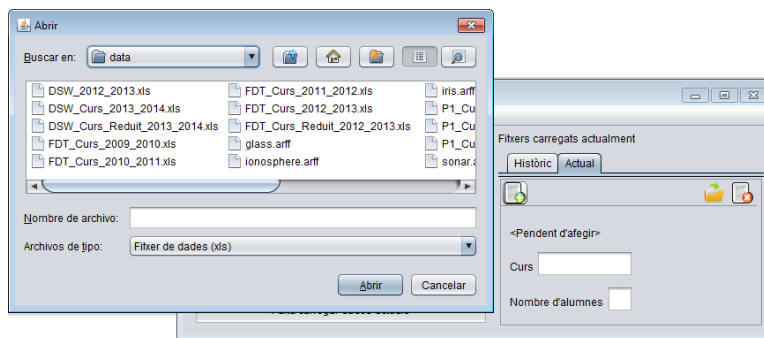
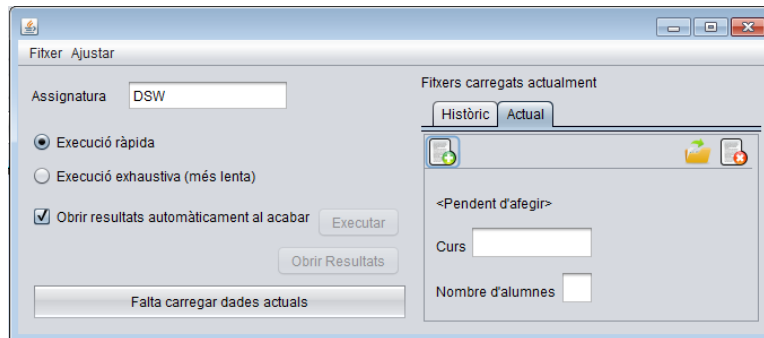
Si es vol comprovar les dades carregades es pot obrir amb el botó corresponent (segon començant per la dreta), i si es volen eliminar les dades s'utilitza el primer botó començant per la dreta.

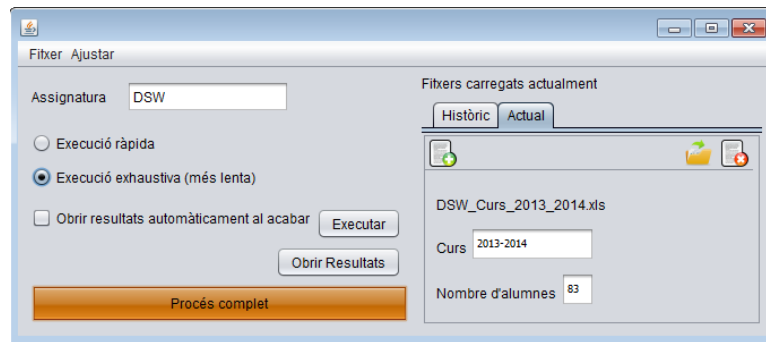
Seguidament repetirem els passos per a carregar les dades actuals. Per a això canviem a la pestanya "Actual".

En aquest cas es carrega un únic fitxer. D'aquest fitxer es mostrarà el nom, el curs (si té el format del nom adequat, "Assignatura.Curs") i el nom de l'assignatura.

De la mateixa manera que amb l'històric, es pot obrir aquest fitxer o esborrar les dades carregades.

Un cop carregades totes les dades podem passar a realitzar l'execució. Podem escollir una execució simple amb el CBR una execució de tota la bateria de proves abans de fer la predicció. Es pot escollir que el programa obri automàticament el fitxer de sortida o bé simplement calcular-lo i realitzar els passos posteriors manualment.





Finalment, s'activa el botó d'obrir resultats per visualitzar-los en qualsevol moment, o es selecciona al menú l'opció de guardar resultats.

