



25-5-2015

Web Interface to Learn Machine Learning

Facultad d'Informàtica de Barcelona
Universitat Politècnica de Catalunya - BarcelonaTech

DEGREE IN COMPUTER SCIENCE



Author: [Sònia Sandalinas Pérez](#)
DIRECTOR: LUÍS ANTONIO BELANCHE

Table of contents

INDEX OF IMAGES:	4
INDEX OF TABLES	6
INDEX OF CODE	7
ABSTRACTS	8
CATALÀ	8
ENGLISH	8
CASTELLANO	9
FIRST PART: INTRODUCTION AND PLANNING	10
1 - INTRODUCTION	11
1.1- <i>State of art</i>	12
1.1.1 - Machine Learning.....	12
1.1.2- R	13
1.1.3 – Rstudio.....	13
1.1.4 - RShiny.....	13
1.1.5 - Current Applications.....	14
2 - SCOPE	16
2.1 - <i>Objectives and requirements</i>	16
2.1.1 - General objectives.....	16
2.1.2 - Requirements	16
2.2 – <i>Scope of TFG</i>	18
3 - METHODOLOGIES	19
3.1 - <i>Working methods</i>	19
3.2 - <i>Tracking tools</i>	19
3.3 - <i>Validation method</i>	20
3.4 - <i>Algorithms and software</i>	20
3.4.1 - Programming languages:.....	20
3.4.2 - <i>Web application</i> :.....	21
4 - TEMPORARY PLANNING	22
4.1 - <i>Project phases</i>	22
4.1.1 - Planning and feasibility of the project.....	22
4.1.2 - Review and study of the concepts of Machine Learning (10/10/2014 - 19/12/2014)	22
4.1.3 – Information Collection and study of new technologies(19/12/2014 - 2/01/2015)	23
4.1.4 - Design of the web application (02/01/2015 - 22/01/2015)	23
4.1.5 – Programming and testing application (22/01/2015 - 15/04/2015)	23
4.1.6 - Final part of the project(16/04/2015 - 25/05/2015)	24
4.2 - SUMMARY PROJECT TIME	24

Web Interface to learn Machine Learning

<i>Review and study the concepts of Machine Learning</i>	<i>24</i>
<i>Information Collection and study of new technologies.....</i>	<i>24</i>
<i>Design of the web application.....</i>	<i>24</i>
<i>Programming and testing application</i>	<i>24</i>
<i>Final part of the project</i>	<i>24</i>
<i>4.3 – Modified Gantt chart.....</i>	<i>24</i>
5 - ECONOMIC MANAGEMENT AND BUDGET.	24
<i>5.1 - Costs detection.....</i>	<i>24</i>
<i>5.2 - Human resources costs</i>	<i>25</i>
<i>5.3 – Hardware Costs</i>	<i>26</i>
<i>5.4 – Software costs</i>	<i>26</i>
<i>5.5 – Generals costs</i>	<i>27</i>
<i>5.6 - Variations of the initial budget</i>	<i>28</i>
<i>5.7 – Total budget.....</i>	<i>28</i>
6 - SUSTAINABILITY AND SOCIAL COMMITMENT	29
<i>6.1 Assessment of the sustainability of the project.....</i>	<i>29</i>
<i>6.2 - Sustainability matrix</i>	<i>30</i>
SECOND PART: DESCRIPTION AND IMPLEMENTATION.....	31
1 - DEFINITION OF THE MAIN CONCEPTS.....	32
<i>1.1 - Description of the problem</i>	<i>32</i>
2 - DEFINITION OF THE ASSOCIATE TASKS.....	34
<i>2.1 - Classification</i>	<i>35</i>
<i>2.1.1 - Concepts.....</i>	<i>35</i>
<i>2.1.2 - General approach to solving a classification problem.....</i>	<i>37</i>
<i>2.2 - Regression</i>	<i>39</i>
<i>2.2.1 - Concepts.....</i>	<i>39</i>
<i>2.2.2 - General approach to solving a regression problem.....</i>	<i>41</i>
<i>2.2.3 - Effectiveness of a regression model.....</i>	<i>42</i>
<i>2.3 - Clustering</i>	<i>42</i>
<i>2.3.1 - Concepts.....</i>	<i>42</i>
<i>2.3.3 - Effectiveness:</i>	<i>48</i>
<i>2.4 - Feature Extraction.....</i>	<i>48</i>
<i>2.4.1 - Concepts:.....</i>	<i>48</i>
3 - DESIGN, IMPLEMENTATION AND SATISFACTION REQUIREMENTS	50
<i>3.1 – Design.....</i>	<i>50</i>
<i>3.2 – Implementation and Satisfaction of Requirements.....</i>	<i>54</i>
<i>3.2.1 - Administrator / Professor:.....</i>	<i>54</i>
<i>3.2.2 - User:</i>	<i>58</i>
4 - TECHNICAL SKILLS ACHIEVEMENT	63
5 - EXAMPLE APPLICATION EXECUTION.....	65
<i>5.1 – Execution example part of Login.....</i>	<i>65</i>
<i>5.2 - Execution example part of learn</i>	<i>66</i>
<i>5.3 - Execution example part of test</i>	<i>69</i>

Web Interface to learn Machine Learning

THIRD PART: IMPROVEMENTS AND CONCLUSIONS	74
1 - NEXT IMPROVEMENTS	75
2 - CONCLUSIONS.....	76
REFERENCES.....	77
GLOSSARY.....	78
ANNEXES	79
USER MANUAL	79
ADMINISTRATOR MANUAL.....	79
GANTT DIAGRAM CHANGED.....	79

Index of Images:

<i>Image 1 - K-means learn application.</i>	17
<i>Image 2: Web application for learning statistics</i>	17
<i>Image 3: Definició gràfica de la tasca de Classificació.</i>	36
<i>Image 4: Learn part of application</i>	37
<i>Image 5: Test part of application</i>	37
<i>Image 6: Graphic definition of classification task</i>	38
<i>Image 7: Example of iris dataset.</i>	39
<i>Image 8: Record to classify.</i>	40
<i>Image 9: Resolution of classification problem.</i>	41
<i>Image 10: Result Matrix of Iris Dataset.</i>	42
<i>Image 11: Dataset where X_2 depends on X_1.</i>	43
<i>Image 12: Record where we want to find value for x_2 .</i>	44
<i>Image 13: Result of apply a regression model to dependent variable X_2</i>	45
<i>Image 14: Example of clusters.</i>	46
<i>Image 15: Kmeans example of Iris dataset (with $K = 3$)</i>	47
<i>Image 16: Hierarchical Clustering example</i>	48
<i>Image 17: Example of Center Based Clustering</i>	49
<i>Image 18: Example of density based clustering</i>	50

Web Interface to learn Machine Learning

<i>Image 19: PCA example on Iris Dataset</i>	52
<i>Image 20: Application design</i>	54
<i>Image 21: Database implementation</i>	55
<i>Image 22: Table “testusuario”</i>	55
<i>Image 23: Table “Test”</i>	56
<i>Image 24: Table “grupopreguntas”</i>	56
<i>Image 25: Table “Preguntes”</i>	56
<i>Image 26: Table “Respostes”</i>	57
<i>Image 27: Login Interface</i>	69
<i>Image 28: Main Screen without selecting anything.</i>	69
<i>Image 29: Main Strcture of Learn Task</i>	70
<i>Image 30: Summary tab of an algorithm</i>	71
<i>Image 31: DataTable of Iris Dataset</i>	72
<i>Image 32: Main Interface of test part.</i>	73
<i>Image 33: Sidebar of test panel</i>	73
<i>Image 34: Main questions panel.</i>	74
<i>Image 35: Main Panel of results</i>	75
<i>Image 36: Main Results with BarPlot</i>	75

Index Of Tables

<i>Table 1: Summary of Project Phases</i>	<i>27</i>
<i>Table 2: Human Resources Costs</i>	<i>28</i>
<i>Table 3: Net Costs of the Worker</i>	<i>29</i>
<i>Table 4: Hardware Costs for developing the project</i>	<i>29</i>
<i>Table 5: Software Costs for developing the project</i>	<i>30</i>
<i>Table 6: Other expenses related with the project.</i>	<i>31</i>
<i>Table 7: Variations of the initial budget</i>	<i>31</i>
<i>Table 8: Summary of the project budget.</i>	<i>32</i>

Index Of Code

<i>Code 1 : Implementation of the test part</i>	<i>57</i>
<i>Code 2: Implementations to change between learn and test part</i>	<i>57</i>
<i>Code 3: Reactive functions getTest and getLearn</i>	<i>58</i>
<i>Code 4: Different models included</i>	<i>58</i>
<i>Code 5 : Implementation of the LogIn System.</i>	<i>59</i>
<i>Code 6: Credentials Checking on database</i>	<i>59</i>
<i>Code 7: Implementation of a selectable attribute in a field.</i>	<i>60</i>
<i>Code 8: Dropdown implementation with task selection</i>	<i>61</i>
<i>Code 9: Conditional Panel implementation.</i>	<i>61</i>
<i>Code 10: TabPanel Implementation.</i>	<i>62</i>
<i>Code 11: RenderPlot Implementation</i>	<i>62</i>
<i>Code 12: Interface implementation of test part</i>	<i>63</i>
<i>Code 13: Reactive function implementation that retrieve a random test from DB</i>	<i>63</i>
<i>Code 14: Query implementation to database to retrieve the results of some user.</i>	<i>64</i>
<i>Code 15: RenderPlot task implementation</i>	<i>64</i>

Abstracts

Català

Començarem per explicar el concepte de machine learning, aquest és una disciplina a mig camí entre la informàtica i l'estadística, que té com a objectiu desenvolupar algoritmes i tècniques que permetin a les diferents màquines aprendre, és a dir, es tracta de crear un programa que generalitzi comportaments a partir de dades empíriques d'un procés.

Avui en dia en molts camps de la societat està present el machine learning com per exemple la medicina, el big data, el reconeixement de la veu...

Es per aquest motiu que molts alumnes tenen curiositat i volen saber més sobre els conceptes de machine learning, en algunes de les assignatures de la FIB.

Aquest TFG és una interfície web amigable que està dirigit precisament aquests alumnes que volen fer alguna d'aquestes assignatures, per ajudar-los en el seu aprenentatge d'aquesta matèria i que els serveixi com a guia.

L'objectiu principal és doncs, que els alumnes tinguin una eina per el suport del aprenentatge de l'assignatura, amb exemples de cadascú dels algoritmes i un apartat de test per poder evaluar els seus coneixements.

English

First of all, machine learning is a science between informatics and statistics which main objective is that a machine learns with some algorithms

Today, In the actual society, machine learning is present in different places, for example: medicine, Big Data and Voice recognition

Web Interface to learn Machine Learning

For this reason, there are a lot of students curious about it, who want to learn more about the concepts of machine learning in some subject of FIB.

This Final degree project is a friendly web interface that is created just for this kind of students, to help them in the learning of this subject and to have a guide.

The most important objective is, that the above mentioned students have a tool to support their learning on this subject, with examples of each algorithm and one section for testing their knowledge about machine learning

Castellano

En primer lugar explicaré el concepto de machine learning que es una disciplina entre la informática y la estadística, su principal objetivo es, desarrollar algoritmos y técnicas que permitan a las maquinas a aprender.

Hoy en día es una una ciencia que está presente en muchos lugares, como por ejemplo, en la medicina, el big data y el reconocimiento de voz entre otros.

Es por eso que algunos de los alumnos de la FIB, tienen curiosidad por aprender estas técnicas en algunas de las asignaturas que se ofrecen en la facultad.

Este TFG es una interfaz web amigable que está pensada precisamente para cubrir esa necesidad, para poder ayudarlos en su aprendizaje y que les sirva como guía.

Dicho esto, el objetivo principal es que los alumnos tengan una herramienta de soporte para el aprendizaje de la materia, con ejemplos de varios algoritmos y con un apartado de autoevaluación para demostrar sus conocimientos.

FIRST PART: INTRODUCTION AND PLANNING

Web Interface to learn Machine Learning

1 - Introduction

This project responds to the offer proposed by Luís Antonio Belanche, Professor of the department of computer science, exposed in “racó”

The objective of this project is to increase the number of students who want to learn about machine learning. Therefore, it is necessary to have a support tool to develop this subject.

On one hand the development of this TFG could help other universities and/or faculties offering subjects of the same kind, on the other hand, the project could help different internships in their jobs because it is a web page.

The main objective of this project is to get a web application - interactive, usable and user friendly - so everybody can have a support for learning machine learning.

This web application has two main sections:

1. Learning section: In this part the user has different algorithms that may be executed. Through different graphics, the user can see the results, as well as, read an explicative text and a table containing the used data for the execution.

These data will have been selected in forehand for the users.

The user can choose different tasks to do before selecting the algorithm:

- a. Classification
- b. Regression
- c. Clustering
- d. Feature Extraction

Web Interface to learn Machine Learning

2. Test Section: In this section, the user can choose different levels of difficulty (easy, medium and difficult) and a specific number of questions to be evaluated with a machine learning test. Finally, users can check their results in form of text or graph, whenever they want.

1.1- State of art

We will explain the required concepts and tools we will use for our project.

1.1.1 - Machine Learning

Machine learning is a discipline between computer science and statistics, which aims to develop algorithms and techniques that allow different machines to learn. It consists in creating a program that generalizes behaviour from empirical data of a process [\[1\]](#).

Machine Learning applications:

- Search engine
- Medical diagnostic
- Detection of credit card fraud
- Analysis of the stock market
- Classification of DNA sequences
- Speech recognition
- Robotics
- Data Mining
- Big Data

In the field of machine learning there are different types of algorithms:

Supervised, unsupervised, basic knowledge, structured knowledge, inductive, deductive, symbolic and connexionist [\[2\]](#).

Web Interface to learn Machine Learning

1.1.2- R

R is an object-oriented programming language, interpreted and not compiled.

The syntax of R [3] is very simple and intuitive.

It is a language used for statistical and graphical analysis.

R is based on the S language and is open-source. [4]

Being a language used for statistical analysis, it is used in many fields: biomedical research, bioinformatics and the financial mathematics.

As Ross Ihaka, one of the creators of R, said:

R: A language for data analysis and graphics

1.1.3 - Rstudio

RStudio is an integrated development environment(IDE) for R. RStudio is formed by a console, a syntax-highlighting editor that supports direct code execution and tools for managing the workspace [5].

1.1.4 - RShiny

Shiny is a framework for interactive web applications in R. The main feature is reactive programming. [6]

Some of the advantages of shiny are:

- Creation of usable web applications with few lines of code without using “JavaScript”

Web Interface to learn Machine Learning

- Shiny applications have a similar concept to “Google Drive” because they are “live”, this means it is an application that modifies the output for any change the user makes in the inputs, without having to refresh the browser.
- They operate in any environment with “R”.
- It contains “Bootstrap” framework [\[7\]](#)
- It contains predesigned “widgets” to display the output of the “Plots (renderPlot)”, “Tables (renderTable)” or text (“renderPrint”).

1.1.5 - Current Applications

Nowadays, if you start searching you can find different methods for learning machine learning.

These methods, doesn't have online support, for example:

- Academic Articles. ([Genetic Algorithms and Machine Learning](#))
- Books ([Bishop, C.M , Pattern recognition and machine learning](#))
- Publications and Presentations ([Machine Learning and Data Mining](#))
- On-site lessons in the faculty. ([FIB - APA](#))

We also know there is online media, for learning some of the techniques of machine learning.

Web Interface to learn Machine Learning

For example, in the following image we find an application to check the operation of “K-MEANS”[\[8\]](#)

Iris k-means clustering

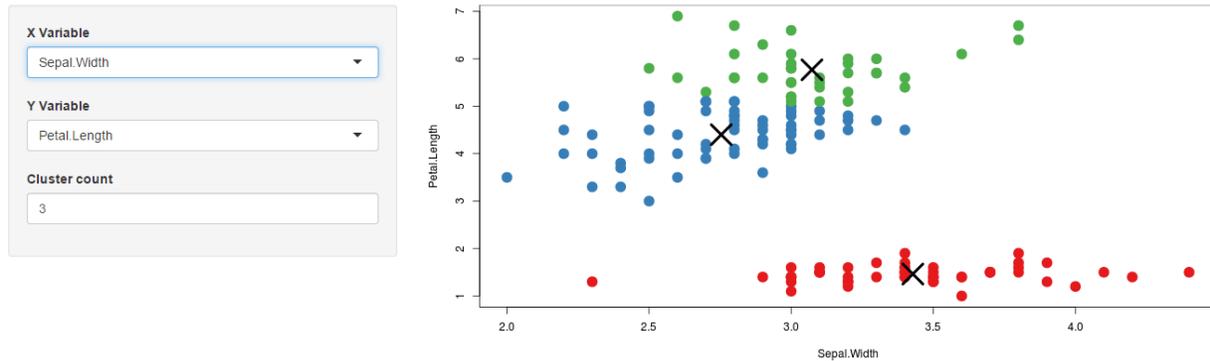


Image 1 - K-means learn application.

Also there are some webs, where you can try to learn statistical. The following image shows one of this webs. [IHateStatistics](#) .



avans
hogeschool

ERASMUS
UNIVERSITY

Universiteit Utrecht

NHTV
University of Applied Sciences

Image 2: Web application for learning statistics

Web Interface to learn Machine Learning

But if you want a tool that has an overview of all kinds of algorithms we have not found any. That's why we believe there is a need to develop this tool, to meet the need for all students who want to learn this subject.

2 - SCOPE

2.1 - Objectives and requirements

The main objective of the project is to create a web application, which enables the learning of machine learning.

2.1.1 - General objectives

Moreover, we have the following objectives:

- Provide an application that is useful and usable.
- Facilitate monitoring of students in the subject.
- Users can evaluate their knowledge.

2.1.2 - Requirements

As for the development project we have followed the SCRUM methodology.

The project requirements are defined by separating all stakeholders and through user stories.

In different UH (user history), if not indicated, we assume that the requirements are functional

Web Interface to learn Machine Learning

A user history follows the following structure:

As a ____ (1) ____, want that the application ____ (2) ____, for ____ (3) ____ [9]

1. User, person who uses the application or administrator
2. Functionality, non-functional requirements...
3. Objective to be achieved.

2.1.2.1 - Administrator / Professor:

- As an administrator, I want the application to have two different sections “learn” and “test”, so we can offer a good service to users.
- As an administrator, I want the application to have different machine learning algorithms for the users, so they can do different tests.
- As an administrator, I want the application to allow having different users and a control of them, so it can be usable for everybody.
- As an administrator, I want the application to be scalable and easy to change, so everybody can maintain it in production environment (Not functional).

2.1.2.2 - User:

- As a user, I want that the application allows me to enter and access the whole features by introducing username and password.
- As a user, I want that the application allows me choose between different dataset, for testing different dataset for the same algorithm.
- As a user, I want that the application allows me to choose one option between different tasks (classification, regression, clustering or feature extraction) for testing different algorithms
- As a user, I want that the application allows me to execute different algorithms to see its performance.

Web Interface to learn Machine Learning

- As a user, I want that the application allows me to swap between graphics, text and table in the execution of each algorithm for checking each of the values.
- As a user, I want that the application allows me to swap between the section of “learn” and the section of “test”, in order to access both of them.
- As a user, I want that the application allows me to evaluate myself in the different test levels (“easy”, “medium” and “difficult”) for letting me know my level in the subject.
- As a user, I want that the application allows me to check all my results in the test part any time.
- As a user, I want that the application allows me to see the graphics of the results, in order to check them in more detail.
- As a user, I want that the application responds with an acceptable time (less than 5 seconds) in the execution of the algorithm, for not having to wait time in front of the screen (Not functional)
- As a user, I want that the application responds efficiently with the changes of the parameters for improving interactivity with the application. (Not functional)

2.2 – Scope of TFG

The application is interactive with the user, have all the necessary sections in order to follow the subject by user, serves as a support method for learning different methods of the subject, and finally to learn machine learning.

This application is separated into two parts, the first one is the learning part and the second one is an evaluation method in order to check the knowledge of the user.

The first part has different fields of machine learning (Regression, Classification, Feature Extraction and Clustering) and the second part is the general purpose.

3 - Methodologies

3.1 - Working methods

The methodology we chose for developing this project, SCRUM [10], was successful because by performing small tasks and short development cycles, we obtained a more realistic view of the project and we could detect and solve problems in a shorter period of time. We had an important feedback from the tutor of the project, with whom we met every three weeks.

The roles within the SCRUM methodology were the following:

1. “SCRUM MASTER”: In our case, the director of the project, Luís Antonio Belanche.
2. “PRODUCT OWNER”: the product owner is the director of the project as well.
3. “DEVELOPMENT TEAM”: In our case, this is integrated by me, Sònia Sandalinas Pérez.

As can be seen, the director of the project shares two positions. According to the methodology, this isn't correct, but in this case we have this situation because there are just two members on this project.

3.2 -Tracking tools

We have ruled out the option of a GitHub repository, because we believe that having only one developer it's not necessary. This tool would involve too many expenses.

Web Interface to learn Machine Learning

To control the different versions of the project we used a tree structure in DropBox, each version with different name.

3.3 -Validation method.

Given all the tests and controls that have been made in each version, thanks to the control system DropBox and the meetings with the tutor, we have accomplished the project with all its objectives.

To check the correct usability of the application, there have been different application tests as we have been developing. These tests have been done throughout the implementation phase of the project.

3.4 - Algorithms and software

For this project we had to choose alternatives to cover two distinct elements: On one hand, a programming language with the necessary power to run machine learning algorithms, on the other hand, showing the results in a user-friendly web interface.

3.4.1 - Programming languages:

To run the machine learning algorithms, at a low cost in time and complexity, we did an analysis of alternatives for different programming languages.

In this analysis, we chose four different programming languages: R, Java, Matlab and C++.

Web Interface to learn Machine Learning

Of these four, the first to be dismissed were Java and C++ because of their complexity to develop machine learning algorithm with them. This would also increase execution time.

Once we had on the table R and Matlab [11], we discarded Matlab because it wasn't an open source solution.

Finally, we chose R because it integrated libraries for machine learning and it was more optimized than the others, as well as being an open source solution.

3.4.2 - Web application:

For the realization of the web application, we needed to create a friendly interface for the end user. We had four options: R (Shiny), Python, PHP and Node.js.

First, we ruled out PHP and node.js because the integration with R was quite complex and didn't offer us a great power to the graphics interface, as they were "back-office" languages.

We were between Python and Shiny framework for web applications in R.

After some research for each of the alternatives, we found that shiny offered us an easier integration, especially by the fact of having reactive functions (1) and was organized in modules (2), more than having integrated the display CSS3 framework "bootstrap".

4 - Temporary planning.

Regarding the initial planning, the calendar had an important delay respect what was intended in the first instance. The accorded times in the different phases of the project were correct. But I have not been able to invest all the time that we had originally planned to carry out the project throughout the course.

For this reason, planning was delayed a bit (this is why we presented on the 25th of May and not in April as we had expected).

4.1 - Project phases

4.1.1 - Planning and feasibility of the project.

This phase was done at the beginning of the academic year, i.e., the GEP. This phase was divided into the following aspects:

1. Scope
2. Temporary planning.
3. Economic management and sustainability
4. Preview
5. Contextualization and bibliography
6. Specifications
7. Final documentation

4.1.2 - Review and study of the concepts of Machine Learning (10/10/2014 - 19/12/2014)

Web Interface to learn Machine Learning

At this stage, machine learning concepts were reviewed. They were studied in the subject APA (Aprentatge Automatic Machine Learning) and essential for the realization of the project. The new concepts of machine learning necessary for the application were studied, but not explained in the subject.

With these revised and studied concepts, we had everything necessary to develop the technical part of the project.

4.1.3 - Information Collection and study of new technologies(19/12/2014 - 2/01/2015)

At this stage we studied the tools to develop the project (RShiny, RStudio) and we collected information, examples and manuals, to have the necessary knowledge of the tools we needed.

4.1.4 - Design of the web application (02/01/2015 - 22/01/2015)

In this phase our application was designed. It included the graphic part and the software architecture part.

4.1.5 - Programming and testing application (22/01/2015 - 15/04/2015)

At this stage we developed the application in the two parts: the test and learn ones. We did it with the knowledge reviewed in the previous phases.

Web Interface to learn Machine Learning

We considered these two phases together, as it was a project with small goals. We made programming and testing in parallel, as well as finding the errors in a fast way, to fix them in an easier way.

4.1.6 - Final part of the project(16/04/2015 - 25/05/2015)

In this last stage the software product was already done. It was only necessary to create a user manual, the project report and the corresponding presentation that will be held before court.

4.2 - Summary project time

Stage	Hours
Planning and feasibility of the project	75
Review and study the concepts of Machine Learning	150
Information Collection and study of new technologies	30
Design of the web application	30
Programming and testing application	150
Final part of the project	60
TOTAL	495

Table 1: Summary of Project Phases

4.3 – Modified Gantt chart

The Gantt Chart is in the annex.

5 - Economic management and budget.

5.1 - Costs detection

We can summarize the costs involved in this project in four: human resources, hardware, software and general resources.

Web Interface to learn Machine Learning

The specified costs below are the forecasts made for the project, which have suffered minor variations which is explained at the end of this chapter.

5.2 - Human resources costs

This project has been developed by a single person, therefore has been commissioned to make “project manager”, developer, developer tester and software engineer. So we differentiate the 495 hours they did:

Person	Estimated hours	Price per hour (gross)	Estimated Total
Project manager	75	50 €	3750 €
Developer	315	35 €	11025 €
Developer Tester	75	30 €	2250 €
Software engineer	30	40 €	1200 €
Total	495	155 €	18225 €

Table 2 : Human Resources Costs

Although costs are thought by working hours rather than months, we will assume that there is a single payment of € 18225 raw and therefore it is necessary to all applicable withholdings.

Resulting salary of the worker	Price
Total gross salary	18.225
-4,70% Contributions to social security	856,57
-1,55% Quote unemployment	282.48

Web Interface to learn Machine Learning

-18% IRPF	3280.5
TOTAL	13805.45

Table 3: Net Costs of the Worker

5.3 – Hardware Costs

To develop this project, we needed a notebook. Following is the cost of hardware:

Product	Price	Units	Useful Life	Estimated Amortization
Asus A550LD-XX375-H	699 €	1	4	87.37 €
Total	699 €			87.37 €

Table 4: Hardware Costs for developing the project

5.4 – Software costs

In order to make the project we needed a series of software. Here we find the costs of it:

Product	Price	Units	Useful Life	Estimated Amortization
Adobe Reader X	0€	1	-	0€

Web Interface to learn Machine Learning

Windows 8.1	0€ (included in notebook)	1	-	0€
Microsoft Office 2013	139€	1	2	34.75€
Dropbox	0€	1	-	0€
Drive (Google)	0€	1	-	0€
RStudio	0€	1	-	0€
R	0€	1	-	0€
Shiny (R)	0€	1	-	0€
Total	139€	-	-	34.75€

Table 5: Software Costs for developing the project

5.5 – Generals costs

Apart from the project's expenses, salaries, hardware and software, we have other general expenses, as detailed below.

Concept	Estimated monthly cost	Estimated totally cost
Premises	450 €	2700 €
Water	30 €	180 €
Electricity	50 €	300 €
Internet	40 €	240 €
Total	570 €	3420 €

Web Interface to learn Machine Learning

Table 6: Other expenses related with the project.

5.6 - Variations of the initial budget

The project costs have not changed compared to the initial planning because the hours have been the same. The general costs: electricity, water, local ... have increased by one month extension due to project delivery. The budget is the following:

Concept	Initial Estimated Cost	Final Estimated Cost
Local	2700 €	3150 €
Water	180 €	210 €
Electricity	300 €	350 €
Internet	240 €	280 €
Total	3420 €	3990 €

Table 7: Variations of the initial budget

5.7 – Total budget

The final budget of the project, after changes explained in the previous section, is as follows:

Concept	Cost
Human Resources	13805.45€
Hardware	87.37€
Software	34.75€
General costs	3990€
Total	17917.57€

Web Interface to learn Machine Learning

Table 8: Summary of the project budget.

6 - Sustainability and social commitment

6.1 Assessment of the sustainability of the project.

The economic viability of this project is relatively high, as the costs invested in the realization of it have been minimal and the profitability that can be obtained is very high, both in an academic environment and also in a possible working environment, as has discussed above.

The material costs are minimal, because it is a programming project, the only cost has been the purchase of the laptop to develop the application. Therefore, the environmental effect is not high as this investment in the laptop will continue to be profitable within four years.

When the laptop life ends, it can be donated to different organizations to take advantage of it and this will have a lower environmental cost.

Finally, the social viability of this project is also good, as its implementation could improve the life of the teacher, providing classes, as well as helping the student to learn in a more dynamic and fast way

At having a section of "test" the application could serve as an alternative method of assessing an impact on a percentage of the note.

Regarding the economic aspects, in this project we have not had any collaboration with any entity or company or academic.

In the event that this project is located in a business and not in academic content, I think it could be developed following the budget.

The time of completion of each task is consistent according to importance and planning is sufficiently tight to carry out the project in less time.

Web Interface to learn Machine Learning

As for the social aspects, as we have said this project is aimed at students of any subject of Machine Learning and is not a very large group of society. But it could be applied in many other fields using Machine Learning. The direct impact of the project is low but considering all areas in which Machine Learning is used, the impact can be considerable.

6.2 - Sustainability matrix

¿sustainable?	Economic	Social	Environmental
Planning	Economic viability	Improvement in quality of life	Resource Analysis
Assessment	8	10	8
Result	Final Cost vs Prevision	Impact in a social environment	Resource consumption
Assessment	8	10	9
Risks	Adaptation to changes of scenery.	Socials Damage	Enviromment Damage
Assessment	-2	0	-1
Total Valuation	50		

SECOND PART: Description and Implementation

1 - Definition of the main concepts.

1.1 - Description of the problem

Our problem is to develop a web application for learning machine learning.

This application has different parts that are defined briefly in this section.

1. Login system

In our application there is a login system that consists of a database created with "MySQL" and controlled by "MySQL Workbench" software, the user will have a user name consisting of an email and a password. These data will be generated by the administrator and users will be supplied with them.

Once the user enters the name and password, the system validates if the user exists in the database and if the given data is correct, then gives the user access to the application.

Web Interface to learn Machine Learning

Login

Account

Password

Login

Image 3 : Login System

2. Learning part :

Once the user has entered with their credentials correctly, the learn part will appear.

In this part the user will have all the functionality to have the support for learning machine learning, in the first instance, the user will only see a drop-down list, but as you progress in steps the user will arrive to a screen as shown in the figure below.

Web Interface to learn Machine Learning

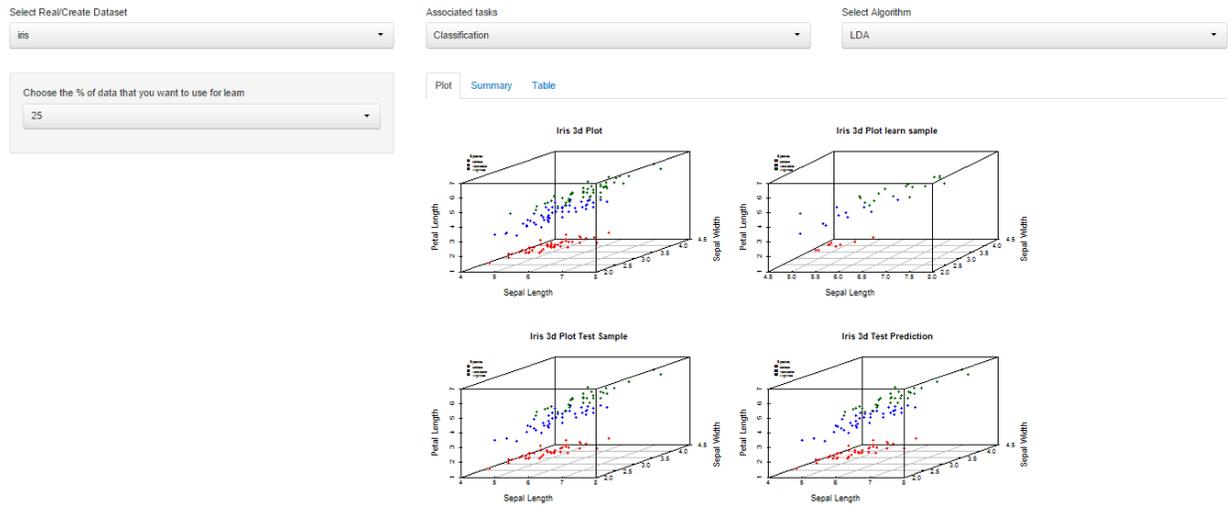


Image 4 : Learn part of application

3. Testing part :

In this part, the user can test the concepts received in the subject lessons as well as the support given by the application.

The user will observe the results in graphical and text format.

The screenshot shows the 'Test' part of the application. It has two dropdown menus: 'Choose the number of questions' (set to '5') and 'Choose the complexity of test' (set to 'medium'). Below these are three buttons: 'START TEST' (green), 'SHOW MY RESULTS' (blue), and 'PLOT MY RESULTS' (blue). The main area is titled 'Test of Machine Learning' and contains a section for 'Pregunta 1' with the instruction 'Please Select:'. There are three radio button options: 'Respuesta 1', 'Respuesta 2', and 'Respuesta 3'. Below the options is a text input field and a 'Next' button.

Image 5 : Test part of application

2 - Definition of the Associate Tasks

Web Interface to learn Machine Learning

We have classified the machine learning algorithms on different tasks. [\[12..21\]](#) :

- Classification
- Regression
- Clustering
- Feature Extraction

Below we will explain each of them.

2.1 - Classification

2.1.1 - Concepts

In machine learning classification call, the task we use for each element of a collection is assigned it to a category or class. This is a widespread problem that includes various applications. Some examples include the detection of spam emails, categorizing benign or malignant cells based on MRI results, as well as the classification of different galaxies, among others.

This is a supervised learning task.(3).

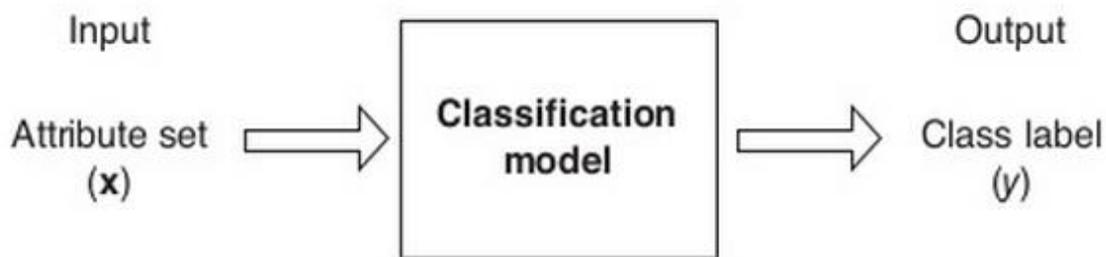


Image 6: Graphic Definition of Classification task

The objective of the classification task is to accurately predict the class or category of each of the items in the collection.

Web Interface to learn Machine Learning

The input data for a classification task is a collection of records, each record is characterized by a tuple (x, y) where x is the set of attributes and y is the class attribute or attribute category

	sepal.Length	sepal.width	Petal.Length	Petal.width	Species
56	5.7	2.8	4.5	1.3	versicolor
22	5.1	3.7	1.5	0.4	setosa
136	7.7	3.0	6.1	2.3	virginica
102	5.8	2.7	5.1	1.9	virginica
63	6.0	2.2	4.0	1.0	versicolor
21	5.4	3.4	1.7	0.2	setosa
133	6.4	2.8	5.6	2.2	virginica
121	6.9	3.2	5.7	2.3	virginica
20	5.1	3.8	1.5	0.3	setosa
57	6.3	3.3	4.7	1.6	versicolor

Image 7: Example of Iris Dataset.

In this example the set of attributes includes: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width and the attribute of the class in this case would be Species.

The attributes shown in the table are numbers except the class attribute that is always a discrete attribute. This is a key feature that distinguishes the classification of the regression (a task which will be explained in the next section).

In our case we have used a classification model to predict the class label of unknown records. Classification models also have another utility, serving as an explanatory tool to differentiate between objects of different classes.

We can say that a classification model is like a black box that automatically assigns a class when it has a set of attributes with an attribute of unknown class.

We assume that we have the following characteristics of a flower(iris) to classify.

	sepal.Length	sepal.width	Petal.Length	Petal.width	species
1	5.1	3.5	1.4	0.2	?

Web Interface to learn Machine Learning

Image 8: Record to classify

To determine the class of the record, we used a classification model obtained from the data of Image 10.

2.1.2 - General approach to solving a classification problem.

A classification technique is a systematic approach for building models of classification using a set of input data. Examples: LDA, QDA, Naive Bayes, Neural Networks, Support Vector Machines, Nearest Neighbour and Decision trees among others.

Each technique uses a learning algorithm to identify a model that best fits in the relation between the sets of attributes and the attribute of the input class.

A good classification model must be adjusted to the input data, but also have to correctly predict the class attribute records that may not have considered.

It is necessary to find the middle ground to not overfitting the data.

Therefore, the most important thing in a classification model is to generate models with good generalization ability, i.e. models that correctly predict the class attribute for unknown samples.

Phases of a classification algorithm:

1. Learn/Training Set: The input set, where all samples have a defined attribute class and therefore know. This set is the one that is used to create user's own learn model (Classification model).
2. Test Set: The data set the user wants to get the class attribute, as this is unknown.
3. Application of the classification model: A model obtained from the input set and the classification algorithm can predict the class attributes of all records in the test set.

Web Interface to learn Machine Learning

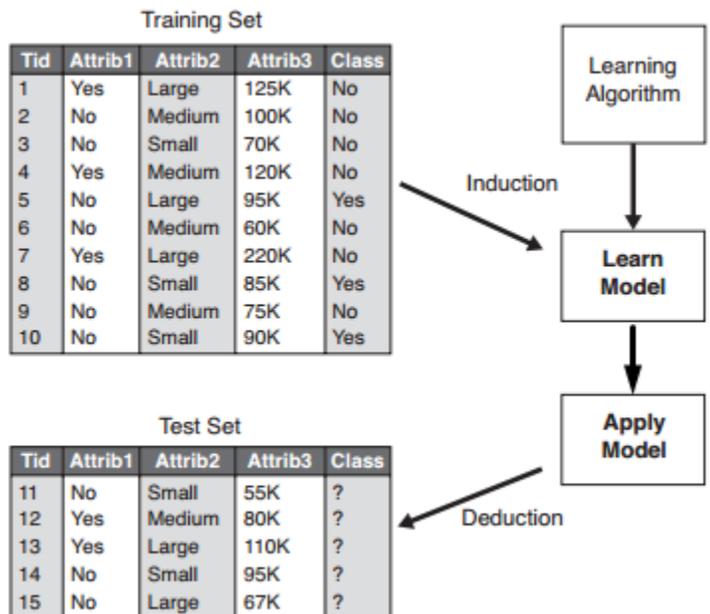


Image 9: Resolution of Classification problem.

Both data, the Learn set and the Test Set are extracted from the same dataset (4).

In the case of Test data, we have to remove the class attribute. This attribute is saved to check the effectiveness of the model later, as it is the attribute that is predicted.

2.1.3 - Effectiveness of a classification model

To evaluate the effectiveness of our model we have:

1. Results Matrix:

It consists of the matrix of Test we previously predicted, so we can see how the class attribute for each of the different model classes have evolved.

Web Interface to learn Machine Learning

```
mydata.predict_test setosa versicolor virginica
setosa              12          0          0
versicolor          0          14          0
virginica            0          0          12
```

Image 10: Results Matrix Of Iris Dataset.

In this observation can see that the classification model is very good, since all predictions have been correct, but in most cases, there are values outside the diagonals of the matrix (which is an error).

2. Effectiveness: This is calculated using the average number of correct, i.e., the number of class attributes that are predicted correctly, divided by the total number of items in the Test.

$$\underline{\underline{Effectiveness = \frac{hits}{Total}}}$$

2.2 - Regression

2.2.1 - Concepts

Given a set of data, we can define a set containing two types of variables for each record:

- Independent variables

Web Interface to learn Machine Learning

- Dependent variables

We can say that in Machine Learning, regression is the task for obtaining a dependent variable from the other variables in the record.

The objective of task regression is to predict the value of a dependent variable from the values of the other variables in the record.

The input data for a regression task is a collection of records. Each record is categorized by a tuple (x, y), where x is the set of attributes and y is the dependent variable.

This is a supervised learning task.

	x1	x2
44	0.88	0.726003817
7	0.14	0.004925153
74	0.98	1.066866777
103	0.56	0.166581594
138	1.26	1.386318207
43	0.86	0.437772675
21	0.42	0.380602954
126	1.02	0.976484887
69	0.88	1.076548379
29	0.58	0.158505952

Image 11: Dataset where X2 depends on X1.

In this example "X2" is a variable that depends on "X1", therefore is the variable that we want to predict the regression model.

A regression model can say that is like a black box that automatically assigns a value to an unknown variable.

Let's suppose we have the following characteristics:

	x1	x2
1	0.02	?

Web Interface to learn Machine Learning

Image 12: Record where we want to find value for X2

To determine the value of "X2" of this register we can use a regression model obtained from the data of Image 11.

2.2.2 - General approach to solving a regression problem

Examples of regression techniques are: linear regression, Neural Networks, Support Vector Machines and General Linear Model among others.

Each technique uses a learning algorithm to identify the model that best fits the relationship between the set of independent variables and the dependent variable.

Phases of a regression algorithm:

1. Learn/Training Set: The input set, where all samples are a set of the independent variables we know. This set is used to create the model learn (regression model).
2. Test Set: The data set containing the dependent variable the user wants to obtain, because it is unknown.
3. Application of regression model: From the obtained input set model and algorithm of regression, we can predict values of the dependent variable of all records in the test set.

Both data from Learn set and Test Set was taken from the same dataset.

In the case of Test data, there was an elimination of the variable to predict. This value was saved for later, to be able to check the effectiveness of the model as it was the predicted value.

Web Interface to learn Machine Learning

2.2.3 - Effectiveness of a regression model

To evaluate the effectiveness of our model we have:

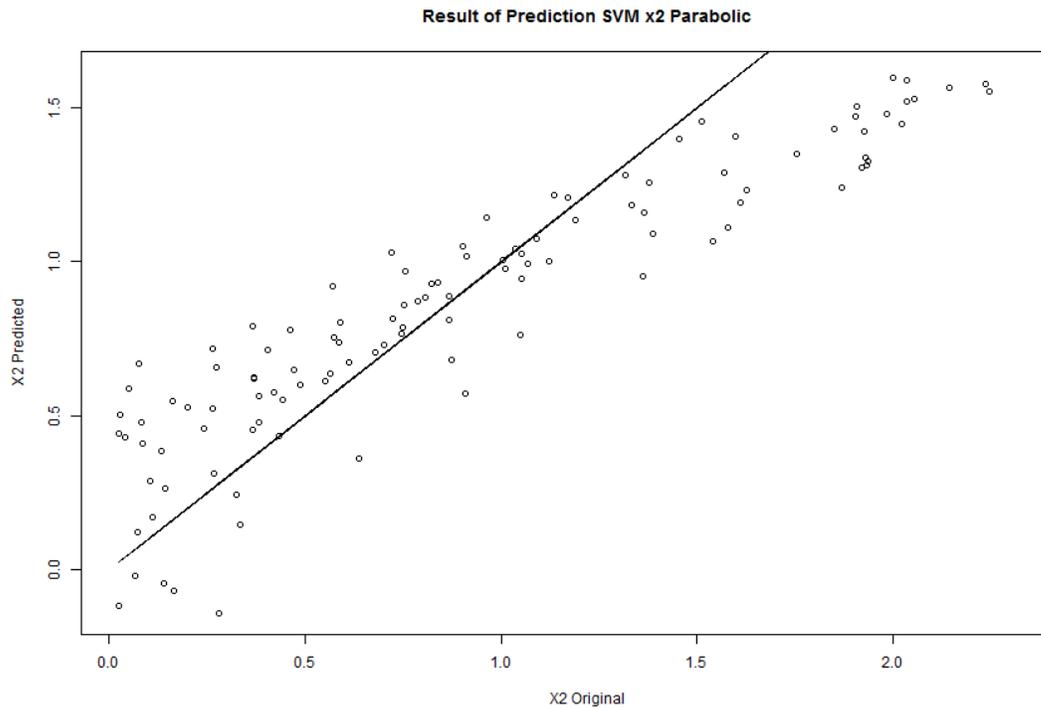


Image 13 : Result of apply a regression model to dependent variable X2

In this observation we can see that the regression model is an approximation, since the accumulation of points is around the straight line that would be a perfect model.

The points that are on the line, are predicted successfully, whereas the points that have major distance imply a worse model.

2.3 - Clustering

2.3.1 - Concepts

Web Interface to learn Machine Learning

We can define the clustering as the allocation of a set of observations into subsets (clusters) so observations in the same group are similar.

same group are

The clustering is an unsupervised learning task (6).

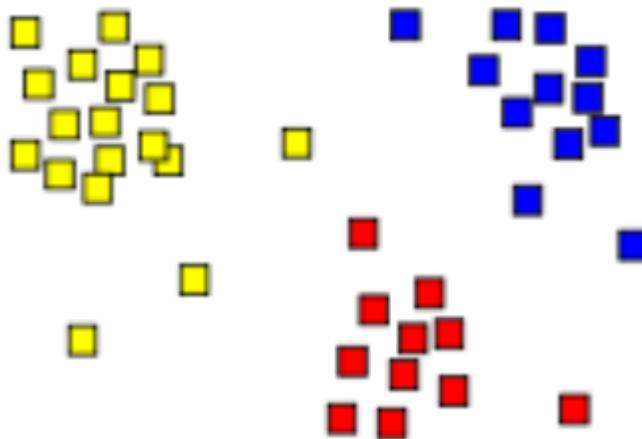
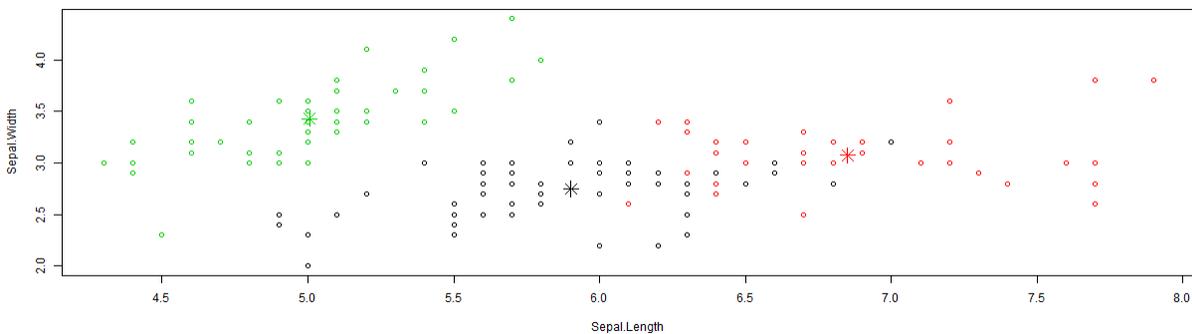


Image 14: Example of clusters.

Clustering algorithms is based on classifying or grouping the records based on their characteristics in subgroups, for example, Kmeans is grouped into a K groups or clusters where K is a positive integer



Web Interface to learn Machine Learning

Image 15: Kmeans example of Iris dataset (with $K = 3$)

The computational cost of finding the optimal partition of N records in K groups is NP-hard.

We can group the clustering in 4 different types:

- Hierarchical clustering

In this method, the data form a tree diagram shows the relationship between the objects according to a proximity matrix. The root node of the tree represents the entire set, and each leaf node is a single data.

Intermediate nodes describe how objects are close together.

Each cluster is obtained by cutting the tree at different levels. In this way we can distinguish two ways to generate clusters: root cutting the leaves (divisive) or conversely (agglomerative)

Web Interface to learn Machine Learning

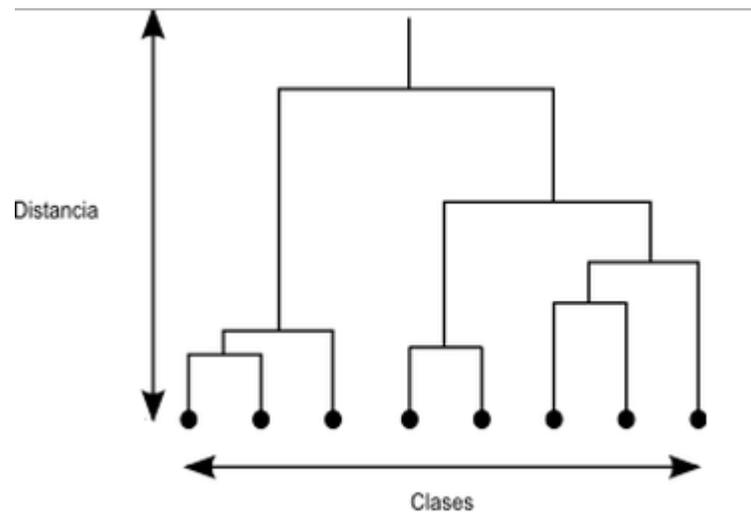


Image 16: Hierarchical Clustering example

- Based clustering centres

It is an iterative algorithm to divide a data set into a number of clusters specified by the user. It starts by assigning the object representing each cluster (centre) randomly taken from the dataset grouping.

In each repetition, data is classified by assigning it to the nearest cluster.

Once the whole data has been allocated, each centre is recalculated.

The process continues until no cluster varies.

Web Interface to learn Machine Learning

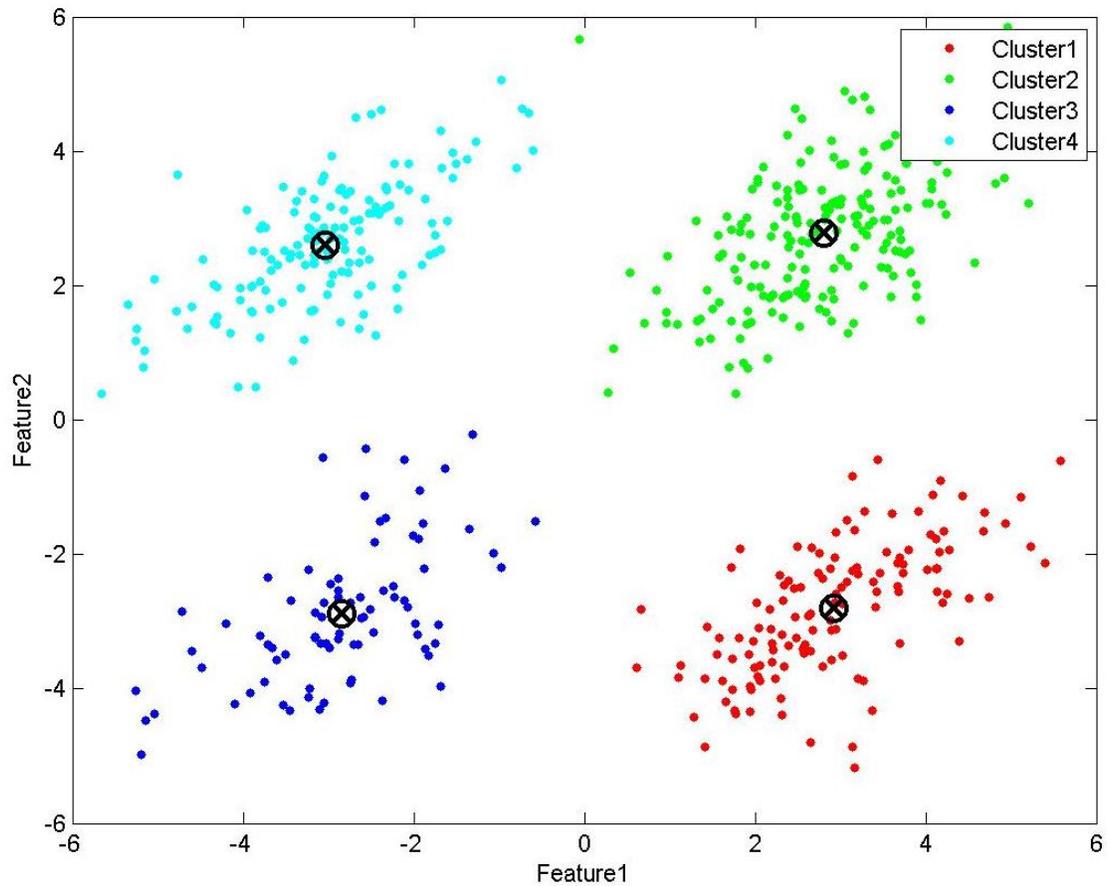


Image 17: Example of Centre Based Clustering.

- Distribution based Clustering

The distribution based clustering model is the most closely related to the statistical grouping.

Clusters can be defined as a group of objects that have a similar distribution.

Web Interface to learn Machine Learning

A desirable property of such clustering is similar to the way datasets generate artificial objects by sampling a random distribution.

- Density based Clustering

The clustering based on density can be defined as some areas where the density is higher in comparison to the rest of the dataset.

The objects of these scattered areas, which are required to separate the clusters, are usually regarded as noise and border points.

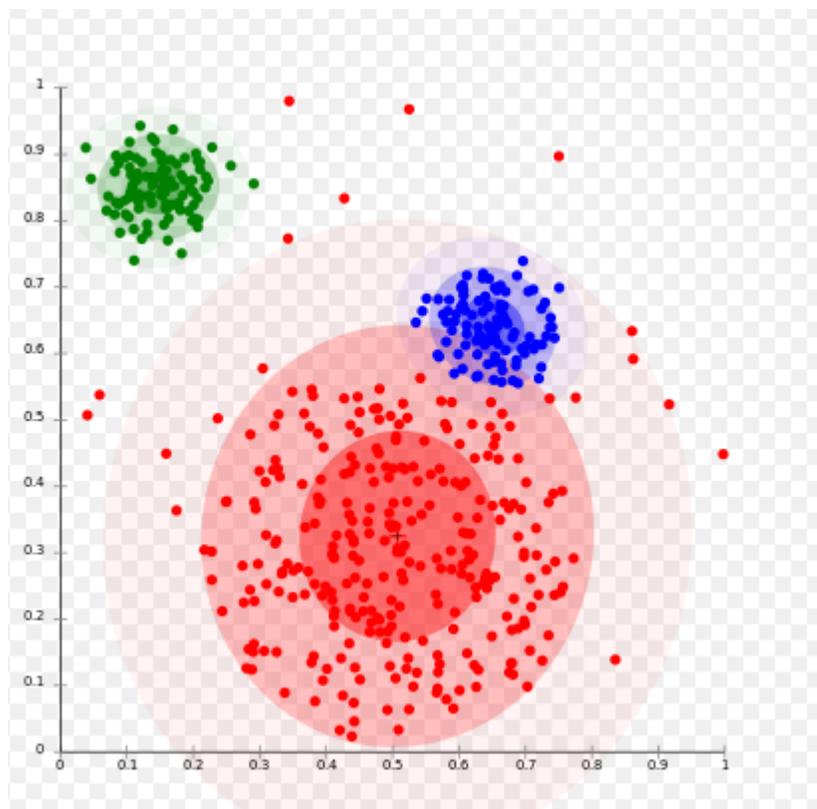


Image 18: Example of density based clustering

Web Interface to learn Machine Learning

2.3.3 - Effectiveness:

A model of clustering is better than another because it is an unsupervised algorithm, we don't know the solution.

To find an effective clustering algorithm, it usually depends on the type of data. It will be found through experimentation.

2.4 - Feature Extraction

2.4.1 - Concepts:

Feature extraction is a method that aims to extract useful information (features) and to eliminate redundant information. It can be performed after a process (such as classification).

Feature extraction allows to work on the issue of pattern recognition using a smaller amount of computer resources, by analysing the whole data. This function should not reduce the accuracy of our method. To make a good feature extraction, we must eliminate redundant or irrelevant information, reduce the dimensionality of the problem and maximize the most representative aspects of the data. It will allow us to make an effective selection of features and a subsequent classification.

Examples of feature extraction methods are: LDA, MCA and PCA among others.

Web Interface to learn Machine Learning

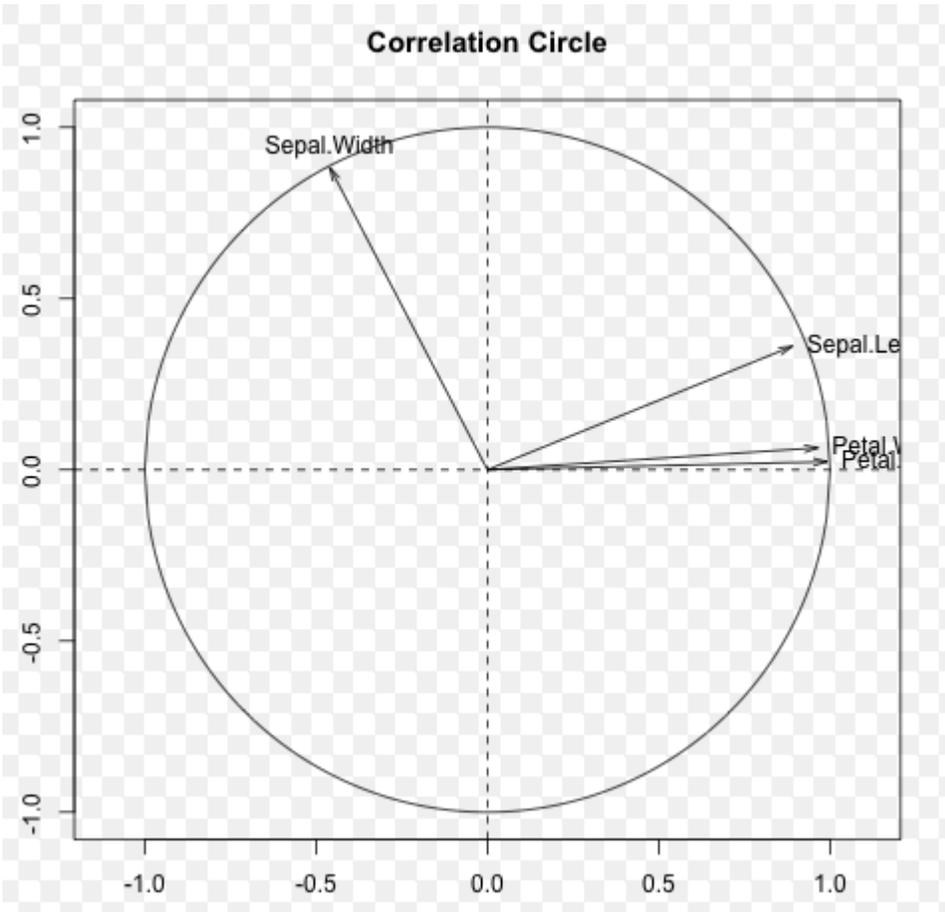


Image 19: PCA example on Iris Dataset

3 - Design, implementation and satisfaction requirements

3.1 – Design

The system we used for our application was a modification of the MVC system.

We divided the three layers architecture in two parts: the first one consists of the functionality of view and controller which we delegate to Shiny, as it has the required power to meet all of our requirements. The second one consists in generating a data model based on a RDBMS (Relational database management system), which in our case has been MySQL.

In the next picture you can see the structure of our application.

Web Interface to learn Machine Learning

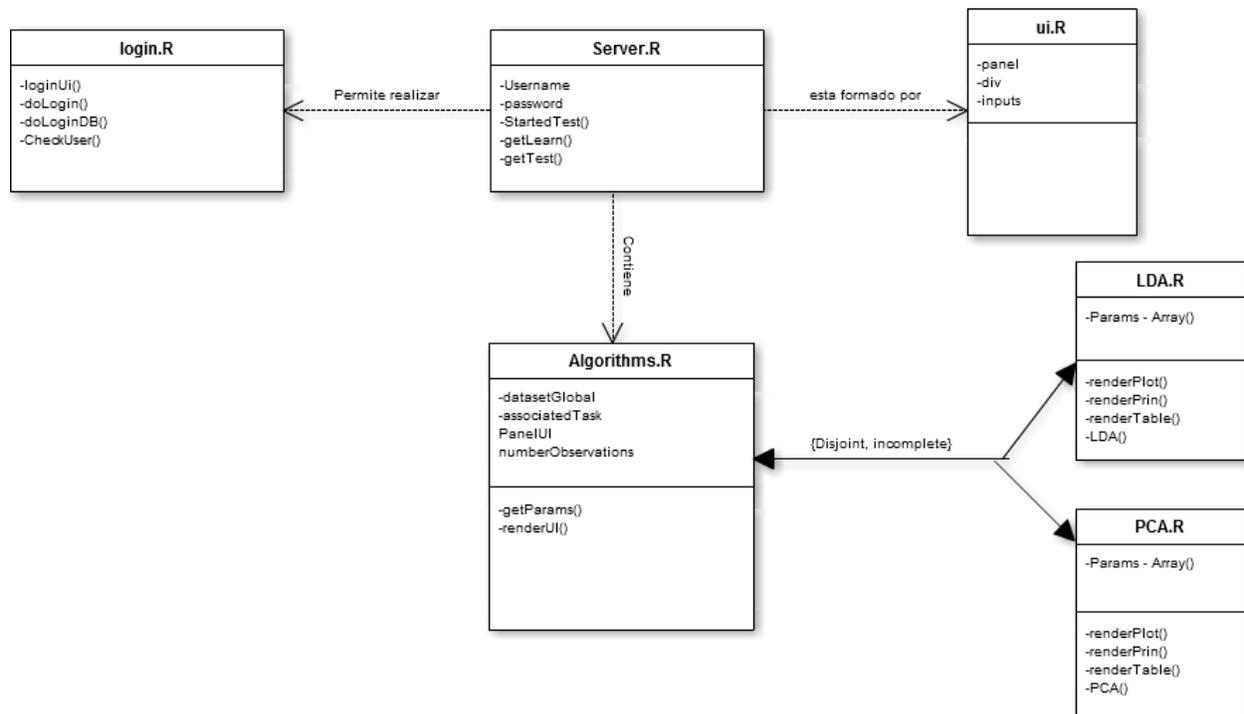


Image 20: Application design

As you can see, we delegate the view and the controller to Shiny. These parts are "Server.R" and "ui.R."

We have a module "Server.R" responsible for managing the entire application, since all interactions have to go through it.

Moreover, we have a main user interface that is generated through the module named "ui.R" as seen on the image. "Server.R" and "ui.R" are connected, then every user interaction causes the release of an event in our Server to control it.

Apart from the two main modules ("Server.R" and "ui.R"), we have a module that handles the system "logging". In this module, as you can see, we have different functions: some to create an interface and others formed by the logic of the system with the database.

Web Interface to learn Machine Learning

Finally, we have different modules that can be categorized into "Algorithms.R". In these modules, as in the case of "logging", we also have creations of all interfaces and logic algorithm. In this module we will generate all the other modules for each of the machine learning algorithms.

The second part is the structure of the database. This structure has been created and designed considering the needs of the project.

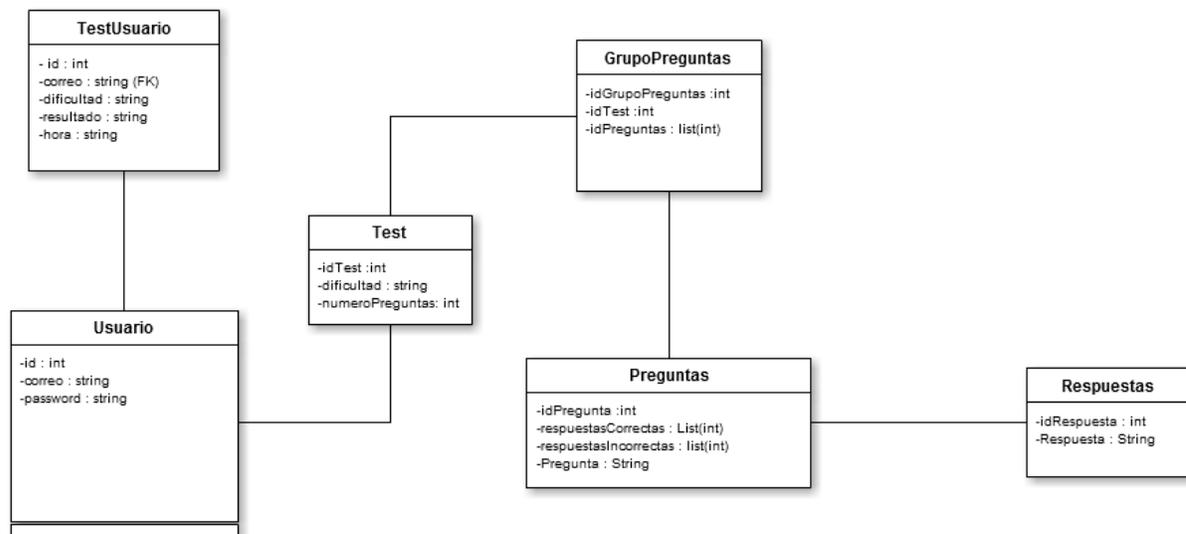


Image 21: Database implementation

Our database is made up of six tables, five of which are used for carrying out the test and the other to manage users.

Following is explanation of each table.

In the first table ("testusuario") we store all information related to the user test, that has been done during the execution of the application.

Web Interface to learn Machine Learning

Column	Type
◇ id	int(11)
◇ correo	varchar(250)
◇ dificultad	varchar(250)
◇ resultado	varchar(250)
◇ hora	varchar(200)

Image 22: Table “testusuario”

This table will only be accessed if a user makes a test.

In this table we'll save a score, that will be shown when the user presses the button “show my results” or “plot my results”

The remaining tables are necessary to do a dynamic test. The user can vary the number of questions and answers.

Firstly, the table "test" where we store all the tests that have been created with an ID

◇ idTest	int(11)
◇ dificultad	varchar(45)
◇ numeroPreguntas	int(11)

Image 23: Table “Test”

Second, once we have a definite test, for this "test" we have to define a group of questions.

◇ idgrupoPreguntas	int(11)
◇ idTest	varchar(45)
◇ idPreguntas	varchar(250)

Image 24: Table “grupopreguntas”

Web Interface to learn Machine Learning

As you can see in the picture above, a group of questions is formed by a string of questions.

◇ idPregunta	int(11)
◇ respuestasCorrectas	varchar(220)
◇ respuestasIncorrectas	varchar(220)
◇ pregunta	varchar(500)

Image 25: Table “Preguntas”

And every question consists of a string of correct answers, wrong answers and a question in text format.

And finally, the table of answers, which comprises the answer in text format.

◇ idRespuesta	int(11)
◇ respuesta	varchar(500)

Image 26: Table “Respuestas”

3.2 – Implementation and Satisfaction of Requirements

As explained in the first part, we have a number of requirements our project has to meet. Therefore, we will go through them one by one and tell how it fits within our system resolution.

3.2.1 - Administrator / Professor:

- 1.1 - As an administrator, I want that the application has two different sections “learn” and “test”, so we can offer a good service to the users.

Web Interface to learn Machine Learning

- This requirement has been met through two different implementations, learn and test parts. It has also been met because we are using reactive functions, so every user can change between learn and test part automatically.

```
testUI <- function () {  
  div(class = "",  
    selectInput("questions", "Choose the number of questions", c("5", "10", "15")),  
    selectInput("complexity", "Choose the complexity of test", c('easy', 'medium', 'hard')),  
    div(class = "",  
      shiny::actionButton("start_test", "START TEST", styleclass = 'success'),  
      shiny::actionButton("show_history", "SHOW MY RESULTS", styleclass = 'primary'),  
      shiny::actionButton("show_plot", "PLOT MY RESULTS", styleclass = 'primary')  
    )  
  )  
}
```

Code 1 : Implementation of the test part

As you can see, there is similar nomenclature of bootstrap and html5. We have used "action_buttons" for different functions.

```
output$mainPage <- renderUI({  
  doLoginBD()  
  doLogout()  
  getTest()  
  getLearn()  
})
```

Code 2: Implementations to change between learn and test part

As shown in the picture above, on our core module ("server") we have reactive calls to detect any click in the user buttons "learn" or "test".

Web Interface to learn Machine Learning

```
getTest <- reactive({
  if (!is.null(input$test) && input$test > 0) {
    isolate(
      loginData$TestMode <<- TRUE
    )
  }
})

getLearn <- reactive({
  if (!is.null(input$learn) && input$learn > 0){
    isolate(
      loginData$TestMode <<- FALSE
    )
  }
})
```

Code 3: Reactive functions getTest and getLearn

Finally, these checks and using a global variable shared with the server, can switch between "learn" and "test".

- 1.2 - As an administrator, I want that the application has different machine learning algorithms for the users, so they can do different tests.
- This requirement has been met through the implementation of different algorithms in the application.

```
source("PCA.R", local=T)
source("LDA.R", local=T)
source("QDA.R", local=T)
source("MCA.R", local=T)
source("NB.R", local=T)
source("MLP.R", local=T)
source("SVM.R", local=T)
source("KMEANS.R", local=T)
```

Code 4: Different models included

Web Interface to learn Machine Learning

As you can see in our core module we include different modules for each of the algorithms.

- 1.3 - As an administrator, I want that the application allows to have different users and a control of them, so the application would be usable for everybody.
- This requirement is fulfilled through a system of "logging" in the application. This "logging" is achieved via a module integrated into the "login.R" and a function that checks the user credentials to the database.

```
# Login user interface ----
loginUI <- function (){
  div(class= "",
      code,
      textInput("account", "Account",value="example@example.com"),
      passwordTextInput("pwd",label = "Password"),
      shiny::ActionButton("login", "Login", styleclass='success')
  )
}
```

Code 5 : Implementation of the Login System.

```
#login base de datos
doLoginBD <- reactive ({
  if (!is.null(input$login)) {
    if (input$login > 0) {
      con = dbConnect(MySQL(), user='root', password='1234', db='TFG', host='localhost')
      prueba <- input$account
      query = paste("select * from usuarios where correo =",prueba,"",sep='')
      result2<-dbGetQuery(con,query)
      dbDisconnect(con)
      pass <- input$pwd
      if(identical(pass,result2$password)){
        loginData$Account <- result2$correo
        loginData$LoggedIn <- TRUE
      }
    }
  }
})
```

Code 6: Credentials Checking on database

- 1.4 - As an administrator, I want that the application to be scalable and easy to change, so everybody can maintain it in production environment (Not functional).

Web Interface to learn Machine Learning

- This requirement has been met thanks that the implementation was done in modules as they are easily modifiable. A web system can also be scalable whenever the user simply wants to add a new instance horizontally and automatically doubling the number of users allowed.

3.2.2 - User:

- 2.1 -As a user, I want that the application allows me to enter by username and password for accessing to all features.
- This requirement is shared with the manager; it is met by the above.
- 2.2 -As a user, I want that the application allows me choose between different dataset, for testing different dataset for the same algorithm.
- This requirement has been met through the implementation of a system of selectable attributes.

```
column(4,br(),selectInput
      (inputId = "datasetSelection",
       label = "select Real/Create Dataset",
       choices = c("select Dataset","iris","decathlon","parabolic","fibonacci","tea","hobbies"),
       selected = selectedDataset()
      ),
```

Code 7: Implementation of a selectable attribute in a field.

- 2.3 - As a user, I want that the application allows me to choose one option between different tasks (classification, regression, clustering or feature extraction) for different testing algorithms

Web Interface to learn Machine Learning

- This requirement is fulfilled by implementing an attribute selection (dropdown)

```
selectInput  
(inputId = "typeSelection",  
  label = "Associated tasks",  
  choices = c("Select Type", "Classification", "Regression", "Feature Extraction", "Clustering"),  
  selected = selectedType()  
)
```

Code 8: Dropdown implementation with task selection.

- 2.4 - As a user, I want that the application allows me to execute different algorithms I have to see the performance.
- This requirement is fulfilled by, first of all, a drop-down list (attribute selection) where the user can choose the type of algorithm. It must be chosen according to the task previously selected. This is achieved using a "conditionalPanel".

```
conditionalPanel(  
  condition = "input.datasetSelection != 'select Dataset'  
  && input.typeSelection != 'select Type'  
  && input.datasetSelection != 'tea'  
  && input.datasetSelection != 'hobbies'  
  && input.typeSelection == 'Classification',  
  selectInput  
  (inputId = "algorithm2",  
   label = "select Algorithm",  
   choices = c("select", "LDA", "QDA", "NBAYES", "MLP", "SVM"),  
   selected = selectedAlgorithm2()  
  )  
)
```

Code 9: Conditional Panel implementation.

Once the user chooses the algorithm in question, it runs the corresponding module thus creating the appropriate interface.

Web Interface to learn Machine Learning

- 2.5 - As a user, I want that the application allows me to swap between graphics, text and table in the execution of each algorithm for checking the values of each.
- This requirement is fulfilled through the use of a "TabSetPanel", which can create tabs within a panel, so we just have to create a tab for each functionality. Shiny operates reactively; when the user changes some tab, the corresponding code will be executed.

```
mainPanel(  
  tabsetPanel(type = "tabs",  
             tabPanel("Plot", plotoutput("plot_kmeans")),  
             tabPanel("Plot original DataSet", plotoutput("summary_kmeans")),  
             tabPanel("Table", tableoutput("table_kmeans"))  
  )  
)
```

Code 10: TabPanel Implementation.

```
output$plot_mlp <- renderPlot({  
  datasetSelected <- input$datasetselection  
  typeAlgorithm <- input$typeselection # Clasificación o Regresión / Classification/ Regresion  
  porcentaje_obs <- as.numeric(input$observaciones)  
  decay <- as.numeric(input$decay)  
  size <- as.numeric(input$size)  
  maxit <- as.numeric(input$maxit)
```

Code 11: RenderPlot Implementation

- 2.6 - As a user, I want that the application allows me to swap between the section “learn” and the section “test” to access both.
- This requirement is shared with the administrator; it is met by the requirement mentioned in 1.1.
- 2.7 - As a user, I want that the application allows me to evaluate myself in the different tests (“easy”, “medium”, “difficult”) for letting me know my level in the subject.

Web Interface to learn Machine Learning

- This requirement has been met by an implementation that interacts between the module and the database server. First of all, take the server settings that the user has chosen. Once the user presses "START TEST", a reactive function is called from the server and is responsible for retrieving a random test from database and show it on the interface. When the user finishes the test, another reactive function is executed to calculate the result of the test.

```
testUI <- function (){  
  div(class= "",  
    selectInput("questions","Choose the number of questions", c("5","10", "15")),  
    selectInput("complexity","Choose the complexity of test",c('easy','medium','hard')),  
    .....
```

Code 12: Interface implementation of test part

```
startedTest<- reactive({  
  if(!is.null(input$start_test) && input$start_test > 0){  
    con = dbConnect(MySQL(), user='root', password='1234', db='TFG', host='localhost')  
    numPreguntas <- as.numeric(input$questions)  
    dificultad <- input$complexity  
    #Se escoge un Test Aleatorio segun los parametros de los usuarios  
    query1 = paste("select idTest from test where dificultad =",dificultad, "",sep='')  
    query2 = paste("and numeroPreguntas=",numPreguntas, "",sep='')  
    query = paste(query1,query2,sep='')  
    result2 <- dbGetQuery(con,query)  
    ids <- result2$idTest  
    .....
```

Code 13: Reactive function implementation that retrieve a random test from DB

- 2.8 - As a user, I want that the application allows me to check all of my results in the test part, for recovering any time the results.
- This requirement is fulfilled by a reactive function that is released when the user presses the button "SHOW MY RESULTS", which searches the database of the whole users and displays the results in an interface.

Web Interface to learn Machine Learning

```
output$myresults <- renderPrint({
  if(!is.null(input$show_history) && input$show_history > 0){
    con = dbConnect(MySQL(), user='root', password='1234', db='TFG', host='localhost')
    query = paste("SELECT dificultad, resultado,hora from testusuario where correo='", input$account,"'",sep='')
    results <- dbGetQuery(con,query)
    dbDisconnect(con)
    print(results)
  }
});
```

Code 14: Query implementation to database to retrieve the results of some user.

- 2.9 - As a user, I want that the application allows me to see graphics of the results, for checking them into more detail.
- This requirement is fulfilled like the requirement 2.8, the difference is that the results extracted from the database are printed with "barplot"

```
tot <- colMeans(p)
text(mp, tot + 3, format(tot), xpd = TRUE, col = "blue")
barplot(p$resultados_numeric, xlab = "Results of each test" , ylab = "score" ,
text0 <- paste("Results of Test " , dificultad, sep="")
title(main = text0, font.main = 4)
```

Code 15: RenderPlot task implementation

- 2.10 - As a user, I want that the application responds with an acceptable time (less than 5 seconds) in the execution of the algorithm, for not waiting time in front of the screen (Not functional)
- This requirement has been fulfilled because I have used the best implementation of the machine learning algorithms from R libraries. The execution time is also less than 4s taking into account that printing graphics will be the most expensive task.
- 2.11 - As a user, I want that the application responds efficiently with the changes of the parameters for improving interactivity with the application. (Not functional)

Web Interface to learn Machine Learning

- This requirement has been met thanks to the RShiny ability and its reactive functions. In case any change is made, the user is able to monitor it in real time.

4 - Technical skills Achievement

For the development of our project, we had a series of technical skills to achieve. These are defined below with the reasons we have been attained.

1. CCO1.3: Define, evaluate and select platforms of development and production for hardware and software for developing applications and services of varying complexity.
 - a. Level of achievement required: A little
 - b. Level of achievement acquired: A little
 - c. Justification: From the study of the different technology. This study was performed using a comparison of ("benchmarks") both programming languages and development environments (IDE). It has also tested in a successful way, that the software can run with a not so big hardware.
2. CCO2.1: Demonstrate knowledge of the basics, paradigms and own smart; analyse, design and build systems, services and applications using these techniques in any scope of technical systems.
 - a. Level of achievement required: quite
 - b. Level of achievement acquired: quite
 - c. Justification: It has been achieved by studying the different techniques of Machine Learning. These are necessary for the proper development of the application. The

Web Interface to learn Machine Learning

application also works correctly (refer to Example of application execution), we can say that we have reached the technical skill required.

3. CCO2.3: Develop and evaluate interactive systems, present complex information and its application to solve problems of individual computer interaction design
 - a. Level of achievement required: quite
 - b. Level of achievement acquired: quite
 - c. Justification: It has been achieved with the development of the application, as this is interactive with the user through the implementation of Shiny. Data also shows that for the user it is a useful and easy way to understand information.

4. CCO2.4: Demonstrate knowledge and develop computational learning techniques; design, implement applications and systems that use them, including those dedicated to the automatic extraction of information and knowledge from large volumes of data.
 - a. Level of achievement required: Deep
 - b. Level of achievement acquired: Deep
 - c. Justification: It has been achieved through the development of all Machine Learning techniques to handle large volumes of data, such as for example "K-means clustering", "NEURAL NETWORKS" and "SUPPORT VECTOR Machines". These techniques can be checked with execution of the application.

5. CCO2.6: Designing and implementing graphical applications, virtual reality, augmented reality and video games.
 - a. Level of achievement required: A little
 - b. Level of achievement acquired: A little
 - c. Justification: This skill has been achieved through the implementation of 3D plots in the application, along with the development of graphics.

Web Interface to learn Machine Learning

6. CCO3.1: Implement critical code following criteria runtime,
efficiency and safety.
 - a. Level of achievement required: quite
 - b. Level of achievement acquired: quite
 - c. Justification: It has been acquired through correct implementation, taking into account aspects of modularity but without neglecting execution times higher than could produce mass of this. They have used best practice at the time of development.

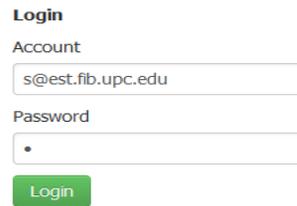
7. CCO3.2: Programming considering hardware architecture, both in assembler and high level.
 - a. Level of achievement required: A little
 - b. Level of achievement acquired: A little
 - c. Justification: It has been possible thanks to the test application to different computers. This application will also rise to scienceNodes portal and therefore will be tested in a real production environment.

5 - Example application execution

5.1 – Execution example part of Login

The first thing we find when the application starts is the "Login" interface, where the user can see how fields are required, then credentials must be inserted and press "Login"

Web Interface to learn Machine Learning



Login
Account
s@est.fib.upc.edu
Password
•
Login

Image 27: Login Interface

Once user has entered these two fields, the system will launch the "query" against the database to see if the user exists and if its credentials are correct. If so, the system will give access to the main application screen.

MACHINE LEARNING



Select Real/Create Dataset
Select Dataset ▼

User: s
Test Mode **Logout**

Image 28: Main Screen without selecting anything.

In this image you can see the interface that the user will see once authorized.

5.2 - Execution example part of learn

Web Interface to learn Machine Learning

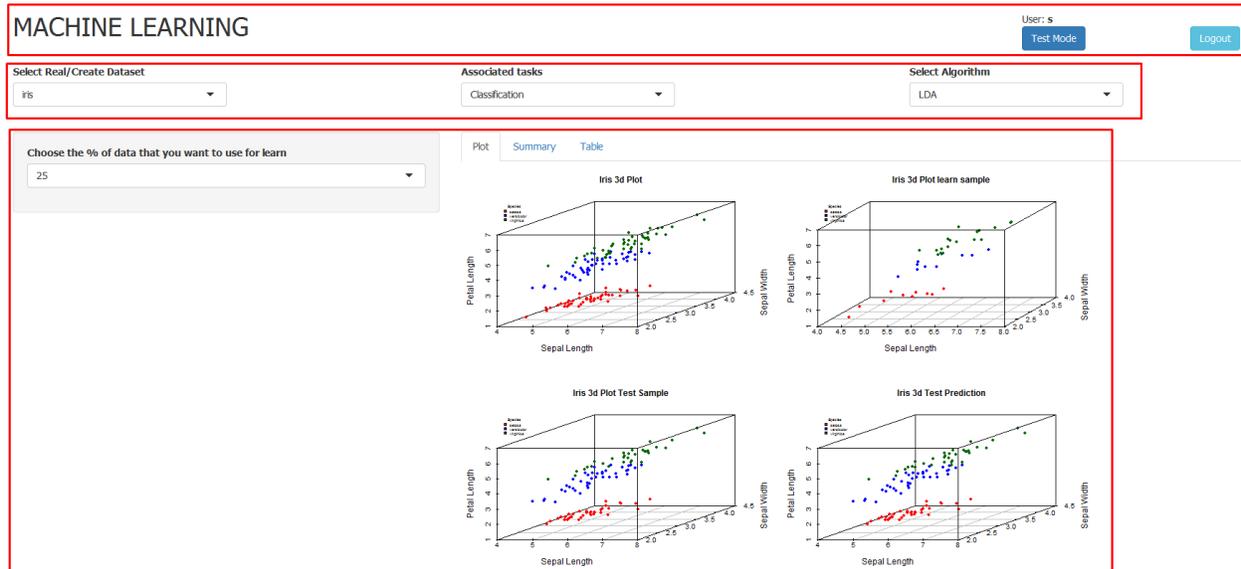


Image 29: Main Structure of Learn Task

The first part consists of a header which includes user's name, username and two buttons to switch to test mode and exit the application. This header is present in both interfaces.

As you can see in the picture above, we have made a selection of dataset (iris), we also have subsequently chosen as task to make "Classification" and finally the algorithm chose was the "LDA".

The charging time to show the results of LDA with default parameters has been less than 3 sec.

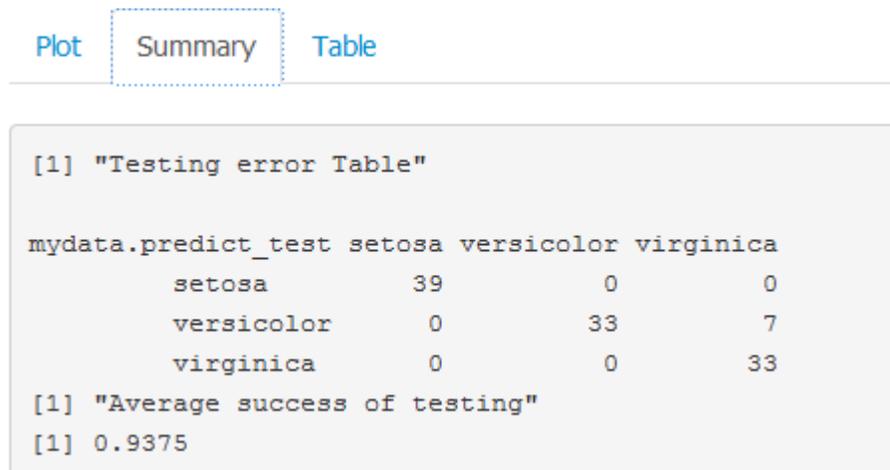
Once we have the results, we can modify the list of parameters on the sidebar panel to recalculate the algorithm in real time.

Web Interface to learn Machine Learning

As you can see the interaction with the user is very fluid, you can switch between all tabs easily and quickly.

Each of the tabs are:

- PlotTab:
 - In this tab, the user can check the results of the algorithm, typically in 4 different "plots". One is the original dataset, second the data used for learning, consistent in the original test, third the data test and last a 'plot "with the predicted data.
- SummaryTab:
 - In this tab the user can obtain useful information about the algorithm. It usually gets the error matrix method and the "success rate"



```
[1] "Testing error Table"

mydata.predict_test setosa versicolor virginica
  setosa           39         0         0
  versicolor       0         33         7
  virginica        0         0         33

[1] "Average success of testing"
[1] 0.9375
```

Image 30: Summary tab of an algorithm

- TableTab:
 - In this tab you can check the value of the dataset we have chosen to make the algorithm.

Web Interface to learn Machine Learning

Plot Summary **Table**

	Sepal.L.	Sepal.W.	Petal.L.	Petal.W.	Sp
1	5.10	3.50	1.40	0.20	setosa
2	4.90	3.00	1.40	0.20	setosa
3	4.70	3.20	1.30	0.20	setosa
4	4.60	3.10	1.50	0.20	setosa
5	5.00	3.60	1.40	0.20	setosa
6	5.40	3.90	1.70	0.40	setosa
7	4.60	3.40	1.40	0.30	setosa
8	5.00	3.40	1.50	0.20	setosa
9	4.40	2.90	1.40	0.20	setosa
10	4.90	3.10	1.50	0.10	setosa

Image 31: DataTable of Iris Dataset

5.3 - Execution example part of test

In this part we keep the "header" that we explained in the section of "learn" by modifying the button that redirects to another section.

This is the main screen of test

Web Interface to learn Machine Learning

Machine Learning Test

Choose the number of questions

5

Choose the complexity of test

easy

START TEST SHOW MY RESULTS PLOT MY RESULTS

User: s

Learn Mode

Logout

Image 32: Main Interface of test part.

As the main part of the learn section, the main test screen can be divided into a side panel (sidebar) and a central panel where it will appear questions, results or plots as desired by the user.

On the side panel there are two drop-down lists: the first one containing the number of questions the user wants to do in the test, and the second list with the level of difficulty of it.

I must say that the difficulty lies in the number of responses to the question. In the case of an “easy” difficulty test, there are only 2 answers and just one answer will be correct. In the case of a test of “medium” difficulty the number of answers will be 3 and also a single correct. And finally in the case of a test of “difficult” difficulty the number of answers will be between 3 and 5 and in this case the correct answers can be from 0 to all answers.

We also have three buttons, the first to start the test and the other to display the results (both text and plot).

Web Interface to learn Machine Learning

Choose the number of questions

5 ▼

Choose the complexity of test

easy ▼

START TEST SHOW MY RESULTS PLOT MY RESULTS

Image 33: Sidebar of test panel

If the user press “Start test”, a function is activated, which will show the questions one by one, go ahead and answer when the user clicks on the next button. This process continues until the end of the test.

Test of Machine Learning

Pregunta 2

Please Select:

Respuesta 1

Respuesta 2

Next

Image 34: Main questions panel.

Web Interface to learn Machine Learning

Other functionality is activated when pressing “showmyresults”, which shows all results for the current active user.

	dificultad	resultado	hora
1	easy	Total result of your test : 2/5	14-03-15 20:52:16
2	easy	Total result of your test : 5/5	14-03-15 21:07:58
3	easy	Total result of your test : 4/5	14-03-15 21:09:48
4	easy	Total result of your test : 3.5/5	14-03-15 21:09:53
5	easy	Total result of your test : 2/5	14-03-15 21:12:38
6	easy	Total result of your test : 2/5	14-03-15 21:12:44
7	medium	Total result of your test : 2/5	14-03-15 21:13:18
8	easy	Total result of your test : 10/10	03-04-15 17:42:27
9	easy	Total result of your test : 9.5/10	03-04-15 17:42:35
10	easy	Total result of your test : 3/15	03-04-15 18:12:33
11	easy	Total result of your test : 3/5	12-04-15 14:31:12
12	easy	Total result of your test : 3.5/5	12-04-15 14:31:20

Image 35: Main Panel of results

Finally, we have another similar feature to the previous one, but this will show the results of tests on a barplot.

Web Interface to learn Machine Learning

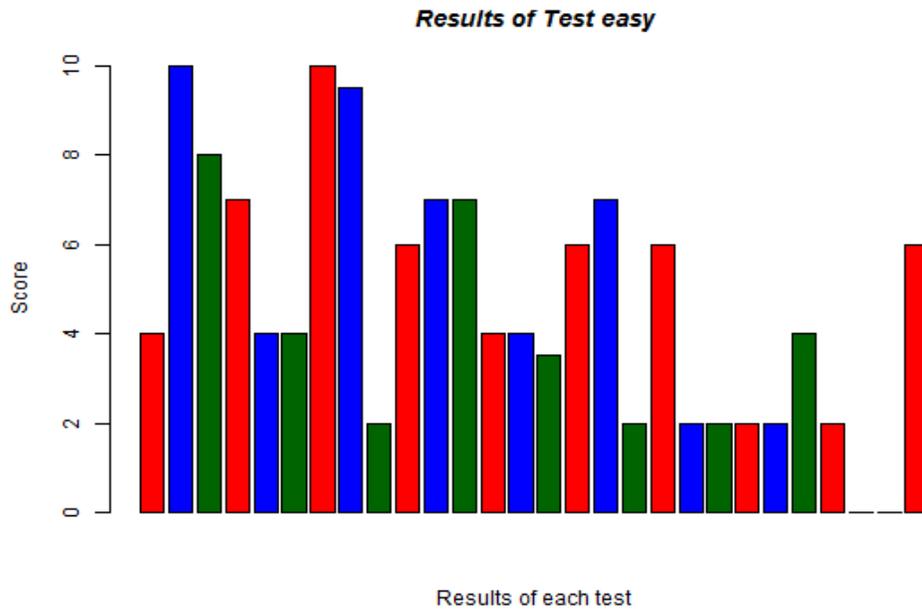


Image 36: Main Results with BarPlot

THIRD PART: IMPROVEMENTS AND CONCLUSIONS

Web Interface to learn Machine Learning

1 - Next improvements

Although we are pleased with the development of the project in general. This has many ways of continuity, many improvements that can be made and time issues that we could not do.

- We can implement a system of "gamification" to the test, consisting of the acquisition of trophies or "badges" for the user to overcome some challenges.
- It can incorporate a series of questions of Machine Learning since now we only have a mock template.
- Both test part and learn part can be improved by incorporating algorithms or even make the interface friendlier.
- Incorporate combined methods such as a method of Feature Extraction & classification at the same time.
- The user can set up and test its personal data.
- Upload the application in the portal science department LSI (<http://science.lsi.upc.edu/>)

2 - Conclusions

The project objectives were to create a web interface that enables to learn Machine Learning, in addition, this application should be useful, usable and facilitate the monitoring of the student's evolution in the course. Then the user could evaluate their knowledge, therefore the interface resulting from this project was appropriately solved, although initially proposed improvement objectives.

This is a user-friendly interface, enabling the user improve its knowledge of Machine Learning.

The final interface is very similar to what we thought initially. It has minor changes to make it easier for the user, and to improve usability.

As can be seen in the section on Future Improvement, there is still work to do because the application isn't absolutely perfect.

During the project we had some problems, such as understanding how internally shiny works and reactive functions. Otherwise the system "login" where database costs more than it was initially thought. Finally, we had a problem planning to find a job that will significantly reduce the time that I can use daily to the TFG.

Finally, as personal conclusions we can say that this project was very interesting because it has given me experience in a field that was attractive to me, besides the satisfaction this application will be used by end users, i.e. will be released, probably because that application appears in the portal science department LSI (<http://science.lsi.upc.edu/>).

Web Interface to learn Machine Learning

It has also been an enriching experience both academic and professional, as this is a field that is growing in the market today, and I have learned and have ideas for new tools.

References

- [1] MACHINE LEARNING- <http://personal.disco.unimib.it/Vanneschi/McGrawHill - Machine Learning -Tom Mitchell.pdf> - Consulted at 05/10/2014
- [2] APRENTATGE AUTOMÀTIC - <http://www.lsi.upc.edu/~bejar/apren/docum/trans/00-introAprendizaje-eng.pdf> - Consulted at 05/10/2014
- [3] http://cran.r-project.org/doc/contrib/rdebut_es.pdf - Consulted at 06/10/2014
- [4] <http://www.r-project.org/> - Consulted at 06/10/2014
- [5] <http://www.rstudio.com/products/rstudio/> - Consulted at 06/10/2014
- [6] <https://github.com/rstudio/shiny/blob/master/README.md> - Consultat a 07/10/2014
- [7] <http://getbootstrap.com/> - Consulted at 22/10/2014
- [8] <http://shiny.rstudio.com/gallery/kmeans-example.html> - Consulted at 15/10/2014
- [9] <http://www.pmoinformatica.com/2012/10/plantillas-scrum-historias-de-usuario.html> - Consulted at 10/01/2015
- [10] SCRUM EN 15 MINUTOS. <http://es.slideshare.net/rodrigoiscrum-en-15-minutos> - Consulted at 06/10/2015
- [11] MATLAB - <http://es.mathworks.com/products/matlab/> - Consulted at 05/11/2014
- [12] ALPAYDIN, E , INTRODUCTION TO MACHINE LEARNING , MIT PRESS , 2010 , ISBN:026201243X.
http://cataleg.upc.edu/record=b1375315~S1*cat
- [13] BISHOP, C.M , PATTERN RECOGNITION AND MACHINE LEARNING , SPRINGER , 2006 , ISBN:0387310738.
http://cataleg.upc.edu/record=b1298151~S1*cat
- [14] CHERKASSKY, V.; MULIER, F , LEARNING FROM DATA: CONCEPTS, THEORY, AND METHODS , JOHN WILEY , 2007 , ISBN:0471681822. http://cataleg.upc.edu/record=b1352176~S1*cat
- [15] HAYKIN, S.S , NEURAL NETWORKS AND LEARNING MACHINES , PRENTICE HALL , 2009 , ISBN:0131471392.

Web Interface to learn Machine Learning

http://cataleg.upc.edu/record=b1342098~S1*cat

[16] [MITCHELL, T.M, MACHINE LEARNING, THE MCGRAW-COMPANIES, 1997, ISBN:0070428077.](#)

[HILL](#)

http://cataleg.upc.edu/record=b1133468~S1*cat

[17] [FEATURE EXTRACTION](#) - http://en.wikipedia.org/wiki/Feature_extraction - Consulted at 5/03/2015

[18] [CLASSIFICACIÓ](#) - <http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf> – Consulted at 8/03/2015

[19] [REGRESIÓ](#) http://en.wikipedia.org/wiki/Regression_analysis - Consultat a 09/03/2015 - Consulted at 12/03/2015

[20] [CLUSTERING](#) – <http://www.cs.upc.edu/~bejar/apren/docum/trans/08-clustering-eng.pdf> - Consulted at 15/03/2015

[21] [CLUSTERING](#) - http://en.wikipedia.org/wiki/Cluster_analysis - Consulted at 15/03/2015

Glossary

(1) - **Reactive function** → A reactive function is a function that is called automatically from client's side when a user interaction occur. So we can generate real time results for every user input.

(1.1) - **Input** → Action and value of a field in application to the algorithm.

(2) **Module** → A class or piece of code in R language.

(3) - **Supervised learning** → It is said that a supervised learning problem happens after a sample training ("learn"). They are known values (x, y) building a function that allows to predict the unknown value of "y" for a new observation "x."

(4) - **dataset** → Dataset input of our algorithms

Web Interface to learn Machine Learning

(5) – **Unsupervised learning** → In unsupervised learning, unlike supervised, it only has values of "x" without knowing a value of "y".

Annexes

User Manual

Administrator Manual.

Gantt Diagram Changed

USER MANUAL

This is the user manual to understand the flow of application and to have the basics for starting to use the application in the correct way.

*Web interface to
learn machine
learning*

INDEX

1 - LOGGING	2
2 - HEADER	3
3 - LEARN PART	4
4 - TEST PART	9

Logging to application

1 - Logging

To start using the application first of all you need to login with your credentials (These credentials should be given to you by your teacher).

When you open the application, the login layout will popup to you, please complete the fields to login.

The screenshot shows a login interface. At the top, the word "Login" is displayed. Below it, the label "Account" is positioned above a text input field containing the text "sonia.sandalina:". A red rectangular box highlights this input field, with a red number "1" to its right. Below the "Account" field, the label "Password" is positioned above a password input field filled with ten black dots. A red rectangular box highlights this password field, with a red number "2" to its right. At the bottom of the form is a green button with the text "Login" in white.

1. Here you need to type your username
2. Here you need to type your password

LEARN AND TEST PARTS

2 - HEADER

If credentials are correct ,the system will redirect you to the first page of application.

MACHINE LEARNING



Here is the **header** of application where you can see your username (1) , you can change between learn/test mode(2), and you have the option to logout(3) from application.

3 - LEARN PART

Now we start to explain how to use the **learn** part.

- 1) First you need to select between Real datasets (iris, decathlon, tea, hobbies) or created datasets by system (parabolic, Fibonacci)

Select Real/Create Dataset

Select Real/Create Dataset

- Select Dataset
- iris
- decathlon
- parabolic
- fibonacci
- tea
- hobbies

- 2) When you select one of this datasets another selected list will appear at the right . In this you need to select what associated task to do with the dataset selected (Classification, Regression, Feature Extraction or Clustering). **Not all datasets are valid to do all tasks.**

Associated tasks

Associated tasks

Select Type

- Select Type
- Classification
- Regression
- Feature Extraction
- Clustering

3) Depending on associated task selected, another selected field will appear at the right, at this moment you will select the algorithm to perform with dataset selected.

a. For example : Iris → Classification → LDA

Select Real/Create Dataset: iris

Associated tasks: Classification

Select Algorithm: LDA

As we said, depending on associated task, algorithms can vary.

- If we select Classification :
 - LDA
 - QDA
 - NBAYES
 - MLP
 - SVM

Associated tasks: Classification

Select Algorithm: LDA

- If we select Regression :
 - MLP
 - SVM

USER MANUAL

Associated tasks

Regression

Select Algorithm

Select|

Select

MLP

SVM

- If we select Feature Extraction
 - PCA
 - LDA
 - MCA (Only “Tea”, “Hobbies”)

Associated tasks

Feature Extraction

Select Algorithm

Select|

Select

PCA

LDA

Associated tasks

Feature Extraction

For "Tea" and "Hobbies" Datasets

Select Algorithm

Select|

Select

MCA

- If we select Clustering
 - KMEANS

Associated tasks

Clustering

Select Algorithm

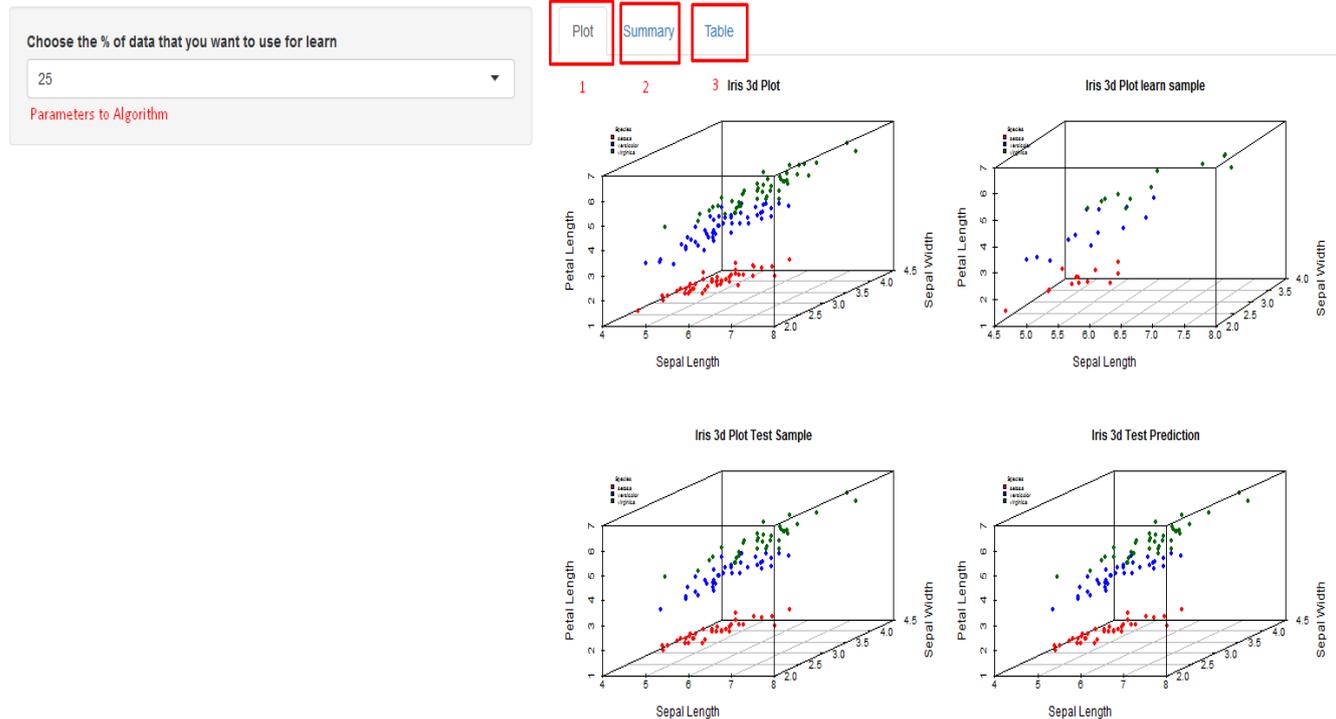
KMEANS

So, now, we going to explain how to change parameters and how to visualize results.

Imagine for example, that we select “Iris” , “Classification” , and “LDA” as algorithm.

When we click to LDA, a new panel will be rendered, see the next image.

USER MANUAL



Mostly, you can change parameters to visualize how the parameters affect to our dataset.

You have 3 tabs:

- 1) The first tab is where you can see the result of the algorithm by graphics.
- 2) The second tab, summary, is where you can see the results of the algorithm in text format.
- 3) The last tab, “Table”, is where you can see the entire dataset.

If you change the parameter to another value(for example 50%), the graphics and the rest of tabs will upload with the new result.

All tabs are clickable, and you can change between them without losing the results.

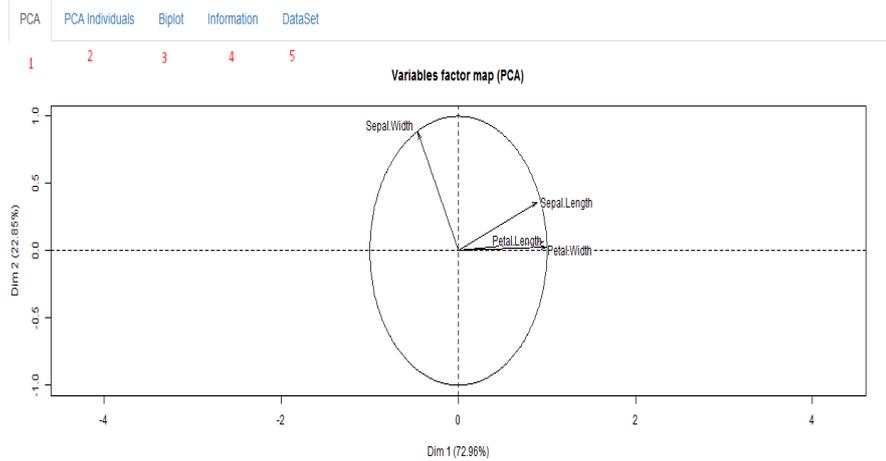
Another example :

We are going to select “iris” , “feature extraction” and “PCA”

As we can see in the next image, is a little different than LDA from classification, this is because we need to show you different results in different plots, but the usability is the same, you can change between tabs as you want.

USER MANUAL

This method is used to find the causes of variability in a dataset and sort them by importance



One , two and Third tab, are graphics to see how PCA resolve the dataset, the fourth tab is a summary of the result and the fifth tab is the same as “table” , the dataset printed.

4 - TEST PART

Now , we start to explain the **test** mode

First of all, to enter to “test” mode, you need to click the button at top right , see the image below.



When you click to this button you will be redirected to the test layout.

Machine Learning Test

Choose the number of questions

Choose the complexity of test



Here you can perform different tests, to evaluate your knowledge about machine learning.

At left you can see two selected fields, one of them is the number of questions of the test (**5,10,15**) , and the other is the complexity (**easy, medium, hard**)

As you can see, there are three buttons.

- **START TEST** : If you click in this button, you start a new test with the parameters selected

USER MANUAL

Choose the number of questions
5

Choose the complexity of test
easy

START TEST **SHOW MY RESULTS** **PLOT MY RESULTS**

Test of Machine Learning

Welcome to Machine Learning Test

Next

Now you need to push “next” to start the test, and first question will appear, when you answer this question click another time to next, when the test is finished, your results will be shown

Test of Machine Learning

Pregunta 1

Please Select:

- Respuesta 1
- Respuesta 2

Next

Test of Machine Learning

Your results

```
[1] "Question 1 has: CORRECT ANSWER"  
[1] "Question 2 has: CORRECT ANSWER"  
[1] "Question 3 has: CORRECT ANSWER"  
[1] "Question 4 has: INCORRECT ANSWER"  
[1] "Question 5 has: INCORRECT ANSWER"  
[1] "Total result of your test : 3/5"
```

Next

SHOW MY RESULTS : If you click in this button, all your tests will be shown at the right.

USER MANUAL

Choose the number of questions
5

Choose the complexity of test
hard

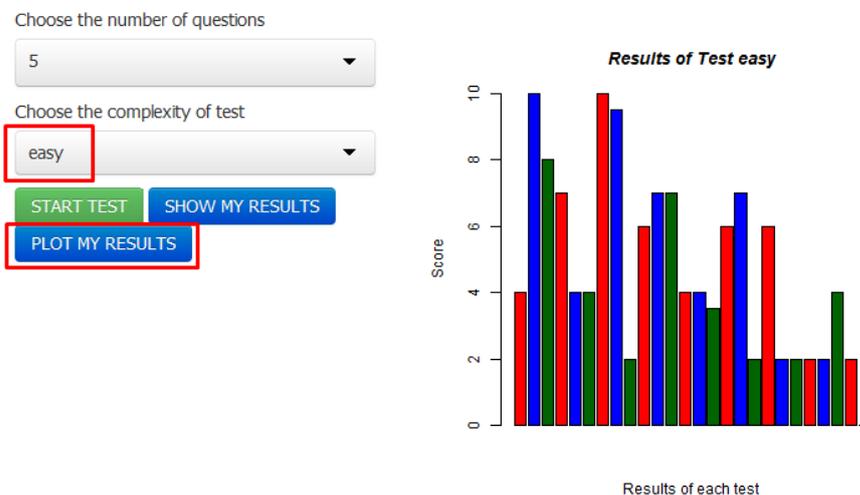
START TEST SHOW MY RESULTS PLOT MY RESULTS

	dificultad	resultado	hora
1	easy	Total result of your test : 2/5	14-03-15 20:52:16
2	easy	Total result of your test : 5/5	14-03-15 21:07:58
3	easy	Total result of your test : 4/5	14-03-15 21:09:48
4	easy	Total result of your test : 3.5/5	14-03-15 21:09:53
5	easy	Total result of your test : 2/5	14-03-15 21:12:38
6	easy	Total result of your test : 2/5	14-03-15 21:12:44
7	medium	Total result of your test : 2/5	14-03-15 21:13:18
8	easy	Total result of your test : 10/10	03-04-15 17:42:27
9	easy	Total result of your test : 9.5/10	03-04-15 17:42:35
10	easy	Total result of your test : 3/15	03-04-15 18:12:33
11	easy	Total result of your test : 3/5	12-04-15 14:31:12
12	easy	Total result of your test : 3.5/5	12-04-15 14:31:20
13	easy	Total result of your test : 3.5/5	12-04-15 14:31:24
14	easy	Total result of your test : 4/10	12-04-15 14:31:43
15	easy	Total result of your test : 6/15	12-04-15 14:31:57
16	easy	Total result of your test : 3.5/10	12-04-15 14:32:06

As you can see in the image, all of your results, with complexity, result and date will show when you press “Show my results”

- PLOT MY RESULTS : Depending on complexity that are selected, a plot with total of results will appear

If you press in “plot my results” with complexity “easy” or “medium” or “hard” a plot with all of your tests that satisfy the criteria will appear.



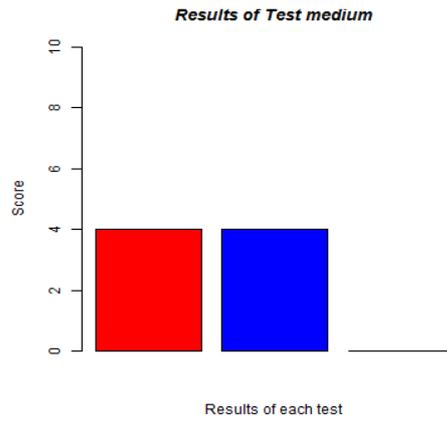
USER MANUAL

Choose the number of questions

Choose the complexity of test

START TEST SHOW MY RESULTS

PLOT MY RESULTS

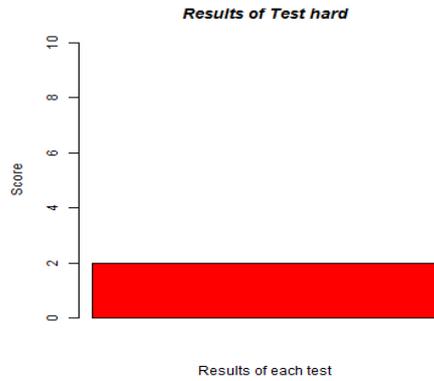


Choose the number of questions

Choose the complexity of test

START TEST SHOW MY RESULTS

PLOT MY RESULTS



ADMIN MANUAL

This manual will help you to create a database for the web application :
"Aprendre Aprenentatge"

*How to Create
Database and
upload questions
and tests*

INDEX

1 – MYSQL	2
2 – WORKBENCH	2
3 – WORKBENCH INSTANCE	3
4 - SCHEMA	5
CREATE SCHEMA QUERY	5
5 – TABLES	5
CREATE TABLE QUERIES.	5
CREATE TEST REGISTER	7
CREATE GRUPOPREGUNTAS REGISTER	7
CREATE PREGUNTAS REGISTER	8
CREATE RESPUESTAS REGISTER	8
CREATE USERS	9

REQUIREMENTS

1 - MYSQL

First of all , you need to download the last version of MySQL.

You can download here : <http://www.mysql.com/downloads/>

In order to facilitate the process to integrate with web application, we suggest that you should create a user “root” with password “1234”.

2 - WorkBench

We strongly recommend to use MySQL Workbench or PhpMyAdmin or something similar to create the schema, in this manual we explain how to create through text queries in MySQL Workbench 6.2.

Download link : <https://dev.mysql.com/downloads/workbench/>

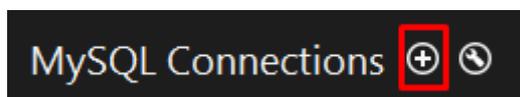
CREATING THE INSTANCE

3 – Workbench Instance

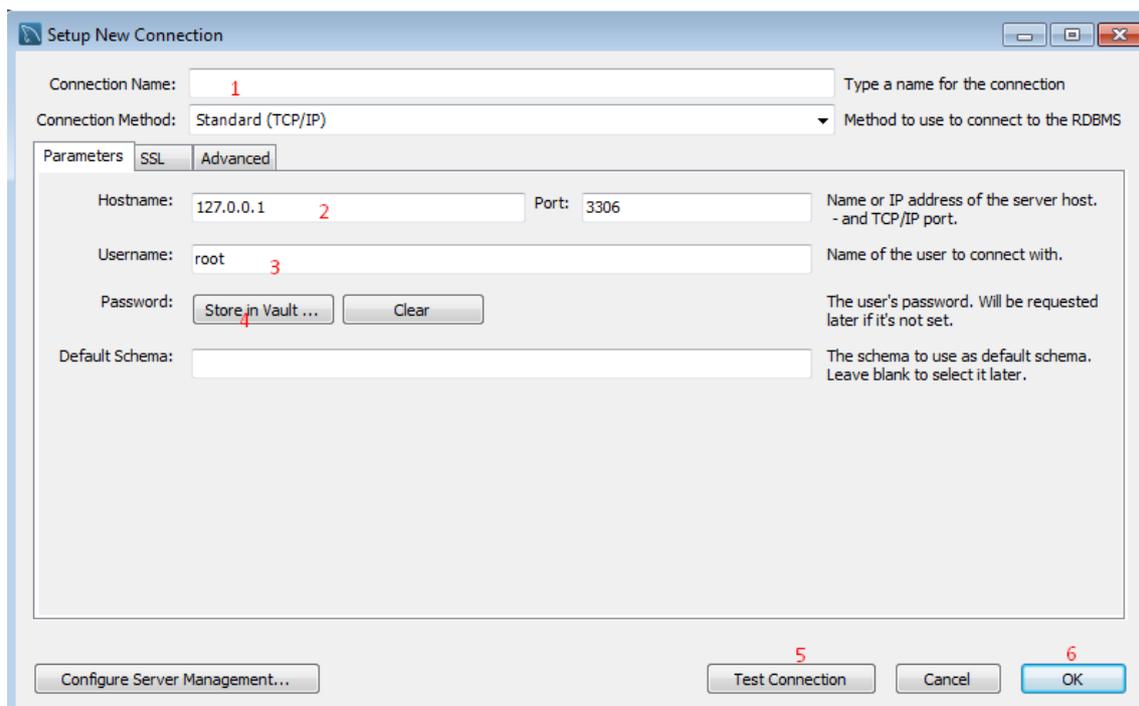
In order to create schema, please open your “MySQL Workbench”



When opened, please add a new connection.



When you click in add new connection, another window will appear.

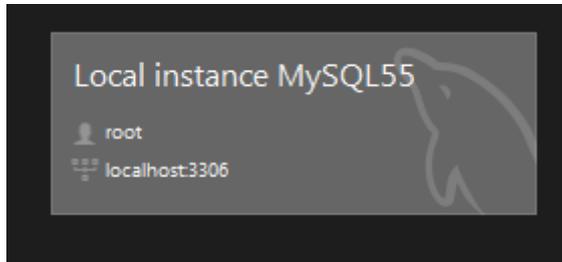


Here you can add your new connection:

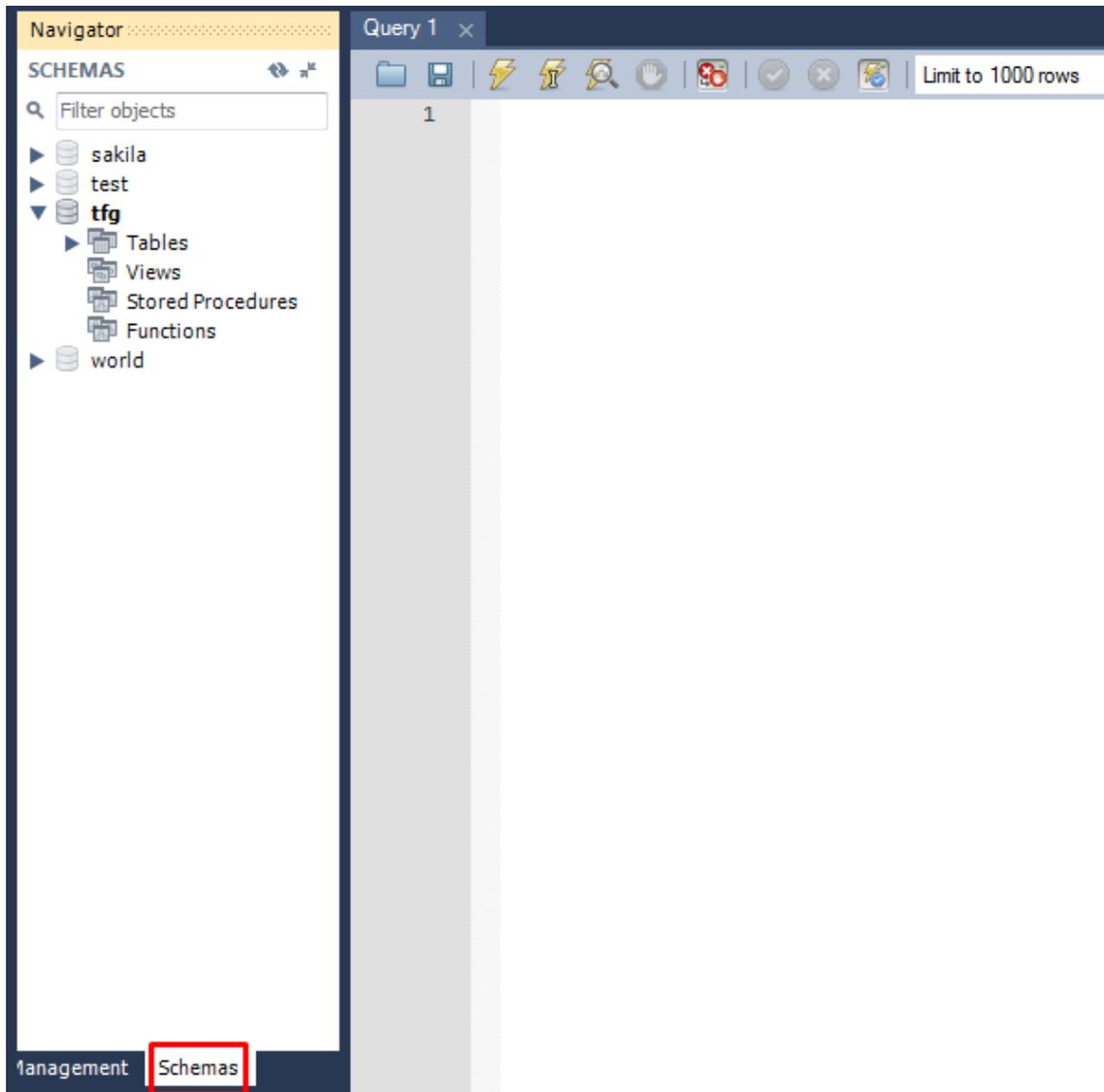
1. Connection name : A name that helps you to identify instance, for example “LOCAL INSTANCE”
2. Hostname : If you are in development, 127.0.0.1 is perfect, if you want to point to some server, here you need to write the public ip address.
3. Username : As we said in requirements, we recommend you to assign this value to “root”.

Admin MANUAL

4. When you press “Store in Vault...” a new window when you need to write the password (1234) appear.
5. Finally , test your connection, is everything is ok go to point 6
6. Save connection and use it.



Now, you need to log in our instance with double click and then, another window will appear.



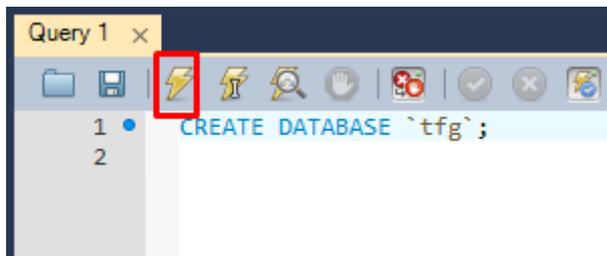
Creating the SCHEMA

4 - Schema

Now that you are logged to mysql workbench, you need to create schema for this please write the next sentence in right place of your workbench and press the beam icon.

Create schema query

```
CREATE DATABASE `tfg`;
```



5 - Tables

Now, we're going to create all needed tables.

Create table queries.

```
CREATE TABLE `usuarios` (  
  `idusuarios` int(11) NOT NULL AUTO_INCREMENT,  
  `correo` varchar(220) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  PRIMARY KEY (`idusuarios`),  
  UNIQUE KEY `idusuarios_UNIQUE` (`idusuarios`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `grupopreguntas` (  
  `idgrupoPreguntas` int(11) NOT NULL,  
  `idTest` varchar(45) NOT NULL,  
  `idPreguntas` varchar(250) NOT NULL,  
  PRIMARY KEY (`idgrupoPreguntas`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

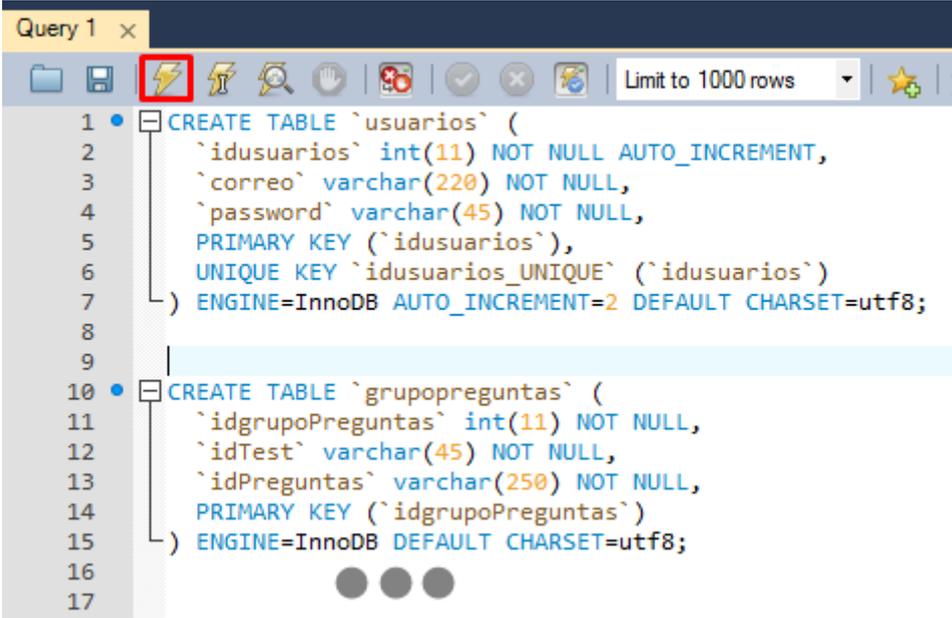
Admin MANUAL

```
CREATE TABLE `preguntas` (  
  `idPregunta` int(11) NOT NULL,  
  `respuestasCorrectas` varchar(220) DEFAULT NULL,  
  `respuestasIncorrectas` varchar(220) DEFAULT NULL,  
  `pregunta` varchar(500) NOT NULL,  
  PRIMARY KEY (`idPregunta`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `respuestas` (  
  `idRespuesta` int(11) NOT NULL,  
  `respuesta` varchar(500) NOT NULL,  
  PRIMARY KEY (`idRespuesta`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `test` (  
  `idTest` int(11) NOT NULL AUTO_INCREMENT,  
  `dificultad` varchar(45) NOT NULL,  
  `numeroPreguntas` int(11) NOT NULL,  
  PRIMARY KEY (`idTest`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `testusuario` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `correo` varchar(250) NOT NULL,  
  `dificultad` varchar(250) NOT NULL,  
  `resultado` varchar(250) NOT NULL,  
  `hora` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `usuarioId_idx` (`correo`),  
  KEY `testId_idx` (`dificultad`),  
  KEY `correo_index` (`correo`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=55 DEFAULT CHARSET=utf8;
```



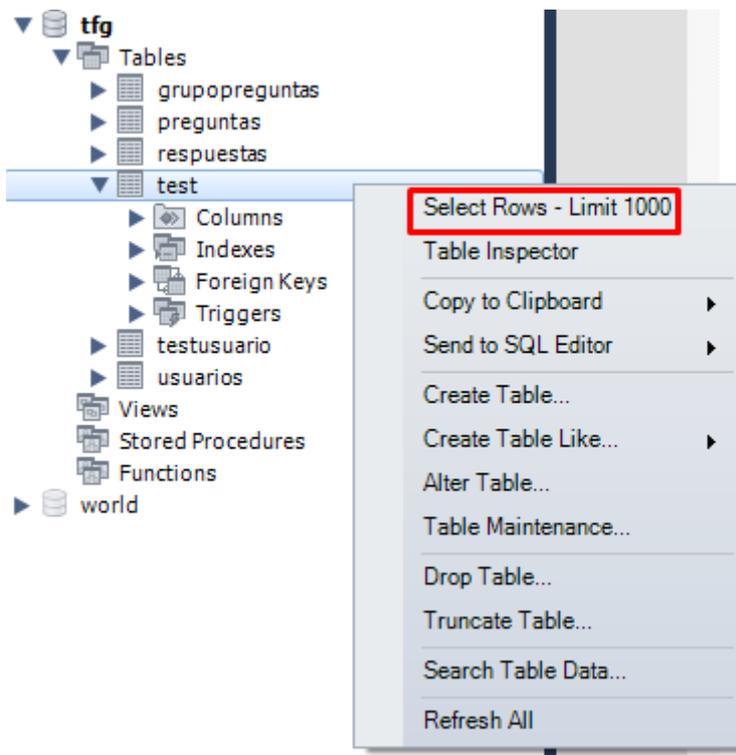
The screenshot shows a MySQL query editor window titled "Query 1". The toolbar includes icons for file operations, a lightning bolt icon (highlighted with a red box), search, and other standard database tools. The "Limit to 1000 rows" dropdown is visible. The SQL code in the editor is as follows:

```
1 CREATE TABLE `usuarios` (  
2   `idusuarios` int(11) NOT NULL AUTO_INCREMENT,  
3   `correo` varchar(220) NOT NULL,  
4   `password` varchar(45) NOT NULL,  
5   PRIMARY KEY (`idusuarios`),  
6   UNIQUE KEY `idusuarios_UNIQUE` (`idusuarios`)  
7 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
8  
9  
10 CREATE TABLE `grupopreguntas` (  
11   `idgrupoPreguntas` int(11) NOT NULL,  
12   `idTest` varchar(45) NOT NULL,  
13   `idPreguntas` varchar(250) NOT NULL,  
14   PRIMARY KEY (`idgrupoPreguntas`)  
15 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
16  
17
```

CREATING TESTS AND QUESTIONS

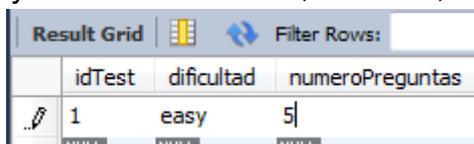
Now we start to explain how to use the create 1 Test and assign them some questions and answers.

For this example we suppose we want an “easy” test with 5 questions.



Create Test register

- First, you need to right click on “test” and select “Select rows...”
- Then , you need to write “id,dificultad,numeropreguntas” on right panel



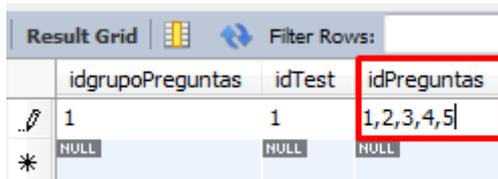
- And to save this, press “Apply” in bottom right



Create GrupoPreguntas register

Admin MANUAL

- For insert a new “grupopreguntas”, you need to do the same as “test” , the next image show an example of input.

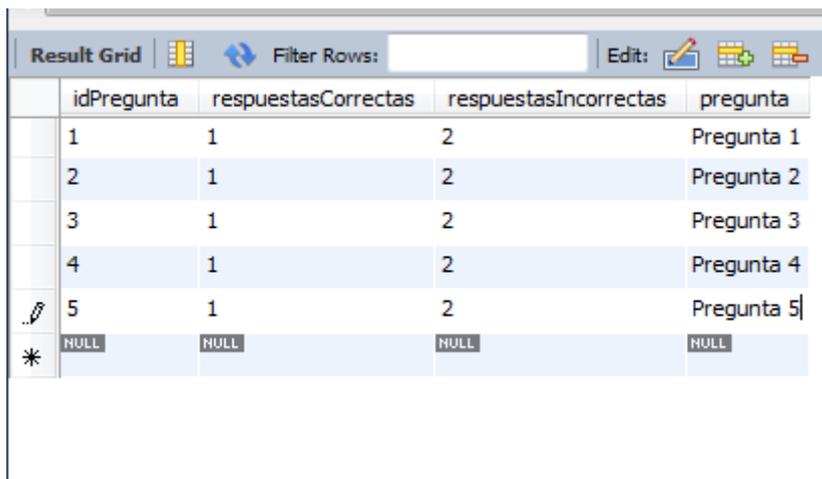


	idgrupoPreguntas	idTest	idPreguntas
✎	1	1	1,2,3,4,5
*	NULL	NULL	NULL

“idPreguntas” is a string in Comma separated format of id’s of questions in table “preguntas”.

Create Preguntas register

- For create a “Preguntas” register , you need to do the same, but here, you need to insert five register, one per question, and id’s of question should be the same as “idPreguntas” in table “grupoPreguntas”, in the next image you will se an example.



	idPregunta	respuestasCorrectas	respuestasIncorrectas	pregunta
	1	1	2	Pregunta 1
	2	1	2	Pregunta 2
	3	1	2	Pregunta 3
	4	1	2	Pregunta 4
✎	5	1	2	Pregunta 5
*	NULL	NULL	NULL	NULL

Create Respuestas register

- And for finish, we need to complete the table “Respuestas” with the text of answer for each id in “preguntas.respuestasCorrectas” and “preguntas.respuestasIncorrectas”. See the next image for example.

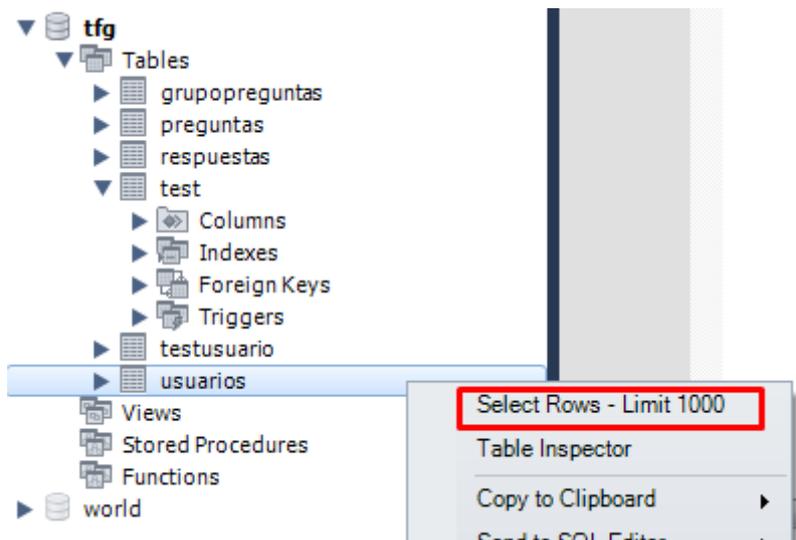
Result Grid		Filter Rows:
	idRespuesta	respuesta
	1	Respuesta 1
	2	Respuesta 2
*	NULL	NULL

When all this inserts are finished, you can go to the application and perform a “easy” test with 5 questions.

CREATE USERS

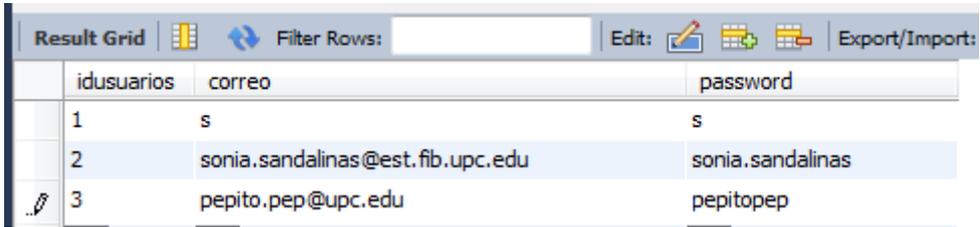
Create users

If you want to create users to grant acces to web application, you need to go to “usuarios” table.



Admin MANUAL

Right click in this table and choose “Select Rows ...” , On the bottom right panel you can insert all the records that you want. See the next image for example



	idusuarios	correo	password
1		s	s
2		sonia.sandalinas@est.fib.upc.edu	sonia.sandalinas
3		pepito.pep@upc.edu	pepitopep

When you finish to add register, don't forget to press “Apply” button to save all.

Gantt

