BACHELOR THESIS

IMPLEMENTING NEW INTERPRETATION ORIENTED TOOLS IN KLASS TO SUPPORT DECISION MAKING BASED ON LOGISTIC REGRESSION

Supervisor :

Dr. Karina Gibert

Candidate:

Mandadapu Lavanya

Implementing new Interpretation oriented tools in KLASS to support decision making based on Logistic regression

Mandadapu Lavanya

February 2018 - June 2018

This if for you, Jahnavi Mandadapu

Acknowledgements

First of all, I would like to thank my thesis supervisor, Dr. Karina Gibert, for the willingness, the constant support, the advice and the help, at any time of the day and the night during these months that I spent in Barcelona.

Thanks to SASTRA university for believing in me and giving me this opportunity to do my Bachelor thesis in Barcelona. Thanks to FIB for accepting me as an exchange student.

Thanks to Bea, Luis, Carlos and Johnny for all the support and help that you have provided during this thesis and thanks for being such sweet colleagues in this journey. Thanks to Toni Font for solving all the technical issues I have faced during the installation of softwares.

Thanks to Vinnu and RamaKrishnan for being there with me and for ensuring that good times keep flowing. Thanks to the flatmates and friends that I met during this experience, because they made it as good as it was, and easier than it would've been otherwise.

Finally, last but not least, the deepest thanks go to my family, who never stopped believing in me, for supporting me, for paying rent and letting me live where I felt like and for all they are, will be and will do hereafter.

Contents

Chapter 1

Introduction

In the context of problem solving, decision making involves choosing between possible solutions. Using the data available on a particular problem, we can process and choose the right decision. To get useful knowledge from available data we use Knowledge discovery from databases (KDD).

Taking decisions on data which has both independent and dependent variables is risky. We need to find strong relation between these variables for accurate decisions when dependent variable is qualitative. One way is to build discriminant models. This means to be able to recognize the class of a new object provided that a proper discriminant model has been previously trained. Among the available discriminant models, logistic regression is a proposal coming from Statistical field.

Logistic regression permits to determine a qualitative outcome by analyzing a dataset in which there are one or more independent variables.The outcome (often binary) is used to explain the relationship between one dependent binary variable and one or more independent variables. For these reasons, useful application of logistic regression analysis can be found for many fields, some of which are :

- Healthcare : Finding factors related with myocardial infarction from the data of an entire population followed within a period of 10 years.
- Fraud Detection : To recognize fraudulent credit card usages from a database of Internet credit card transactions before approval of operations.

In intelligent decision support systems, logistic regression can be used as an internal component that predicts the expected class of a new case to which certain decisions or actions will be assigned. In fact, logistic regression (as other classifiers coming from the machine learning fields) contribute in any context where a diagnosis is required for future treatment.

In this project a post-processing of logistic regression that helps to understand and visualize the situation of cases with regards to the plausible diagnosis is presented and treated over a real data set.

1.1 Project Goals

This project is part of an almost 30-year long research line of combination of statistics methods and Artificial Intelligence for Knowledge Discovery in ill-structured real domains, conducted by Phd Karina Gibert from the Department of Statistics and Operations Research of the Universitat Politècnica de Catalunya-BarcelonaTech (UPC) of Barcelona,Spain. The developments achieved along this years are implemented in a application called **JavaKlass**, which is currently in its 19th version.

JavaKlass was conceived as a system to support knowledge discovery from data in complex domains. Along the years it has been including new functionalities that come from very first versions, mainly oriented to clustering with heterogeneous data, towards high level operations such as automatic interpretation of clusters and concepts induction from clusters and so on. In this particular context, the work [Gibert 2013] is developing the EBLR (Embedded binary logistic regression) methodology that is able to combine several logistic regressions, trained after a clustering process, to provide a certain interpretation of what classes mean. The paper also proposes a visualization of this results in what its called a Profile Assessment Grid (PAG), which is a kind of cubic representation of the logistic regression results for a particular situation in which only 4 classes have to be assessed, under the hypothesis that these classes correspond to an ordinal qualitative variable, that means they can be properly sorted in ascendant or descendant way.

This work shows the implementation of all required functionalities to upgrade KLASS to incorporate the EBLR methodology and preliminary visualization of the PAG. To that purpose some specific secondary goals arise.

In practice, the new functionalities that this project brings to the Java-KLASS are the following:

- New Data management functionalities, which includes :
    - generation of dummy variables for a selected qualitative variable
    - generation of a new aggregated data matrix from the original dataset according to an aggregation factor or qualitative variable
- New Modelling functionalities :
    - Multiple linear regression
    - Binary logistic regression
- Model assessment functionalities :
    - goodness of fit indicators for Multiple linear regression and logistic regression.
    - graphical residual analysis.

- Model evaluation functionalities, which include :
    - evaluation of linear and logistic regression models over a certain dataset and generation of new variables with predicted values.
- Model management functionalities:
    - Some required functionalities to deal with all previously trained models available for a certain dataset.
- Input / output operations :
    - exporting and importing MLR files.
    - exporting and importing LogReg files.

We will present them one by one, explaining how and why they were designed and realized, with screenshots and examples of usage.

In the end, we will perform a few case studies, investigating on the behaviour of the modelling functionalities and benchmarking the results with R. KLASS allows to visualize exhaustive LaTex reports, so we will present the actual results that the package produced.

The structure of this work is, first we will introduce the reference system KLASS and methods and concepts needed to understand multiple linear regression and logistic regression then we will explain data management functionalities followed by modelling functionalities and Input/ Output functionalities.

Chapter 2

The Reference System KLASS

2.1 Introduction to KLASS

KLASS is a software originally designed by karina Gibert which first appeared in her Master thesis in 1991 [Gibert 91] and later in her PhD thesis [Gibert 94]. In its original version, it was a system oriented to the automatic classification of ill-structured domains, implemented in LISP over the Unix operating system. Over the years, however, it has grown and changed, in order to widen its range of functionalities and stay up to date with the new methods and algorithms that have been developed in the research group. Portability was a major factor of enhancement, too, as after more than a decade of development, it was necessary for KLASS to be easily transferred to other operating systems, for various reasons. Among them:

- the decision of UPC to cancel the LISP licence in the early 90s;
- the different policies outside UPC about computer equipment;
- the need to easily send the system to other teams and/or end-users for it to work properly without the requirement of a Lisp license;
- the discomfort of having to send the whole source code for an outside user to be able to use the system, as Lisp is an interpreted language that doesn't permit to generate executable files.

For these reasons a new version of KLASS was developed in Java, named **Java-KLASS** . Currently [Gibert 05][Gibert 08], Java-KLASS offers functionalities for:

- representation of the data matrices;
- data management and sampling;
- Knowledge base management;
- ontology management;
- extended descriptive statistics of data and 3D visualization;
- automatic reasoning;
- interoperability of methods;
- dynamic analysis[SCAM+10];
- automatic interpretation of classes[GCV12][GRSA13] ,both graphically[GC14][GGRRS08] and conceptually[gibert 2014];
- KLASS works with wide range of distance metrics including mixed distance metrics [MATHWARE 1997], semantic distances;

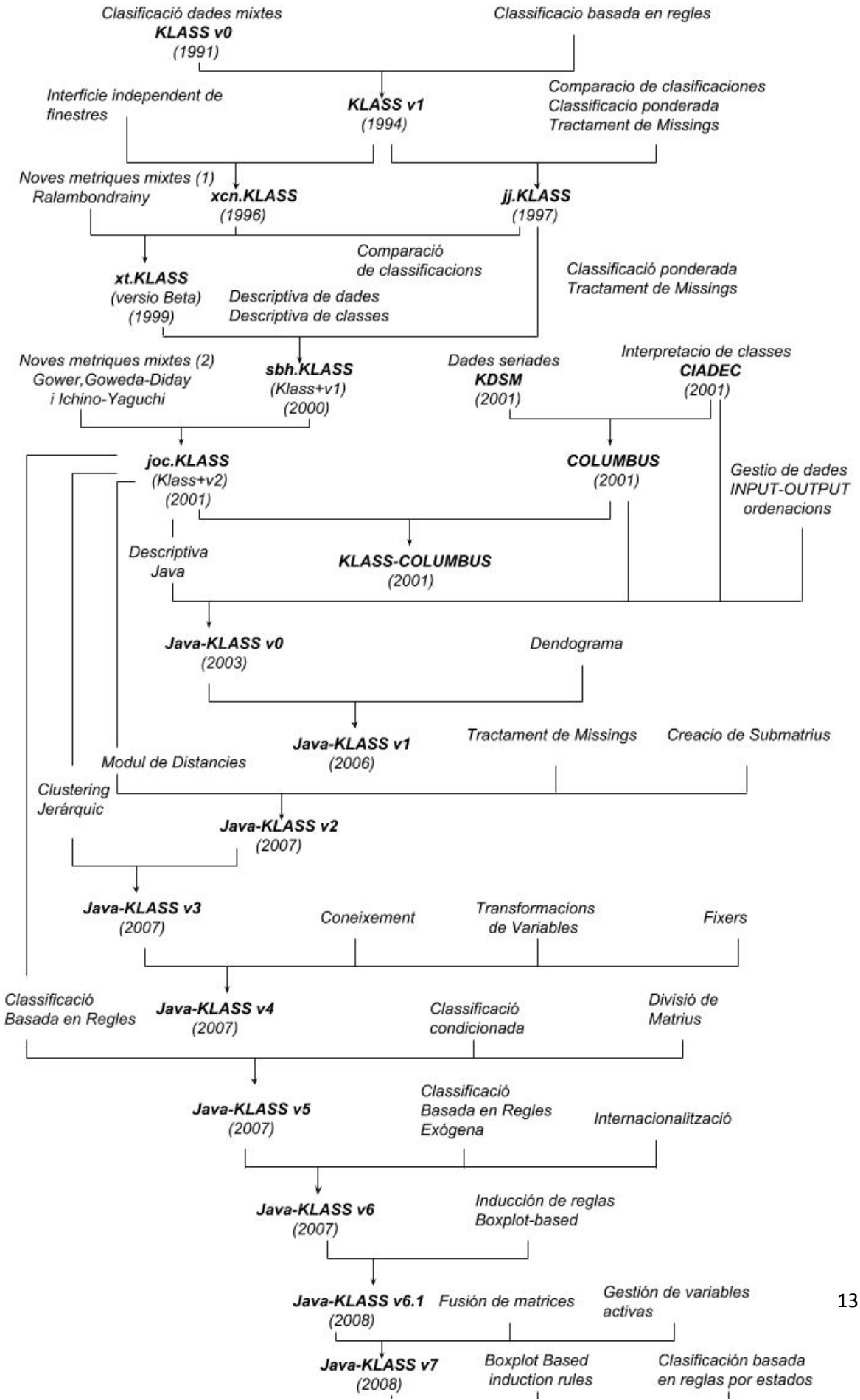
- automatic clustering including hierarchical clustering, with some extensions like clustering based on rules or ontology based clustering, DBSCAN, CURE and OPTICS clusterings[MS 2014];
- heterogeneous system management including data, knowledge visualization among others.

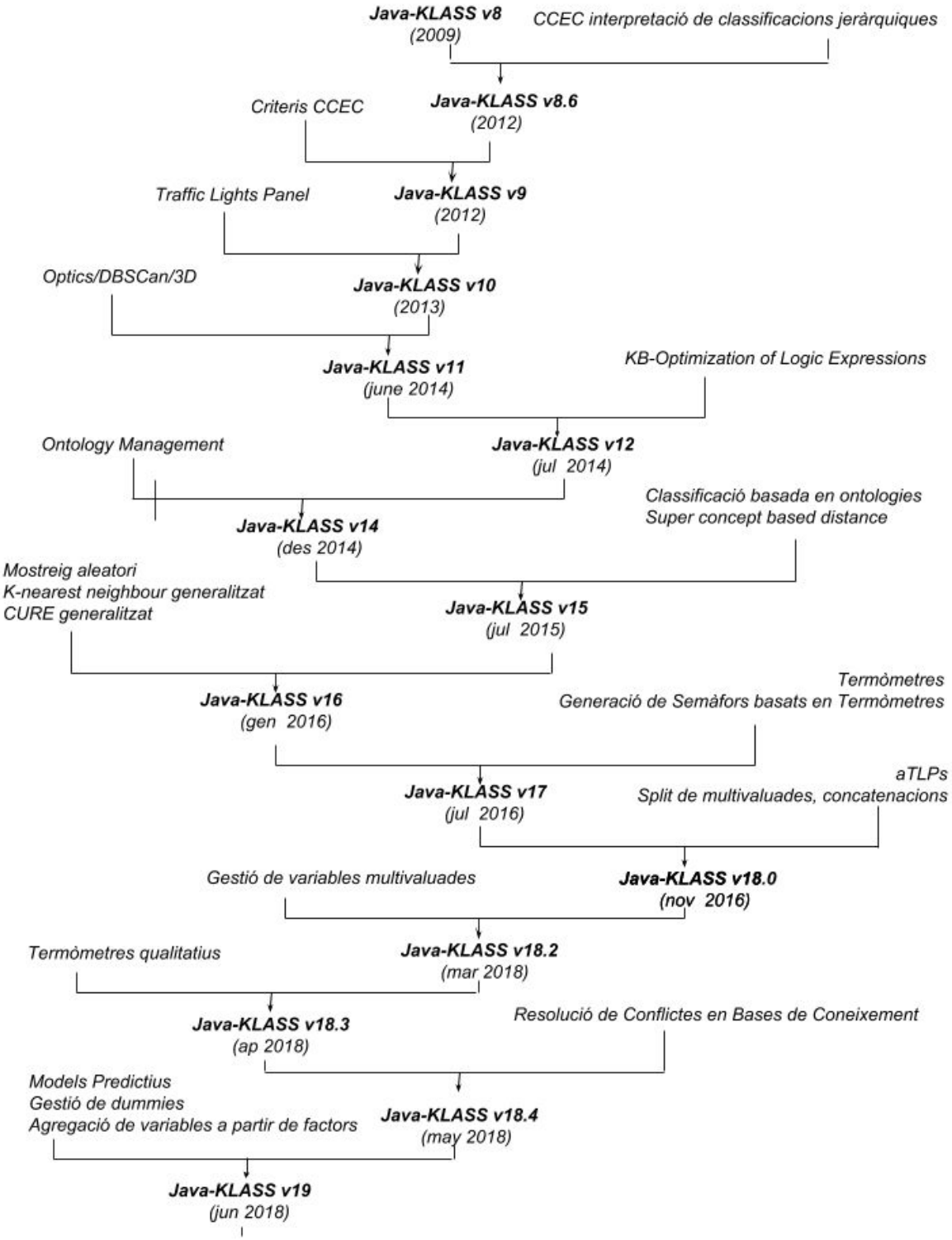
2.2 Chronology of KLASS

- Feb. 1991 KLASS v0. Thesis Karina Gibert. \ KLASS. Study a system of aid treatment Statistical large databases. "Classify data matrices heterogeneous mixed with distance.
- Nov. 1994 KLASS v1. Thesis Karina Gibert. " The use of information Symbolic in the automation of the treatment statistical domains little Structured . " It is an extension of KLASS v0. It incorporates the rule-based classification . [Gibert 94]
- Jul. 1996 KLASS v1.1. PFC Xavier Castillejo. It incorporates an interface window KLASS.v1 independent with a system that facilitates the use of KLASS from SUN and from PC to users who do not know Lisp and UNIX. We will call xcn.KLASS to the kernel Lisp of this new version and xcn.i in interface C.
- Oct. 1997 jj.KLASS . PFC Juan José Márquez and Juan Carlos Martín. Add new options to the KLASS.v1 version data processing However , the possibility of working with objects and implements a weighted non-parametric test comparison rankings.
- Set 1999 KLASS v1.2. PFC Xavier Tubau (version β). Incorporate to the version xcn.KLASS module jj.KLASS comparison rankings, metrics and Ralambondrainy mixed formulation prepared three more for subsequent implementation. We will call xt.KLASS to the kernel Lisp of this new version and xt.i to the associated C interface .
- 1999-2000 KLASS + v1. PFC Sílvia Bayonne . Final fusion of the version xt.KLASS with jj.KLASS . It also incorporates a non - descriptive data module , also of the classes resulting refocusing towards a purpose KLASS moregeneral and less specialist. We will call sbh.KLASS to the kernel Lisp of this new version and sbh.i to the associated C interface .
- 2000-2002 KLASS + v2. PFC Josep Oliveras. Add to sbh.KLASS metrics mixed pending (Gower, Gowda- Diday i Ichino-Yaguchi). We will call juego.KLASS to this new version .
- 2000-2003 jr.KLASS +. Doctoral Thesis Jorge Rodas. Integrates KLASS + v.2 and Columbus, which is introduced more go ahead ...
- 2000-2003 Research Anna Salvador and Fernando Vázquez. Development of CIADEC, which is introduced more forward.

- 2002-2003 Java-KLASS v0. PFC Ma del Mar Colillas. Java version of the module of descriptive analysis and integration with CIADEC and Columbus.
- 2003-2005 Java-KLASS v0.22. Collaboration with Mar Colillas. Extension of the descriptive analysis and introduction Tool Data Management (definition of sorts in the reports, possibility of several arrays of objects in the system simultaneously , change of active matrix).
- 2005-2006 Java-KLASS v1.0. Collaboration with Mar Colillas. Includes reading and viewing of dendograms isolated and the generation of partitions from them.
- 2006-2007 Java-KLASS v2.0. PFC Jose Ignacio Mateos. Expanding Java-KLASS a module for calculating distances to different type of data matrices, including combining information qualitative and quantitative , treatment of missings and creation of submatrices .
- 2006-2007 Java-KLASS v3.0. PFC Roberto Tuda . It includes a classification module automatic by methods hierarchical , using all distances implemented in v2.0 and an option to study aggregations of objects step by step. It creates the option to select the working directory by default. You add the option to add it and save it objects with weight
- 2006-2007 Java-KLASS v4.0. PFC Laia Riera Guerra. Introduction , management and evaluation of Bases of Knowledge . Expanding Java-KLASS a module that allows transformation of variables discretions , recodings and calculations arithmetic with numerical variables . Finally, this version includes the definition of submatrius via logical filters on Objects , the editing of meta information of the variables of the array , elimination of variables and import of files in format . dat standard.
- 2007 Java-KLASS v5.0. PFC Andreu Raya. Includes conditional classification, classification based on rules and functionalities division of database management classification tree (or dendograms) associated with different data matrices.
- 2007 Java-KLASS v6.0 Work of investigation Tutored Alejandro García. Classification based on exogenous rules . Intentionalization and location of three languages (Catalan , English and Spanish). Fusion of dies .
- 2008 Java-KLASS v6.4. Work Master Alfons Bosch Sansa, Patricia García Giménez, Ismael Sayyad Hernando. Boxplot-based discretization, Boxplot-based Induction rules.
- 2008 Doctoral thesis Alejandra Perez. Characterization by conditioning successive method that induces Automatically go to concepts associated to the classes discovered.
- 2008 Doctoral Thesis Gustavo Rodríguez. Classification based on rules by states that allows systems analysis Dynamic.

- 2008: Java-KLASS v7.0 : TrT Alejandro García Rudolph . Fusion of dies and management of active variables.
- 2009: Java-KLASS v8 .: Master Thesis by Ester Lozano. Criteria Best Local Concept and no close world assumption of CCEC. PT Alejandro García Rudolph . Classification based on rules by states .
- 2010: Java-KLASS v8.1: Practice SISPD. Narcissus Maragall . Boxplot Based Induction Rules.
- 2012: Java-KLASS v8.6: Practice SISPD. Pau CCEC methodology.
- 2012: Java-KLASS v9: Practice SISPD. Marco Villegas. CCEC Criteria .
- 2013 Java-KLASS v10: Practice SISPD. Emili Boronat . Traffic Light Panel.
- 2014: Java-KLASS v11: Final Project of Engineering Career FIB computing Sheila dock. DBSCAN, OPTICS, 3D Visualization.
- 2014: Java-KLASS v12: Practice SISPD. Jonathan Moreno. Optimization of expressions logics.
- 2015 Java-KLASSv15: Practices IKPDI + SISPD Sergio Santamaria and Daniel Gibert et alt practices Management of ONTOLOGIES , distances semantic. Classification based on ontologies .
- 2016 Java-KLASSv16: TFG Valerio Di Matteo (U. La Sapienza , Rome, Italy). Sampling and Scalability: Generation of random variables, sampling on random data matrix, Nearest k- Neighbor , CURE.
- June 2016 Java-KLASSv17: TM David Canudes + practice IKPD des2015: Management thermometers + automation TLPs
- 2016 Nov Java-KLASSv18 IKPD practices : Implementation of TLPs scored. First infrastructures to manage multivalued variables (deployment and concatenation)
- 2 018 Mar Java-KLASSv18 .2 : TM Luis Daniel Pérez Tamayo: Management of multivalued variables and consolidation previous work
- 2018 may Java-KLASSv18.3: TM Carlos Luis Jordan and TM Johnny Avila : Thermometers Qualitative and Connected with traffic lights , and reorganization of all the methods of induction of concepts.
- 2018 June Java-KLASSv19 : Lavanya Mandadapu : adding predictive models, dummies management and variables aggregation through factoring.


Clasificació dades mixtes
KLASS v0
(1991)
Classificacio basada en regles
Interficie independent de finestres
KLASS v1
(1994)
Comparacio de clasificaciones
Classificacio ponderada
Tractament de Missings
Noves metriques mixtes (1)
Ralambondrainy
xcn.KLASS
(1996)
jj.KLASS
(1997)
Comparació de classificacions
xt.KLASS
(versio Beta)
(1999)
Descriptiva de dades
Descriptiva de classes
Classificació ponderada
Tractament de Missings
Noves metriques mixtes (2)
Gower,Goweda-Diday
i Ichino-Yaguchi
sbh.KLASS
(Klass+v1)
(2000)
Dades seriades
KDSM
(2001)
Interpretacio de classes
CIADEC
(2001)
joc.KLASS
(Klass+v2)
(2001)
COLUMBUS
(2001)
Gestio de dades
INPUT-OUTPUT
ordenacions
Descriptiva
Java
KLASS-COLUMBUS
(2001)
Java-KLASS v0
(2003)
Dendograma
Java-KLASS v1
(2006)
Tractament de Missings
Creacio de Submatrius
Modul de Distancies
Clustering
Jerárquic
Java-KLASS v2
(2007)
Java-KLASS v3
(2007)
Coneixement
Transformacions
de Variables
Fixers
Classificació
Basada en Regles
Java-KLASS v4
(2007)
Classificació
condicionada
Divisió de
Matrius
Java-KLASS v5
(2007)
Classificació
Basada en Regles
Exógena
Internacionalització
Java-KLASS v6
(2007)
Inducción de reglas
Boxplot-based
Java-KLASS v6.1
(2008)
Fusión de matrices
Gestión de variables
activas
Java-KLASS v7
(2008)
Boxplot Based
induction rules
Clasificación basada
en reglas por estados


Java-KLASS v8
(2009)
CCEC interpretació de classificacions jeràrquiques
Java-KLASS v8.6
(2012)
Criteris CCEC
Java-KLASS v9
(2012)
Traffic Lights Panel
Java-KLASS v10
(2013)
Optics/DBSCan/3D
Java-KLASS v11
(june 2014)
KB-Optimization of Logic Expressions
Java-KLASS v12
(jul 2014)
Ontology Management
Java-KLASS v14
(des 2014)
Classificació basada en ontologies
Super concept based distance
Java-KLASS v15
(jul 2015)
Mostreig aleatori
K-nearest neighbour generalitzat
CURE generalitzat
Java-KLASS v16
(gen 2016)
Termòmetres
Generació de Semàfors basats en Termòmetres
Java-KLASS v17
(jul 2016)
aTLPs
Split de multivaluades, concatenacions
Java-KLASS v18.0
(nov 2016)
Gestió de variables multivaluades
Java-KLASS v18.2
(mar 2018)
Termòmetres qualitatius
Java-KLASS v18.3
(ap 2018)
Resolució de Conflictes en Bases de Coneixement
Java-KLASS v18.4
(may 2018)
Models Predictius
Gestió de dummies
Agregació de variables a partir de factors
Java-KLASS v19
(jun 2018)

2.3 KLASS structure

In order to understand how the functionalities that we implemented work and interact with each other and with the rest of the software, it is important to know the structure of the JavaKlass.

It is a java application that was implemented as a “layered” structure, in order to separate the graphic interface from the actual methods that perform the computation. It is constituted by three main packages:

- **jKlass.iu** : this package forms the “interface” layer of the system. Here, we can find the classes that draw graphic panels with which users interact with the Klass. They can only perform graphic interactions, store and call methods that perform actual computation and then eventually producing results.
- **jKlass.nucli** : this is the “core” layer of the system. this contains the methods which actually perform the actions desired by the user. There are two core classes inside this layer which are:
- **GestorKlass** : Klass can work with multiple matrices at the same time and eventually with multiple users. This class manage to store information about the actual matrix in use and consequently calling the methods corresponding to the correct instance of data to process.
- **GestorMatriu** : this is the class that represents a single data matrix and holds the methods that perform concrete actions on it, managing its properties and objects and providing information about it when needed.Normally, these methods are called from GestorKlass. GestorMatriu contains the data matrix and all the context to work with it, like associated metadata, knowledge bases, ontologies, dendograms, traffic light panels or models.
- **jKlass.util** : this package contains classes which are used for the configuration management of the options and parameters of the functionalities and for the communication with the operating system.

Chapter 3

Methods and concepts of interest

In this chapter we will introduce all the methods used and the concepts needed to understand the development of this project.

3.1 Dummy Variables

Dummy variables are the set of binary variables that are generated from a qualitative variable. Number of dummies that are generated for a qualitative variable depends upon the number of modalities of the reference qualitative variable. Dummy variables plays an important role in statistical modelling because this is the way in which qualitative variables enter into the classical statistical models like linear regression and logistic regression.

If **X** is a data matrix containing $X_1, X_2 .....X_k$ variables, in which some of them are qualitative.

$$X = (X_1, X_2 .....X_k)$$

Let us take a qualitative variable X_k. The variable X_k takes values in its domain D_k that contains $\{m_1, m_2 .....m_s\}$ values.

$$D_k = \{m_1, m_2 .....m_s\}$$

When we generate dummies for this qualitative variable X_k, we will get new variables which are $X_{m_1}, X_{m_2}, ...X_{m_s}$.

These generated variables are binary variables having values 1 and 0.

$$X_{m_s} = \begin{cases} 1, if\ X = m_s \\ 0, otherwise \end{cases} \qquad \forall\ s = (1 : S)$$

In this project we have convenience to generate dummies as numerical variables or qualitative variables (logical or nominal).

3.2 Multiple Linear Regression

Main focus of regression methods is to find the relationship between a dependent variable and independent variables(s) and formulate the linear equation between dependent and independent variable(s). If the regression analysis takes

place between one dependent numerical variable and one independent numerical variable, then it is called as linear simple regression. Linear regression model with one dependent variable and more than one independent variables is called as multiple linear regression. Multiple linear regression analysis is formulated as in the following [Baron 2014]:

$$y_i = \beta_0 + \beta_1 x_{1_i} + ... + \beta_K x_{K_i} + \varepsilon$$

Y= Dependent variable

X_i= Independent variables

β_i*= Coefficients*

ε= Error

Estimates of the model parameters

- SSR is the "regression sum of squares" and quantifies how far the estimated sloped regression line, y_i, is from the horizontal "no relationship line," the sample mean or y [STAT 501].

$$SSR = \sum_{i=1}^{n} (\widehat{y_i} - \overline{y})^2$$

- SSE is the "error sum of squares" and quantifies how much the data points, y_i, vary around the estimated regression line, $\widehat{y_i}$.

$$SSE = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2$$

- SSTO is the "total sum of squares" and quantifies how much the data points, y_i, vary around the global mean, $\overline{y}$.

$$SSTO = \sum_{i=1}^{n} (y_i - \overline{y_i})^2$$

- $MSE = \frac{SSE}{n-k}$,estimates σ^2, the variance of the errors. In the formula, *n* stands for the sample size, *k* is number of β coefficients in the model (including the intercept).
- Residual standard error, also named as the regression standard error , which estimates σ is

$$S = \sqrt{MSE}$$

- Residual term is calculated as

$$e_i = y_i - \widehat{y_i}$$

- Coefficient of Determination, R-Squared can be calculated using

$$R^2 = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO}$$

Qualitative variables enter into multiple linear regression as dummy variables. They are generated using Generate Dummy functionality that we have developed in KLASS.

3.3 Graphical Residual Analysis in linear regression

Residual analysis plays an important role in validation of regression models fitted from data. In this thesis we will assume a simplified residual analysis containing:

- Histograms of residuals, predicted values and numerical explanatory variables
- Boxplots of residuals, predicted values and numerical explanatory variables.
- Statistical summaries of residuals, predicted values and numerical explanatory variables.
- Bar charts of qualitative explanatory variables
- Frequency tables of qualitative explanatory variables.
- Residual plots:
    - Residuals on vertical axis and predicted value on horizontal axis.
    - Residuals on vertical axis and independent numerical variables on horizontal axis.

If points in the residual plots are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data. Otherwise, missing regressors or heteroskedasticity or non linearities occur and model is not valid.

3.4 Logistic Regression

The central mathematical concept that underlies logistic regression is *logit*. logit is the natural logarithm of the odds ratio. If we plot the simplest case of linear regression for one independent variable and one dichotomous dependent variable, we will get two parallel lines each corresponding to a value of the dichotomous dependent variable [vanderbilt] . (See figure 1).

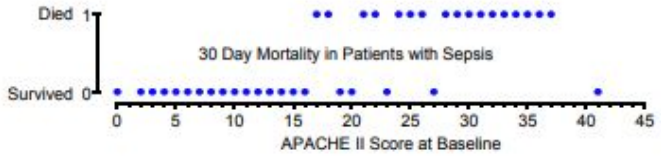


Figure 1 : Plot between one independent variable and dichotomous dependent variable. Source [Vanderbilt].

Because the two parallel lines are difficult to be modelled with an ordinary least squares regression equation, one may create categories of the predictor and compute the mean of the outcome variable for the respective categories. This plot results in a S-shaped or Sigmoidal curve (see figure 2). This curve cannot be described by a linear equation for two reasons

1. The extremes do not follow a linear trend.
2. The errors are neither normally distributed nor constant across the entire range of the data and follows the logistic function.

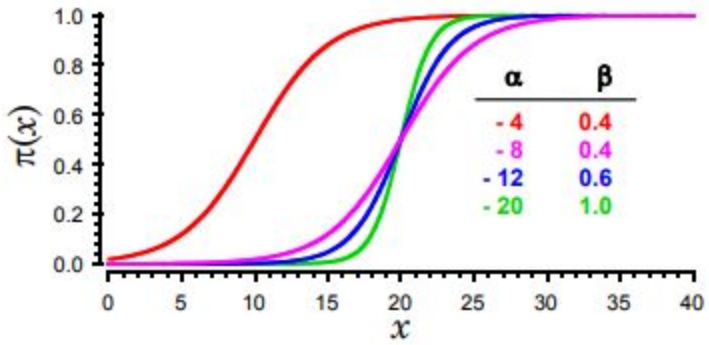


Figure 2: Sinusoidal Curves. Source [Vanderbilt]

Logistic regression solves these problems by applying the logit transformation to the dependent variable [Montgomery 2012]. Logistic regression equation having multiple independent variables can be given by the following equation

$$logit(y_i) = ln\left(\frac{P_i}{1-P_i}\right) = \beta_0 + \beta_1 x_{1_i} + ... + \beta_K x_{K_i}$$

y=Dependent variable

x_i =*Independent variables*

P= $P(Y|X_1, X_2, ...X_K)$, being Y the outcome of interest and X_k are the independent variables.

$$P_i = \frac{e^{\beta_0+\beta_1 x_{1_i}+....+\beta_K x_{K_i}}}{1+e^{\beta_0+\beta_1 x_{1_i}+....+\beta_K x_{K_i}}}$$

Odds and Odds Ratio

There are algebraically equivalent ways to write the logistic regression model.

$$\frac{P_i}{1-P_i} = exp(\beta_0 + \beta_1 x_{1_i} + ... + \beta_K x_{K_i}) \quad (1)$$

The above equation describes the odds of being in the current category of interest. By definition, odds for an event is P / 1-P , where P is the probability of the event occurrence. From expression (1) it is easy to see that the logistic regression can be reduced to a multiple linear regression problem as well.

$$log\left(\frac{P_i}{1-P_i}\right) = \beta_0 + \beta_1 x_{1_i} + ... + \beta_K x_{K_i}$$

However in this project we used external library that is fitting logistic regression model directly. (See section 3.8).

3.5 Residual Deviance

Deviance is the measure of goodness of fit of the logistic regression model.Higher numbers always indicates the bad fit.The residual deviance shows how well the response variable is predicted in the model with the inclusion of intercept and the independent variables.

$$D = 2\sum_{i=1}^{n}\left(y_i ln\left(\frac{y_i}{\hat{P}_i}\right) + (1-y_i) ln\left(\frac{(1-y_i)}{(1-\hat{P}_i)}\right)\right)$$

P =*predicted probability*

y =*dichotomous dependent variable*

3.6 Embedded Binary Logistic regression

As mentioned in section 1.1, for PAG we need to have a combination of combined binary logistic regressions that can be nested according to the natural ordering of the classes. EBLR proposes this [Gibert 2013].

EBLR is a formal method to recognise the class of a new object, provided that a previous set of clusters has been found (P) and a total order (R) exists over the clusters. From a statistical point of view, the class can be considered as a qualitative variable to be fitted. The EBLR consists of nesting several binary logistic regressions to find the better prediction of a new case.

- Defining a family of embedded sets of objects, according to the R ordering being l the total set of objects to be analyzed.

$$\boldsymbol{l}_0 = \boldsymbol{l}, \qquad \boldsymbol{l}_k = \boldsymbol{l} \setminus \bigcup_{(c \prec C_k)} c, \quad k = 1{:}\,\xi.$$

- For the particular case of 4 classes, this is defining 4 datasets.

$$l_0 = l \ , \quad l_1 = l \setminus l_0$$

$$l_2 = l \setminus (l_0 \cup l_1), \quad l_3 = l \setminus (l_0 \cup l_1 \cup l_2), \quad l_4 = l \setminus (l_0 \cup l_1 \cup l_2 \cup l_3)$$

being,

$$l_1 = \{ i \,\varepsilon\, l\} \quad , \quad l_2 = \{ i \,\varepsilon\, l \ : \ i \notin C_1\} \quad , \quad l_3 = \{ i \,\varepsilon\, l \ : \ i \in C_3 \cup C_4\}$$

$$l_4 = \{ i \,\varepsilon\, l \ : \ i \in C_4\}$$

- Define Π_k = p(i $\in$ $C_k \mid l_k$), k = 1: ξ − 1. This estimates k_1, a binary logistic regression model that can recognize C_1 class. For k_2 it estimates the probability of being in C_2 when it is known that object is not in C_1 and so on.
- Find $\widehat{P_k}$(i) to estimate Π_k; apply binary logistic regression to the set of objects l_k and the total set of variables X. The final model, $\widehat{p_k}$(i) will only involve x_p and will enable detection of C_k when the larger classes (upon R) have been previously discarded:

$$\hat{p}_k(i) = \frac{e^{\hat{\beta}_0 + \sum_{l=1}^{K} \hat{\beta}_l X_l}}{\left(1 + e^{\hat{\beta}_0 + \sum_{l=1}^{K} \hat{\beta}_l X_l}\right)}.$$

- Given a threshold ε = [0, 1],

$$\hat{Q}(i) = \begin{cases} C_k & \text{such that } \forall k' < k, \hat{p}_{k'}(i) < \varepsilon \,\&\, \hat{p}_k(i) \geq \varepsilon \\ C_\xi, & \text{otherwise} \end{cases}$$

EBLR_estimate calculates the logistic models for all classes and EBLR_use uses the results of EBLR_estimate to predict a class for new objects.

```
proc EBLR_estimate(I, P, X, ξ):
     Sample=I,  k = 1
     While (k < ξ − 1) do
           p̂_k = logisRegr(C_k, I_k, X)
           Sample = (Sample \C_k)
           k = k + 1
     endWhile
     return (p̂_1, p̂_2, ..., p̂_ξ), X_P
endproc
```

Table 1: Algorithm of the EBLR_estimate method. Source [gibert 2013].

```
proc EBLR_use(i, P, X_P(i), ξ, ε, (p̂_1, ..., p̂_ξ)):
     p=0, k=1
     While (p < ε) or (k < ξ) do
           p= p̂_k(i) = e^{β̂_k X_P(i)} / (1 + e^{β̂_k X_P(i)})
           if p ≥ ε then Class= C_k
                    else, k = k + 1
           endIf
     endWhile
     return Class
endproc
```

Table 2: Algorithm of EBLR_use method. Source [Gibert 2013].

3.7 Profile Assessment Grid

Profile assessment grid (PAG) represents a simple post-processing of the EBLR method, that enables non expert end-users to use the model. It is a graphical transformation of EBLR_use that aims to provide a frame to visualize the most likely profile of a new object in a unitary cube, using a minimal set of relevant variables, and currently works for a case of 4 ordered profiles.

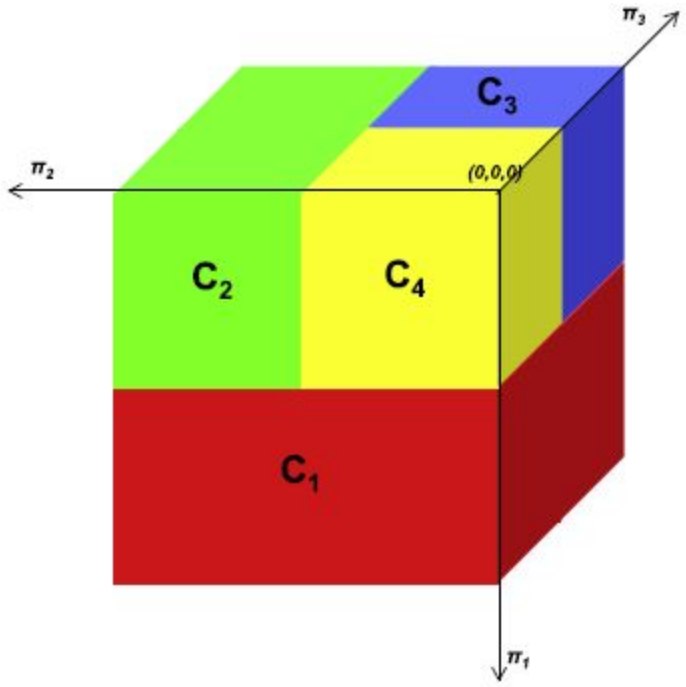


Figure 3: Profile's Assessment grid ($\varepsilon = 0.5$)

The PAG is built by associating three orthogonal axes with the three EBLR logistic models that are required to discriminate among the four profiles (the Π_2 being represented on the X axis, the Π_3 on the Z axis, the Π_1 on the Y axis) and placing the origin in the top-right-front corner of the cube, for visual purposes. The threshold ε is used to label, and conveniently colour regions of the cube assigned to each profile, according to the conditionals used in the EBLR process

3.8 Development Tools

Java under Eclipse, KLASS and JSAT are the only tools that are going to be used. Eclipse is flexible and serves as a good platform for java projects.

The main of our project is to develop and incorporate new modelling functionalities into KLASS. We will develop several modules corresponding to the different new functionalities mentioned in the section 1.1. Part of this will be developed from scratch. To develop our modelling modules we want to use already existing machine learning library, so that we can develop this project in less time by reducing the risk of implementation errors. There exist relatively few general purpose machine learning libraries for use. We have identified some of the libraries in which multiple linear regression and logistic regression are implemented in java such as *JSAT[Raff 17] ,Weka[waikato 97] ,Java-ML[Abeel 2009]* . As we know, KLASS is a software that is entirely developed in java. If we incorporate a library that has been written in java, it will be easy to use the algorithms. The library that suits our project in a best way is JSAT. when a larger dataset is taken, JSAT analyzed the data faster than Weka.

Java Statistical Analysis Tool (JSAT) [Raff 2017] is a machine learning library written in java 6 and has no dependencies, making it easy to integrate into KLASS without conflicts. JSAT includes numerous algorithms for classification and regression, clustering, feature selection and engineering. The part that we are interested is *Regression.* Before incorporating into KLASS, we checked *muliplelinearregression* and *logisticregression* modules separately, we used *iris.dat* as input and benchmarked with R. Results are similar. Then, *we* included a folder called *regressionjsat* in *jklass.nucli* in order to use *MultipleLinearRegression* and *Logisticregression* present in *jsat.regression*. There is a method called *train* in both *MultipleLinearRegression* and *LogisticRegression* which computes coefficients given a dataset as input. This dataset must contain dependent variable at last and only works with numerical values. They have used least squares solution to calculate β values or regression coefficients. From JSAT only the coefficients are taken, predictions, residuals, model parameters and deviance are calculate using the formulas given in the previous section.

We will use LaTeX too for writing reports. KLASS produces LaTeX outputs sometimes. So LaTeX is also indirectly linked to the project development. LaTeX is more comfortable if we have a lot of mathematical equations in our report.

Chapter 4

Methodology

This chapter will cover the detailed explanation of methodology that has been used to make this project complete and working well.

- We have started with the implementation of generate dummy functionality into KLASS. This functionality helps us to generate dummy variables for a qualitative variable. Through dummies, qualitative variables enter into the Multiple linear regression and Logistic regression modules. This is the main reason behind the development of generate dummy functionality.
- At first, we have generated only the numerical dummies, which of course needed for our multiple linear regression analysis according to the library specifications. Later on we have added qualitative dummies both logical and nominal into our functionality. We have also added an option that controls the prefix names of new variables, so that 1 coming from different dummies can be properly distinguished in common representations. This will be useful for further multivariate functions which are not tackled in this project.
- Once Generate dummy functionality is working well, we moved to the development of multiple linear regression module.
- As mentioned in the section 3.8, we incorporated JSAT into KLASS. JSAT only works with numerical variables which can also be binary. So, at first our multiple linear regression module contains only numerical variables and we benchmarked with R. It includes the creation of a new java object containing the regression model.
- Later on we used dummies to incorporate qualitative variables and benchmarked with R.
- Once our module is ready we then created a LaTeX document that shows the results obtained from the multiple linear regression analysis of a particular MLR model.
- We then moved to the implementation of Logistic regression into KLASS. At first, we created a qualitative binary variable containing 1's and 0's to use as response variable in logistic regression by using the recode option from KLASS. As like multiple linear regression implementation, we worked with the numerical variables at first, benchmarked the results with R and included qualitative variables trough dummies.
- Then, we created a java method which converts a qualitative binary response variable (WorkingDay, NonWorkingDay) data to 1's and 0's, taking first modality as 1 and second modality as 0, because JSAT only takes

binary response variable which contains 1's and 0's. So connection between KLASS and JSAT is completed.

- Once our Logistic regression module is working well, we created a LaTeX document that shows the results obtained from the logistic regression analysis of a particular LogReg model.
- We have used logistic regression functionality present in JSAT to implement our logistic regression module. But, as mentioned in section 3.4, odds ratio, log odds ratio are important in the generation of coefficients of logistic regression model. To showcase the odds, ratio, odds ratio and log odds ratio we decided to develop Qualitative Aggregation functionality into KLASS.
- At first, we have developed single factor aggregation, in which there will be one aggregation factor and one binary selected variable. We have created a module for generating a new data matrix in KLASS which contains aggregated explanatory variable in first column and log odds ratio, odds, size etc., as rest of the columns. In a second step we enlarged this functionality by including multiple aggregation factors and implemented qualitative aggregation functionality into KLASS.
- Main aim of our thesis is to develop a visualization tool (PAG) mentioned in section 3.7. As the connection between KLASS and JSAT was not immediate and required internal data structure and formats transforms. During the development of Qualitative aggregation module, we realized that we need more time to implement PAG present in [Gibert 2013]. So, we decided to implement a pseudo PAG using the functionalities in KLASS which more or less resembles like PAG present in [Gibert 2013].
- To implement pseudo-PAG we need predicted values from three logistic regression models generated from a dataset. We then developed model evaluation and model management functionalities into KLASS that helps us to generate the predicted variables, needed for pseudo PAG, whenever a model is given as input.
- We then decided to store our MLR and LogReg models outside the current session to get model fits persistence. This way we can import the models and can evaluate whenever needed including new case datasets. For exporting and importing MLR and LogReg models we have developed Input Output functionalities into KLASS namely Export MLR, Export LogReg, Import MLR and Import LogReg.
- Finally we used 3D functionality present in KLASS under descriptive menu to view our pseudo-PAG.

The next chapter provide technical detail on the development of all the tasks described above.

Chapter 5

New Data Management Functionalities

5.1 Generate Dummy

5.1.1 Functionality description.

The generate dummy functionality allows the user to generate dummy variables for the selected qualitative variable. User can select which type of variables to generate;

- Numerical
- Logical
- nominal

User can determine whether the new column names should have the prefix or not, using *prefix* selection. By default descriptive analysis will be generated for the newly created dummy variables. User can also control this selection.

5.1.2 Interface design and specification

The functionality is available on *Data* drop-down menu, and its panel is generated by the class *PanelGenerateDummy.java*, in the *jKlass.iu* package.

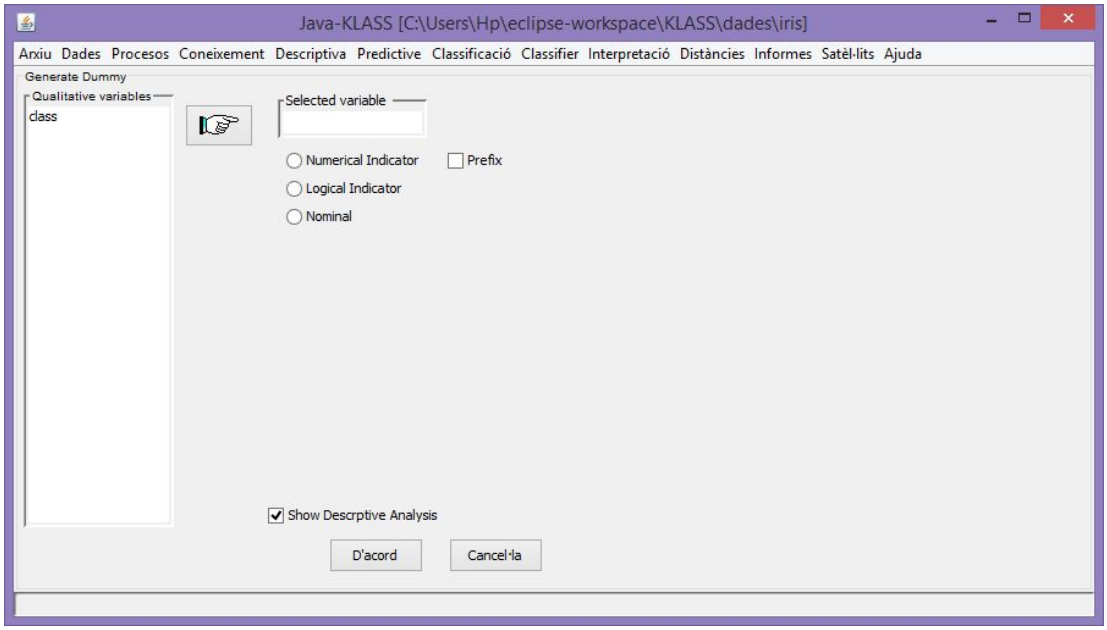
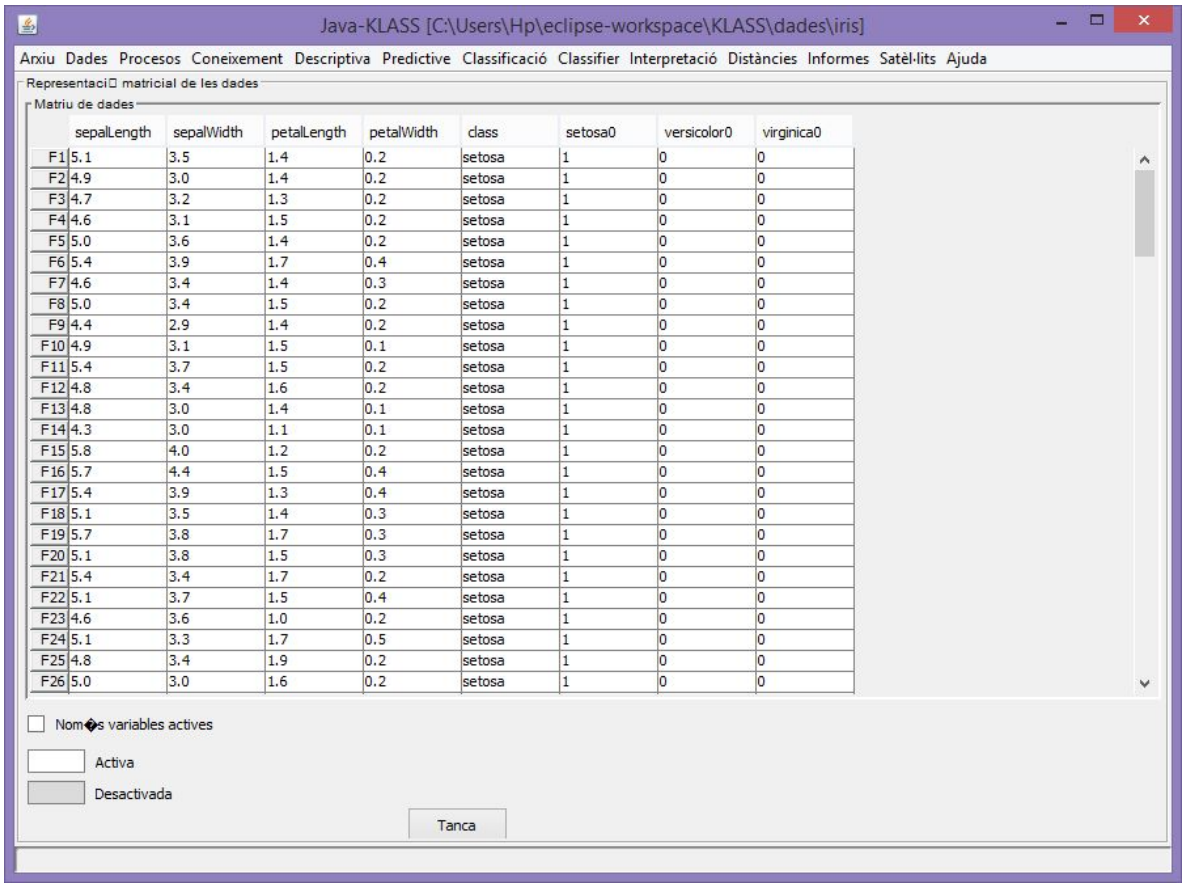


Figure 4: View of Generate dummy Panel.

Next we will present each element of the panel explaining its task.

- **Qualitative Variables**: This shows all the qualitative variables present in the current matrix.
- **Selected variable**: User can select one of the variables present in the qualitative variables List. That variable will appear in this field. At a time user can select only one variable.
- **Numerical Indicator**: On clicking this radio button, numerical dummies will be generated. These dummies are added as new columns to the data matrix. In figure 5 setosa0, virginica0 and versicolor0 columns are numerical columns that are generated on selecting *classs* variable present in *iris.dat* and choosing *Numerical Indicator* radio button.

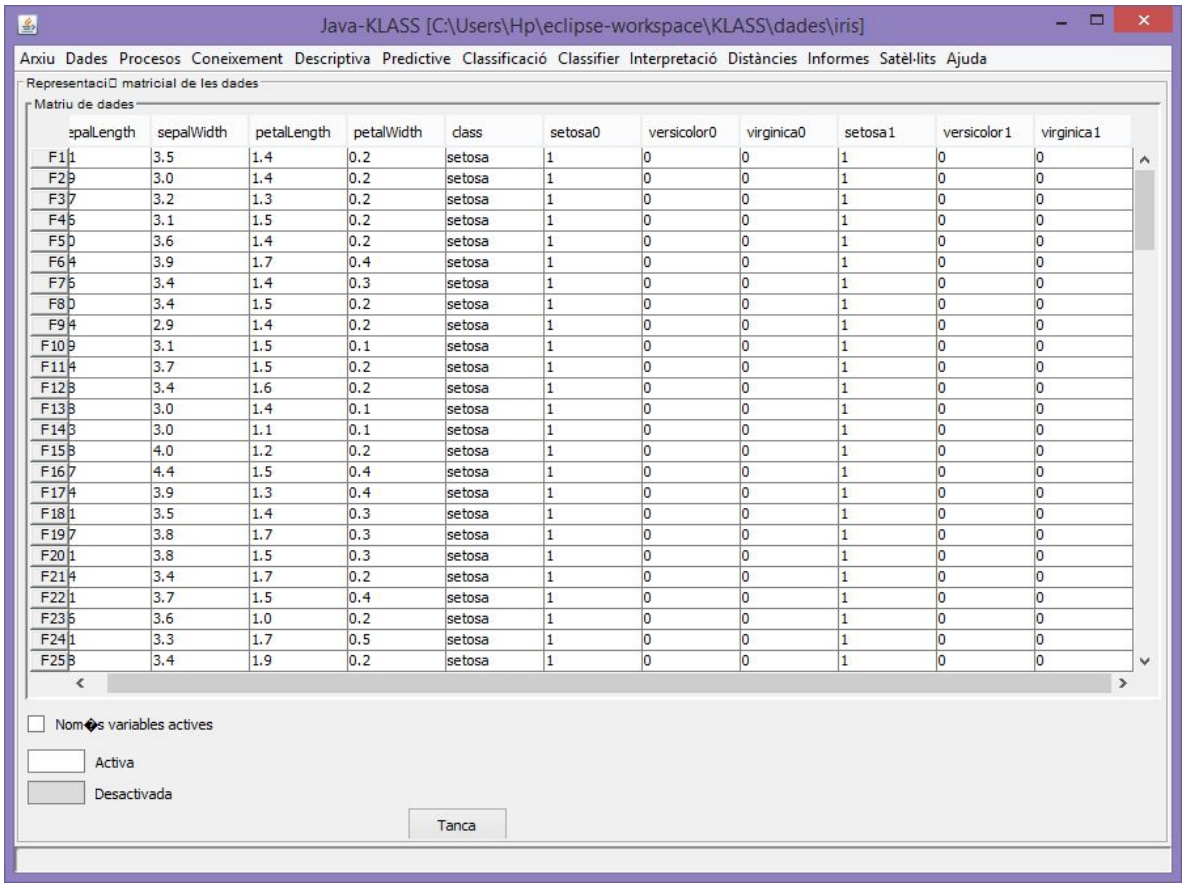


	sepalLength	sepalWidth	petalLength	petalWidth	class	setosa0	versicolor0	virginica0
F1	5.1	3.5	1.4	0.2	setosa	1	0	0
F2	4.9	3.0	1.4	0.2	setosa	1	0	0
F3	4.7	3.2	1.3	0.2	setosa	1	0	0
F4	4.6	3.1	1.5	0.2	setosa	1	0	0
F5	5.0	3.6	1.4	0.2	setosa	1	0	0
F6	5.4	3.9	1.7	0.4	setosa	1	0	0
F7	4.6	3.4	1.4	0.3	setosa	1	0	0
F8	5.0	3.4	1.5	0.2	setosa	1	0	0
F9	4.4	2.9	1.4	0.2	setosa	1	0	0
F10	4.9	3.1	1.5	0.1	setosa	1	0	0
F11	5.4	3.7	1.5	0.2	setosa	1	0	0
F12	4.8	3.4	1.6	0.2	setosa	1	0	0
F13	4.8	3.0	1.4	0.1	setosa	1	0	0
F14	4.3	3.0	1.1	0.1	setosa	1	0	0
F15	5.8	4.0	1.2	0.2	setosa	1	0	0
F16	5.7	4.4	1.5	0.4	setosa	1	0	0
F17	5.4	3.9	1.3	0.4	setosa	1	0	0
F18	5.1	3.5	1.4	0.3	setosa	1	0	0
F19	5.7	3.8	1.7	0.3	setosa	1	0	0
F20	5.1	3.8	1.5	0.3	setosa	1	0	0
F21	5.4	3.4	1.7	0.2	setosa	1	0	0
F22	5.1	3.7	1.5	0.4	setosa	1	0	0
F23	4.6	3.6	1.0	0.2	setosa	1	0	0
F24	5.1	3.3	1.7	0.5	setosa	1	0	0
F25	4.8	3.4	1.9	0.2	setosa	1	0	0
F26	5.0	3.0	1.6	0.2	setosa	1	0	0

Figure 5: Numerical Dummy Generation

- **Logical Indicator** : On selecting this radio button, qualitative dummies will be generated. The generated new columns will have the 1 or 0 values.In figure 6 setosa1, virginica1, versicolor1 are the new qualitative columns generated on selecting *class* as selected variable from *iris.dat* and choosing

Logical Indicator selection. The transformation that KLASS manages for these variables is that they are qualitative. So all operations for qualitative variables are suitable (like table of frequencies or bar charts).



	epalLength	sepalWidth	petalLength	petalWidth	class	setosa0	versicolor0	virginica0	setosa1	versicolor1	virginica1
F1	1	3.5	1.4	0.2	setosa	1	0	0	1	0	0
F2	9	3.0	1.4	0.2	setosa	1	0	0	1	0	0
F3	7	3.2	1.3	0.2	setosa	1	0	0	1	0	0
F4	6	3.1	1.5	0.2	setosa	1	0	0	1	0	0
F5	0	3.6	1.4	0.2	setosa	1	0	0	1	0	0
F6	4	3.9	1.7	0.4	setosa	1	0	0	1	0	0
F7	6	3.4	1.4	0.3	setosa	1	0	0	1	0	0
F8	0	3.4	1.5	0.2	setosa	1	0	0	1	0	0
F9	4	2.9	1.4	0.2	setosa	1	0	0	1	0	0
F10	9	3.1	1.5	0.1	setosa	1	0	0	1	0	0
F11	4	3.7	1.5	0.2	setosa	1	0	0	1	0	0
F12	8	3.4	1.6	0.2	setosa	1	0	0	1	0	0
F13	8	3.0	1.4	0.1	setosa	1	0	0	1	0	0
F14	3	3.0	1.1	0.1	setosa	1	0	0	1	0	0
F15	8	4.0	1.2	0.2	setosa	1	0	0	1	0	0
F16	7	4.4	1.5	0.4	setosa	1	0	0	1	0	0
F17	4	3.9	1.3	0.4	setosa	1	0	0	1	0	0
F18	1	3.5	1.4	0.3	setosa	1	0	0	1	0	0
F19	7	3.8	1.7	0.3	setosa	1	0	0	1	0	0
F20	1	3.8	1.5	0.3	setosa	1	0	0	1	0	0
F21	4	3.4	1.7	0.2	setosa	1	0	0	1	0	0
F22	1	3.7	1.5	0.4	setosa	1	0	0	1	0	0
F23	6	3.6	1.0	0.2	setosa	1	0	0	1	0	0
F24	1	3.3	1.7	0.5	setosa	1	0	0	1	0	0
F25	8	3.4	1.9	0.2	setosa	1	0	0	1	0	0

Figure 6: Qualitative dummy generation on selecting Logical Indicator.

- **Nominal** : On selecting this radio button, qualitative dummies will be generated having values with modality 1 or 0. Nominal and Logical Indicator both generates qualitative dummy variables. But, the main difference is the values inside the generated columns. In the figure 7, setosa2, virginica2 and versicolor2 are the newly generated dummies having class as the selected variable from iris.dat and choosing nominal selection.
- **Prefix** : This can be selected only if one of the Numerical Indicator, Logical Indicator or Nominal is selected. Otherwise panel will produce an error message. On selecting prefix checkbox column names of the newly generated variables will be class.setosa3 , class.virginica3 and class.versicolor3. This columns are generated on selecting class as selected variable from iris.dat and selecting numerical radio button with prefix checkbox. see figure 7

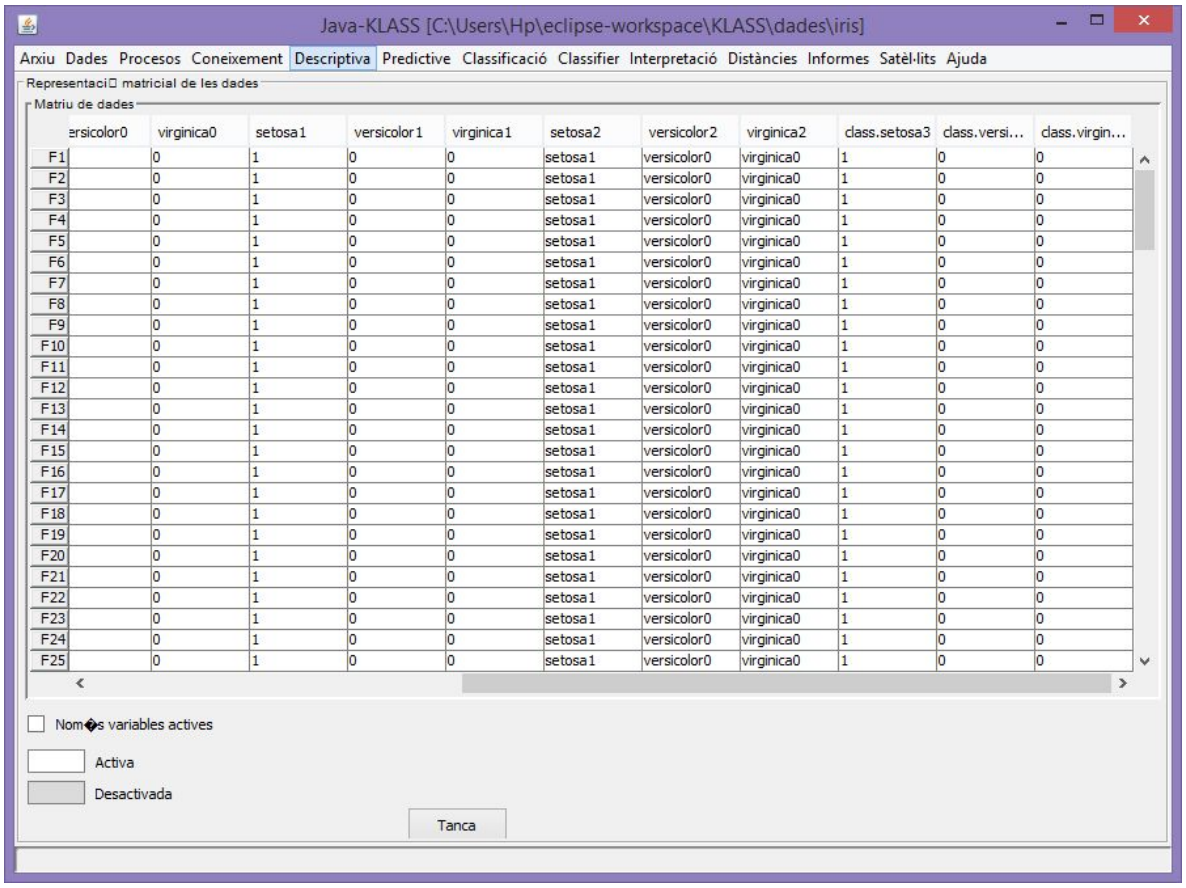
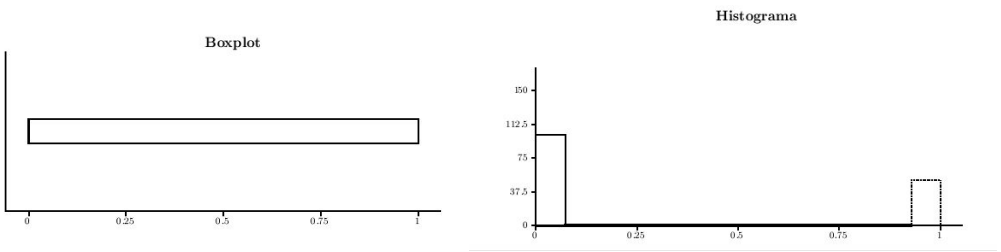


Figure 7: Nominal dummies and prefix generation example

- **Show descriptive analysis** : On selecting this checkbox, we can see the descriptive analysis of the selected variable and the newly generated dummies.If Numerical Indicator is selected then descriptive analysis shows histogram, Boxplot, Estadictis sumaries, Resum en 5 numeros of the newly generated variables. See figure 8.If either Logical Indicator or Nominal is selected then descriptive analysis shows us Diagrama de barres and taula de frequencies of the newly generated variables.See figure 9.

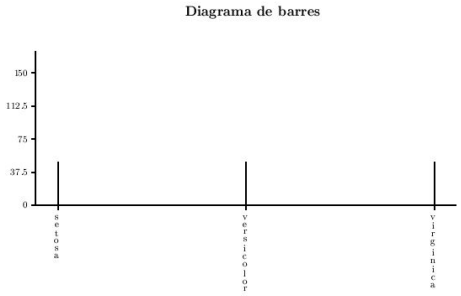


Resum en 5 números

min	Q1	Me	Q3	max
0	0	0	1	1

Estadístics sumaris	
Nombre d'objectes	150
Nombre de dades mancants	0
Nombre d'observacions útils	150
Mitjana	0.3333
Mediana	0
Primer quartil (Q1)	0
Tercer quartil (Q3)	1
Mínim	0
Màxim	1
Quasi-desviació típica	0.473
Coeficient de variació	1.4142

Figure 8 :Descriptive analysis of Numerical dummy variable *virginica0*



Taula de freqüències				
Modalitats	Freq. absol.	Freq. acum.	Freq. relat.	Freq. rel. acum.
setosa	50	50	0.3333	0.3333
versicolor	50	100	0.3333	0.6667
virginica	50	150	0.3333	1
dades mancants	0	N = 150	0	

Figure 9: Descriptive analysis of Qualitative dummy variable *virginica3*

5.1.3 Implementation

In order to develop the Generate Dummy functionality we have added following classes to the KLASS packages that are mentioned in 2.3.

- *PanelGenerateDummy:* This class calls the panel to generate dummy variables, given a qualitative variable.This class is present in jKlass.iu package.
- *generateDummyMLR*: Computes the modalities and calls *generateDummyVar* present in *MatriuDades*. This is present in jKlass.nucli.
- *generateDummyVar:* Calls *generateDummyVarBinary* present in class *Operacio*.This is present in jKlass.nucli.

- *generateDummyVarBinary*: Based on the option that user choose, this generates new columns in *Matriu de dades*.This is present in jKlass.nucli.

If D'accord button is clicked without selecting selected variable then proper error message will be generated.

5.2 Qualitative Aggregation

5.2.1 Functionality description

The Qualitative Aggregation functionality allows the user to generate new Data Matrix, having aggregation factor as a first column and rest of the columns can be selected from one of the options that appear in the panel, which are :

- Counts
- Size
- Ratio
- Odds
- Odds Ratio
- Log Odds Ratio

5.2.2 Interface design and specification

The functionality is available on *Data* drop-down menu, and its panel is generated by the class *PanelBDaggregation.java* , in the *jKlass.iu* package.

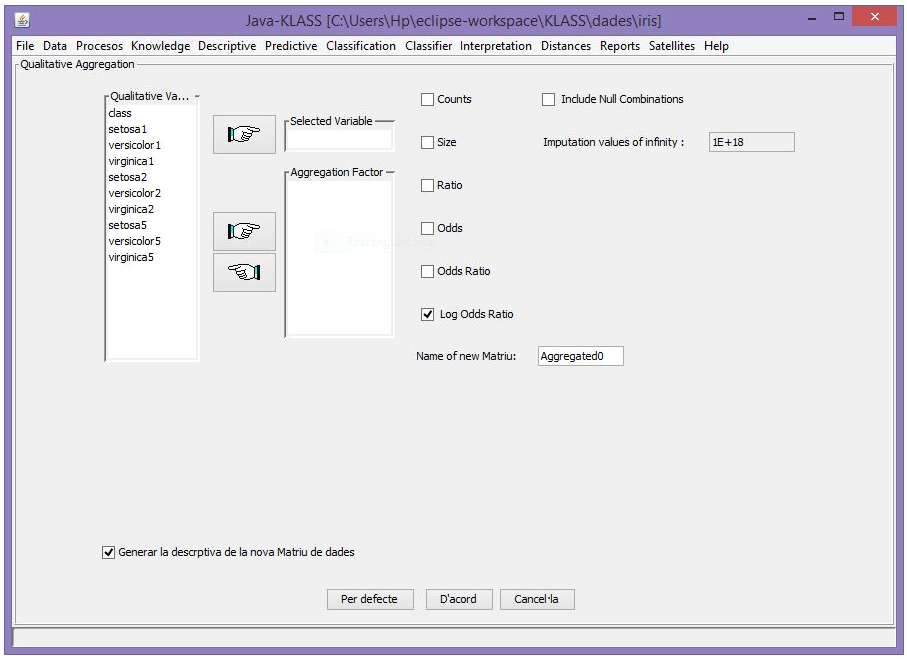
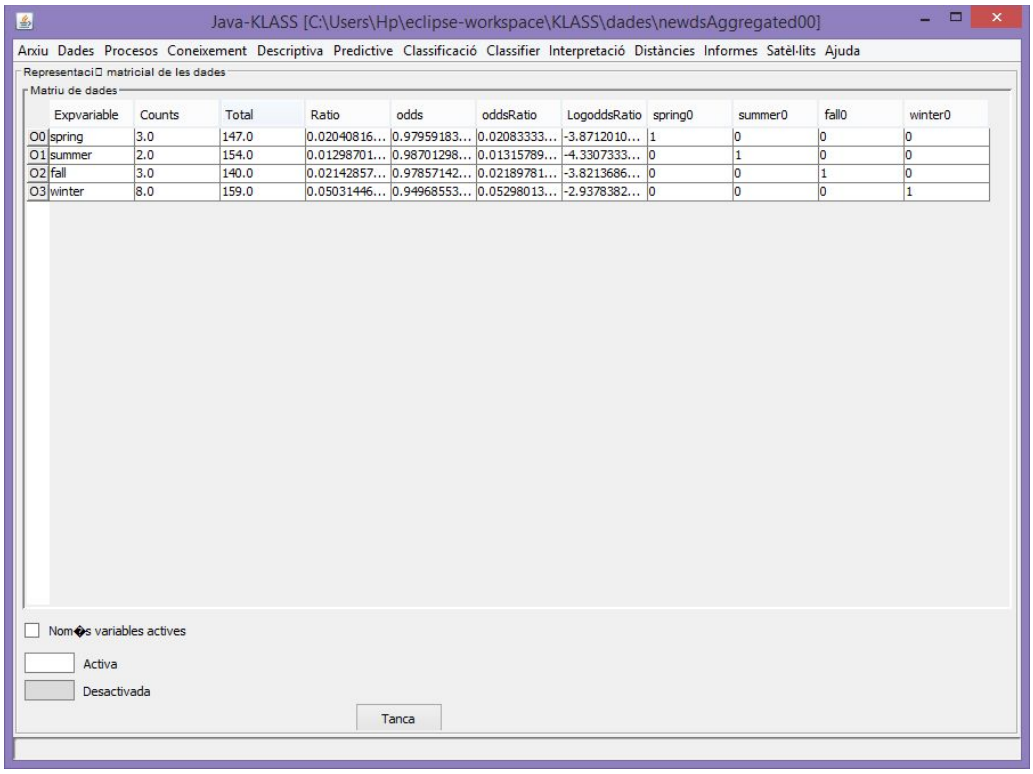


Figure 10: View of Qualitative Aggregation panel

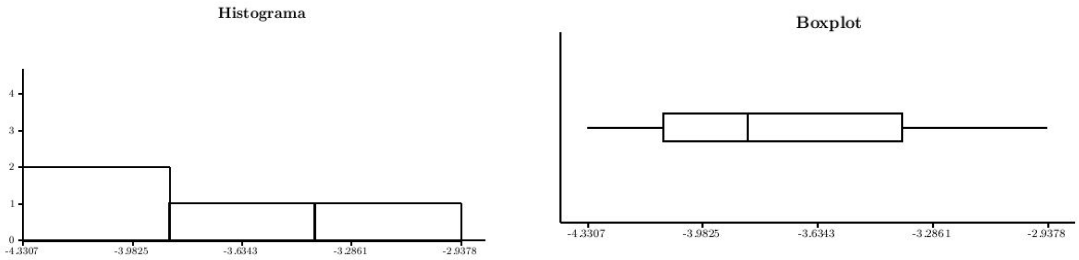
Next we will present each element of the panel explaining its task.

- **Qualitative Variables**: This field contains all the qualitative variables present in the current data matrix. To perform qualitative aggregation user must choose from this list.
- **Selected variable** : This must contain only two modalities.This is considered as the response variable that helps us to generate the counts.
- **Aggregation Factor**: We can select one or more qualitative variables to be our explanatory variables. In case if we select one qualitative variable then we will generate dummies and perform Table of frequencies present in bivariant java object to get the frequencies which are considered as counts in our case. If we select more than one qualitative variable then they will be concatenated to get a single explanatory variable and Taula de frequencies will be performed.
- **Name of new Matriu** : User can name the generated new matrix in the textfield provided. Default name will be the Aggregated0. where 0 is the count for that particular session.
- **Include Null combinations**: Sometimes there will be 0 counts, which leads to infinity value for logs odds ratio. To avoid such conditions by default we are removing null count rows. But, if user wants to include those rows then he can do that by enabling this checkbox.
- **Imputation values of Infinity** : If user enables Include null combinations then the positive and negative infinity values in log odds ratio are replaced by 1E+18. User can change this imputation value, once include null combinations is enabled.
- **Descriptive Analysis of new Matrix:** By checking this user gets a latex report which has univariant description of all the columns present in the new matrix. Figure 12 shows univariant analysis that are generated for log odds ratio.
- **Per defecte** : On clicking this we can select all options which are counts,size,ratio,odds,oddsratio,logoddsratio and Include null combinations.log odds ratio and show descriptive analysis are enabled by default.
- **D'accord** : On clicking this button a new matrix will generate in Matriu de dades. First column of the matrix will always be single explanatory variable that is taken from the aggregation factor., and rest of the columns to be generated are selected from the panel. Figure 11 shows the new matrix with all options being selected.
- **Cancel-la :** This clears the window.



	Expvariable	Counts	Total	Ratio	odds	oddsRatio	LogoddsRatio	spring0	summer0	fall0	winter0
O0	spring	3.0	147.0	0.02040816...	0.97959183...	0.02083333...	-3.8712010...	1	0	0	0
O1	summer	2.0	154.0	0.01298701...	0.98701298...	0.01315789...	-4.3307333...	0	1	0	0
O2	fall	3.0	140.0	0.02142857...	0.97857142...	0.02189781...	-3.8213686...	0	0	1	0
O3	winter	8.0	159.0	0.05031446...	0.94968553...	0.05298013...	-2.9378382...	0	0	0	1

Figure 11 :View of new generated matrix



Estadístics sumaris	
Nombre d'objectes	4
Nombre de dades mancants	0
Nombre d'observacions útils	4
Mitjana	-3.7403
Mediana	-3.8463
Primer quartil (Q1)	-4.101
Tercer quartil (Q3)	-3.3796
Mínim	-4.3307
Màxim	-2.9378
Quasi-desviació típica	0.582
Coeficient de variació	-0.1348

Figure 12: Univariant descriptive analysis of Log odds ratio.

5.2.3 Implementation

In order to develop the Qualitative Aggregation functionality we have added following classes to the KLASS packages that are mentioned in 2.3.

- *PanelBDaggregation.java* : This class calls the panel to perform qualitative aggregation.This class is present in jKlass.iu package.
- *generateBDaggregation*: this method is present in GestorMatriu and called by the generateBDaggregation method present in GestorKlass. This returns List of Lists which contains name of the explanatory variable with the respective counts value. This counts values are taken from the bivariant object present in KLASS when we pass aggregated factor and selected variable.
- *createMatriuLogistic :* This method call the GestorMatriu constructor with the name of new matix and list of lists, which are returned from generateBDaggregation. This method is present in GestorKlass.
- *GestorMatriu*: This helps to generate new objects and properties that are needed to generate new matrix.
- *afegirPropietatLogisti*c : returns propietats for the new matrix and adds to the propietats of the new GestorMatriu.
- *afegirObjecteLogistic* : returns objectes for the new matrix and adds to the objectes of the new GestorMatriu.
- *computLogodds*: Given the imputation value, list of choices and counts, this method which is present in the GestorMatriu computes the values of selected choices and adds these values as new columns in the new matrix.
- *addColAggre:* Given the column name and the values. This method creates the new column.

Chapter 6

New Modelling Functionalities

We have created a new menu item inside KLASS called Predictive to include our modelling functionalities.We have created two modelling functionalities which are

- Multiple linear regression
- Logistic regression

6.1 Multiple Linear Regression

6.1.1 Functionality description

This functionality allows the user to perform multiple linear regression analysis over the current matrix. In order to perform this analysis user must select a response variable, which is numerical and one or more explanatory variable(s) which can be numerical or qualitative.

Once response and explanatory variables are selected, user can select what calculations should be performed on this variables.Following are the options that user can select

- Coefficients
- R-Squared
- S-Squared
- Predicted
- Residuals
- Fer dummies visibles
- Graphical residual Analysis

6.1.2 Interface design and specification

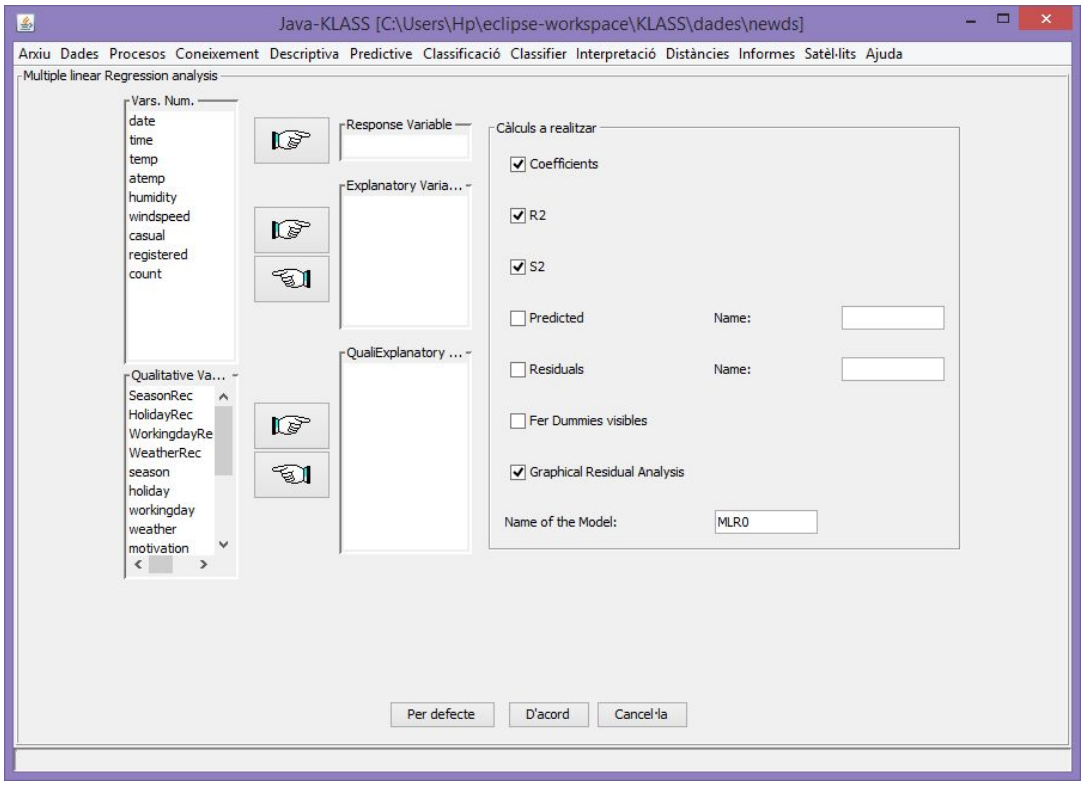


Figure 13 : View of Multiple linear regression panel

Now we will present each element of the panel explaining its task.

- **Vars.Num** : This field shows the list of numerical variables present in the current matrix.
- **Qualitative variables** : This field shows the list of qualitative variables present in the current matrix.
- **Response variable** : This must be a numerical variable and is selected from Vars.Num . This serves as the dependent variable.
- **Explanatory variables**: These are independent variables and can be both numerical or qualitative. This must be selected from Vars.Num list in the panel.
- **Quali Explanatory variables** : These are also independent variables and are qualitative. This must be selected from Qualitative variables list in the panel.
- **Coefficients** : Coefficients are calculated by selecting this. By default , this is made selected. We have used JSAT to get the coefficients.

- **Predicted**: Once the coefficients are generated, we can calculate the predicted values using the formulas given in section 3.2.
- **Residuals** : Once we have the predicted values, by subtracting original response variable values with predicted values, we can get the residuals.

When predicted and residual options are selected, then they will generate new columns in the data matrix with the names given in the corresponding name field present in the panel. By default, the names are pre and res followed by the count value.

When selected R-squared and S-Squared values are also computed using the formulas given in section 3.2.Remember, we have only taken coefficients from JSAT and rest of the calculations are implemented from scratch.

- **Fer Dummies visibles** : The multiple linear regression module of JSAT only works with numerical variables. So, If user selects Quali Explanatory variables, then numerical dummies are generated using Numerical Indicator in Generate Dummy panel. User can decide whether these intermediate dummy columns should be visible once he exit from the multiple linear regression module. If he want them to be visible then he must select this checkbox otherwise all the intermediate columns generated by this will be eliminated from the data matrix.

We have created a list of Multiple linear regression objects in GestorMatriu. For each multiple linear regression analysis an object will be created and added to this list. The object contains

- Response variable
- Explanatory variables
- Coefficients
- R-squared
- S-squared
- Modelterms; which is a hashmap with the coefficients as keys and explanatory variables as values.
- Name of the model

In the panel the value present in the name of the model field will be given in to the name of the model present in the MLR class.Using this name we can access particular multiple linear regression model. By default, name will be MLR followed by the counter. User can change this name if desired.

- **Graphical residual analysis** : On selecting this, Univariant analysis of predictions, residuals and bivariant analysis with the residuals on X-axis and all the explanatory variables and predicted on Y-axis will take

place.Figure 14 shows the graphical residual analysis of Bicingclass.dat dataset, we have taken response variable as temp and explanatory variables as windspeed,registered and season.

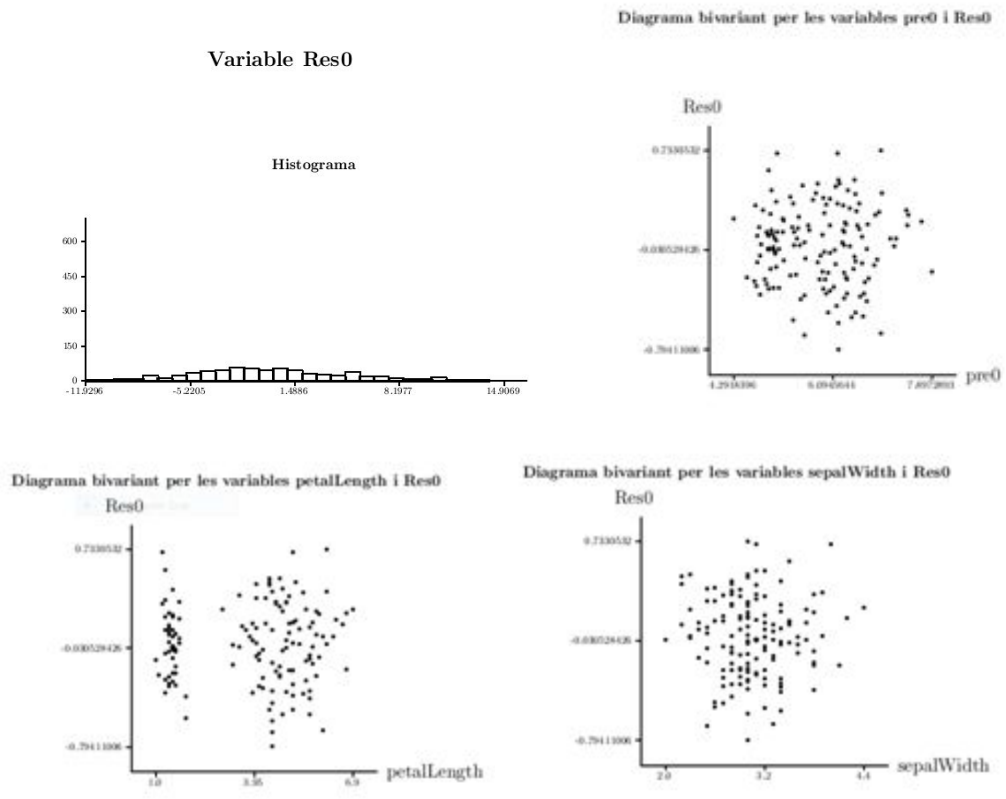


Figure 14: Example of Graphical residual analysis

After selecting the response variable, Explanatory variables and proper calculations, when we press D'accord it will produce a LaTex document which contains the results of the analysis. The structure of the output has been designed in the following way

- Information about input parameters
- Information about method of fitting
- Information about model estimated parser
- Information about coefficients
- Goodness of fit indicators
- If graphical residual analysis is selected then univariant and bivariant analysis will also be added to this LaTex file.

Anàlisi Regressiò Lineal Multiple

Response Variable : temp
Explanatory Variable :
windspeed
registered
SeasonRec
The minimum least squares estimation criterion is used. See results below

Regression Equation

$$temp = 10.9205 + 0.0352 \times windspeed + 0.0095 \times registered + 9.7393 \times summer + 15.608 \times fall$$
$$+3.5474 \times winter$$

Coefficients

Variable	Coefficient
Intercept	10.9205
windspeed	0.0352
registered	0.0095
summer	9.7393
fall	15.608
winter	3.5474

R Squared Value : -0.0811
S Squared Value : 21.623

Figure 15: Multiple linear regression results display format

We have created a simple parser that converts the data present in MLR java object into the the regression equation present in the figure 15.

Proper error messages will appear at the bottom of the panel in the following cases:

- no response variable is selected.
- no explanatory variable(s) is selected.
- calculations are selected without selecting response and explanatory variables.
- there exists a property with the specified name in name field.

6.1.3 Implementation

In order to develop the Multiple linear regression functionality we have added following classes to the KLASS packages that are mentioned in 2.3.

- *PanelPredictive:* This class calls the panel to perform multiple linear regression analysis.This class is present in jKlass.iu package.
- *ferMLRPredictive:* Given the list of choices, this method calls the JSAT train method and gets coefficients, calculates predicted , residuals , R-squared and S-squared values. This is present in GestorMatriu.
- *constructdata* : This method creates and returns new matriu dades with the structure useful to call multiplelinearregression in JSAT.This is present in MatriuDades class.
- addMLRpredictions: this method is used to create new column in the current matrix given the name of the column and the values.This is present in Operacio class.
- *generarLtxUnivarsPerVarsMLR:* This method writes the results of multiple linear regression analysis into the LaTex file.This is present in GeneradorTex class. In this method we have created the structure given in figure 15.
- getyvalue and getxvalue: these two are the methods that I have written in multiplelinearregression.java class present in JSAT.getyvalue returns the responsevariable values and getxvalue returns explanatory values.
- MLR : This is a constructor which is used to put the values that are generated into the MLR object.

MLR.java is present in jKlass.nucli package.

6.2 Logistic Regression

6.2.1 Functionality description

This functionality allows the user to perform logistic regression analysis over the current matrix. In order to perform this analysis user must select a response variable, which is qualitative and one or more explanatory variable(s) which can be numerical or qualitative.

Once response and explanatory variables are selected, user can select what calculations should be performed on this variables.Following are the options that user can select

- Coefficients
- Deviance
- Predicted
- Model Validation

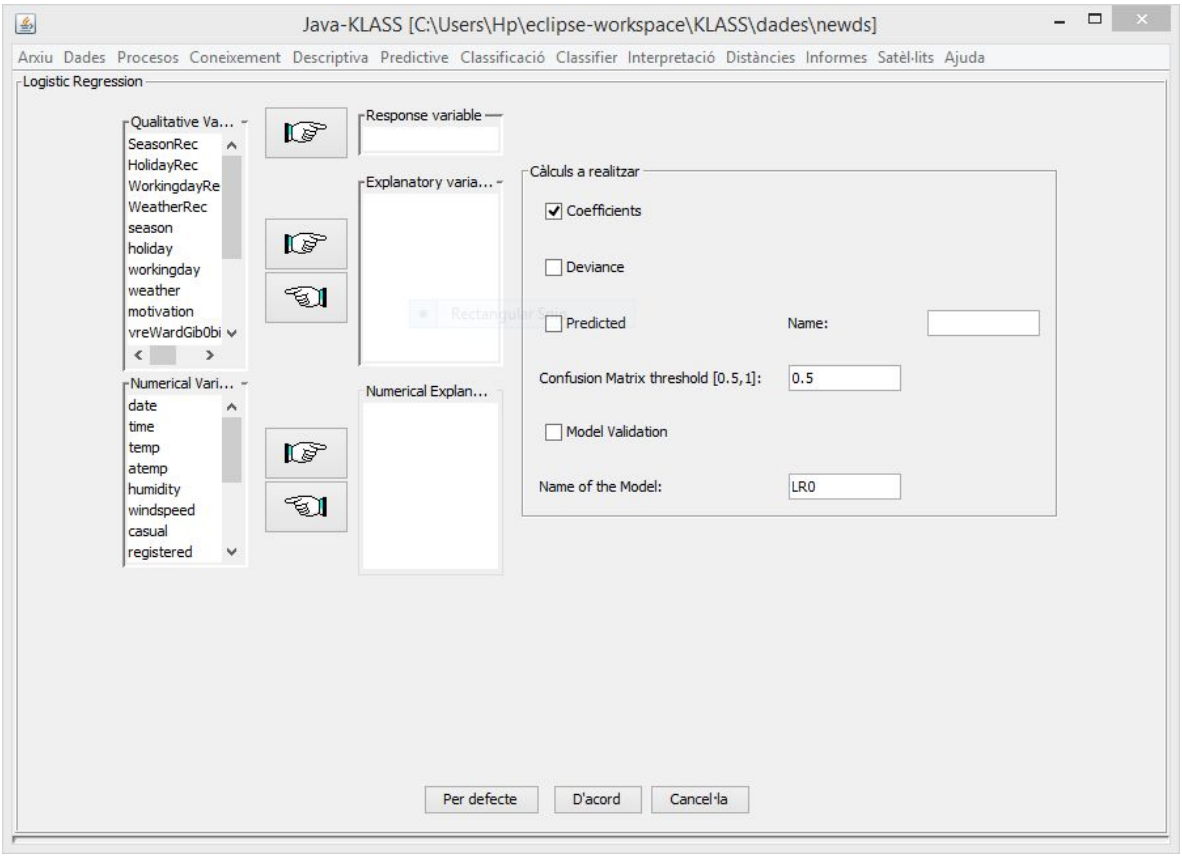


Figure 16 : View of Logistic regression panel

6.2.2 Interface design and specification

Now we will present each element of the panel explaining its task.

- ***Qualitative Variables****:* This contains all the qualitative variables present in the current matrix.
- **Numerical Variables** : This contains all the numerical variables present in the current matrix.
- ***Response variable*** : We can select a single qualitative binary variable as our response variable. This acts as the dependent variable in the analysis.
- **Explanatory variables**: These are independent variables and can be both numerical or qualitative. This must be selected from Qualitative variables list in the panel. If these are selected then dummies will be generated for them using Numerical Indicator in Generate Dummy panel and these new columns are considered as Numerical Explanatory variables instead of actual qualitative explanatory variables.
- **NumericalExp variables** : These are also independent variables and are numerical. This must be selected from Numerical variables list in the panel.
- **Coefficients** : Coefficients are calculated by selecting this. By default , this is made selected. We have used JSAT to get the coefficients.
- **Predicted:** Once the coefficients are generated, we can calculate the predicted values using the formulas given in section 2.3. These predicted values are displayed as a new column in the data matrix and the name of this column will be the name given in the corresponding name field present in the panel.By default the name is pre followed by the counter.User can change this name.
- ***Deviance* :** Once we have the predicted values, we can compute the deviance. we have implemented a function that calculates deviance using the formula given in section 2.3.
- ***Confusion Matrix Threshold[0.5,1]*** : We apply this threshold value on the predicted values in order to create a new binary column which contains 0's and 1's. we then create a confusion matrix between this new binary column and the response variable. This matrix helps us to understand the relation between the response variable and the predicted values. By default, threshold value is taken as 0.5. User can change this value, but the range should be in between 0.5 and 1.

- **Model Validation** : Once this is selected, then only the confusion matrix is displayed in the pdf that is generated by clicking D'accord button.See figure 17.

Taula de contingència per les variables CMpredi1 i Responsevariable0

Contingut de la casella: efectius absoluts

CMpredi1 \ Responsevariable0	0	1	útils	mancants
1	171	413	584	0
0	16	0	16	0
útils	187	413	600	
mancants	0	0		0

Figure 17: Confusion Matrix with threshold 0.5

We have created a list of logistic regression objects in GestorMatriu. For each logistic regression analysis, an object will be created and added to this list. The object contains

- Response variable
- Explanatory variables
- Coefficients
- Modelterms; which is a hashmap with the coefficients as keys and explanatory variables as values.
- Name of the model

In the panel the value present in the name of the model field will be given in to the name of the model present in the LR class.Using this name we can access particular logistic regression model. By default, name will be LR followed by the counter. User can change this name if desired.

Anàlisi de la Regressio Logistica

Response Variable : WorkingdayRec
Explanatory Variable :
temp
atemp
humidity
SeasonRec
HolidayRec
Deviance : 694.6073634097105

Regression Equation

$$log\Big(\frac{p(WorkingdayRec = true)}{p(WorkingdayRec = false)}\Big) = 0.4433 + 0.2654 \times temp - 0.2056 \times atemp + 0.0044 \times humidity$$
$$-0.6349 \times true - 0.7226 \times summer + 0.0606 \times fall$$
$$-5.4241 \times winter$$

Coefficients

Variable	Coefficient
Intercept	0.4433
temp	0.2654
atemp	-0.2056
humidity	0.0044
true	-0.6349
summer	-0.7226
fall	0.0606
winter	-5.4241

Figure 18: view of Logistic regression display format

Proper error messages will appear at the bottom of the panel in the following cases:

- no response variable is selected.
- no explanatory variable(s) is selected.

- calculations are selected without selecting response and explanatory variables.
- response variable is not binary.
- there exists a property with the specified name in name field.

6.2.3 Implementation

In order to develop the logistic regression functionality we have added following methods and classes to the KLASS packages that are mentioned in 2.3.

- *PanelLogistic*: This class calls the panel to perform logistic regression analysis.This class is present in jKlass.iu package.
- *generateLogistic* : This method is present in GestorMatriu. This will call train method present in JSAT logisticregression class and gets coefficients and based on the selection it calls *computeDeviance*.
- *computeLogpredictions:* This method calculates the logpredictions as a new column when predictions column is given as input.
- *computeDeviance:* This method calculates the deviance when logpredictions and response variable are given as input.
- *generarLtxLogistic:* This method writes the results of logistic regression analysis into the LaTex file.This is present in GeneradorTex class. In this method we have created the structure given in figure 18.
- getyvalue and getxvalue: these two are the methods that I have written in multiplelinearregression.java class present in JSAT.getyvalue returns the responsevariable values and getxvalue returns explanatory values.
- LR : This is a constructor which is used to put the values that are generated into the LR object.

LR.java is present in jKlass.nucli package.

Chapter 7

New Model Evaluation and Management Functionalities

7.1 Evaluate Multiple Linear Regression

7.1.1 Functionality description

This functionality allows the user to select one of the multiple linear regression models that are present in the current GestorMatriu. When a model is selected this panel generates a new column in data matrix which contains the predicted values that are computed from the Multiple linear regression equation.

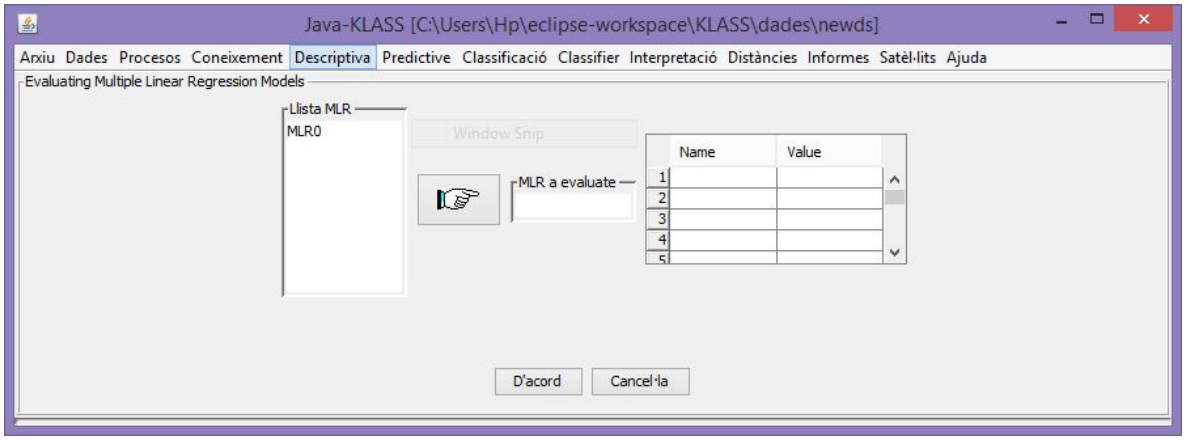


Figure 19: View of evaluate Multiple regression models panel

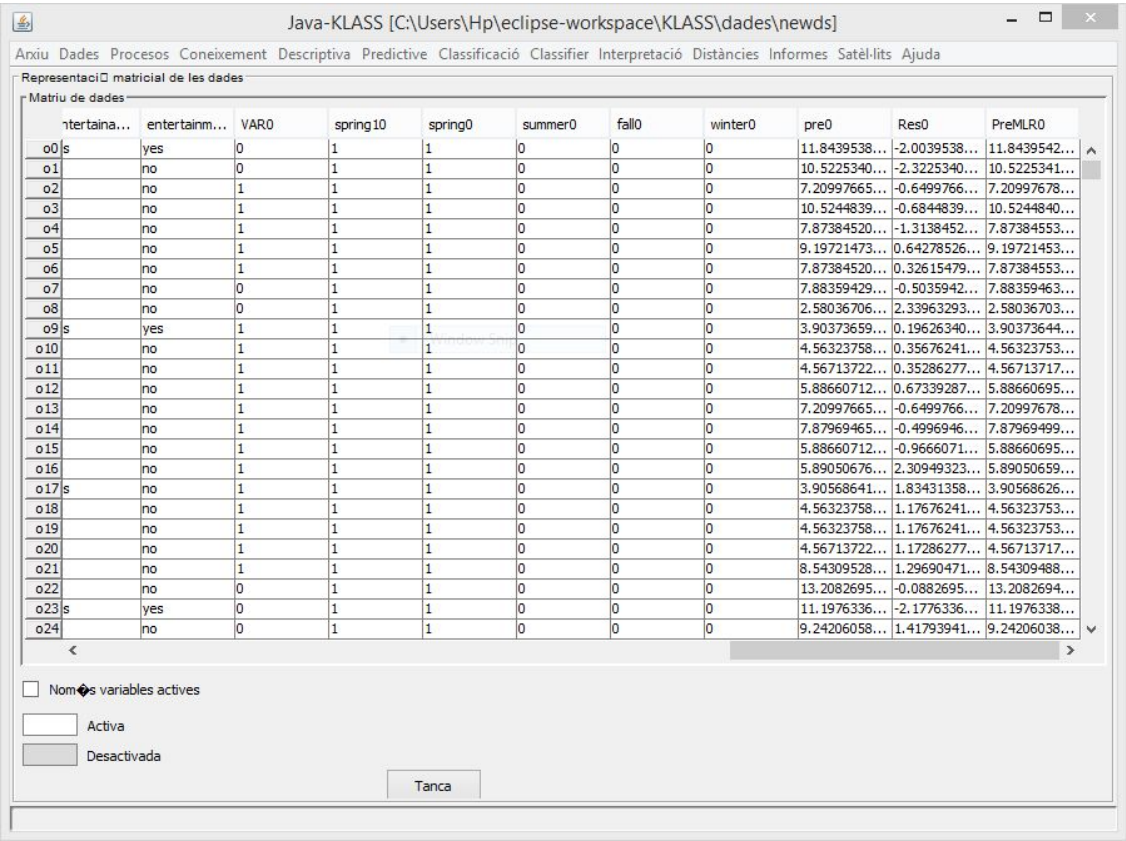
7.1.2 Interface design and specification

- **Llista MLR**: This field shows all the multiple linear regression models present in the current GestorMatriu.
- **MLR a evaluate**: The selected MLR object will be shown here and the response variable and the explanatory variables with associated coefficients are displayed in a table present on the right side.
- In figure PreMLR0 is the new column that is generated when we evaluate MLR0 object which contains temp as response variable and atemp,casual,seasonrec as explanatory variables.

7.1.3 Implementation

In order to develop the Evaluate Multiple linear regression functionality we have added following methods and classes to the KLASS packages that are mentioned in 2.3.

- *PanelEMLR:* This class calls the panel to evaluate multiple linear regression models.This class is present in jKlass.iu package.
- *computeEquationMLR:* This method which is present in GestorMatriu helps to creates new predicted column using the coefficients and explanatory variables present in the selected model.



	ntertaina...	entertainm...	VAR0	spring10	spring0	summer0	fall0	winter0	pre0	Res0	PreMLR0
o0	s	yes	0	1	1	0	0	0	11.8439538...	-2.0039538...	11.8439542...
o1		no	0	1	1	0	0	0	10.5225340...	-2.3225340...	10.5225341...
o2		no	1	1	1	0	0	0	7.20997665...	-0.6499766...	7.20997678...
o3		no	1	1	1	0	0	0	10.5244839...	-0.6844839...	10.5244840...
o4		no	1	1	1	0	0	0	7.87384520...	-1.3138452...	7.87384553...
o5		no	1	1	1	0	0	0	9.19721473...	0.64278526...	9.19721453...
o6		no	1	1	1	0	0	0	7.87384520...	0.32615479...	7.87384553...
o7		no	0	1	1	0	0	0	7.88359429...	-0.5035942...	7.88359463...
o8		no	0	1	1	0	0	0	2.58036706...	2.33963293...	2.58036703...
o9	s	yes	1	1	1	0	0	0	3.90373659...	0.19626340...	3.90373644...
o10		no	1	1	1	0	0	0	4.56323758...	0.35676241...	4.56323753...
o11		no	1	1	1	0	0	0	4.56713722...	0.35286277...	4.56713717...
o12		no	1	1	1	0	0	0	5.88660712...	0.67339287...	5.88660695...
o13		no	1	1	1	0	0	0	7.20997665...	-0.6499766...	7.20997678...
o14		no	1	1	1	0	0	0	7.87969465...	-0.4996946...	7.87969499...
o15		no	1	1	1	0	0	0	5.88660712...	-0.9666071...	5.88660695...
o16		no	1	1	1	0	0	0	5.89050676...	2.30949323...	5.89050659...
o17	s	no	1	1	1	0	0	0	3.90568641...	1.83431358...	3.90568626...
o18		no	1	1	1	0	0	0	4.56323758...	1.17676241...	4.56323753...
o19		no	1	1	1	0	0	0	4.56323758...	1.17676241...	4.56323753...
o20		no	1	1	1	0	0	0	4.56713722...	1.17286277...	4.56713717...
o21		no	1	1	1	0	0	0	8.54309528...	1.29690471...	8.54309488...
o22		no	0	1	1	0	0	0	13.2082695...	-0.0882695...	13.2082694...
o23	s	yes	0	1	1	0	0	0	11.1976336...	-2.1776336...	11.1976338...
o24		no	0	1	1	0	0	0	9.24206058...	1.41793941...	9.24206038...

Figure 20: New column PreMLR0 is generated when we evaluate MLR0 object

7.2 Evaluate Logistic Regression

7.2.1 Functionality description

This functionality allows the user to select one of the multiple linear regression models that are present in the current GestorMatriu. When a model is selected this panel generates two new columns in data matrix which contains the predictions and log prediction values that are computed from the logistic regression equation.

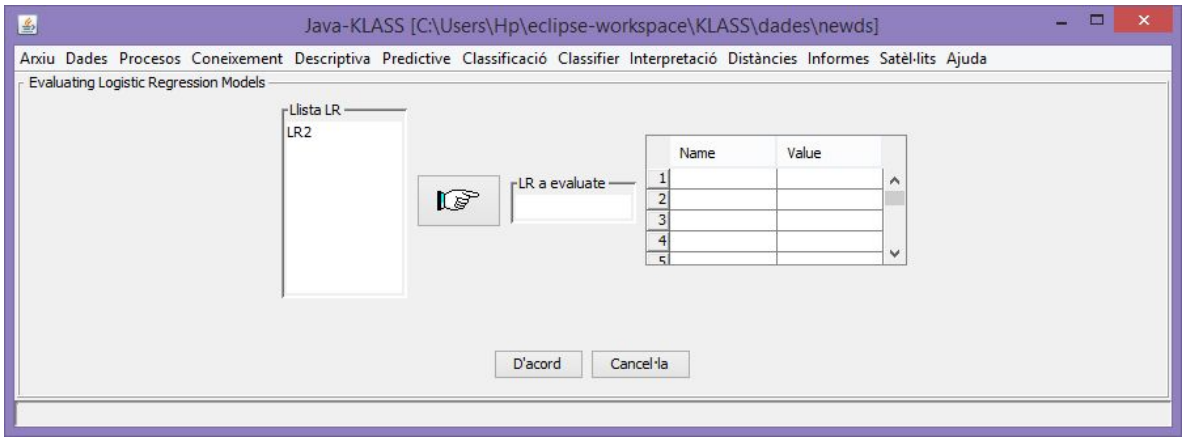


Figure 21: View of Evaluate Logistic regression models panel

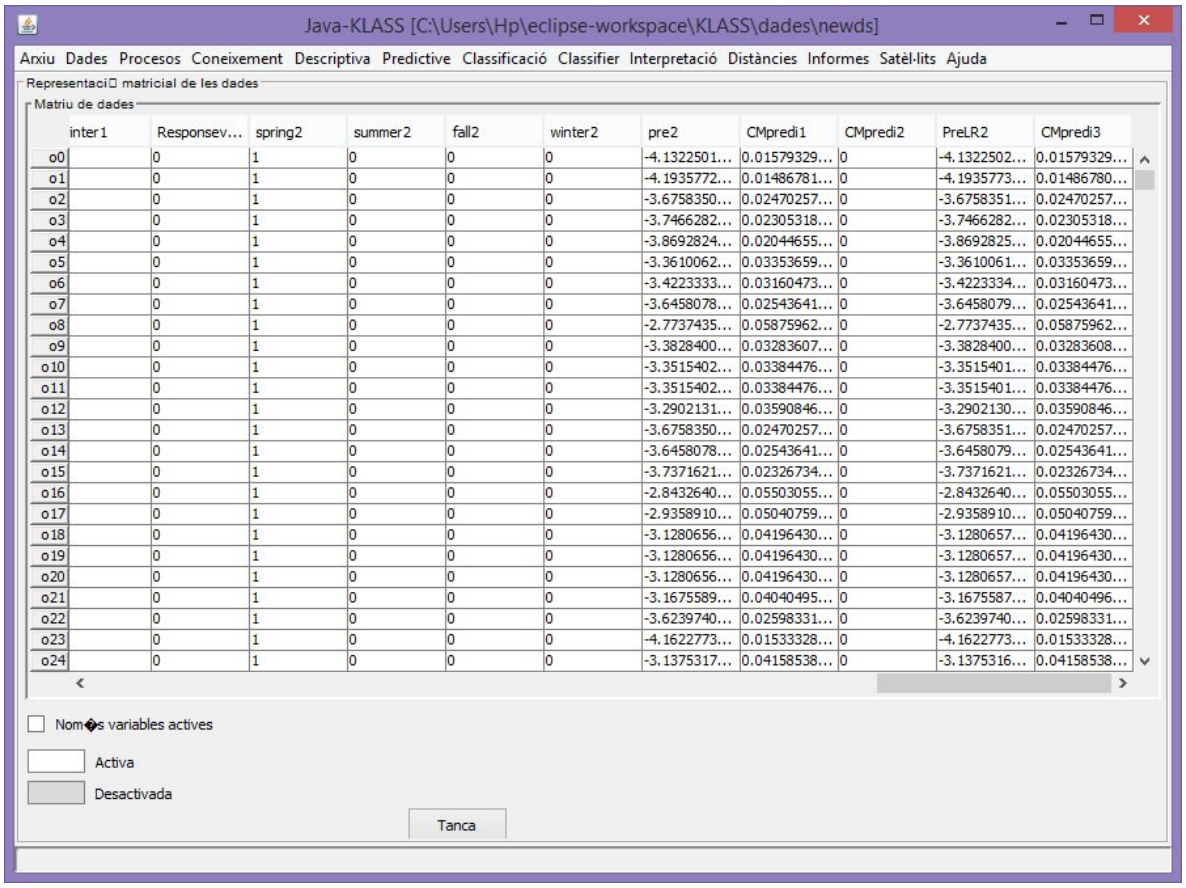
7.2.2 Interface design and specification

- **Llista LR:** This field shows all the logistic regression models present in the current GestorMatriu.
- **LR a evaluate**: The selected LR object will be shown here and the response variable and the explanatory variables with associated coefficients are displayed in a table present on the right side.
- In figure PreLR2 and CMpredi3 are the new columns that are generated when we evaluate LR2 object which contains HolidayRec as response variable and atemp,temp, seasonRec as explanatory variables.

7.2.3 Implementation

In order to develop the Evaluate logistic regression functionality we have added following methods and classes to the KLASS packages that are mentioned in 2.3.

- *PanelELR:* This class calls the panel to evaluate logistic regression models.This class is present in jKlass.iu package.
- *computeEquationLR:* This method which is present in GestorMatriu helps to creates new predicted column using the coefficients and explanatory variables present in the selected model.
- *computeLogprediction :* This method which is present in GestorMatriu helps to creates new log predicted column using the coefficients and explanatory variables present in the selected model.This method is called from the *computeEquationLR* method.



	inter1	Responsev...	spring2	summer2	fall2	winter2	pre2	CMpredi1	CMpredi2	PreLR2	CMpredi3
o0		0	1	0	0	0	-4.1322501...	0.01579329...	0	-4.1322502...	0.01579329...
o1		0	1	0	0	0	-4.1935772...	0.01486781...	0	-4.1935773...	0.01486780...
o2		0	1	0	0	0	-3.6758350...	0.02470257...	0	-3.6758351...	0.02470257...
o3		0	1	0	0	0	-3.7466282...	0.02305318...	0	-3.7466282...	0.02305318...
o4		0	1	0	0	0	-3.8692824...	0.02044655...	0	-3.8692825...	0.02044655...
o5		0	1	0	0	0	-3.3610062...	0.03353659...	0	-3.3610061...	0.03353659...
o6		0	1	0	0	0	-3.4223333...	0.03160473...	0	-3.4223334...	0.03160473...
o7		0	1	0	0	0	-3.6458078...	0.02543641...	0	-3.6458079...	0.02543641...
o8		0	1	0	0	0	-2.7737435...	0.05875962...	0	-2.7737435...	0.05875962...
o9		0	1	0	0	0	-3.3828400...	0.03283607...	0	-3.3828400...	0.03283608...
o10		0	1	0	0	0	-3.3515402...	0.03384476...	0	-3.3515401...	0.03384476...
o11		0	1	0	0	0	-3.3515402...	0.03384476...	0	-3.3515401...	0.03384476...
o12		0	1	0	0	0	-3.2902131...	0.03590846...	0	-3.2902130...	0.03590846...
o13		0	1	0	0	0	-3.6758350...	0.02470257...	0	-3.6758351...	0.02470257...
o14		0	1	0	0	0	-3.6458078...	0.02543641...	0	-3.6458079...	0.02543641...
o15		0	1	0	0	0	-3.7371621...	0.02326734...	0	-3.7371621...	0.02326734...
o16		0	1	0	0	0	-2.8432640...	0.05503055...	0	-2.8432640...	0.05503055...
o17		0	1	0	0	0	-2.9358910...	0.05040759...	0	-2.9358910...	0.05040759...
o18		0	1	0	0	0	-3.1280656...	0.04196430...	0	-3.1280657...	0.04196430...
o19		0	1	0	0	0	-3.1280656...	0.04196430...	0	-3.1280657...	0.04196430...
o20		0	1	0	0	0	-3.1280656...	0.04196430...	0	-3.1280657...	0.04196430...
o21		0	1	0	0	0	-3.1675589...	0.04040495...	0	-3.1675587...	0.04040496...
o22		0	1	0	0	0	-3.6239740...	0.02598331...	0	-3.6239740...	0.02598331...
o23		0	1	0	0	0	-4.1622773...	0.01533328...	0	-4.1622773...	0.01533328...
o24		0	1	0	0	0	-3.1375317...	0.04158538...	0	-3.1375316...	0.04158538...

Figure 22: New columns PreLR2 and CMpredi3 are generated when we evaluate LR2 object

Chapter 8

New Input Output Functionalities

8.1 Exporting and Importing MLR files

8.1.1 Functionality description

When multiple linear regression analysis is performed on a dataset then it will create a MLR model object. To use this model even after the session is closed we are going to create a file which contains

- Name of the model
- response variable
- number of model terms and then the model terms present in the MLR object.

This files are given a .MLR extension and are exported into resultats folder present in KLASS\dades.

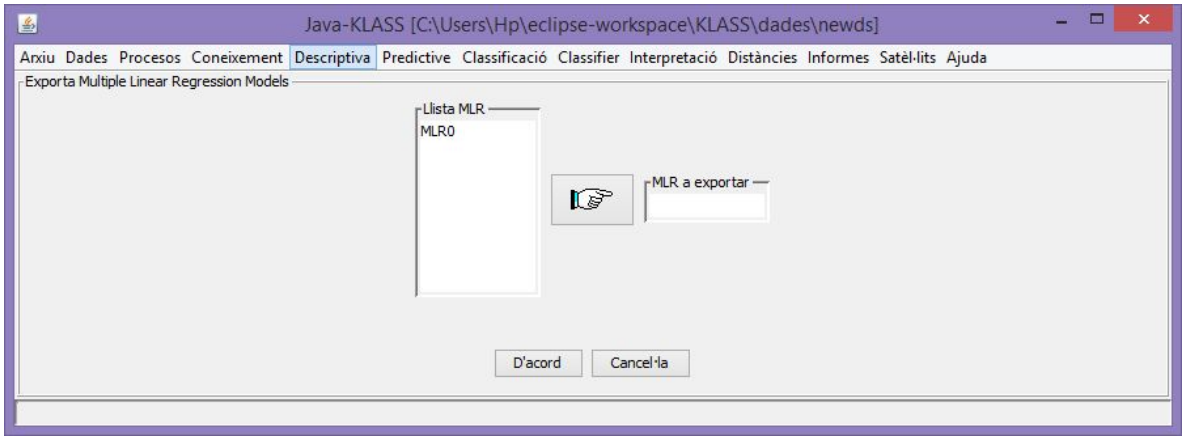


Figure 23: Export Multiple linear regression models

If user want to import the MLR files, then he must go to the Export MLR menuitem present in Arxiu drop-down menu. Once it is clicked then all the .MLR extension files present in KLASS\dades\resultats will be open.When user selects one of this files then the data is retrieved from the file, MLR object is created and added to the list of MLR objects present in GestorMatriu.

8.1.2 Implementation

In order to develop the this we have added following methods and classes to the KLASS packages that are mentioned in 2.3.

- *PanelExportMLR:* This class calls the panel to export MLR objects.This is present in jKlass.iu package.
- *jMenuImportarMLR_actionPerformed*: This method helps to read the .MLR files and creates MLR objects.It is present in FrPrincipal.java.
- *addTolrobjects :* This method which is present in the GestorMatriu adds the newly created MLR object into the list of MLR objects.

8.2 Exporting and Importing LR files

8.2.1 Functionality description

When logistic regression analysis is performed on a dataset then it will create a LR model object. To use this model even after the session is closed we are going to create a file which contains

- Name of the model
- response variable
- number of modelterms and the modelterms present in the LR object.

This files are given a .LOR extension and are exported into resultats folder present in KLASS\dades.

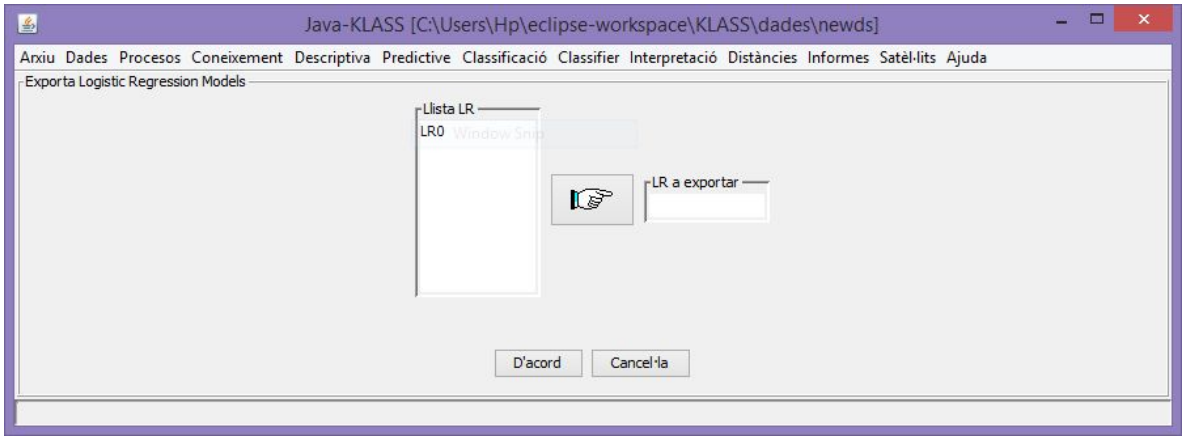


Figure 24: Export Logistic regression models panel

If user want to import the LR files, then he must go to the Export LogReg menuitem present in Arxiu drop-down menu. Once it is clicked then all the .LOR extension files present in KLASS\dades\resultats will be open.When user selects one of this files then the data is retrieved from the file, LR object is created and added to the list of LR objects present in GestorMatriu.

8.2.2 Implementation

In order to develop the this we have added following methods and classes to the KLASS packages that are mentioned in 2.3.

- *PanelExportLR:* This class calls the panel to export MLR objects.This is present in jKlass.iu package.
- *jMenuImportarLR_actionPerformed*: This method helps to read the .LOR files and creates LR objects.It is present in FrPrincipal.java.
- *addTolrobjects :* This method which is present in the GestorMatriu adds the newly created LR object into the list of LR objects.

Chapter 9

Testing

9.1 Validation of Multiple linear regression

The dataset that we have used to validate multiple linear regression module is *iris.dat* [UCI 88] . This dataset contains 3 classes of 50 instances each, where each class refers to a 3 different species of iris flower (Setosa, Virginica and Versicolor) . The rows being the samples with 150 flowers from IRIS genera and the columns being: Sepal Length, Sepal Width, Petal Length, Petal Width and Species.

Now we will perform multiple linear regression analysis for three cases in which

- Case 1: Sepal length as response variable and petal length, petal width and sepal width as explanatory variables that means we are considering only numerical explanatory variables.
- Case 2: Sepal length as response variable and specie as explanatory variable that means only qualitative explanatory variable.
- Case 3: Sepal length as response variable and Sepal width, Petal length, species as explanatory variables. As species is a qualitative explanatory variable, dummies are generated for species and first dummy will be automatically ignored, such that the MLR considers as a reference group with effect included in β_0 . We are estimating the model

Case 1 :

$$Sepal\ length = \beta_0 + \beta_1 Sepal\ width + \beta_2 Petal\ length + \beta_3 Petal\ Width$$

where Versicolor is a binary variable that evaluates 1 for versicolor flowers and 0 to others. Similarly Virginica is a binary variable that evaluates 1 for virginica flowers and 0 to others.Once analysis is done , we will get the results in a LaTex file. (see Table 3)

Coefficients

Variable	Coefficient
Intercept	1.8451
sepalWidth	0.6549
petalLength	0.7111
petalWidth	-0.5626

Table 3: Coefficients obtained in KLASS for case 1

In table 4 the results obtained from R package confirm that we get same results in KLASS (apart from some precision issues).

JSAT is being used double float. So we presume that one precision is got with JSAT, although we open a future line to better check this.

```
> y=Irisdata[,1]
> x1=Irisdata[,2]
> x2=Irisdata[,3]
> x3=Irisdata[,4]
> x4=Irisdata[,5]
>  res<-lm(y~x1+x2+x3)
>  coefficients(res)
(Intercept)          x1          x2          x3
  1.8559975   0.6508372   0.7091320  -0.5564827
```

Table 4: R code and Coefficients obtained in R for case 1

Case 2 :

The regression equation that we are working in this case 2 is:

$$Sepal\ length = \beta_0 + \beta_1 Versicolor + \beta_2 Virginica$$

Coefficients

Variable	Coefficient
Intercept	5.006
versicolor	0.93
virginica	1.582

Table 5: Coefficients obtained in KLASS for case 2

```
> y=Irisdata[,1]
> x1=Irisdata[,2]
> x2=Irisdata[,3]
> x3=Irisdata[,4]
> x4=Irisdata[,5]
>  res<-lm(y~x4)
>  coefficients(res)
      (Intercept) x4Iris-versicolor  x4Iris-virginica
            5.006             0.930             1.582
>
```

Table 6: R code and Coefficients obtained in R for case 2

Case 3:

The regression equation that we are working in this case 3 is:

$$Sepal\ length = \beta_0 + \beta_1 Sepal\ width + \beta_2 Petal\ length + \beta_3 Versicolor + \beta_4 Virginica$$

Coefficients

Variable	Coefficient
Intercept	2.3886
sepalWidth	0.4339
petalLength	0.7748
versicolor	-0.9552
virginica	-1.3928

Table 7: Coefficients obtained in KLASS for case 3

```
> y=Irisdata[,1]
> x1=Irisdata[,2]
> x2=Irisdata[,3]
> x3=Irisdata[,4]
> x4=Irisdata[,5]
>  res<-lm(y~x1+x2+x4)
>  coefficients(res)
      (Intercept)                x1                x2 x4Iris-versicolor  x4Iris-virgini
ca
        2.3903891         0.4322172         0.7756295        -0.9558123        -1.39409
79
```

Table 8: R code and Coefficients obtained in R for case 3

We can see that in all the three cases the coefficients we got when we performed MLR in R are same with the coefficients we got when we perform MLR in KLASS. Even the model parameters, predicted and residuals also matched with those of R.

Now, we can say that our Multiple linear regression module is working well. We have tested our module even with other datasets like Bicingclass.dat, bicingordenate.dat etc. In all the cases, we have benchmarked our results with R.

9.2 Validation of Logistic regression

The dataset that we have used to validate our Logistic regression module is Bicingclass.dat. This data contains 600 rows with 23 columns. These are bicing trips from Montreal city in Canada. It has both numerical and qualitative variables. Some of the numerical variables are temperature, humidity, date, time, windspeed

etc. and some of the qualitative variables are Season {Spring, Summer, Fall, winter}, Holiday {yes, No}, Workingday {yes, No} etc.

Response variable in logistic regression should have only two modalities. In bicingclass.dat Holiday and Workingday has only two modalities. We are now going to perform logistic regression having response variable as Workingday and explanatory variables as temperature and humidity. When the logistic regression analysis is completed, we can see the results in a LaTex file called as LogReg.tex.

Coefficients

Variable	Coefficient
Intercept	0.5208
temp	0.0108
humidity	0.0009

Table 9: Coefficients obtained in KLASS for a logistic model

Now we will perform the logistic regression in R and see whether the results are similar or not. Below figure shows the coefficient values that are obtained in R on performing logistic regression on bicingclass.dat dataset.

```
> y=newds[,13]
> x1=newds[,4]
> x2=newds[,6]
> res<-glm(y~x1+x2,family=binomial)
> coefficients(res)
 (Intercept)           x1           x2
0.5207297927 0.0109719180 0.0008890489
>
```

Table 10: R code and coefficients of LogReg on bicingclass.dat

We can see that the coefficients we got when we performed LR in R are similar with the coefficients we got we we perform LR in KLASS. Now we will see the residual deviance in both cases. (See table 11 for KLASS results and Table 12 for R package results and the results are same).

Response Variable	: WorkingdayRec
Explanatory Variable :	
	temp
	humidity
Deviance	: 743.5537503432047

Table 11: Residual deviance obtained in our module

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 744.51  on 599  degrees of freedom
Residual deviance: 743.55  on 597  degrees of freedom
AIC: 749.55

Number of Fisher Scoring iterations: 4
```

Table 12: Residual deviance obtained in R

Chapter 10

Application to functionality of elderly patients and decision support

A sample of 20 control subjects and 76 neurological patients (affected by spinal cord injury, Parkinson's disease, stroke or depression) and recovering in an Italian hospital were assessed using the WHODAS-II disability assessment scale. The WHO-DASII assessment scale is proposed by the World Health Organization (WHO) to measure the degree of patient disability and covers both physical (self-care, moving, eating or standing-up) and mental (understanding, communicating or life activities) health factors. Version 3.1a contains 96 Likert items related to the ICF classification.

Profiles of functional disability: Four profiles from a previous work were identified by using innovative clustering techniques:

- low (31 self-dependent subjects);
- intermediateI, (24 subjects with a low to moderate degree of physical and emotional disability);
- intermediateII, (6 subjects with moderate to severe disability without emotional problems);
- high (32 subjects with the highest degree of disability).

Surprisingly, these profiles were associated with decreasing levels of functionality in the patient rather than being directly linked to the cause of the disability itself.

As expected, patients of the same profile are qualitatively homogeneous and will presumably respond to the same rehabilitative treatment. From the clinical point of view it is very important to help doctors to properly place a new patient in the taxonomy in a very friendly way. This classification enables a quick and accurate decision about the most suitable treatment for that patient. Such efficient management will contribute to increase the success of therapies. Each profile is associated with a prototypical rehabilitation treatment designed by experts.

In the paper [Gibert 2013] EBLR has been used to find a reduced set of items from WHO-DASII, that can significantly characterise the four profiles. Firstly the whole sample of 96 patients was used to estimate the outer logistic model pHigh. Secondly, the 64 patients not labelled as high were used for the second logistic model pIntII . Finally, the remaining 58 patients provided the logistic equation pIntI to discriminate intermediateI (high values of pIntI) from low (low values of pIntI).

$$p_{High} = \frac{e^{-35.93+1.70*B2+3.35*B4+3.98*B9+2.20*S4}}{(1+e^{-35.93+1.70*B2+3.35*B4+3.98*B9+2.20*S4})}$$

$$p_{IntII} = P(i \in IntermediateII | i \notin High) = \frac{e^{-2.63-1.37*B4+1.30*S9}}{(1+e^{-2.63-1.37*B4+1.30*S9})}$$

$$p_{IntI} = P(i \in IntermediateI | i \notin High \wedge i \notin IntermediateII) = \frac{e^{-13.20+2.20*B9+1.89*S2+1.40*S5}}{(1+e^{-13.20+2.20*B9+1.89*S2+1.40*S5})}.$$

Table 13 :Prediction values from [Gibert 2013]

The PAG visualises the equations described above and suggests an evaluation of the patient using a minimal set of seven relevant items from the WHO-DASII. Here, ε = 0.5 has been used.

Getting predictions with KLASS

The first thing we do in this thesis is to use all new functionalities implemented in KLASS to associate the three logistic equations estimated from data to the current data matrix. Note that in this application we will try to visualize the matrix participating in the trial. However, the same models could be associated with new data sets for further evaluation as well, just using the new functionalities added to KLASS in this final degree work.

The functionality of *EBLR_estimate* method from [Gibert 2013] is achieved by the *computeLogprediction* method present in KLASS and described in section 6.2.3. As like *EBLR_estimate* , *computeLogprediction* calculates the predicted values needed to perform PAG. But, *computeLogprediction* calculates predicted values for the current model in the current dataset.

To get 3 predicted values that are needed to perform 3D visualization, we have performed the series of steps which are :

1. First, we specified the models given in table 13 in 3 LogReg files as figures 25, 26 and 27. These LogReg files contains the structure mentioned in section 8.2.1

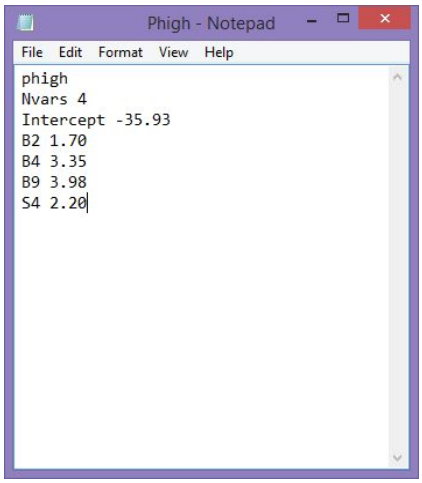


Figure 25: Phigh.LOR file

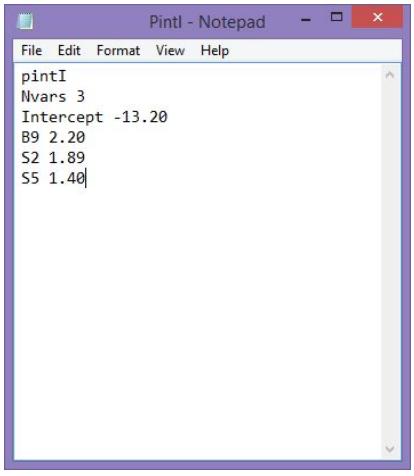


Figure 26 : PintI.LOR file

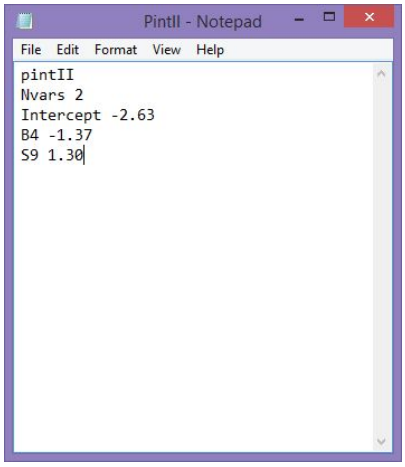


Figure 27: PintII.LOR file

2. Now we have 3 LogReg files present in the KLASS\dades\resultats and when we import them into KLASS then 3 LOR models will be created and added to the list of logistic regression models present in GestorMatriu.

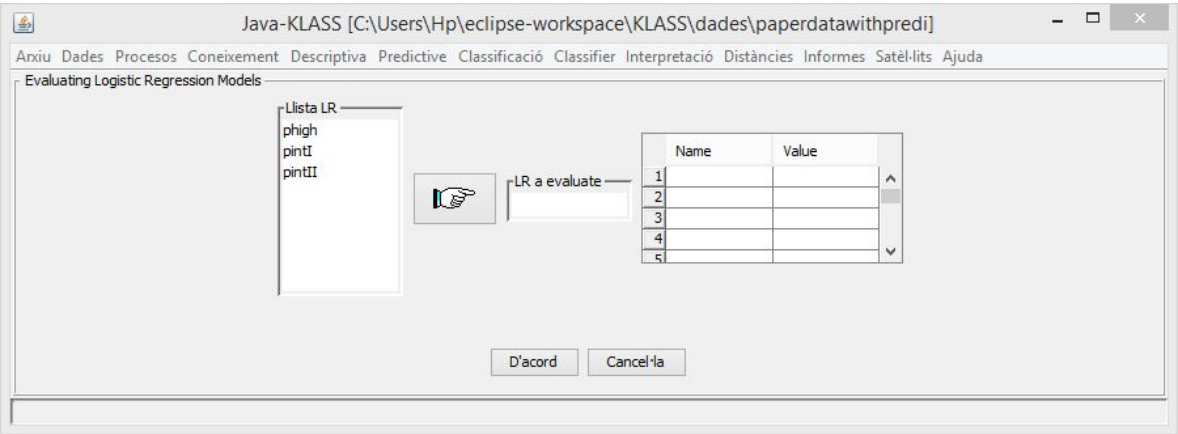
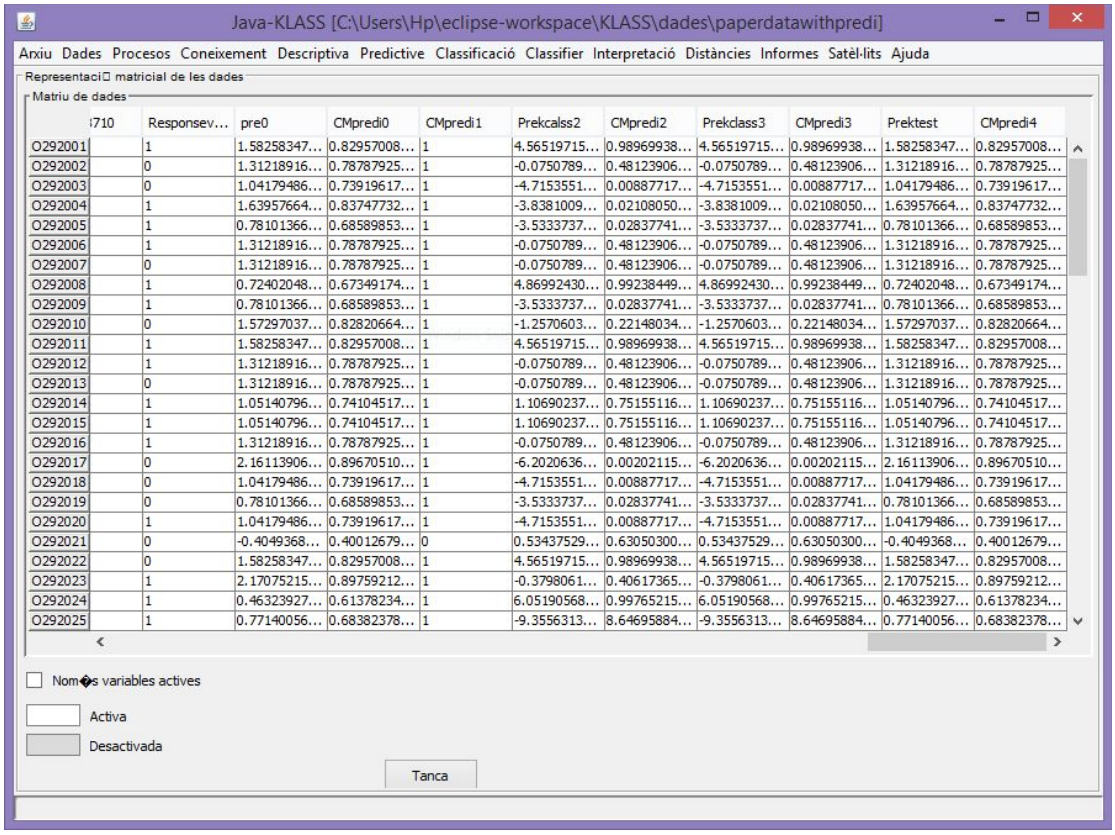


Figure 28: phigh, pintI and pintII models present in Gestor Matriu for evaluation.

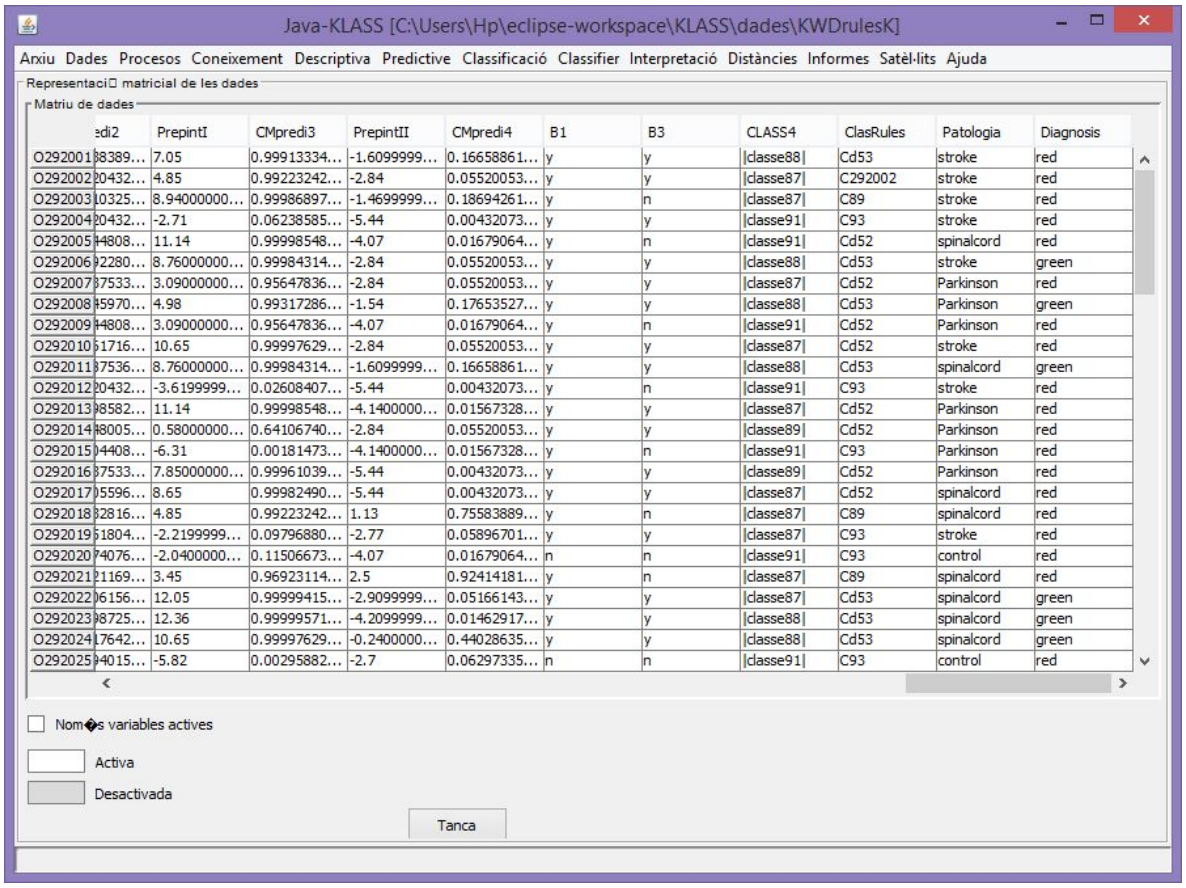
3. Perform evaluation on phigh, pintI and pintII using evaluate logistic regression functionality. This will give us 3 new log predicted variables. In figure 27 we can find CMpredi2, CMpredi3 and CMpredi4 which are the new log predicted variables which will be X , Y and Z axis respectively for a 3D visualization. CMpredi provide $log\left(\frac{\hat{p}}{1-\hat{p}}\right)$ and PreKclass2, PreKclass3 and PreKtest which are $\hat{p}$ are the predicted variables from which we have generated our log predicted variables.
4. This CMpredi2, CMpredi3, CMpredi4 are the values required to locate a patient in the PAG. Since we are unable to cover in this project the implementation of the PAG cube. We can use the 3D representation functionality of KLASS available in descriptive menu, to make a 3D visualization of all patients in the sample, in a cube with the same axis the PAG has. To do that we need a T column to indicate the color of each point assigned according to PAG. The resulting correspond to a pseudo PAG projecting the same information. The color instead of leaving on the walls of the cube as in original paper [Gibert 2013] is transferred to the patient's points itself. That color is determined by EBLR_use algorithm described in section 3.6.
5. BC Diagnosis is codified in a file called BCDiagnosis0.5.reg that keeps for all other applications provided that the predicted variables for the 3 classes are named as CMpredi2, CMpredi3 and CMpredi4 in KLASS dataset, also for the diagnosis with other ε is only the matter of changing the threshold in BC Diagnosis.

6. If we perform 3D visualization, which is present in *descriptive* menu, taking CMpredi2 in X , CMpredi3 in Y, CMpredi4 in Z and Diagnosis0.5 in T , then we will get a pseudo PAG which looks like in figure 29 (with ε=0.5) and figure 30 (with ε=0.3).



	i710	Responsev...	pre0	CMpredi0	CMpredi1	Prekcalss2	CMpredi2	Prekclass3	CMpredi3	Prektest	CMpredi4
O292001		1	1.58258347...	0.82957008...	1	4.56519715...	0.98969938...	4.56519715...	0.98969938...	1.58258347...	0.82957008...
O292002		0	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292003		0	1.04179486...	0.73919617...	1	-4.7153551...	0.00887717...	-4.7153551...	0.00887717...	1.04179486...	0.73919617...
O292004		1	1.63957664...	0.83747732...	1	-3.8381009...	0.02108050...	-3.8381009...	0.02108050...	1.63957664...	0.83747732...
O292005		1	0.78101366...	0.68589853...	1	-3.5333737...	0.02837741...	-3.5333737...	0.02837741...	0.78101366...	0.68589853...
O292006		1	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292007		0	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292008		1	0.72402048...	0.67349174...	1	4.86992430...	0.99238449...	4.86992430...	0.99238449...	0.72402048...	0.67349174...
O292009		1	0.78101366...	0.68589853...	1	-3.5333737...	0.02837741...	-3.5333737...	0.02837741...	0.78101366...	0.68589853...
O292010		0	1.57297037...	0.82820664...	1	-1.2570603...	0.22148034...	-1.2570603...	0.22148034...	1.57297037...	0.82820664...
O292011		1	1.58258347...	0.82957008...	1	4.56519715...	0.98969938...	4.56519715...	0.98969938...	1.58258347...	0.82957008...
O292012		1	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292013		0	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292014		1	1.05140796...	0.74104517...	1	1.10690237...	0.75155116...	1.10690237...	0.75155116...	1.05140796...	0.74104517...
O292015		1	1.05140796...	0.74104517...	1	1.10690237...	0.75155116...	1.10690237...	0.75155116...	1.05140796...	0.74104517...
O292016		1	1.31218916...	0.78787925...	1	-0.0750789...	0.48123906...	-0.0750789...	0.48123906...	1.31218916...	0.78787925...
O292017		0	2.16113906...	0.89670510...	1	-6.2020636...	0.00202115...	-6.2020636...	0.00202115...	2.16113906...	0.89670510...
O292018		0	1.04179486...	0.73919617...	1	-4.7153551...	0.00887717...	-4.7153551...	0.00887717...	1.04179486...	0.73919617...
O292019		0	0.78101366...	0.68589853...	1	-3.5333737...	0.02837741...	-3.5333737...	0.02837741...	0.78101366...	0.68589853...
O292020		1	1.04179486...	0.73919617...	1	-4.7153551...	0.00887717...	-4.7153551...	0.00887717...	1.04179486...	0.73919617...
O292021		0	-0.4049368...	0.40012679...	0	0.53437529...	0.63050300...	0.53437529...	0.63050300...	-0.4049368...	0.40012679...
O292022		0	1.58258347...	0.82957008...	1	4.56519715...	0.98969938...	4.56519715...	0.98969938...	1.58258347...	0.82957008...
O292023		1	2.17075215...	0.89759212...	1	-0.3798061...	0.40617365...	-0.3798061...	0.40617365...	2.17075215...	0.89759212...
O292024		1	0.46323927...	0.61378234...	1	6.05190568...	0.99765215...	6.05190568...	0.99765215...	0.46323927...	0.61378234...
O292025		1	0.77140056...	0.68382378...	1	-9.3556313...	8.64695884...	-9.3556313...	8.64695884...	0.77140056...	0.68382378...

Figure 29: CMpredi2, CMpredi3 and CMpredi4 are new log predicted variables.



edi2	PrepintI	CMpredi3	PrepintII	CMpredi4	B1	B3	CLASS4	ClasRules	Patologia	Diagnosis
O292001 38389...	7.05	0.99913334...	-1.6099999...	0.16658861...	y	y	|classe88|	Cd53	stroke	red
O292002 20432...	4.85	0.99223242...	-2.84	0.05520053...	y	y	|classe87|	C292002	stroke	red
O292003 10325...	8.94000000...	0.99986897...	-1.4699999...	0.18694261...	y	n	|classe87|	C89	stroke	red
O292004 20432...	-2.71	0.06238585...	-5.44	0.00432073...	y	y	|classe91|	C93	stroke	red
O292005 44808...	11.14	0.99998548...	-4.07	0.01679064...	y	n	|classe91|	Cd52	spinalcord	red
O292006 92280...	8.76000000...	0.99984314...	-2.84	0.05520053...	y	y	|classe88|	Cd53	stroke	green
O292007 87533...	3.09000000...	0.95647836...	-2.84	0.05520053...	y	y	|classe87|	Cd52	Parkinson	red
O292008 45970...	4.98	0.99317286...	-1.54	0.17653527...	y	y	|classe88|	Cd53	Parkinson	green
O292009 44808...	3.09000000...	0.95647836...	-4.07	0.01679064...	y	n	|classe91|	Cd52	Parkinson	red
O292010 61716...	10.65	0.99997629...	-2.84	0.05520053...	y	y	|classe87|	Cd52	stroke	red
O292011 87536...	8.76000000...	0.99984314...	-1.6099999...	0.16658861...	y	y	|classe88|	Cd53	spinalcord	green
O292012 20432...	-3.6199999...	0.02608407...	-5.44	0.00432073...	y	n	|classe91|	C93	stroke	red
O292013 98582...	11.14	0.99998548...	-4.1400000...	0.01567328...	y	y	|classe87|	Cd52	Parkinson	red
O292014 48005...	0.58000000...	0.64106740...	-2.84	0.05520053...	y	y	|classe89|	Cd52	Parkinson	red
O292015 94408...	-6.31	0.00181473...	-4.1400000...	0.01567328...	y	n	|classe91|	C93	Parkinson	red
O292016 87533...	7.85000000...	0.99961039...	-5.44	0.00432073...	y	y	|classe89|	Cd52	Parkinson	red
O292017 05596...	8.65	0.99982490...	-5.44	0.00432073...	y	y	|classe87|	Cd52	spinalcord	red
O292018 32816...	4.85	0.99223242...	1.13	0.75583889...	y	n	|classe87|	C89	spinalcord	red
O292019 61804...	-2.2199999...	0.09796880...	-2.77	0.05896701...	y	y	|classe87|	C93	stroke	red
O292020 74076...	-2.0400000...	0.11506673...	-4.07	0.01679064...	n	n	|classe91|	C93	control	red
O292021 21169...	3.45	0.96923114...	2.5	0.92414181...	y	n	|classe87|	C89	spinalcord	red
O292022 06156...	12.05	0.99999415...	-2.9099999...	0.05166143...	y	y	|classe87|	Cd53	spinalcord	green
O292023 98725...	12.36	0.99999571...	-4.2099999...	0.01462917...	y	y	|classe88|	Cd53	spinalcord	green
O292024 17642...	10.65	0.99997629...	-0.2400000...	0.44028635...	y	y	|classe88|	Cd53	spinalcord	green
O292025 94015...	-5.82	0.00295882...	-2.7	0.06297335...	n	n	|classe91|	C93	control	red

Figure 30: Data Matrix with Diagnosis variable which we will take as T in 3D vvisualization.

Implementing EBLR_use in KLASS through KnowledgeBase

The *Avaluar BC* functionality present in KLASS helps us to generate the T categorical variable that directly use EBLR_use algorithm and produces a new variable with the result of each predicted variable (see figure 28). We take the advantage of the knowledge base management functionalities already available in KLASS to implement the EBLR_use algorithm. In fact no new code is required since EBLR can be formalized directly for a knowledge base.

BC Diagnosis 0.5

We have generated a knowledge base with four rules taking $\varepsilon\, as\, 0.5$. In figure 28, the diagnosis value for a patient having an id 202924 is green. Now if we observe the CMpredi2,CMpredi3 and CMpredi4 values for this id, we will see that CMpredi2 and CMpredi3 value is less than 0.5 and CMpredi4 value is greater than 0.5. r_3 in the knowledge base is satisfied for this id. So the value green is given in the diagnosis for 202924 id.

$$r0 : CMpredi2 \geq 0.5 \longrightarrow red$$

$$r1 : (CMpredi2 < 0.5) \wedge (CMpredi3 \geq 0.5) \longrightarrow yellow$$

$$r2 : (CMpredi2 < 0.5) \wedge ((CMpredi3 < 0.5) \wedge (CMpredi4 \geq 0.5)) \longrightarrow green$$

$$r3 : (CMpredi2 < 0.5) \wedge ((CMpredi3 < 0.5) \wedge (CMpredi4 < 0.5)) \longrightarrow blue$$

Table 14: Knowledge Base that replaces EBLR_use for $\varepsilon = 0.5$

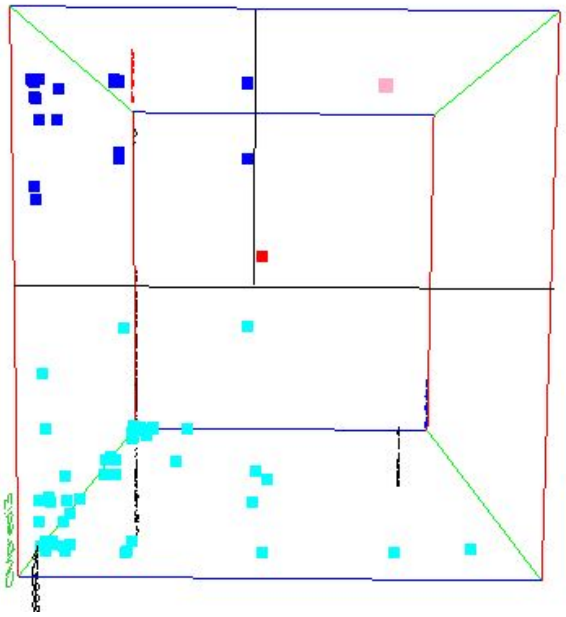


Figure 31: Pseudo PAG for $\varepsilon = 0.5$

we can see 4 different regions present in pseudo PAG which resembles the 4 regions present in the PAG. Majority of class high is given as impaired sample of patients. Moving ε more or less, conservative diagnosis will be given.

BC Diagnosis 0.3

We have generated a knowledge base with four rules taking $\varepsilon\, as\, 0.3$. The *Avaluar BC* functionality present in KLASS helps us to generate a categorical variable for this knowledge base. (See table 15)

$$r0 : CMpredi2 < 0.3 \longrightarrow red$$

$$r1 : (CMpredi2 \geq 0.3) \wedge (CMpredi3 < 0.3) \longrightarrow yellow$$

$$r2 : (CMpredi2 \geq 0.3) \wedge ((CMpredi3 \geq 0.3) \wedge (CMpredi4 < 0.3)) \longrightarrow green$$

$$r3 : (CMpredi2 \geq 0.3) \wedge ((CMpredi3 \geq 0.3) \wedge (CMpredi4 \geq 0.3)) \longrightarrow blue$$

Table 15: Knowledge Base when $\varepsilon = 0.3$

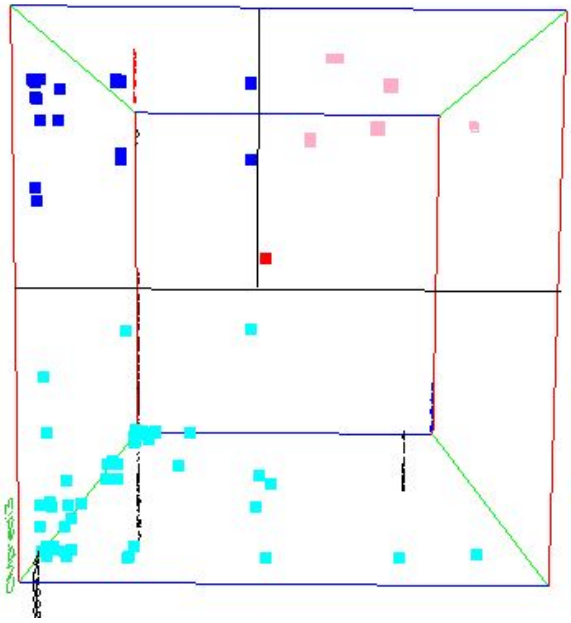


Figure 33: Pseudo PAG with $\varepsilon = 0.3$

In this case with $\varepsilon = 0.3$, we can observe that there are more C_3 class.

If we add colors to the walls of this Pseudo-PAG then we will get a 3D graph which resembles like PAG (see figure 34)

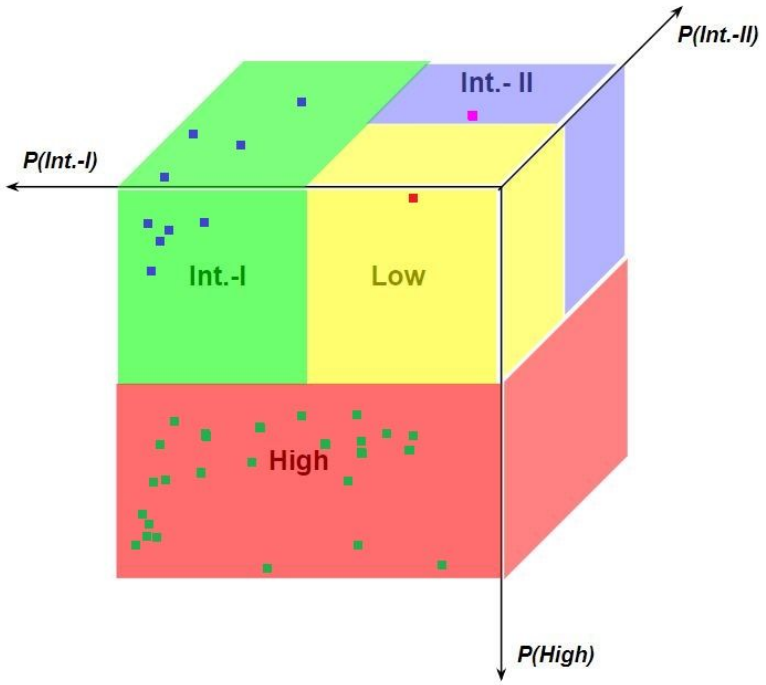


Figure 34: Pseudo PAG with colors for $\varepsilon = 0.3$

Generation of EBLR Logistic regression from scratch

In this we used previously found LogReg to show how pseudo-PAG can be visualized using functionalities that KLASS offers, also the probability to fit the LogReg models internally as well. To implement the EBLR_estimate algorithm we need to follow the sequence of steps described below:

- Our response variable is Class4. But response variable in logistic regression must be a binary variable.
- From the original data set, perform logistic regression analysis by taking Class4.class91 as response variable which is generated by using recode functionality in KLASS and B2,B4,B7 as numerical explanatory variables save the model.
- Now, select a submatrix by removing all the samples that contains class88 modality in Class4 variable.
- Perform logistic regression on this sub matrix with the response variable being Class4.class88 as response variable and B2,B4,B7 as explanatory variables and save the model.
- Now, select a submatrix from the above matrix by removing all the samples that contains class87 modality in Class4 variable.

- Perform logistic regression on this sub matrix with the response variable being Class4.class87 as response variable and B2,B4,B7 as explanatory variables and save the model.
- Now we have 3 LOR files present in the KLASS\dades\resultats and when we import them into KLASS then 3 LOR models will be created and added to the list of logistic regression models present in GestorMatriu.
- Perform evaluation of logistic regression on the 3 models using evaluate logistic regression functionality. This will give us 3 predicted variables required for the 3D visualization.

Chapter 11

Conclusions

All the functionalities that we implemented produced more than satisfying results and have been integrated well with the rest of the application and this launches release javaKlass v19. Now KLASS has all the functionalities that we have mentioned in 1.2. As we benchmarked our results with R, we can confirm that our modules are working well and are producing correct results. KLASS is now able to learn predictive probabilities for a certain number of clusters and visualize the assigned class of a patient through the EBLR methodology in a 3D graph giving a first support to the diagnosis and follow up of patients In my third year of B.Tech I have Data Analysis and Data Mining course. The concepts present in that course helped me to understand the multiple linear regression and logistic regression. I have been programming in java for past two years and it helped me to implement the methods in more efficient way.

I have been introducing logistic regression into KLASS as a means to build PAG. So, we have prioritize all the elements required to build the PAG and there is an important missing part which is the inference on the models. But, as we can export the logistic regression models outside KLASS, we can easily enter them into R and can run all the inferencies and now we can eliminate non significant terms from the model files (.LOR or .MLR) while the inference of the module being infected, out of the scope of the paper.

Chapter 12

Future Works

- In future, we can do the real implementation of the PAG procedure that are the sequence of calls to the existing functionalities in the KLASS.
- We can do the modification of the 3D representation that incorporates the wall with the colors in the cube.
- We can include classical statistical modelling that makes inference on coefficients of logistic regression models and the goodness of fit indicators.
- In future, it would be interesting to implement the PAG for more than 4 classes.
- We can add new functionality that automatically generates BC Diagnosis given the 3 probability variables and threshold (trivial using current form of BC Diagnosis).

List of Figures

List of Tables

Bibiliography

1. [Gibert 2013] K.Gibert, G.Rodriguez-Silva, R.Annicchiarico (2013) *Post-processing: bridging the gap between modeling and effective decision-support. The profile assessment grid in human behavior.*
2. [Gibert 91] K. Gibert. Klass. *estudi d'un sistema d'ajuda al tractament estadstic de grans bases de dades.* Master's thesis, Universitat Politecnica de Catalunya, 1991.
3. [Gibert 94] K. Gibert. *L'us de la informacio simbolica en l'automatitzacio del tractament estadstic de dominis poc estructurat*s. PhD thesis, EIO Dep, UPC, Barcelona, Spain, 1994.
4. [Gibert 05] K. Gibert and R. Nonell. *Descriptive statistics with klass. supporting latex documents elaboration. In Proceedings of the 3rd World Conference on Computational Statistics and Data Analysi*s, page 90, 2005.
5. [Gibert 08] K. Gibert and R. Nonell. *Pre and postprocessing in klass. In Proceedings of the iEMSs IVth International Congress of Environmental Modeling and Software (DM-TES'08 Workshop)*, 2008.
6. [GCV 12] K. Gibert, D. Conti, and D. Vrecko. *Assisting the end-user in the interpretation of profiles for decision support. an application to wastewater treatment plants. Environmental Engineering and Management Journal,* 11(5):931{944, January 2012.
7. [GRSA 13] K. Gibert, G. Rodrguez-Silva, and R. Annicchiarico. *Post-processing: Bridging the gap between modelling and effective decision-support. the prole assessment grid in human behaviour. Mathematical and Computer Modelling*, page 16331639, April 2013.
8. [Gibert 14] K. Gibert. *Automatic generation of classes interpretation as a bridge between clustering and decision making. International Journal of Multicriteria Decision Making*, 4(2), 2014.
9. [GC 14] K. Gibert and D. Conti. *On the understanding of proles by means of post-processing techniques: an application to financial assets*. International Journal of Computer Mathematics, 2014.
10. [GGRRS 08] K. Gibert, A. Garca-Rudolph, and G. Rodrguez-Silva. *The role of kdd support-interpretation tools in the conceptualization of medical proles: an application to neurorehabilitation*. Acta Informatica Medica, 16(4):178{182, December 2008.
11. [MS 14] S. Molla Santiago. *Generalitzacio de metodes de density-based clustering a dades mixtes*. Master's thesis, Universitat Politecnica de Catalunya, June 2014.
12. [Raff 17] Edward Raff (2017 April) *JSAT: Java Statistical Analysis Tool, a Library for Machine Learning.*
13. [Baron 2014] Baron, M, Chapman & Hall *Probability and statistics for computer scientists* - , 2014. ISBN: 9781439875902

14. [Montgomery 2012] DC Montgomery, EA Peck, GG Vining. *Introduction to linear regression analysis*, 2012.
15. [SCASM+10] L. Salvador-Carulla, Jose Alberto Salinas, M. Martn, M. Gran,K. Gibert, M. Roca, and A. Bulbena. *A preliminary taxonomy and a standard knowledge base for mental-health system indicators in spain. International Journal of Mental Health Systems,* 4(1):1{28, 2010.
16. [Mathware] K. Gibert, Ulises cortes. *Weighting quantitative and qualitative variables in clustering methods*. 1997.
17. [Waikato] https://www.cs.waikato.ac.nz/ml/weka/
18. [vanderbilt] http://www.mc.vanderbilt.edu/gcrc/workshop_files/2004-11-12.pdf
19. [UCI 88] https://archive.ics.uci.edu/ml/datasets/iris

.