Mathematical Techniques on Machine Learning

Eleftherakis Stavros

University of Crete, Department of Mathematics and Applied Mathematics

05/03/2020

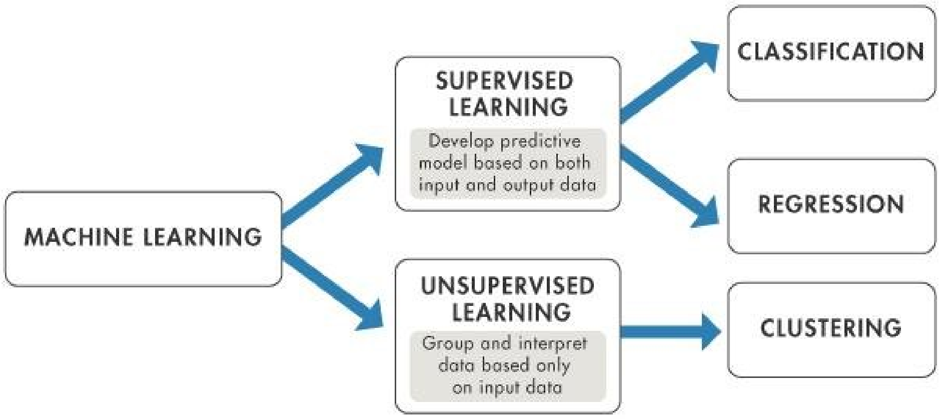
EXAMINATION COMMITTEE PAGE

Committee Chairperson: Georgios Kosioris
Committee Members: Georgios Kosioris, Panagiotis Chatzipantelidis, Michael Taroudakis

Abstract

For sure Machine Learning is an ascending academic field and researchers has put much effort during the last decades. In the first of part of this Thesis we will cover a contradiction of PCA and LDA, thus applying them to both the famous Iris Dataset and also a dataset consisted of a Greek team's football players and their attributes. In the beginning, we will perform a data reduction task with both PCA and LDA and after that, using LDA as a supervised algorithm we will solve a three-class classification problem in order to classify a new player(transfer target)as good, medium or bad transfer. Interpreting the results of PCA loadings we will try to characterize which team's attributes are the strongest, highlighting that our results coincide with Feature Selection Techniques(Univariate Selection with Anova Test). Moreover,we will understand that LDA can be seen both as a supervised and unsupervised technique and also can easily overcome the SSS(small sample size) problem.

After that we will analyze some famous supervised ML algorithms and more specifically Decision Trees, Random Forests and K-NN, thereby applying all of them for Educational purposes. More specifically, we will formulate a binary classification problem in order to predict which percentage of both our Departments' active students will manage to graduate or not, thus understanding that Applied Mathematics direction will face a difficult situation. Simple Statistical tools like Pearson Correlation Coefficient and more technical ones like one-way Anova test will be also useful in order to analyze the students' academic behaviour. Especially, the last one will show us that female students, who study in the above Department are more diligent than the male ones, thus tearing down the saying that men perform better than women in Mathematics.


MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Contents

Chapter 1

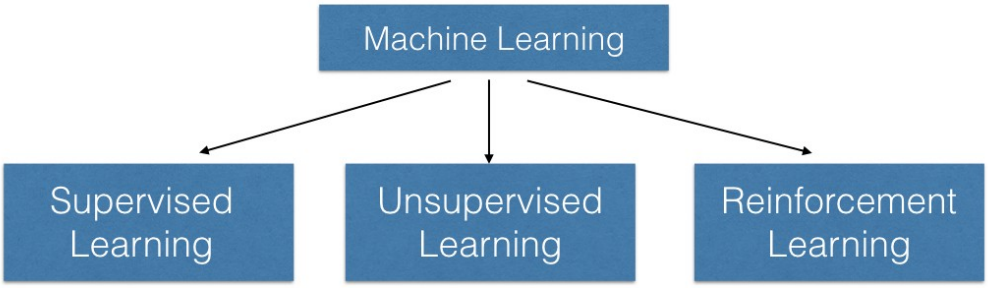
Introduction

1.1 Background

During the last decades scientists' research have focused on Data Analysis and Machine Learning. In practice Machine Learning can be seen as a subset of Artificial Intelligence, where the other important subset is Deep Learning. It is difficult to give a certain definition for Machine Learning but we can say that: "Machine Learning is the scientific study of algorithms and statistical tools that computer science use in order to perform a specific task effectively without using explicit instructions relying on patterns and inference instead. In practice, ML algorithms build a mathematical model based on training data(samples) in order to make predictions for unseen data. Machine Learning algorithms are used in a wide variety of applications such as: Cancer Prediction, Bioinformatics, Evaluation of Educational procedures, Email Filtering, Image Processing, Environmental purposes, Business Problems, Social Problems and many others. Typically, the mathematical background that we use is not considered as a difficult one since we build up our knowledge, mostly, based on Linear Algebra, Calculus, Information Theory ,Probability Theory, Statistics and Optimization. Based on the above, somebody may claim that the construction of a ML model is an easy task, something that it does not hold since we have to take into consideration many factors and combine them efficiently. It needs a deep knowledge of many Mathematical concepts, high computational abilities, scientific mature and for sure much practice and experience.

To be more specific, Machine Learning includes three different types of techniques: Unsupervised Learning, Supervised Learning and Reinforcement Learning. The first two have been analyzed well during the last years whereas the third approach is a newer one and is very efficient for different type of applications than the first two. For instance, reinforcement learning techniques can help us create automated vehicles(for instance cars that can drive and park alone) using a very large amount of data. In this Thesis we will cover the first two approaches and especially Supervised one. The major difference is that in the unsupervised ML approach we do not know the class of each observation whereas in the supervised we take the class of the observations into consideration. The goal of an Unsupervised technique can be data reduction and data visualization, whereas the goal for the supervised techniques that we will analyze is classification. The most common supervised technique is PCA and we will compare and contrast this technique with LDA which can be considered both as a supervised and an unsupervised one(Part 1 of this Thesis). We will apply these techniques to both the famous Iris Dataset(Ficher, 1936) and also a football players dataset that we created alone. We will interpret the results, thus understanding that LDA is a stronger technique, thereby overcoming easily the famous SSS(small sample size) problem.In the second part of this work, we will analyze some well-known supervised Machine Learning algorithms and more specifically Decision Trees, Random Forests and K-NN. We will understand what is a hyperparameter and also suggest a way of Hyperparameter Tuning(GridSearch mehtod) in order to find the optimal hyperparameters for these algorithms, thereby optimizing their performance. Moreover, we will analyze the interesting topic of "Automated ML", thus understanding the different existing methods for training our data. We will apply K-NN, Decision Trees, Random Forests and LDA(supervised approach) to two different datasets that are consisted of the students that were enrolled in either the Department of Mathematics or the Applied Mathematics one since 2009. We will try to predict the percentage of active students that will manage to graduate or not(binary classification problem). For performance metrics, except of the accuracy score will also use ROC curves and Precision Recall Curves and their AUCs, in order to find which of the four supervised algorithms is the most appropriate for the classification task(more reliable prediction). Apart from that we will use some simple statistical tools such as descriptive statistics or Correlation Coefficients and some more complicated like Anova Test. They will help us understand the students' academic behaviour and also the application of Anova test will show us that female students have a better performance than men students.

I have to point out that this Thesis started a year ago(February, 2019, when I had a completely different scientific background. Part 1 had been completed until September, and Part 2 have been constructed from September until now. To be honest, I am not satisfied with what I have covered in the first part of my Thesis, since my analysis is typical and the applications are simple. I am completely satisfied with the second Part of the Thesis, since I studied more interesting and useful Machine Learning concepts and and applied them to a much more interesting application. So, comparing Part 1 and Part 2 we can see the one year process of a student that started studying Machine Learning from a zero point.


Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning

Part I

Chapter 2

PRINCIPAL COMPONENTS ANALYSIS

2.1 History of PCA

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set.

PCA was invented in 1901 by Karl Pearson as an analogue of the principal axis theorem in mechanics.It was later independently developed and named by Harold Hotelling in the 1930s.Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT) in signal processing, the Hotelling transform in multivariate quality control, proper orthogonal decomposition (POD) in mechanical engineering, singular value decomposition (SVD) of X (Golub and Van Loan, 1983), eigenvalue decomposition (EVD) of XTX in linear algebra or empirical orthogonal functions (EOF) in meteorological science, empirical eigenfunction decomposition (Sirovich, 1987), empirical component analysis (Lorenz, 1956) spectral decomposition in noise and vibration, and empirical modal analysis in structural dynamics.

PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to visualize genetic distance and relatedness between populations. PCA can be done by eigenvalue decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after a normalization step of the initial data. The normalization of each attribute consists of mean centering – subtracting each data value from its variable's measured mean so that its empirical mean (average) is zero – and, possibly, normalizing each variable's variance to make it equal to 1.The results of a PCA are usually discussed in terms of component scores, sometimes called factor scores (the transformed variable values corresponding to a particular data point), and loadings (the weight by which each standardized original variable should be multiplied to get the component score).If component scores are standardized to unit variance, loadings must contain the data variance in them (and that is the magnitude of eigenvalues). If component scores are not standardized (therefore they contain the data variance) then loadings must be unit-scaled, ("normalized") and these weights are called eigenvectors; they are the cosines of orthogonal rotation of variables into principal components or back.[wiki]

Figure 2.1: Karl Pearson 1857-1936

Figure 2.2: Harold Hotelling 1895-1973

2.2 Problem Setting

2.2.1 Introduction

In PCA, we are given a dataset of n observation(called also samples or data points) $X = x_1, ...x_n$ and we are looking for projections $\tilde{x_n}$,that are as similar as possible to the original data points, but which have a significantly lower intrinsic dimensionality. To be more specific,the dataset of n observations $X = (x_1, x_2, ..., x_n)$,$x_n \in R^D$, with mean 0 creates the data covariance matrix:

$$S = \frac{1}{N}\sum_{i=1}^{N} x_n x_n^T \tag{2.1}$$

In addition, we suppose that there exists a low-dimensional compressed representation of x_n:

$$z_n = B^T x_n$$

(2.2)

,$z_n \in R^M$ where we define the projection matrix $B = [b_1, b_2, ..., b_m] \in R^{D\times M}$ We suppose that the columns of B are orthonormal and we seek an M-dimensional subspase $U \subseteq R^D$ with dim(U)=M<D onto which we project the data. We denote the projected data by $\tilde{x_n} \in U$ and their coordinates(with respect to the basis $b_1, ..., b_M$ of U) by z_n. Our aim is to find projections $\tilde{x_n} \in R^D$ (or equivalently the codes z_n and the basis vectors $b_1, ..., b_m$), so that they are as similar to the original data x_n and minimize the loss due to compression.

In PCA, we find a compressed version z of original data x. The compressed data can be reconstructed into $\tilde{x_n}$, which lives in the original data space,but has an intrinsic lower-dimensional representation than x. Practically, the columns $b_1, ..., b_M$ of B form a basis of the M-dimensional subspace in which the projected data $\tilde{x} \in R^D$ live.

Note: In PCA, we consider a linear relationship between the sample x and its low-dimensional code z so that $z = B^T x$ and $\tilde{x} =$ Bz for a suitable matrix B. Based the motivation of thinking PCA as a data compression technique, we can think the above relations as a pair of operations representing encoders and decoders. The linear mapping represented by B can be thought of a decoder, which maps the low-dimensional code $z \in R^M$ back into the original data space R^D. Similarly, B^T can be thought of an encoder, which encodes the original data x as a low-dimensional (compressed) code z.

In the next section, we will maximize the variance of the data,in order to find low-dimensional representations that retain as much information as possible and minimize the compression loss.

2.3 Maximum Variance Perspective

2.3.1 Direction with Maximal Variance

We maximize the variance of the low-dimensional code using a sequential approach. We start by seeking a single vector $b_1 \in R^D$,that maximizes the variance of the projected data. Practically,we aim to maximize the variance of the first coordinate z_1 of z $\in R^M$,so that

$$\mathrm{V}_1 = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} z_{1n}^2 \text{ (2.3)}$$

is maximized, where we exploited the i.i.d. assumption of the data and defined z_{1n} as the first coordinate of the low-dimensional representation $z_n \in R^M$ of $x_n \in R^D$.
Note that the first component of z_n is given by

$$\mathrm{z}_{1n} = b_1^T x_n \text{ (2.4)}$$

We substitute (2.2) into (2.1), so:

$$V_1 = \frac{1}{N}\sum_{i=1}^{N}(b_1^T x_n)^2 = \frac{1}{N}\sum_{i=1}^{N} b_1^T x_n x_n^T b_1 = b_1^T(\frac{1}{N}\sum_{i=1}^{N} x_n x_n^T)b_1 = b_1^T S b_1 \tag{2.5}$$

where S is the data covariance matrix,which we defined in the first section.
We restrict all solutions to $\|b_1\|^2 = 1$, which results in a constrained optimization problem in which we seek the direction along which the data varies most.
With the restriction of the solution space to unit vectors the vector b_1 that points in the direction of maximum variance can be found by the constrained optimization problem:

$$\underset{b_1}{max} b_1^T S b_1$$

$$\text{subject to } \|b_1\|^2 = 1 \text{ (2.6)}$$

We solve easily the problem (2.5),using our basic knowledge of optimization theory. Firstly we obtain the Lagrangian L(b_1, λ_1)=$b_1^T S b_1 + \lambda_1(1 - b_1^T b_1)$ After that,we calculate the partial derivatives of L with respect to b$_1 and \lambda_1$ and we set them equal to zero so:

$$\frac{\partial L}{\partial b_1} = 2b_1^T S - 2\lambda_1 b_1^T = 0, \frac{\partial L}{\partial \lambda_1} = 1 - b_1^T b_1 = 0 \tag{2.7}$$

Solving the relations (2.4),gives us that:

$$\text{Sb}_1 = \lambda_1 b_1 \text{ (2.8)}$$

$$\text{b}_1^T b_1 = 1\text{(2.9)}$$

Therefore,it is clear that b_1 is an eigenvector of the data covariance matrix S and the Lagrange multiplier plays the role of the eigenvalue.The eigenvector property(2.5) allows us to rewrite our variance objective as:
$V_1 = b_1^T S b_1 = \lambda_1 b_1^T b_1 = \lambda_1$ So,the variance of the data projected onto a one-dimensional subspace equals the eigenvalue that is associated with the basis vector b_1 that spans this subspace. **Therefore, to maximize the variance of the low-dimensional code, we choose the basis vector associated with the largest eigenvalue of the data covariance matrix S. This eigenvector is called the *FIRST PRINCIPAL COMPONENT*** . We can determine the effect/contribution of the principal component b_1 in the original data space by mapping the coordinate z_{1n} back into data space, which gives us the projected data point.[reference b]

$$\tilde{x_n} = b_1 z_{1n} = b_1 b_1^T x_n \in R^D \tag{2.10}$$

in the original data space

REMARK: It is clear that following the same procedure we can show that the second principal component is associated with the eigenvector of the second greater eigenvalue, thus capturing variance $V_2 = \sum_{i=1}^{M} \lambda_2$. [reference b]

2.3.2 M-dimensional Subspace with Maximal Variance

We suppose that we have found the first $m - 1$ principal components as the $m - 1$ eigenvectors of S that are associated with the largest $m - 1$ eigenvalues. Since S is symmetric,we know from linear algebra that we can use these eigenvectors to construct an orthonormal eigenbasis of an $(m-1)$ dimensional subspace of R^D.Inductively,we can understand that the m-th principal component is associated with the eigenvector of the m-th largest eigenvalue of the data covariance matrix S and the maximum variance that we can capture with the first M principal components is $V_M = \sum_{m=1}^{M} \lambda_m$.We will quote the whole procedure that verifiies our inductive result based on the online book "Mathematics for Mahine Learning"authored by "Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong" .

Generally, the m-th principal component can be found by subtracting the effect of the first $m-1$ principal components $b_1, ..., b_{m-1}$ from the data, thereby trying to find principal components that compress the remaining information. We achieve this by first subtracting the contribution of the m_1 principal components from the data, similar to (2.7), so that we arrive at the new data matrix:

$$\hat{X} = X - \sum_{i=1}^{m-1} b_i b_i^T X \tag{2.11}$$

,where $X = (x_1, ..., x_n) \in R^{DxN}$ contains the data points as columns vectors.The matrix $\hat{X} = (\hat{x_1}, ..., \hat{x_n}) \in R^{DxN}$ contains the data that only contains the information that has not yet been compressed.

REMARK: (Notation). Throughout this chapter, we do not follow the convention of collecting data $x_1, ..., x_N$ as the rows of the data matrix, but we define them to be the columns of X. This means that our data matrix X is a DxN matrix instead of the conventional NxD matrix. The reason for our choice is that the algebra operations work out smoothly,without the need to either transpose the matrix or to redefine vectors as row vectors that are left-multiplied onto matrices.

To find the mth principal component, we maximize the variance

$$V_m = V_{zm} = \frac{1}{N} \sum_{i=1}^{N} z_{mn}^2 = \frac{1}{N} \sum_{i=1}^{N} (b_m^T x_n)^2 = b_m^T \hat{S} b_m \tag{2.12}$$

subject to $\|b_m\|^2 = 1$,where we followed the same steps as in the previous sections and defined $\hat{S}$ as the data covariance matrix of the transformed dataset $\hat{X} = (\hat{x_1}, ..., \hat{x_n})$.As previously, when we looked at the first principal component alone, we solve a constrained optimization problem and discover that the optimal solution b_m is the eigenvector of $\hat{S}$

that is associated with the largest eigenvalue of $\hat{S}$.
However, it also turns out that bm is an eigenvector of S. It holds that:

$$\hat{S} = \frac{1}{N}\sum_{n=1}^{N} \hat{x_n}\hat{x_n^T} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \sum_{i=1}^{m-1} b_i b_i^T x_n)(x_n - \sum_{i=1}^{m-1} b_i b_i^T x_n) = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T - 2x_n x_n^T \sum_{i=1}^{m-1} b_i b_i^T + \sum_{i=1}^{m-1} b_i b_i^T x_n x_n^T \tag{2.13}$$

,where we exploited the symmetries $x_n^T b_i = b_i^T x_n$ and $b_i x_n^T = x_n b_i^T$ to summarize that

$$- x_n x_n^T \sum_{i=1}^{m-1} b_i b_i^T - \sum_{i=1}^{m-1} b_i b_i^T x_n x_n^T = -2x_n x_n^T \sum_{i=1}^{m-1} b_i b_i^T \tag{2.14}$$

If we take a vector b_m with $\|b_m\| = 1$,that is orthogonal to all the vectors $b_1, ..., b_{m-1}$ and right-multiply b_m to $\hat{S}$ in (2.10),we obtain

$$\hat{S}b_m = \frac{1}{N}\sum_{n=1}^{N} \hat{x_n}\hat{x_n^T} b_m = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T b_m = S b_m = \lambda_m b_m \tag{2.15}$$

Equation (2.12) reveals that b_m is an eigenvector of both $\hat{S}$ and the original data covariance matrix S.In other words,λ_m is the largest eigenvalue of $\hat{S}$ and λ_m is the largest eigenvalue of $\hat{S}$ and λ_m is the largest eigenvalue of $\hat{S}$ and the λ_m is the m-th largest eigenvalue of S and both have the associated eigenvector b_m. This derivation shows that there is an intimate connection between the M-dimensional subspace with maximal variance and the eigenvalue decomposition. With the relation (10.21) and $b_m^T b_m = 1$, the variance of the data projected onto the mth principal component is

$$V_m = b_m^T S b_m = \lambda_m b_m^T b_m = \lambda_m \tag{2.16}$$

This means that the variance of the data, when projected onto an M-dimensional subspace, equals the sum of the eigenvalues that are associated with the corresponding eigenvectors of the data covariance matrix.

Overall, to find an M-dimensional subspace of RD that retains as much information as possible, PCA tells us to choose the columns of the projection matrix B in as the M eigenvectors of the data covariance matrix S that are associated with the M largest eigenvalues.The maximum amount of variance PCA can capture with the first M principal components is

$$V_m = \sum_{m=1}^{M} \lambda_m \tag{2.17}$$

, where the λ_m are the M largest eigenvalues of the data covariance matrix S.**Consequently,the variance lost by data compression via PCA is:**

$$J_m = \sum_{j=m+1}^{D} \lambda_j = V_D - V_M \tag{2.18}$$

Instead of these absolute quantities, we can define the relative variance captured as $\frac{V_M}{V_D}$ and the relative variance lost by compression as $1 - \frac{V_M}{V_D}$

Chapter 3

LINEAR DISCRIMINANT ANALYSIS(LDA)

3.1 History of LDA

The original linear discriminant analysis(LDA) was developed by Sir Ronald Fisher in 1936,when he was the head of of the Department of Eugenics at University College London. He introduced LDA with the famous example of iris dataset that we will explain later.

Linear discriminant analysis (LDA),or normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

LDA is closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made.

Figure 3.1: Sir Ronald Aylmer Fisher 1890-1962

3.2 Introduction

In the previous chapter we analyzed the most famous **unsupervised dimensionality reduction** technique called PCA. Now we will also understand how LDA works and try to copmare and contrast PCA and LDA. The aim of LDA technique is to transform the features into a lower dimensional subspace io order to maximize the class separability. Although the LDA technique is one of the most well-used data reduction techniques, it suffers from two problems. First of all, LDA fails to find the the lower dimensional subspace if the dimensions are much higher than the numbers of data points, which is known as the Small Sample Problem(SSS). In addition,if different classes are not linearly separable, the LDA cannot discriminate between these classes(linearity problem).

In the next sections we will see how LDA Techniques work, thereby explaining the three key steps of LDA. Also we will explain a numerical example to see practically what is going on.

3.3 LDA Technique

The goal of LDA is to project the original data matrix onto a lower dimensional subspace. We point out that the dimension of the lower dimension space is at most c-1, where c equals the number of classes. To fulfil this target we perform a three step procedure. The first step is to calculate the between class variance or between class matrix, which shows the seperability between different classes. The second step is to calculate the within class variance or within class matrix, which shows the distance between the mean and the samples of each class. Finally,we will construct the lower dimensional subspace, which maximizes the between class variance(matrix) and minimizes the within class variance(matrix). We are going to analyze briefly these three steps and represent a numerical example in order to understand how LDA works.

STEP 1: Calculating the between class variance(matrix),(S_B)
The between class variance of the mth class (S_{B_m}) equals the distance between the mean of the mth class (μ_m)and the total mean μ.We are looking for a lower dimensional subspace which maximizes the between class variance.Let's see we can calculate the between class variance.We assume that we are given a dataset of N observations $X = [x_1, x_1, ..., x_N]$,where $x_N \in R^M$.Also,we suppose that our data are partitioned into three classes as follows $X = [\omega_1, \omega_2, \omega_3]$ and each class has five observations(samples),so $n_1 = n_2 = n_3 = 5$.Obviously,the total number of samples $N = \sum_{i=1}^{N}$.In order to calculate S_B,firstly we calculate the separation between classes ,which is $(m_i - m)$,as follows:

$$(m_i - m)^2 = (W^T\mu_i - W^T\mu)^2 = W^T(\mu_i - \mu)(\mu_i - \mu)^T W \tag{3.1}$$

,where m_i is the projection of the mean of the i-th class and is calculated as $m_i = W^T\mu_i$.The term m is the projection of the total mean of all classes and is claculated as $m = W^T\mu$,where W is the transformation matrix of LDA(we will explain exactly what we mean in a few pages).Finally,the term $mu_i \in R^(1 \times M)$,represents the mean of the ith class and $\mu \in R^(1 \times M)$,represents the total mean of all classes.We calculate the terms μ_i and μ as follows:

$$\mu_j = \frac{1}{n_j}\sum_{x_i \in \omega_j} x_i \tag{3.2}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \sum_{i=1}^{c} \frac{n_i}{N}\mu_i \tag{3.3}$$

The term $(\mu_i - \mu)(\mu_i - \mu)^T$ in equation (3.1) represents the between class variance of the ith class which we denoted as S_{B_i}.Substituting $(\mu_i - \mu)(\mu_i - \mu)^T$=$S_{B_i}$ into equation (3.1):

$$(m_i - m)^2 = W^T S_{B_i} W \tag{3.4}$$

In order to calculate the total between class variance we use the relation:

$$S_B = \sum_{i=1}^{c} n_i S_{B_i} \tag{3.5}$$

STEP 2: Calculating the within class variance(matrix) S_W
The within class variance(matrix) of the ith class represents the distance between the mean and the samples of that class.Our aim is to minimize the distance between the projected mean(m_i) and the projected samples(observations) of each class ($W^T x_i$) or equivalently minimize the within class variance.We calculate the within class variance of each class S_{W_j} as follows:

$$S_{W_j=d_j^T d_j(3.6)}$$

,where d_j is given by

$$d_j = \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T \tag{3.7}$$

,where the term x_{ij} represents the ith observation of the j-th class.Prcatically,we use use the term d_j in order to center the data of the j-th class so we can write simply that: $d_j = \omega_j - \mu_j$.In order to calcualte the total within class variance(matrix) we use the next equation:

$$S_W = \sum_{i=1}^{c} S_{W_i} \tag{3.8}$$

,where c represents the total number of classes.

STEP 3: Constructing the lower dimensional subspace
Now that we have calculated the between class variance(matrix) S_B and the within class variance S_W,we can easily calculate the transformation matrix W that we referred to in step 1 using the next equation,which is known as Fisher's criterion.The idea behind the next equation is that we aim to maximize the between class variance while minimizing the within class variance so:

$$J(w) = max_w \frac{w^T S_B w}{w^T S_W w} \tag{3.9}$$

,which equivalently can be written as:

$$S_B w = \lambda S_W w \tag{3.10}$$

where λ represents the eigenvalues of the transformation matrix $S_W^{-1} S_B$,if S_W is invertible.

PROOF
We take the derivative of (3.9) according to w and set it equal to zero,so:

$$\frac{\partial J(w)}{\partial w} = 0 \tag{3.11}$$

Using the quotient rule for derivatives we get that:

$$w^T S_W w S_B w - w^T S_B w S_w w = 0 \Rightarrow S_B w - \frac{w^T S_B w S_W w}{w^T S_W w} = 0 \Rightarrow S_B w = \lambda S_W w \tag{3.12}$$

,where we set that $\lambda = \frac{w^T S_B w}{w^T S_W w}$.
So,beginning from Fisher's criterion(3.9) we proved that it is equivalent to the equation (3.10),which we are going to solve in order to find the optimal eigenvectors w. ■

The eigenvectors represent the direction of the new space that we are constructing and the eigenvalues show the scaling factor.Based on this outlook,each eigenvalue represent on axis of the new space and the associated eigenvalue shows the potential of this eigenvector.The potential of the eigenvector is important in order to achieve the aim of LDA technique,i.e. increase the between class variance and decrease the within class variance of each class.So,the eigenvectors that are associated with the k-highest eigenvalues are used to construct the lower dimensional subspace,which is denoted as V_k, that we seek.After finding the lower dimensional subspace V_k,each observation $x_i \in R^K$ will be represented in this k-dimensional space by projecting it onto V_k as follows:

$$y_i = x_i V_k \tag{3.13}$$

Numerical Example
We will present a numerical example in order to understand better the calculation of the three LDA steps that we saw before.
We are given two classes ω_1 and ω_2 as follows

$$\omega_1 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{bmatrix}$$

$$\omega_2 = \begin{bmatrix} 4 & 2 \\ 5 & 0 \\ 5 & 2 \\ 3 & 2 \\ 5 & 3 \\ 6 & 3 \end{bmatrix}$$

After that we calculate the mean of each class
$\mu_1 = \begin{bmatrix} 3 & 3.6 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 4.67 & 2 \end{bmatrix}$

The total mean is
μ= $\begin{bmatrix}3.91 & 2.727\end{bmatrix}$,since $\mu_{11} = \frac{n_1}{n_1+n_2}3 + \frac{n_2}{n_1+n_2}4.67 = \frac{5}{11}3 + \frac{6}{11}4.67 = 3.91$ and similarly we calculate μ_{12}
Using the relation (3.1) we calculate the between class matrix:
$S_B = n_1(\mu_1 - \mu)^T(\mu_1 - \mu) + n_2(\mu_2 - \mu)^T(\mu_2 - \mu)$= $\begin{bmatrix}7.58 & -7.27\\ -7.27 & 6.98\end{bmatrix}$
Now,in order to calculate the within class matrix we start by centering the data of each class as follows:$d_i = \omega_i - \mu_i$

$$d_1 = \begin{bmatrix}-2 & -1.60\\ -1 & -0.60\\ 0 & -0.60\\ 1 & 1.40\\ 2 & 1.40\end{bmatrix}$$

$$d_2 = \begin{bmatrix}-0.67 & 0\\ 0.33 & -2\\ 0.33 & 0\\ -1.67 & 0\\ 0.33 & 1\\ 1.33 & 1\end{bmatrix}$$

Now we are going to calculate the total within class matrix(step 2) which is the sum of the within class matrices of each class
$S_{W_1} = d_1^T d_1 = \begin{bmatrix}10 & 8\\ 8 & 7.2\end{bmatrix}$ $S_{W_2} = \begin{bmatrix}5.33 & 1\\ 9 & 6\end{bmatrix}$ So,$S_W = S_{W_1} + S_{W_2} = \begin{bmatrix}15.33 & 9\\ 9 & 13.2\end{bmatrix}$ The LDA transformation matrix is given by: $W = S_W^{-1}S_B$ where $S_W^{-1} = \begin{bmatrix}0.11 & -0.07\\ -0.07 & 0.13\end{bmatrix}$,

and as we computed before, S=$\begin{bmatrix}7.58 & -7.27\\ -7.276.98\end{bmatrix}$ So,the transformation matrix
$W =$$\begin{bmatrix}1.36 & -1.31\\ -1.48 & 1.42\end{bmatrix}$
Finally,we calculate the eigenvalues and eigenvectors of W,and we find that $\lambda = 0$ or $\lambda = 2.78$ and the asocciated eigenvectors are v_1=$\begin{bmatrix}-0.69\\ -0.72\end{bmatrix}$ and
v_2=$\begin{bmatrix}0.68\\ -0.74\end{bmatrix}$

It is clear that the second eigenvector v_2 has greater coressponding eigenvalue than the first one,so we choose v_2 in order to construct the lower dimensional subspace that we seek.We know that the projection y_i of the data into this subspace is given by:$y_i = \omega_i v_2$,so:

$$y_1 = \omega_1 v_2 \begin{bmatrix}-0.79\\ -0.85\\ -0.18\\ -0.97\\ -0.29\end{bmatrix}$$ and similarly we find y_2:

$$y_2 = \omega_2 v_2 = \begin{bmatrix} 1.24 \\ 3.39 \\ 1.92 \\ 0.56 \\ 1.18 \\ 1.86 \end{bmatrix}$$

3.4 Problems of LDA Technique

PROBLEM 1: SMALL SAMPLE SIZE(SSS)

Small Sample Size (SSS) is the major problem of LDA technique. This problem results from high-dimensional pattern classification tasks or a low number of training samples available for each class compared with the dimensionality of the sample space. Based on this, we can understand that when the columns of data matrix(features)are more than the rows(samples) of data matrix, LDA technique faces problems.

The most common solution to this problem is the introduction of PCA+LDA technique.In this technique, the original d-dimensional features are first reduced to h-dimensional feature space using PCA, and then the LDA is used to further reduce the features to k-dimensions. The PCA is used in this technique to reduce the dimensions to make the rank of S_W is N c hence, the SSS problem is addressed. However, the PCA neglects some discriminant information, which may reduce the classification performance.

PROBLEM 2:LINEARITY PROBLEM

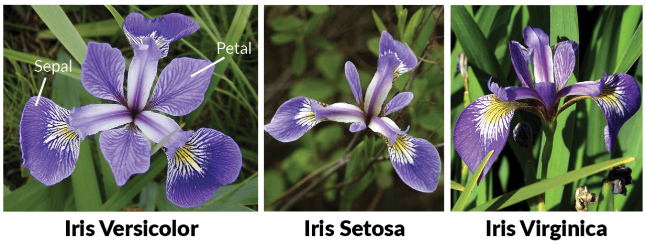
LDA technique is used to to find a linear transformation that discriminates between different classes. However, if the classes are non-linearly separable, LDA can not find a lower dimensional space. In other words, LDA fails to find the LDA space when the discriminatory information are not in the means of classes. This is because the means of the two classes(μ_i) and the total mean(μ) are equal.The mathematical interpretation for this problem is as follows: if the means of the classes are approximately equal, so the S_B and W will be zero. Hence, the LDA space cannot be calculated. The solution to this problem is given by kernel methods, which will be covered in a future work.

Chapter 4

IRIS DATASET

4.1 INTRODUCTION

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.Based on Fisher's linear discriminant model, this data set became a typical test case for many supervised and unsupervised Machine Learing techniques such as PCA and LDA. We will perform both PCA and LDA algorithms to this dataset and explain the results.Iris Dataset is a good example in order to explain the difference between supervised(LDA) and unsupervised(PCA) techniques in data mining.



sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1
5.4	3.7	1.5	0.2
4.8	3.4	1.6	0.2
4.8	3	1.4	0.1
4.3	3	1.1	0.1
5.8	4	1.2	0.2
5.7	4.4	1.5	0.4
5.4	3.9	1.3	0.4
5.1	3.5	1.4	0.3
5.7	3.8	1.7	0.3
5.1	3.8	1.5	0.3
5.4	3.4	1.7	0.2
5.1	3.7	1.5	0.4
4.6	3.6	1	0.2
5.1	3.3	1.7	0.5
4.8	3.4	1.9	0.2
5	3	1.6	0.2
5	3.4	1.6	0.4
5.2	3.5	1.5	0.2
5.2	3.4	1.4	0.2
4.7	3.2	1.6	0.2
4.8	3.1	1.6	0.2
5.4	3.4	1.5	0.4
5.2	4.1	1.5	0.1
5.5	4.2	1.4	0.2
4.9	3.1	1.5	0.1
5	3.2	1.2	0.2
5.5	3.5	1.3	0.2
4.9	3.1	1.5	0.1
4.4	3	1.3	0.2
5.1	3.4	1.5	0.2
5	3.5	1.3	0.3
4.5	2.3	1.3	0.3
4.4	3.2	1.3	0.2
5	3.5	1.6	0.6
5.1	3.8	1.9	0.4
4.8	3	1.4	0.3
5.1	3.8	1.6	0.2

sepal length	sepal width	petal length	petal width
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.5	2.8	4.6	1.5
5.7	2.8	4.5	1.3
6.3	3.3	4.7	1.6
4.9	2.4	3.3	1
6.6	2.9	4.6	1.3
5.2	2.7	3.9	1.4
5	2	3.5	1
5.9	3	4.2	1.5
6	2.2	4	1
6.1	2.9	4.7	1.4
5.6	2.9	3.6	1.3
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1
6.2	2.2	4.5	1.5
5.6	2.5	3.9	1.1
5.9	3.2	4.8	1.8
6.1	2.8	4	1.3
6.3	2.5	4.9	1.5
6.1	2.8	4.7	1.2
6.4	2.9	4.3	1.3
6.6	3	4.4	1.4
6.8	2.8	4.8	1.4
6.7	3	5	1.7
6	2.9	4.5	1.5
5.7	2.6	3.5	1
5.5	2.4	3.8	1.1
5.5	2.4	3.7	1
5.8	2.7	3.9	1.2
6	2.7	5.1	1.6
5.4	3	4.5	1.5
6	3.4	4.5	1.6
6.7	3.1	4.7	1.5
6.3	2.3	4.4	1.3
5.6	3	4.1	1.3
5.5	2.5	4	1.3
5.5	2.6	4.4	1.2
6.1	3	4.6	1.4
5.8	2.6	4	1.2
5	2.3	3.3	1
5.6	2.7	4.2	1.3
5.7	3	4.2	1.2
5.7	2.9	4.2	1.3

sepal length	sepal width	petal length	petal width
6.2	2.9	4.3	1.3
5.1	2.5	3	1.1
5.7	2.8	4.1	1.3
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1
6.3	2.9	5.6	1.8
6.5	3	5.8	2.2
7.6	3	6.6	2.1
4.9	2.5	4.5	1.7
7.3	2.9	6.3	1.8
6.7	2.5	5.8	1.8
7.2	3.6	6.1	2.5
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
5.7	2.5	5	2
5.8	2.8	5.1	2.4
6.4	3.2	5.3	2.3
6.5	3	5.5	1.8
7.7	3.8	6.7	2.2
7.7	2.6	6.9	2.3
6	2.2	5	1.5
6.9	3.2	5.7	2.3
5.6	2.8	4.9	2
7.7	2.8	6.7	2
6.3	2.7	4.9	1.8
6.7	3.3	5.7	2.1
7.2	3.2	6	1.8
6.2	2.8	4.8	1.8
6.1	3	4.9	1.8
6.4	2.8	5.6	2.1
7.2	3	5.8	1.6
7.4	2.8	6.1	1.9
7.9	3.8	6.4	2
6.4	2.8	5.6	2.2
6.3	2.8	5.1	1.5
6.1	2.6	5.6	1.4
7.7	3	6.1	2.3
6.3	3.4	5.6	2.4
6.4	3.1	5.5	1.8
6	3	4.8	1.8
6.9	3.1	5.4	2.1
6.7	3.1	5.6	2.4
6.9	3.1	5.1	2.3
5.8	2.7	5.1	1.9
6.8	3.2	5.9	2.3
6.7	3.3	5.7	2.5
6.7	3	5.2	2.3
6.3	2.5	5	1.9
6.5	3	5.2	2
6.2	3.4	5.4	2.3
5.9	3	5.1	1.8

We will work with Iris Dataset as it is given above,where we mention that the first 50 rows refer to iris-setosa,the second 50 rows refer to iris-versicolor and the final 50 rows refer to iris-virginica.

4.2 IMPLEMENTATION OF ALGORITHMS

We will imply PCA and LDA algorithms to the Iris Dataset in order to distinguish the species from each other.First of all,we will see which are the principal components in each algorithm.After that we will calculate the variance of data that each algorithm achieves.Based on this,we will explain the 'error' of each algorithm.We will use Python programming,thereby importing some basic libraries of Python.The python code is given in the APPENDICES SECTION.

Both PCA and LDA are data reduction techniques.In our dataset,each sample $x_i \in R^4$.Using,PCA and LDA each sample will be expressed in R^2.We explained in the theory section that the variance that each method holds is an indicator for the spread of the data.If we manage to keep a large amount of variance,we minimize the information lost due to compression.Let's compare and contrast the results we get from the two techniques in order to decide which of the two is strongest.

Implying PCA algorithm to our dataset,we find the ratio of the two principal components as follows: (0.92461621,0.05301557)
Practically,the first principal component on x-axis fulfils our target,thus classifying the three types of flowers.

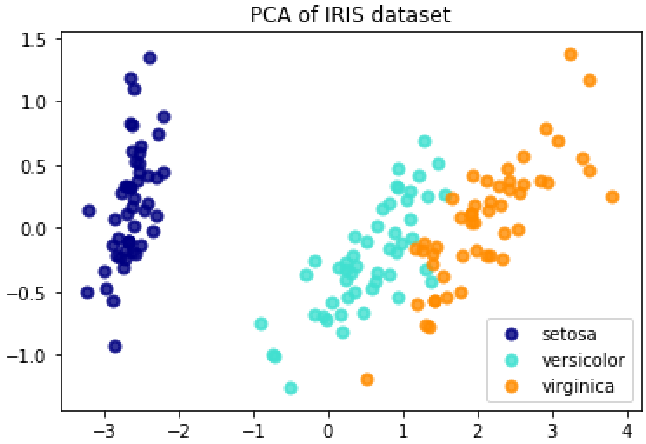


Figure 4.1: PCA CLASSIFICATION WITH TWO PRINCIPAL COMPONENTS

As we can see PCA classifies clearly iris-setosa(blue points) from the other two types of flowers.The classification of iris-versiocolor and iris-virginica is also clear for the most of genes.This result comes from the high amount of variance that PCA keeps. The feature that is more responsible for data reduction is the third feature of our dataset(pca loadings).

Now,let's imply LDA algorithm to Iris Dataset. So, the ratio of two principal components is: (0.99147228,0.0852752). As we can see again the first principal component on the x-axis fulfils the target. Now the variance ratio is higher so we will achieve a better visualization of our data.

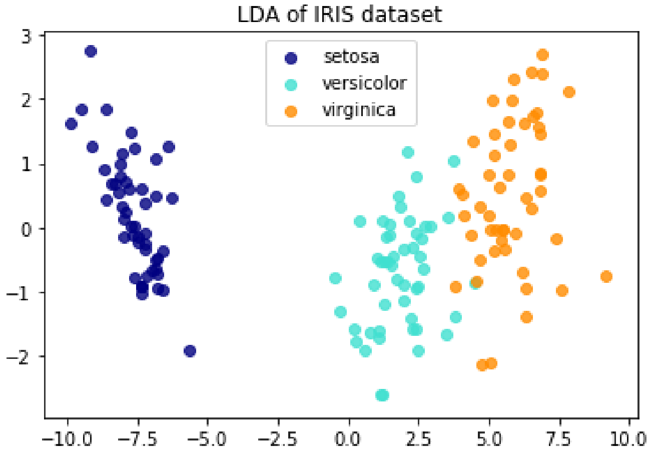


Figure 4.2: LDA CLASSIFICATION WITH TWO PRINCIPAL COMPONENTS

But may we perform another important task with LDA? Since LDA is a supervised technique we have a target for each one of our samples so we will try to train an algorithm in order to classify new samples as setosa, versicolor or virginica. First of all we split our data to training and test set in order to train our algorithm. We will use python train and test split command with 70 percent of our initial data as training set and 30 percent as test set. In order to achieve a more reliable result, we are going to do this 100 times(using a loop outside the command, repeated-hold out method). We will test the results of our algorithm in the test set in order to check if it is strong enough in order to predict unseen data. The accuracy(correct classified/overall) of our algorithm is approximately 97 percent so it is highly accurate and its results are reliable for predictive analysis.

COMPARISON OF TWO METHODS:

The dataset is considered to be a 'good' one since the observations(150) are much more than the features(4), so both methods,as we exprected, gave as an acceptable result. LDA is a little stronger than PCA in this example since it covered larger amount of variance ratio. That's why we say that supervised learning methods(like LDA)are stronger than unsupervised methods(like PCA). After that,we understood that LDA can be seen both as reduction and classification technique. We applied LDA to our dataset in order to classify every new sample that will be given to us and we know that we can perform this task in a highly accurate way.

Chapter 5

FOOTBALL DATA ANALYSIS

5.1 INTRODUCTION

In this chapter we will apply PCA and LDA techniques to a dataset consisted of football players' features. Our task is to classify the 24 football players of a Greek team based on their 36 features. Applying to our data PCA and LDA, we will perform both the reduction and classification tasks, trying to understand which features plays major role in the team's structure. Firstly,we will perform PCA and LDA for all the football players of the team. Our results will be useful for the team since its members(board of directors,coaching staff,fans)will have a clear outlook of the team's level, which is very important for the team's future, thereby influencing major daily tasks such as salaries,transfer policy,etc. In the end, we will again compare and contrast PCA and LDA technique, thus trying to understand which of the two and why worked better.

5.2 DATASET

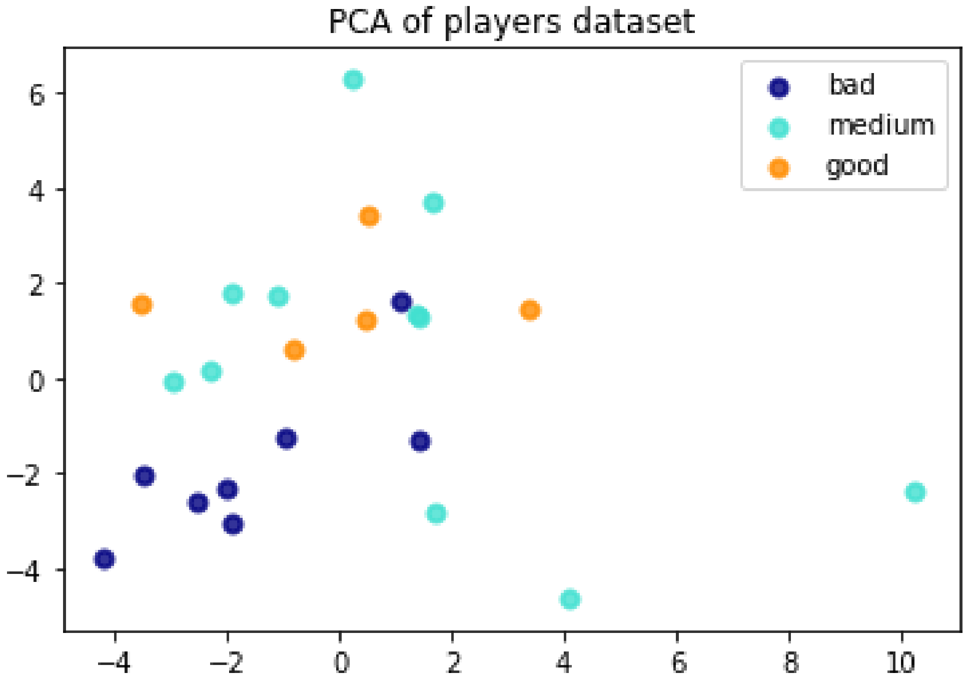
As we said above,our dataset is consisted of 24 samples(footballers)of a Greek team. For each of the 24 football players we have 36 features. It is clear that the dimension of our data matrix is (24x36). For each feature every player has its own grade from 0 to 20. The features come from 3 different categories:technical,mental and physical.Technical abilities consist of the next 14 features: Corner kicks, Crossing, Dribbling, Finishing, First Touch, Free Kick Taking, Heading, Long Shots, Long Throws,Marking, Passing, Penalty Taking, Tackling and Technique. These 14 features are represented in the first 14 columns of our data matrix. Mental abilities consist of the next 14 features(which obviously represent another 14 columns of the data matrix):Aggression, Anticipation, Bravery, Composure, Concentration, Decisions, Determination, Flair, Leadership, Off the ball, Positioning, Teamwork, Vision and Work Rate.As for the physical abilities we have 8 features: Acceleration, Agility, Balance, Jumping Reach, Natural Fitness, Pace,Stamina and Strength.Let's see how our data matrix looks like. We note that the first 9 players are defenders the next 8 players are midfielders and the rest are attackers. For defenders the most important features are: Heading, Marking, Passing, Tackling, Aggresion, Anticipation ,Concentration, Positioning, Jumping Reach. For Midfielders key features are: First Touch, Passing, Technique, Composure, Decisions, Team Work, Work Rate and Balance. Finally attackers should emphasize on: Heading, Penalty Taking, Decisions, Off the Ball, Finishing, Strength. Our target set consists of 3 classes: bad player, medium player and good player. Eight of our players are considered as bad, eleven are considered as medium and the rest(5 players) are considered as good ones. In the next section we will apply PCA and LDA to the whole dataset, trying to visualize it in R^2 and also classify every new player(transfer target) to one of the three categories.

NAME	CORNER	CROSSING	DRIBBLING	TACKLING	FIRST TOUCH	FREE KICK
PLAYER A	7	3	3	7	10	10
PLAYER B	3	4	7	4	11	10
PLAYER C	6	5	5	7	12	6
PLAYER D	9	4	13	8	14	6
PLAYER E	5	7	10	2	10	7
PLAYER F	4	6	12	6	14	9
PLAYER G	6	8	12	7	12	8
PLAYER H	11	11	10	10	6	12
PLAYER I	6	6	7	8	7	9
PLAYER J	10	8	7	7	9	10
PLAYER K	10	8	9	11	10	11
PLAYER L	12	8	8	11	8	13
PLAYER M	6	5	5	3	6	9
PLAYER N	9	7	8	10	12	13
PLAYER O	12	4	9	9	9	7
PLAYER P	12	10	6	12	7	11
PLAYER Q	12	13	8	14	11	11
PLAYER R	8	3	4	8	6	14
PLAYER S	8	9	7	9	7	11
PLAYER T	10	2	4	8	5	6
PLAYER U	9	6	7	9	10	11
PLAYER V	8	13	6	8	12	11
PLAYER W	6	12	6	7	13	10
PLAYER X	7	12	4	5	10	8
PLAYER Y	6	9	6	7	13	8

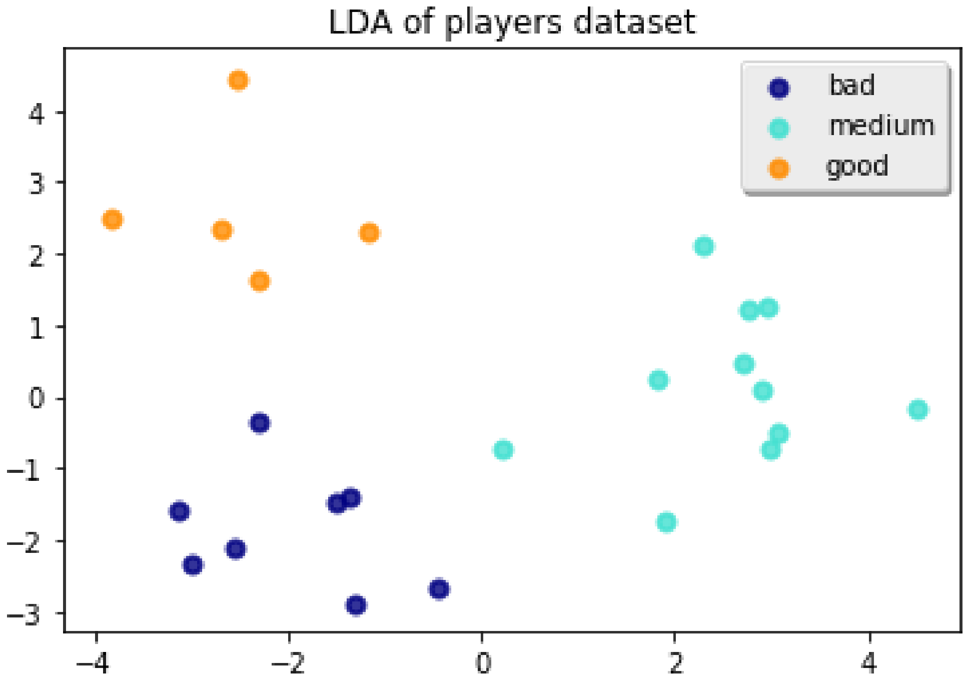
5.3 APPLICATION OF PCA AND LDA

First of all,we will apply PCA and LDA to our dataset as data reduction techniques in order to perform a vizualization of our samples.After that we will use PCA loadings in order to understand which features are the most important(strong features) for this data reduction.Finally we will train a model based on LDA in order to classify new players (transfer targets) into three different classes:”bad player”,”basic player” and ”good player”,so that we can help the team staff to decide if a transfer option will be efficient for the team or no.

As we explained in the theory section,variance is the indicator for the spread of the data.Let’s see how much variance PCA(with two principal components) captures: variance ratio (first two components): [0.29176311 0.18034963].The total variance for the first two principal components is about 0.48,which is a very low amount of variance so we lost much information due to compression. So, PCA cannot achieve an accurate visualization of our data in orhogonal axis.



Now,we will apply LDA to our dataset. In chapter 3,we referred to SSS problem of LDA which comes from the fact that we have much more features than samples. Here we have 24 samples and 36 features but LDA perform greatly as we will see. LDA algorithm captures variance [0.66133583 0.33866417] with its two principal components which means that we keep percentage 99,9 of the information that the original dataset obtained. So we can visualize our observations using LDA.It is very interesting that LDA overcomes the SSS problem in contrast with what theory says.



In my opinion, every work must have some practical results that can be interpreted by common sense. So let's say that I work in this team as a Data Analyst. If the team's President come in my office and ask me what is the progress of my project and I give him the answer:"President, I have applied PCA and LDA for visualization and I showed that LDA keeps higher amount of variance, thereby overcoming the SSS problem.",the most probable way that he will deal with this situation is that he will fire me!!! What practical guidelines can we give to the team's staff? First of all,it is necessary to extract some information about the important features of the players. PCA can give us which are the strong features(the most responsible for data reduction of our dataset. Using the appropriate Python subroutine we see that the top ten features are: Balance, Bravery, teamwork, Aggresion, Agility, off the ball, pace, First Touch, Jumping Reach and Heading. In addition,we will use LDA in order to classify every new transfer target as bad player, medium player or good player. If the accuracy of the algorithm is high it will be an important tool for the transfer policy of the team. We randomly split our dataset into training data(70 percent of the initial) and test data(30 percent of the initial). We train our model and test the predictions on the test set in order to measure the accuracy score. Since we must be as reliable as we can we follow this procedure for 100 times and the accuracy of our algorithm is the mean of all accuracy scores that we had. So,the total accuracy is approximately 71 percent. Not bad if we take into consideration that a trivial algorithm which predicts the majority class would predict

$(11/24) * 100 = 45,83$ percent.

Finally, it would be interesting to know the correlation between some features of the players in order to help the coach with regards to his tactical plan. For instance, if there is a strong correlation between Finishing and Heading the team can play with long balls in order to score. We can perform this task with the use of Pearson correlation coefficient. We will try to find strong or weak correlations which can be very useful for the team's tactical plan. We can either use correlation matrix or more simply examine correlation only between some features that should be correlated. We have the above encouraging results:strong or almost strong correlation between:crossing and finishing(pearson coefficient=0,75),balance and agility(pearson coefficient=0,79),agility and pace(pearson coefficient=0,82) and first touch and positioning(pearson coefficient=0,65). In contrast to the above results, there some weak correlations that may cause problem to the team's tactical plan such as: Corner and Heading(pearson coefficient=0,10), Crossing and Dribbling(pearson coefficient=0,07), passing and teamwork(pearson coefficient=0,05) and agility and stamina(pearson coefficient=0,03). Also,we observed that determination is weakly correlated with most of the features which is a very important problem since determination is a key feature for success. Based on this,we should suggest to the team staff(coach, assistant coach, trainer, etc)some kind of training styles. For instance,since the correlation between corner kicks and heading is low they team faces difficulties in scoring from stationary phases so they have to prepare some training sessions specifically for these features in order to eliminate the problem. Of course we can make many other suggestions but I think that it is not the point of our work.

RESULTS

1. Data Visualization using LDA
2. Important Team's Features (by interpeting the results of PCA loadings)
3. Predictive analysis using LDA
4. Calculation of correlation coefficients between features in order to suggest a training schedule for the team

FUTURE WORK The accuracy of our algorithm for predictive analysis is about 70 percent. It is not bad but I am sure that it could have performed better. The train and split method we used, sometimes called repeated hold out is reliable but I strongly believe that Leave one out cross validation(LOO-CV) could be more efficient in such a dataset since it is consisted of only 25 observations. Apart from that the supervised algorithms that we will introduce in the next part of the Thesis may had performed better than LDA, but as we have already said this Part of the Thesis is about PCA and LDA. Also,we characterized the important features of PCA as important features of our dataset. Practically, PCA is a feature construction technique since we project our data in a lower dimensional space. It is clear that PCA is not considered as feature selection technique. We can use a feature selection techniques like Univariate Feature Selection in order to examine which features are important and compare and contrast the results of this method with the results of PCA. I have done this task and I realized that some features are the same. I believe that it is an interesting result and I would like to investigate it in detail in the future. Is there any correlation between PCA and Feature Selection Ttechniques like Anova Univariate Feature Selection or it happened accidentally in our case?

Chapter 6

ADVANTAGES AND DISADVANTAGES OF PCA AND LDA

Reaching the end of Part 1 it is important to have a clear picture of the prons and cons of our two techniques so that we can compare and contrast them.

ADVANTAGES

1. DATA REDUCTION AND VISUALIZATION: Using both PCA and LDA we achieve dimensionality reduction, through a linear transformation. In a real scenario we have a dataset with very many features, dimensionality reduction can be a crucial preprocessing step in order to visualize our data in a lower dimension subspace.
2. IMPROVES ALGORITHM PERFORMANCE: This practically comes as a second advantage after applying to your data an efficient reduction. If you have a very large dataset with irrelevant features it is clear that the performance of the algorithm is not expected to be the best and also running this algorithm will be a time consuming task.
3. ALGORITHM CONSTRUCTION: As we have said, PCA is an unsupervised technique so the construction of our PCA algorithm is a very easy task and can be performed in all the datasets. LDA construction is a bit more difficult, since we must also take into consideration the class of its sample. If we do not know the class of its sample LDA cannot be applied to the dataset.
4. EASY MATHEMATICAL BACKGROUND: As we saw in the theory section, the main ideas of PCA theory comes from linear algebra, eigenvalues-eigenvectors and

projection to a new subspace, which are considered as simple Mathematical tools. The same holds for LDA, since as we saw in chapter 3 the Mathematical Background is again not so challenging.

DISADVANTAGES

1. INDEPENDENT VARIABLES BECOME LESS INTERPRETABLE: After implementing PCA on the dataset, your original features will turn into Principal Components. Principal Components are the linear combination of your original features. Principal Components are not as readable and interpretable as original features. The same problem holds for LDA we visualize our data to two new axes LD1, LD2 which are not so easily interpretable.
2. INFORMATION LOSS: That is a common problem for all the data reduction techniques and not only PCA and LDA. Obviously we always loose an amount of information but sometimes PCA is very vulnerable to this problem as we saw in the football analysis algorithm.

COMPARISON OF TWO METHODS

The main difference between PCA and LDA is that the first one is an unsupervised technique, whereas the second can be considered both as an unsupervised and as a supervised one. Based on the algorithms that we performed LDA is much more reliable for dimensionality reduction, than PCA since it manages to keep higher amount of variance. Apart from that, LDA can help us perform classification tasks(supervised approach), something that PCA cannot do. Moreover, we saw that practically LDA is not afraid of SSS problem.

As a disadvantage of LDA compared to PCA we can say that the dimensionality reduction is limited. As we saw, the maximum dimension of LDA lower subspace is c-1, where c is the number of classes. So, if we have, for instance, 5 different classes and for each sample, 100 features($x \in R^{100}$) the maximum dimension of LDA lower subspace can be four(5 classes -1). In contrast to this, for this particular case PCA can achieve a dimensionality reduction from R^1 to R^{99}.

Part II

Chapter 7

Some Theory

7.0.1 Decision Trees and Random Forests

Classification Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node[Mitchell].

7.0.2 The Basic ID3 Algorithm

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. This approach is exemplified by the ID3 algorithm (Quinlan 1986) and its successor C4.5 (Quinlan 1993), which form the primary focus of our discussion here. In this section we present the basic algorithm for decision tree learning, corresponding approximately to the ID3 algorithm.Our basic algorithm, ID3, learns decision trees by constructing them topdown, beginning with the question ”which attribute should be tested at the root of the tree?’To answer this question, each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node(i.e., down the branch corresponding to the example’s value for this attribute).The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree.[Mitchel]

In order to find which attribute is the bast classifier we will use a statistical property, called Information Gain, which measures how well a given attribute separates the training examples according to their target classification. We will use the Information Gain property in order to select the best attribute at each step, while growing the the tree.
Definition:ENTROPY Given a collection S, containing positive and negative examples of some target concept, the Entropy of S relative to this boolean classification is :

$$Entropy(S) = -\sum_{i=1}^{k} P(y = y_i)log_2(P(Y = y_i)) \tag{7.1}$$

The number k corresponds to the number of classes and $P(y = y_i)$, is the probability of each class.

Information Gain Formula(IG):Supposing that we have a data collection, let's say S, and some features for each sample, let's say X,Y. We calculate:

$$IG(X) = H(S) - H(S/X) \tag{7.2}$$

$$IG(Y) = H(S) - H(S/Y) \tag{7.3}$$

where H(S) is calculated as before and

$$H(S/X) = -\sum_{i=1}^{k} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i/X = x_j)log_2(P(Y = y_i/X = x_j)) \tag{7.4}$$

In order to decide which feature we use for split we choose the feature that gives us a larger information gain. For instance after calculating (6.2) and (6.3), if $IG(X) > IG(Y)$, we will choose feature X.

The procedure of selecting a new feature and partitioning the training set is repeated for each nonterminal descendant node, thereby using at this time only the training examples that are associated with this node. Any given feature can appear no more than one times along any tree path. This procedure will be repeated until either one of the next two conditions hold:

(a) Every feature has already been selected along this path of the tree

(b) The entropy of training examples associated with this leaf node is zero. To keep it simple, we can understand that there is no meaning in splitting a node if all matching records have the same output value.

REMARK: What should we know about Decision Trees

(a) easy to understand and implement
(b) easy to use
(c) computationally cheap
(d) Information Gain Measure
(e) Can be used both for regression and classification
(f) Vulnerable to overfitting
(g) We must find ways to keep the tree simple(early stopping,fix depth, pruning)

7.1 Random Forests

Based on what we said above, we can easily understand that it is difficult to have an accurate classification task with only one tree. So, why not build a large amount of trees? Now, we can talk about Random Forests which are an ensemble learning method that operate by constructing many decision trees. We use our training set to build some trees and after that we classify our test data based on what the "trees vote". Random Forests method correct the main problem of decision trees, which is overfitting. Let's see what is the procedure of constructing a random forest.

Step 1: Create a bootstrapped dataset
Step 2: Create a decision tree using the Bootstrapped Dataset, but only use a random subset of variables(Hyperparameter) for each split of the tree. For the selected variables,entropy measure can be used in order to select the most accurate variable for the split.
STEP 3: Repeat this procedure with regards to the number of trees you want to construct. The number of trees is also a hyperparameter and a usual default value is 100.
STEP 4: Classification of a new sample. We will classify the sample in each of the trees that we have created. The most frequent predicted class is the final class that the Random Forest predicts(the trees "vote", as we said before). Remark:Why bootstrapped subsets? We know that Decision Trees work very well if they are on small depth. However in real world big data this may not be the case, since we may need to construct a tree of large depth. However, large depth trees are prone to overfitting due to the high variance of the model. This disadvantage of Decision Trees is explored by the Random Forest Model.In the Random Forest algorithm we randomly generate with replacement subsets of samples, which are known as bootstrap samples. After that we build trees for all the bootstrap samples we have created. Each of these Decision Trees is trained separately on these bootstrap samples.This aggregation of Decision Trees is called the Random Forest ensemble and as we have said before the concluding result of the ensemble model(Random Forests) is determined by counting a majority vote from all the Decision Trees. Based on the outlook that each Decision Tree takes a different set of training data

as input, the deviations in the original training dataset do not impact the final result obtained from the aggregation of Decision Trees. As a result, **Bootstrapping as a concept reduces the variance without changing the bias of the Random Forest classifier.** Evaluation of Random Forests: Out of Bag Error
A case in training set is not in the bootstrap sample for about one third of the trees. When we have a large dataset the out of bag samples will be more, thus creating an out of bag dataset. As a first step, we can run each sample of the OOB Dataset through all the decision trees that do not contain the specific sample. After that we estimate how accurate our Random Forest is by the proportion of the out-of-bag samples that were correctly classified by the Random Forest. So, the oob error rate is the error rateof the RF predictor.
TIP: Random Forests does not overfit as we fit more trees.

Gini Index against Information Gain:
In the last subsection we referred to Entropy and Information Gain as a criterion for splitting a tree node. Another criterion of splitting is the famous Gini index, so let's see how it works.Gini index measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. The value of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. A Gini Index of 0.5 denotes equally distributed elements into some classes.

$$Gini = 1 - \sum_{i=1}^{k}(P(Y = y_i))^2 \tag{7.5}$$

,where $P(Y = y_i)$ is the probability of a sample being classified to a particular class. When we build a decision tree, we prefer choosing the feature that has the lowest Gini Index value.
REMARK: Based on the above analysis we can understand that Decision Trees and Random Forests classifiers have a variety of Hyperparameters(max_depth, number of features considered for splitting, splitting criterion,min number of leaf nodes ,number of trees for RF,etc). Using the work Hyperparameters we refer to parameters, which the value is given by the user. It is clear that we need strategies in order to tune our Hyperparameters and that's why we will use a Machine Learning method for Hyperparameter Tuning called GridSearch.

7.2 K-Nearest Neighbor

K-NN is a basic instance-based Machine Learning algorithm, that can be used for both regression and classification. Here, we will use K-NN as a supervised technique for classification, but firstly we will explain how it works. We assume that:

all the samples $x_i \in R^n$ and also the Nearest Neighbors of the sample are defined by the Euclidean Distance. More specifically, we represent a sample as a feature vector:

$$x = (a_1(x), a_2(x), ..., a_n(x)) \tag{7.6}$$

,where a_i represents the value of the i-th feature of the sample x.
We measure the distance between two points x_i, x_j as:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{7.7}$$

Now we define our target function as: $F : R^n toV$, where V is the target set. If we choose $k = 1$, then the 1-Nearest Neighbor algorithm gives the value $f(x_i)$ where x_i is the training instance nearest tox_q. For larger values of k, the algorithm assigns the most common value among the k nearest training examples.Practically K-NN algorithm can be described as:
STEP 1: Splitting our samples into training and test in order to train our classifier.
STEP 2:For every test sample x_j, we calculate the distance between this and k-training samples, using (6.7). We will classify the new sample with regards to the most common class of the k-nearest neighbors.
Remark: It is obvious that the number of neighbors k is a hyperparameter. Again, using Tuning is important in order to find the optimal value of k. In general, when we have a large dataset we choose a large value of k, whereas when we have a small dataset we choose a small one. We highlight that a large value of k is less sensitive to noise.

Let's try to understand how the K-NN algorithm works using a simple graph. In the above graph we have four points(B,D,C,E,F) that correspond to two different classes, the red class and the black class. More specifically,samples B,D are in the red and the other samples in black class. We want to classify the test sample x_test either in red or black class. We can observe that using, for instance, 1-NN the test sample will be classified as red class, whereas if we use a 5-NN it will be classified as black class.

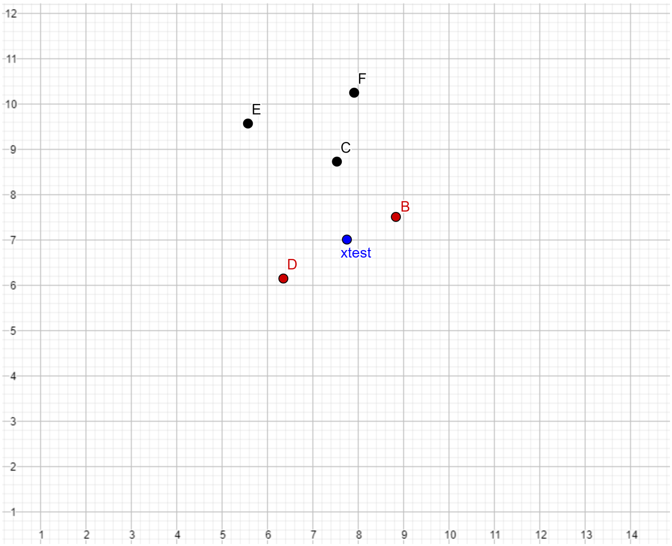


Figure 7.1: K-NN Plot

Major Disadvantage of K-NN: When we apply K-NN the distance, contrary to Decision Trees and Random Forests, between samples is calculated using all the features of the sample. So, if we have a large amount of features but only few of them are relevant to the classification task, K-NN will take into consideration all the features, something that may lead to an inaccurate classification.

7.3 Automated Machine Learning

In this section we will analyze different ways of training our algorithms like Hold-out Method, Repeated Hold-Out Method and some types of K-Fold Cross Validation. In the first part of this work we had used the first two methods, but now we will explain in detail how they work in order to understand the prons and cons of all of them. We will also introduce the GridSearch Method for Hyperparameters Tuning, which is very useful for Trees and K-NN as we had highlighted before.

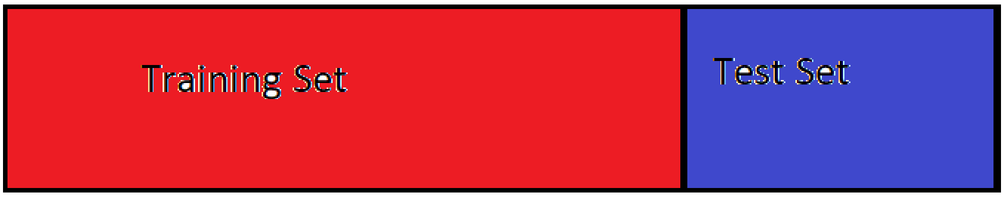


Figure 7.2: Typical Split of Hold-Out Method

7.3.1 (Repeated)Hold-out Method

Definition: Hold out Method is a really simple and computationally efficient method of training our algorithms.The basic disadvantage is that some data are ”lost” to estimation. Typical Splits are: 66,75,80 percent for training set. There is no method for choosing explicitly the size of the training and test sets.
STEP 1: Randomly partition original data in terms of samples STEP 2:Learn on Train Set
STEP 3:Estimate performance on Test Set
The fact that we both choose randomly the size of training and test set and also choose and also partition randomly the samples in training and set is an obvious problem since the results are dependent of our ”guess”. Can we do something better?
Definition: Repeated Hold Out Method
Based on the terminology we used, it is not difficult to understand what this method does. We repeat the procedure of Hold-Out Method and in the end we average out the performance we got for each repetition. As a result we reduce the uncertainty of the estimation and the result is more reliable. However, this method is much more computationally expensive than the simple Hold-Out Method.

Cross Validation

K-fold Cross Validation is the most popular method in order to estimate the skill of a Machine Learning algorithm on unseen data.Indeed, using a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. To be more specific, we split our data into K- folds, where all of our data are used both for training and testing. The number of K is arbitrary but the most common values are 3,5 and 10. It is a conservative and extremely reliable approach,since we use all of our data both for training and testing.If we choose K=N, where N is the number of observations we have, then the method is called Leave one out cross validation(LOO-CV). The basic disadvantage of K-fold CV is that the partition to folds is specific and of course a different partition may change the results.

Based on this outlook, we introduce the idea of Repeated K-fold CV, where we repeat CV with many partitions to folds and get the average. We can use as many repetitions as possible and it is an extremely efficient method for small sample sizes. K-fold Cross Validation Algorithm:
STEP 1: Split the dataset randomly into k groups
STEP 2: For each unique group:

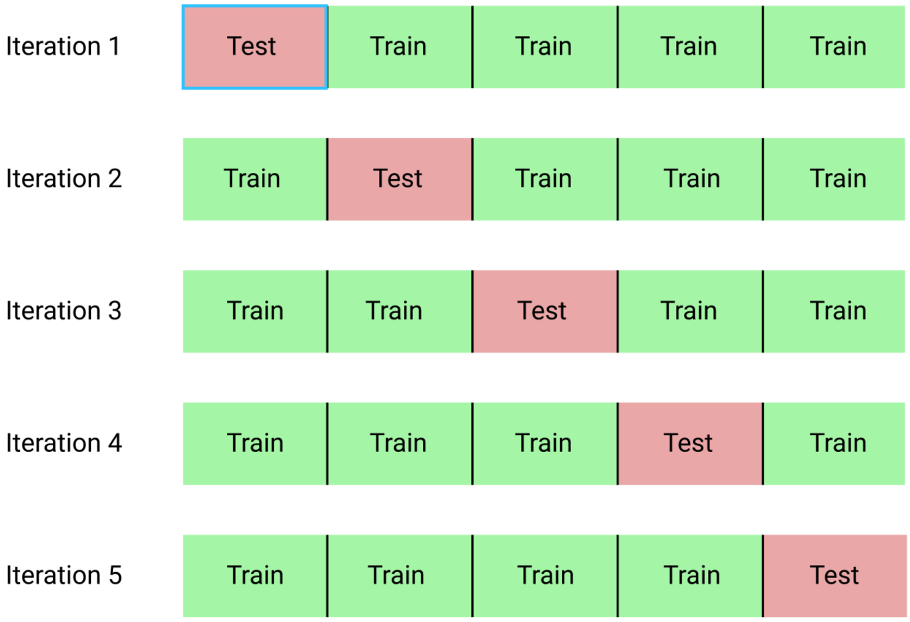
(a) Take the group either as a hold out or test data set

(b) Take the remaining groups as a training data set

(c) Fit a model on the training set and evaluate it on the test set

(d) Calculate the accuracy score for each iteration

STEP 3:Summarize the accuracy of the model calculating the mean of the different accuracy scores



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure 7.3: 5- Fold Cross Validation

Stratified K-Fold Cross Validation
The above technique is identical when some classes are rare.Again, we randomly split to folds while **maintaining the distribution of the classes as close as possible to the one in the full dataset**. All folds should have at least one sample from the rarest class so the maximum value for K is the number of samples of the rarest class. Also, when sample size is small and computational time is no issue it worthwhile using Stratified Repeated K-Fold Cross Validation with K= samples of rarest class.

7.3.2 Hyperparameters Tuning with GridSearch Method

As we have already said Hyperparameters Tuning is a very important procedure for our models. It helps us optimize the performance of our algorithms, thus finding the optimal value of their Hyperparameters. How many trees should a random forest have? What is the optimal depth of these trees? How many features should we use for each split? What is the optimal value of neighbours for K-NN? If we manage to find these optimal values, we will improve the performance of our model so it goes without saying that this prcedure is very important.For this purpose, we will introduce the GridSearch method. It is a simple and easy to implement method for Hyperparameters Tuning.After giving a useful definition, we will understand how it works.
Definition: Configuration is called an instatiation of a learning method f with specific hyperparameter values.
GridSearch Method
STEP 1: Apriori decide which values to try for each hyperparameter. STEP 2: Try all combinations(full-factorial)
Disadvantage: Static hyperparameters search strategy since we predetermine the configurations to try.

Chapter 8

MACHINE LEARNING IN EDUCATIONAL DATA

8.1 Introduction

All we know that Crete is an island and the cost of living is higher compared to the rest of Greece. It is obvious that most students are reluctant to come here for their studies and that is why the enrolment grade in our department has dropped significantly during the last few years. For instance my enrolment grade in 2012 was 17,000 and now the last student can be enrolled with 9,000. But is this such an important problem?Is there any strong correlation between the enrolment grade and the graduation of the student? We will answer this questions and analyze in detail what is happening using Statistical Tools and Machine Learning Techniques. Until now we have used PCA and LDA but for this task since it is very important for us we will try more methods(supervised),like Decision Trees,Random Forests, K-nearest Neighbourhoods. We will also check the performance metrics of our algorithms using either ROC Curves or Precision-Recall Curves.

First of all,we will work with two large datasets one for the Department of Mathematics and one for the Department of Applied Mathematics which are consisted of all the students that were enrolled in our departments since 2009. Each observation(students) has some features like:Enrolment Grade,Way of Enrolment,Gender,success series in the exams,semester,University grade and current status(graduated,dropped off,active). We will work separately for the two datasets since typically the departments are different. To be more specific the feature semester shows the current semester if the student is active, the semester of graduation if the student graduated or the semester that the student was dropped off .The feature:Success Series could be written Exams' Success Series. For instance the student with the higher enrolment grade has Success Series equal to 1,the second equal to 2,etc. Obviously it is correlated with the Enrolment grade. After preprocessing our data we observe that there are many different ways in order to be enrolled in a Greek university except for Greek Enrolment Exams that we are interested in. 19 percent of our samples come from different enrolment categories like Excellent Athletes(Greek Champions), Cyprus students, Students with Health Problems, Older Age students and others.All these observations are dropped off since they are enrolled with no exam or completely different exam(much easier that the majority of students).

8.2 Department of Mathematics

Now we will work with a dataset consisted of 2269 students(samples), who were enrolled in the Department of Mathematics from the Fall Semester of 2009 until now. First of all, we highlight again that there are too many ways(20) of enrolment. The most common category is of course the "Greek Enrolment Exams", in which the 80 percent of our samples belong to(1816 students). The second most common category is the "Second Time Greek Enrolment Exams(10 percent)". If a student fails the exam the first time can have a second chance the next year but only the 10 percentage of these students are enrolled in Greek universities. During the last ten years 169 students came from this category, which is a smaller but still significant amount of students(around 7 percent). Let's see a bar plot about the different students' enrolment categories. We will observe that the category "others" is consisted of many samples. Practically, we gathered many different categories(Health Reasons,Greek expatriates, Athletes,polytechnic families,Social Issues) on this category.

Extracting the samples that were enrolled in the Department in a different way than Greek Enrolment Exam we have 1816 students. We know that 895 of them are totally active, 440 were dropped off, 397 graduated and 84 are in discontinuation of studies. From the number of the dropped off students we observe that 193 of them were dropped off either in the first or the second semester of their studies. This often happened from 2013-2015 since "students' transfers" were permitted and a student that, for instance had been enrolled here at first, could continue his studies in another Department of Mathematics in Greece. Most of these 284 students continued in Athens for economical reasons, because Athens is less expensive than Herakleion. It is clear that these students typically are considered as dropped off but practically they are not since they continued their studies in the same field in a different Greek University. Based on this outlook we will extract theses samples from our dataset. We can have a look at the above bar plots in order to have a better outlook of what we discussed before.

To sum up, the dataset that we will analyze is consisted of 397 graduated students, 247 dropped off and another 895 who are still active and we will try to predict if they will graduate or not.

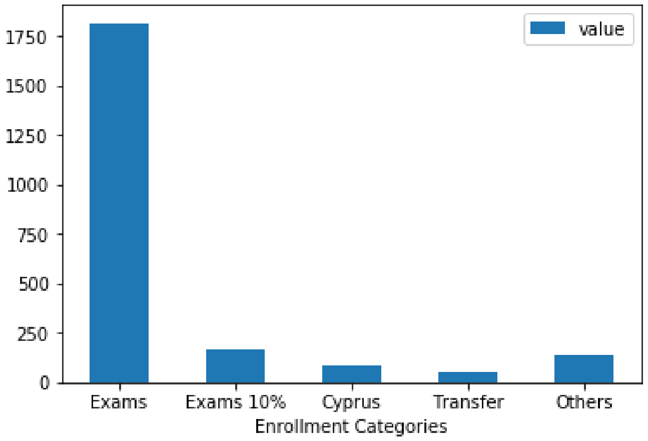


Figure 8.1: Different Enrolment Categories

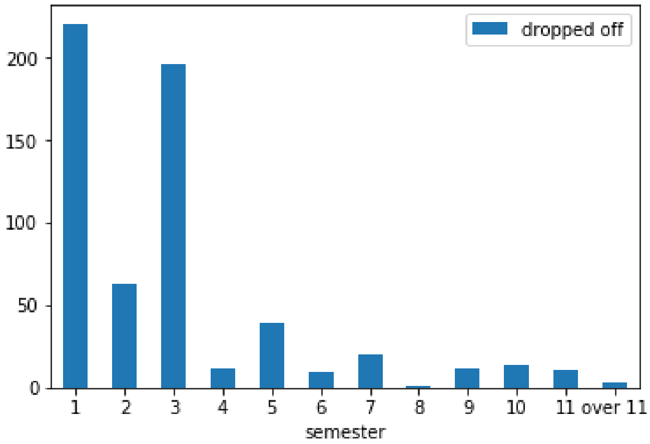


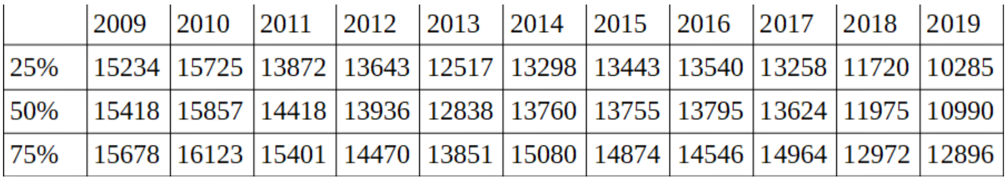
Figure 8.2: dropped off students per semester

8.3 Statistical Analysis

As we have already said in the previous section we will analyze only the behaviour of students(1816 samples), that were enrolled in the Department after succeeding in the "Greek Enrolment Exams". To begin with, 397 of them managed to graduate, 247 were dropped off and 896 are still active. First of all, we will extract the active students, actually our idea is to predict if they are going to graduate or no (see next section) and analyze the rest of dataset, i.e graduated and dropped off students. Our dataset is consisted of 644 samples and for each sample we have the next features:"Gender, "Greek Enrolment Grade"," Success series","Semesters" and for the graduated students we also know their average graduation grade. Our features are not many but are enough in order to have an outlook of a new enrolled student's performance. It is clear that out target is binary, let's say 0 represents dropped off students and 1 represents graduated students. Our plan is to write down some simple descriptive statistics, examine correlation coefficients and also apply ANOVA test to compare Men and Women students.

8.3.1 Descriptive Statistics

Graduated Students:
As for Gender: 157 Men 240 women
As for Enrolment Grade: Average 15.359
As for Graduation Grade: Average 6,74
Men Average Grade: 6,80 , Women Average Grade: 6,70
The average number of semesters men needed in order to graduate is: 11,6 semesters, whereas women needed 11 semesters. Dropped off students: As for Gender: 103 Men, 144 Women As for Enrolment Grade: Average 14.152 As for semesters: Around 60 percent of them were dropped of at the end of third semester.
It is very disappointing that the enrolment grade every year becomes lower and lower. We can take into consideration the above simple matrix which shows the highest enrolment grade limit for the 25, 50 and 75 percent of the students.



	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
25%	15234	15725	13872	13643	12517	13298	13443	13540	13258	11720	10285
50%	15418	15857	14418	13936	12838	13760	13755	13795	13624	11975	10990
75%	15678	16123	15401	14470	13851	15080	14874	14546	14964	12972	12896

8.3.2 Anova Test

Anova test is an important statistical tool that can help us explore the variability of two populations among 1 (one way anova)feature,2 features(2 way anova)or M features(M way anova). Here our populations are consisted of men and women that graduated and we will try to use one way anova according to their average grade. Let's formulate the Anova Test Hypothesis:
H0: Their grade has no significance difference($m_1 = m_2$)

H1:Alternative Hypothesis, the opposite holds Using the appropriate Python subroutine we get a $p_value = 0.18$ which is much larger that a=0.05(significance level), so we can reject Ho. So, there is no significant statistical difference between the average graduation grade of women and men students.

Now let's try another 1-Way Anova. The populations are the same as before but we will examine if there is a significant difference according to the number of semesters they needed to graduate. H0 and H1 are as before. We use again the 1-way Anova test and we get a $p_value = 0,01 < 0,05 = a$ so now we reject the null hypothesis. We can interpret this result as: There is a significance statistical difference between men and women according to the semesters needed to graduate. Taking into consideration that women needed an average of 11 semesters to graduate, whereas men needed 11,6 semesters we can characterize this difference as an important one and conclude that women are better than men with regards to the number of semesters needed for graduation. So, for sure women students are more diligent than men.

8.3.3 Correlation

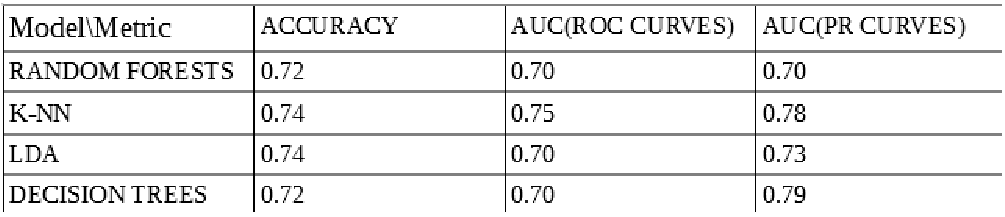
Now we will use Pearson Correlation Coefficient in order to calculate correlations between features or between features and target set. We will begin with our basic set of data, i.e the dataset consisted of both graduated and dropped off students with their features and target. We can see the correlation coefficients above:
Enrolment Grade and Target =0.38
Success Series and Target =0.13
Gender and Target =0.02
Now let's take into consideration only the students that graduated. We can see the correlation Coefficients above:
Enrolment Grade and Graduation Grade: 0.45
Enrolment Grade and Semesters needed for graduation: -0.09
Graduation Grade and Semesters needed for graduation: -0.57
REMARK: The second coefficient with regards to the graduated students is very important for the university community. The enrolment grade is uncorrelated with the semesters needed for graduation something that means that even a student that graduates from school with a low grade, thereby beginning his/her university career with knowledge gaps can graduate from our Department. It shows the high academic level of our Department which helps low grade students to overcome their difficulties. Although, graduation grade is correlated both with semesters needed for graduation and enrolment grade. It is clear and sounds logical that a student with a high grade will also complete his Degree in less semesters than a student with a lower grade and also graduate with a higher grade.

8.4 Predictive Analysis

As we have already said before we have the intention of predicting how many of the active students will manage to graduate. It is clear that supervised ML algorithms is the basic tool for the implementation of this task. We will use Decision Trees, LDA, K-NN and Random Forests. We will also use Performance Metrics like the most typical ROC Curves and also the Precision-Recall Curves since our dataset is imbalanced. It is interesting to examine if the two different type of Performance Metrics will give us something different. As we have already said we will train our models with a dataset consisted of 644 samples from which 397 represent class 1 and 247 represent class 0. As for the features that we will take into consideration: Gender, Semester, Enrolment Grade and Success Series. But is this obvious approach correct?

It took me a long time to understand that this was the problem of my algorithm. Using all these 4 features I got accuracy results around 95 percent. Also the prediction outcome was that half of the students will graduate and half of them will not graduate. It seems to be logical but there is the next very important problem. If we take into consideration the semesters it is clear that from semester 1 until semester 7 all the students, that we know if they have graduated or not, are dropped off. So our models predict that all the active students from semester 1 until semester 7 will be dropped off, something that is completely unrealistic. Here, the reason for overfiting is the very bad problem formulation, that we had done.

So, it is clearly understandable that we will extract the feature: ”semesters”, since it leads to unrealistic predictions. We will work with the other three features and based on these we will try to make some predictions. We will train our model with a 100 times Repeated Holdout Method with a split between training and test set to be 70-30 percent. Using GridSearch Method we found that the optimal depth for Decision Trees and Random Forests is 3, the perfect number of trees for Random Forest Classifier is 150 and the perfect number of neighbors for K-NN is 5. We will also plot ROC Curves and Precision-Recall Curves and calculate the Area Under the Curve(AUC) for both of them. Let’s interpret our results. As for the accuracy it is easily interpretable, since it shows how many of our classifications are correct divided by the overall amount of classifications. For sure, it is an important metric in order to evaluate the performance of a classifier and also it is the most common. If we have an imbalanced dataset like here it is worthwhile to take into consideration other measures. ROC Curves are the most classical one and here is also the most appropriate(we will simply evaluate the results of many confusion matrices for different thresholds), whereas PR curves are sometimes ideal for imbalanced datasets but in our case, based on he outlook that the most common result is the True Positive the Precision measure will give us an optimistic result.($Precision = TP/(TP + FP)$)[Reference m]. So, based on the following table, we conclude that the K-NN classifier is the most reliable and we will analyze its predictions.



Model\Metric	ACCURACY	AUC(ROC CURVES)	AUC(PR CURVES)
RANDOM FORESTS	0.72	0.70	0.70
K-NN	0.74	0.75	0.78
LDA	0.74	0.70	0.73
DECISION TREES	0.72	0.70	0.79

Figure 8.3: Accuracy and Performance Metrics

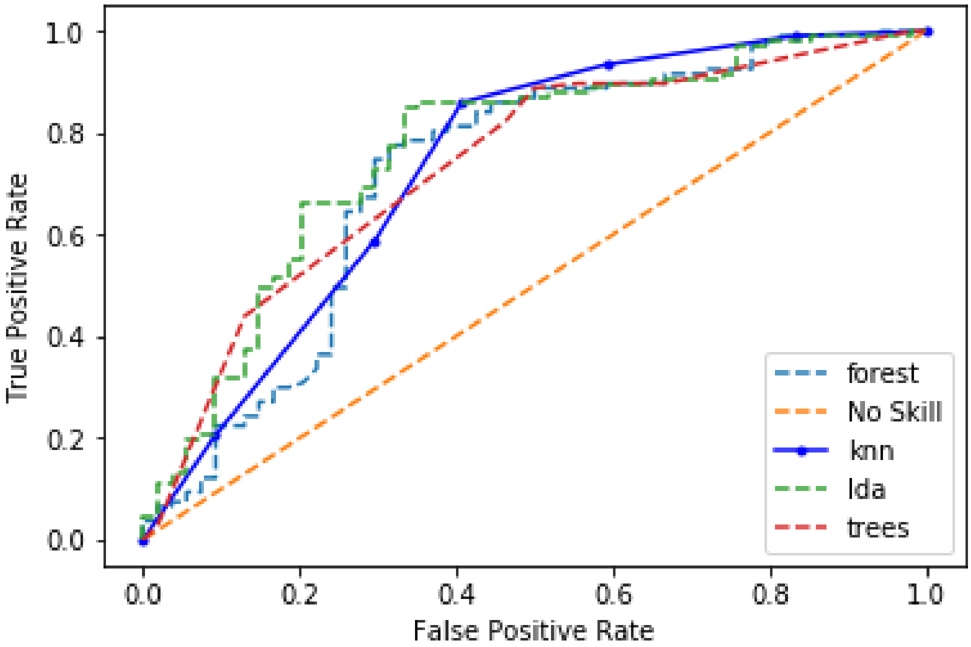


Figure 8.4: ROC CURVES

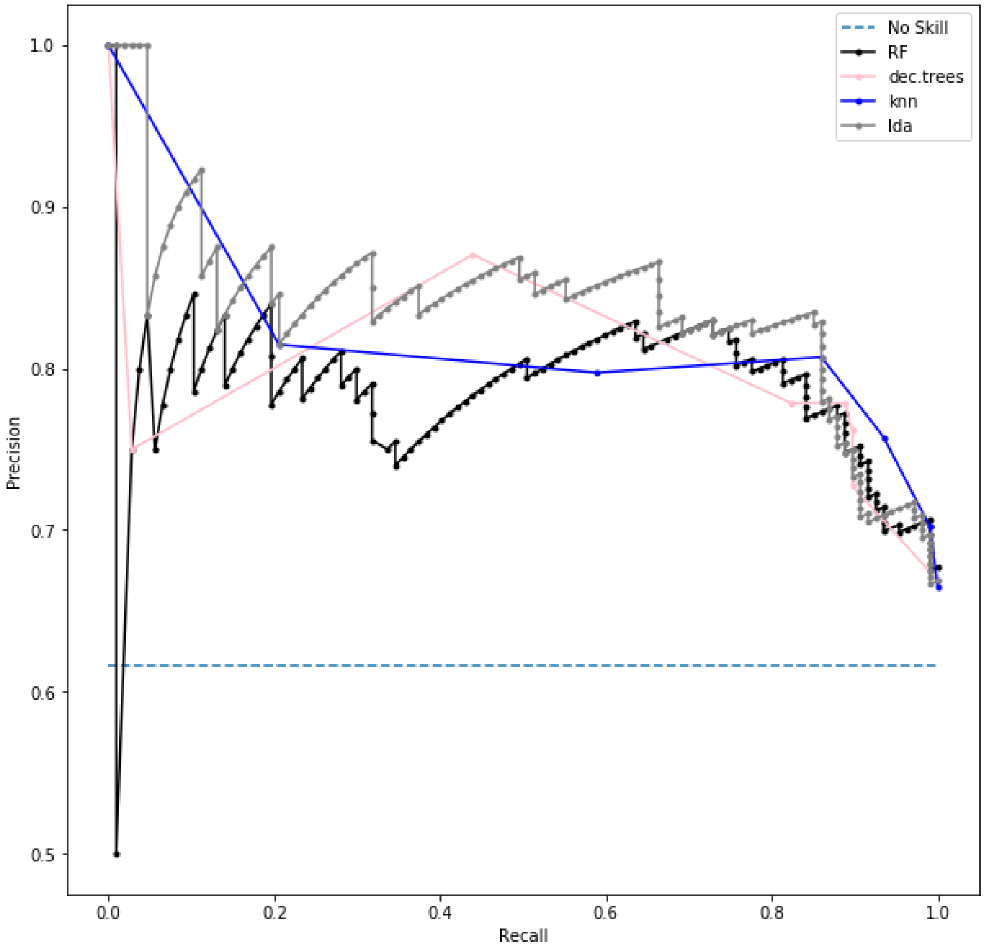


Figure 8.5: Precision-Recall Curves

REMARK: It is obvious that our models are better than a trivial classifier that always predicts the majority class.

As we said before, we will chose the K-NN classifier in order to predict how many of the 895 students will manage to graduate or not. We point out that K-NN(and also all the other 3 algorithms) is a little optimistic, thus having a tendency to predict the class 1 since the majority class of our training data was the positive one. Moreover, we will also examine the performance of the new students that were enrolled in the Department the last two seasons in order to compare and contrast their performance with the old ones' performance. Let's summarize our results:
Older students: 486 will graduate and 161 will not(around 75 percent will graduate)
New students(2018-2019,2019-2020): 163 will graduate and 85 will not.**(around**

65 percent will graduate) Totally, 649 will graduate and 246 will not(around 72 percent will graduate). Again, we must point out that our results are a little optimistic, but we can conclude that the performance of the new students will be for sure worse than the old ones.
REMARK: Somebody may ask us why we did not train our models with a Stratified Cross Validation. We have highlighted that the classes are imbalanced, so based on theory a repeated Stratified K-fold CV makes sense. Indeed, I tried this training method and the results were approximately the same. Based on the outlook that repeated CV and all the familiar to this training methods have a high computational cost, I think that it was not worthwhile to train our models with one of these. Maybe, Repeated Stratified CV could give us a much more reliable result in a more imbalanced situation. Here, with an around 60-40 imbalance between the two classes the simplest and cheapest Repeated Holdout Method is also reliable. Exactly, the same holds for the Applied Mathematics training dataset that we will examine in the next sessions.

8.5 Department of Applied Mathematics

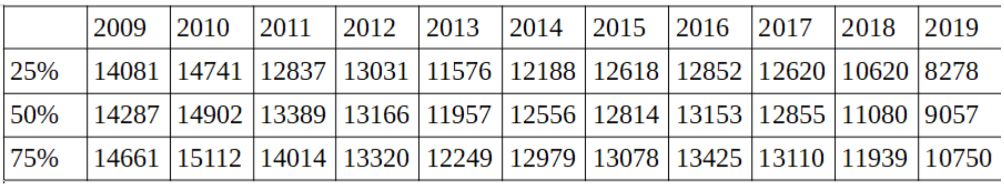
Now, we are going to perform the same tasks for the Applied Mathematics Department. The features of our dataset are the same but there are some other differences. First of all,the total number of enrolled students from 2009 until 2019 are 1795 and 1507 of them have been enrolled with the classical way of Greek Enrolment Exams. As before there is a percentage of students, a little smaller this time, that were enrolled in the Department with a different way. Also, there is no other Department of Applied Mathematics in Greece so there are no ”students' transfers” in this Department. We will only extract the students that were dropped of during their first year(semesters 1 and 2), thereby realizing immediately that they do not like this academic field. It would be boring to make the bar plots as before since there is not an important difference. As for the datasets the training one is consisted of 587 samples: 251 students graduated and 336 were dropped off. Moreover, we have another dataset consisted of 887 active students. It is clear that we have about the same number of active students in this Department, whereas the number of enrolled students is less than before, which means that the performance of students who study in the Applied Mathematics Department is worse than the Mathematics one.

8.6 Statistical Analysis

As before we will try to give some descriptive statistics, correlation coefficients and apply to our dataset an one way Anova test for the populations of male and female students.

8.6.1 Descriptive Statistics

Graduated Students:
As for Gender: 112 Men 139 women
As for Enrolment Grade(graduated students): Average 14098, men average: 14192, women average: 14022 As for Graduation Grade: Average 6,60
Men Average Grade: 6,64 , Women Average Grade: 6,57
The average number of semesters men needed in order to graduate is 12,3 semesters, whereas women needed 11,4 semesters. Dropped off students:
As for Gender: 187 Men, 149 Women
As for Enrolment Grade: Average 13084
As for semesters: Around 75 percent of them were dropped of at the end of third semester.
It is again very disappointing that the enrolment grade every year becomes lower and lower. Let's take a look at the same matrix that we used before.



	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
25%	14081	14741	12837	13031	11576	12188	12618	12852	12620	10620	8278
50%	14287	14902	13389	13166	11957	12556	12814	13153	12855	11080	9057
75%	14661	15112	14014	13320	12249	12979	13078	13425	13110	11939	10750

8.6.2 Anova Test

Again as before we will try to interpret if the difference that we observed before between the average semesters women and men needed in order to graduate can be considered as significant.Using one way test for the populations of men and women and formulating the H0 and H1 as before we get a very small $p_value = 0,0006$ which means that we can reject H0 even for $a = 0,001$. So, the difference that we observed with regards to the semesters that male and female students needed for graduation is statistically significant and we conclude again that female students are more diligent than the male ones.

Trying the same for the average grade we conclude that there is not important difference with regards to this feature($p_value = 0,42$ and we cannot reject H0).

Remark: It is interesting that women managed to perform better than men, even though they were enrolled with a lower average enrolment grade than men.

8.6.3 Correlations

Now we will try to calculate correlation coefficients as before. We will start with the dataset of students that have graduated(251) or dropped off(336).
Enrolment grade and Target= 0.36
Success series and Target= 0.11
Gender and Target= 0.11

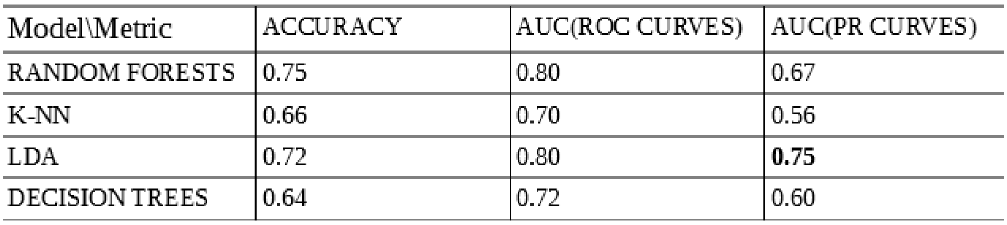
Remark: The strongest feature is the Enrolment Grade but hopefully, for the university, there is not a very strong correlation.

As for only the graduated students: Enrolment grade and Graduation grade = 0.36 **Enrolment grade and Semesters needed for graduation = -0.48**
Graduation grade and Semesters = -0.46

Remark: Comparing and contrasting the correlation coefficients that we calculated for the two Departments we observe that there is an important difference between the correlation of enrolment grade and semesters needed for graduation. In Mathematics Department the same coefficient is 0.09(practically uncorrelated), whereas in the Applied Mathematics equals to 0.48(correlated). The same professors teach at both Departments so they are not responsible for this situation. It is clear both form this and also the descriptive statistics that we have seen before, that the students' level in Applied Mathematics Department is lower that the Mathematics one. The high correlation between the enrolment grade and semesters show that the students who have some knowledge gaps does neither try nor manage to be improved. That coincides with the fact that we have also mentioned above. Although, in the Department of Mathematics have been enrolled more students than in the Applied Mathematics Department the number of active students is almost equal.

8.7 Predictive Analysis

The problem formulation will be as before. We will train our model, using an 100 times Repeated Hold-out method(split for training-test:70-30 percent)and apply the same 4 supervised ML algorithms to the Applied Mathematics Dataset consisted of 251 graduated students and 356 dropped off students. We will try to predict if the 887 active students will graduate or not. Again, the features that we have in order to predict the target are Gender, Success Series and Enrolment Grade. We will again use accuracy score, ROC Curves and Precision-Recall Curves in order to evaluate our results. We have to pint out that our dataset is again imbalanced(251 samples for class 1 and 336 for class 0). It is clear that we are interested in predicting the class 1 but our algorithms will have the tendency to predict class 0. That is why we are going to take into consideration the Precision Recall Curves and the Area under of them as evaluation metric. We point out that that both Precision and Recall does not take into consideration the True Negatives class which here is the most common class. So, in such cases with less positive class samples Area under the PR curves is the most reliable metric of evaluation[Reference n]. Based on the above, we will choose the LDA classifier for our predictions. The following table summarizes our results. We will also plot the ROC curves and PR curves.



Model\Metric	ACCURACY	AUC(ROC CURVES)	AUC(PR CURVES)
RANDOM FORESTS	0.75	0.80	0.67
K-NN	0.66	0.70	0.56
LDA	0.72	0.80	**0.75**
DECISION TREES	0.64	0.72	0.60

Figure 8.6: Metrics for Applied Mathematics Dataset

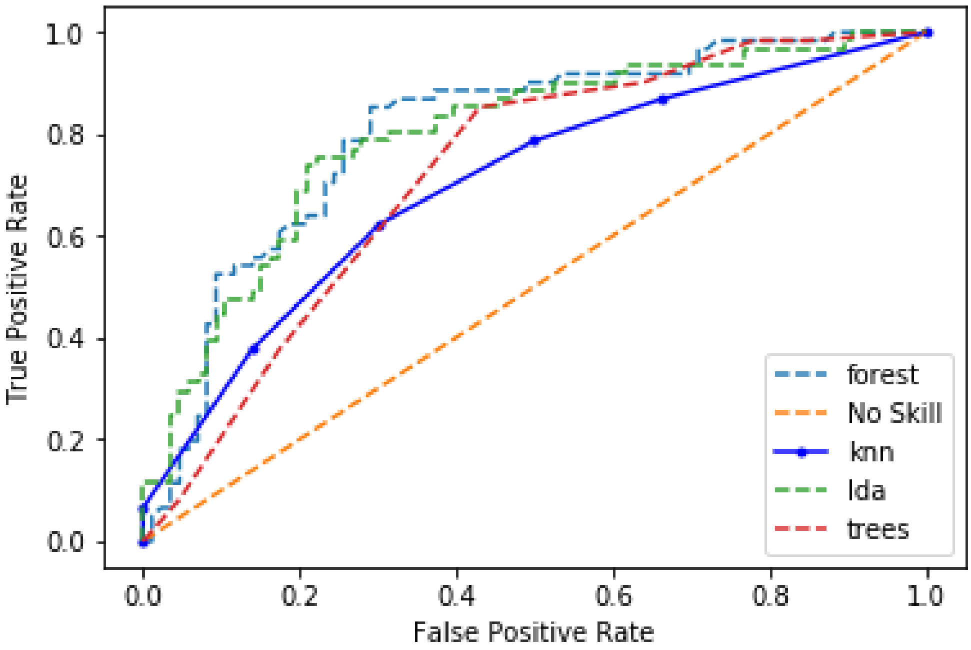


Figure 8.7: ROC Curves for Applied Mathematics Dataset

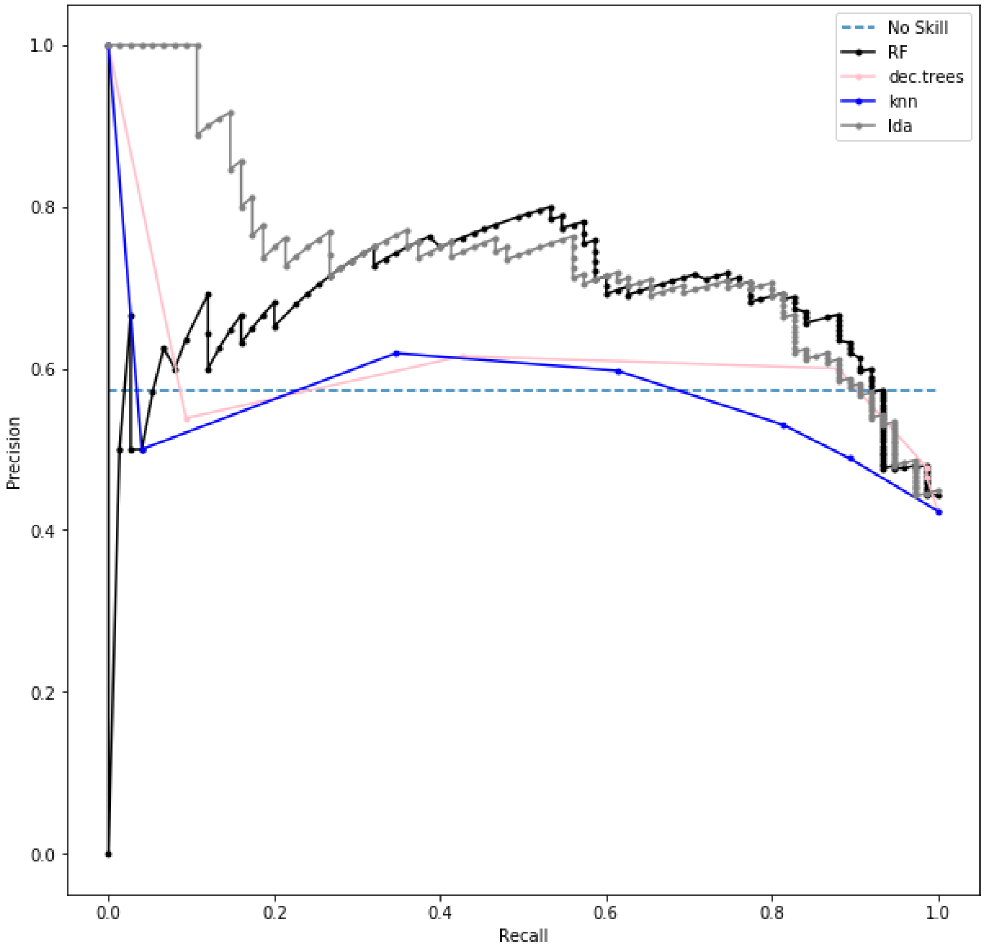


Figure 8.8: PR Curves for Applied Mathematics Dataset

So, for this task LDA classifier is the most reliable and we will analyze its predictions. We recall that we want to predict how many of the 887 active students will graduate. We will also examine the performance of the weakest students that were enrolled in the Department the last season. Let's see our results.
All the active students except for the new ones:
499 will not graduate and 242 will graduate(around 67 percent will not graduate)
New students(2019-2020): 107 will not graduate and 37 will graduate**(around 75 percent will not graduate)**
Totally, 606 will not graduate and only 279 will graduate(around 68 percent failure).
Remark: Analyzing the Features of the 37 new enrolled students that are expected

to complete their Degree we have to highlight that only 10 of them belong to the low grade category(Enrolment garde under: 9.057/20.000). Taking into consideration that 50 percent of our enrolled students(72 students) are considered as low grade students, the percentage of them that will graduate is too low($(10/72)*100 = 13,8$ percent). Based on this, a solution that may help the Department be more competitive is the reduction of the enrolled students.

Remark: It is obvious that the new students are worse than the old ones and this fact will have a negative influence on the graduation rate. Although, the situation for the Applied Mathematics Department is really bad, either with the new students or without them. As for the results, they are a little pessimistic since our training data was consisted of more negative samples. That's why we gave weight to Precision-Recall graphs, so that we can predict more Positives samples. We cannot claim that the percentages of graduated and dropped off students will be totally confirmed, but even if the situation is a little better the Department will probably face the problem of "eternal students".

8.8 Comparison of two Departments

In conclusion, we can say that for sure the situation in Applied Mathematics Department is much worse. The oldest students seem to be weak and the new ones are even weaker. The most important question is what will happen in the future if this downgrade rate continues? For sure, the Department will be forced either to delete students or face the problem of "eternal students". The second situation is more severe than the first since "eternal students" can be considered as a social problem. They just continue their studies for fun and have no intention of working. On the other hand,the Greek State prohibits expelling students. It is clear that an immediate solution is needed, otherwise the situation will get worse.

As for the Mathematics Department, the situation is better. The Department does not face the above problems in such a degree but again it is clear that there is a descending rate during the last two seasons. Indeed, weaker students have managed to graduate and that is why the number of graduation semesters are uncorrelated with the enrolment grade, something that, unfortunately, does not happen in the Applied Mathematics Department.

Finally, the only similarity between the two Departments is an encouraging one. As we saw based both interpreting descriptive statistics and Anova test women students are more diligent than men, thus managing to perform better. Undoubtedly, it is something very interesting and important, showing that women must not be afraid of following a Mathematician career.

Chapter 9

CONCLUSION

Let's briefly examine what we achieved during this project. In the first part of our thesis, we examined the theoretical background of two ML methods: PCA and LDA. After that,we applied successfully both PCA and LDA algorithms to the famous Iris Dataset in order to achieve dimensionality reduction and data visualization.. In addition,we performed a modern approach of ML in football data analysis and we managed to classify all the players of the team using successfully LDA. Using this supervised technique we can classify every transfer target as bad, medium and good player, thus helping the team staff to choose who of the transfer target is better with regards to the team playing level. During this procedure in a completely unknown dataset we managed to have a clear comparison of the two methods. We saw that LDA is much stronger and the well known SSS is not a serious obstacle in the performance of the algorithm. In addition, using correlation coefficient we examined which features are correlated in order to make some proposals to the team staff. Finally we gave a clear comparison of the two methods,thereby explaining their advantages and disadvantages.

The part 2 of this Thesis was for sure more interesting and efficient than the first one. As a theoreticl part we analyzed three supervised Machine Learning K-NN, Random Forests, Decision Trees and Automated ML concepts. Our idea was to apply all of them together with the LDA method to a real dataset of our univeristy's students. The task was much more difficult than before since our results will be taken into consideration by the University community and also, technically speaking, we had to predict the outcome of unseen data(students). We trained our algorithms with the reliable Repeated Holdout method and applied them to both datasets. In order to evaluate their performance we used accuracy score, Roc Curves and PR curves and all of them showed that our classifiers were much better than the trivial one. Supervised Machine Learning Technique and their predictions helped us to show that especially Applied Mathematics Department will face a difficult situation in the future, thus having to deal with many eternal students(75 percent of the active students will not manage to graduate). This result coincide with the fact that the Greek Enrolment Grade is highly correlated with the students' graduation and as a result since the Enrolment Grade becomes lower and lower the students will not manage to graduate. The Mathematics Department will also face a descending situation but the condition is much better, since the Greek Enrolment Grade is not so strongly correlated with the students' graduation. The use of statistical tools helped us analyze the students' academic behaviour and especially show that women perform better than men in both Departments(Anova Test). par The major disadvantage of this work is that we commited a wrong time allocation, as I have already said in the introduction. I think that we lost too much time for the analysis of PCA and LDA and we could have spent this time more efficiently. For instance, we could have analyzed Logistic Regression, SVM and Naive Bayes classifiers which I have already used efficiently in the other project that I completed one month ago. Unfortunately, there was no time to write down a theoretical part for these and that's why they are not included in this Thesis. Especially, Naive Bayes classifier which uses a conditional Independence assumption between features and target could be compared and contrasted with PLDA(probabilistic LDA), which assumes that our data comes from a Normal Distribution. I think that the behaviour of these two classifiers based on the two different assumptions could be a very interesting research topic(and for sure much more interesting than PCA VS LDA).

Reaching the end of this work I am very happy that I am now familiar with a fascinating academic field, which faces a rapid academic growth. Of course, the fact that many ideas and concepts are new established can create confusion, since there are many tools that are under construction. To be honest, this fact was what I liked more even though sometimes make me be frustrated. Of course, every project that helps you improve your knowledge and upgrade your skills is a pleasing one. Also, both this Thesis and also another project that I completed("Machine Learning in Cancer Diagnosis") gave me many ideas for future projects.

Future Work:

During this semester(Spring Semester 2019-2020) I have the intention of completing a project with title: ”Machine Learning VS Deep Learning in Seismology”. As the title says, Machine Learning and Deep Learning concepts will be considered and a contradiction between the two approaches will be performed. At the same time, I will also compare and contrast different methods of correlation coefficients and more specifically:”Pearson VS Spearman” and ”Spearman VS Kendall tau”, since it is interesting to understand in depth the differences between them. For sure, Pearson is the most common method and works efficiently, when our data are linearly separable and I have observed that sometimes Pearson results coincide with ”Spearman” results. On the other hand, if our data are not linearly separable ”Kendall tau” is the most common approach but again I have observed coincidence between the results of Kendal and Spearman(at these times Pearson is outplayed).

ACKNOWLEDGEMENTS

I would like to express my special thanks to my advisor Mr. Kosioris for his help and guidance from my Bachelors' years until the completion of this work. I recognize that he pushed me to study about Machine Learning, since I was a little reluctant in the beginning, and I will always be grateful to him for this reason.

Also, based on the outlook that probably I have completed my whole circle(for now!!) as a student in the Departments of Mathematics and Applied Mathematics, University of Crete, I would like to express my gratitude to all the members of the University community and especially to the professors, that I have collaborated with them all these years.

Finally, it would be an omission to forget expressing my appreciation to the Secretarial Support of both Departments for the Educational Datasets that they provided me with.

Chapter 10

APPENDICES

PYTHON CODE FOR IRIS DATASET(CHAPTER 4):
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.decomposition import PCA
from sklearn.discriminant$_{a}nalysisimportLinearDiscriminantAnalysis$

iris = datasets.load$_{i}ris()$

X = iris.data
print(X)
y = iris.target
print(y)
target$_{n}ames = iris.target_{n}ames$
$print(target_{n}ames)$

pca = PCA(n$_{c}omponents = 2)$
$X_r = pca.fit(X).transform(X)$

lda = LinearDiscriminantAnalysis(n$_{c}omponents = 2)$
$X_r2 = lda.fit(X, y).transform(X)$

print('explained variance ratio (first two components): s' str(pca.explained$_{v}ariance_{r}atio_{)}$)
print('explained variance ratio (first two components): s' str(lda.explained$_{v}ariance_{r}atio_{)}$)
plt.figure()
colors = ['navy', 'turquoise', 'darkorange']

lw = 2

for color, i,target$_{n}ameinzip(colors,[0,1,2],target_{n}ames):plt.scatter(X_{r}[y==i,0],X_{r}[y==i,1],color=color,alpha=.8,lw=lw,label=target_{n}ame)$
$plt.legend(loc=' best',shadow=False,scatterpoints=1)$
$plt.title('PCAofIRISdataset')$

plt.figure()
for color, i, target$_{n}ameinzip(colors,[0,1,2],target_{n}ames):$
$plt.scatter(X_{r}2[y==i,0],X_{r}2[y==i,1],alpha=.8,color=color,label=target_{n}ame)$
$plt.legend(loc=' best',shadow=False,scatterpoints=1)$
$plt.title('LDAofIRISdataset')$

plt.show()

loading$_{s}cores=pd.Series(pca.components_{[}0])$
$sorted_{l}oading_{s}cores=loading_{s}cores.abs().sort_{v}alues(ascending=False)top_{4g}enes=$
$sorted_{l}oading_{s}cores[0:4].index.values$
$print(loading_{s}cores[top_{4g}enes])$

PYTHON CODE FOR FOOTBALL DATASET:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.decomposition import PCA
from sklearn.discriminant$_{a}nalysisimportLinearDiscriminantAnalysis$

X= pd.read$_{e}xcel(r'C:.xlsx')print(X)$
$y=np.array([0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2])$
$print(y)$
$target_{n}ames=['bad',' basic',' good']$

pca = PCA(n$_{c}omponents=2)$
$X_{r}=pca.fit(X).transform(X)$
$lda=LinearDiscriminantAnalysis(n_{c}omponents=2)$
$X_{r}2=lda.fit(X,y).transform(X)$

print('explained variance ratio (first two components):

$$print('explainedvarianceratio(firsttwocomponents):$$

plt.figure()
colors = ['navy', 'turquoise', 'darkorange']
lw = 2

for color, i, target$_{n}ameinzip(colors, [0, 1, 2], target_{n}ames) :$
$plt.scatter(X_{r}[y == i, 0], X_{r}[y == i, 1], color = color, alpha = .8, lw = lw, label = target_{n}ame)$
$plt.legend(loc =' best', shadow = False, scatterpoints = 1)$
$plt.title('PCAofplayersdataset')$
$plt.figure()$
$forcolor, i, target_{n}ameinzip(colors, [0, 1, 2], target_{n}ames) :$
$plt.scatter(X_{r}2[y == i, 0], X_{r}2[y == i, 1], alpha = .8, color = color, label = target_{n}ame)$
$plt.legend(loc =' best', shadow = False, scatterpoints = 1)$
$plt.title('LDAofplayersdataset')$

plt.show()
loading$_{s}cores = pd.Series(pca.components_{[}0])$
$sorted_{l}oading_{s}cores = loading_{s}cores.abs().sort_{v}alues(ascending = False)$
$top_{4g}enes = sorted_{l}oading_{s}cores[0 : 4].index.values$
$print(loading_{s}cores[top_{4g}enes])$

Chapter 11

Bibliography-References

(a) Machine Learning, Tom Mitchell, McGRAW-HILL International Editions, 1997

(b) Mathematics for Machine Learning, March Deisenroth, Aldo Faisal and Cheng Soon Ong, Cambridge University Press Report. 2018

(c) Pattern Recognition and Machine Learning, Christopher Bishop, Springer Editions, 2006

(d) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshbirani and Jerome Friedman, Springer Series in Statistics, Second Edition 2008

(e) Linear Discriminant Analysis: A detailed tutorial, Alaa Tharwat, Tarek Gaber, Abdelhammed Ibrahim and Ella Hassanien, AI Communications 00(20xx) pages 1-22, IOS Press, 2017

(f) Random Forests and Decision Trees, Jehad Ali1, Rehanullah Khan, Nasir Ahmad and Imran Maqsood IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012

(g) Early Programming Education and Career Orientation: The Effects of Gender, Self-Efficacy, Motivation and Stereotypes, Efthimia Aivaloglou, Felienne Hermans, Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pages 679–685), 2019

(h) Bachelor thesis analytics: using machine learning to predict dropout and identify performance factors,J.Nouri, K.Larsson and M.Saqr, International Journal of Learning Analytics and Artificial Intelligence for Education, 2019

(i) Machine Learning Based Classification Approach for Predicting Students Performance in Blended Learning, Celia González, Nespereira Esraa Elhariri-Nashwa and El-Bendary, The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), Egypt 2015

(j) Precision-Recall-Gain Curves: PR Analysis, Peter A.Flach and Meelis Kull, Advances in Neural Information Processing Systems 28, 2015

(k) Comparison of values of Pearson's AND Spearman's Correlation Coefficients on the same sets of data, Jan Hauke ans Tomasz Kossowski, Quaestiones Geographicae 30(2), (2011)

(l) Cross-Validation Methods, Michael WBrowne, Journal of Mathematic Psychology, volume 44-Issue 1, pages 108-132, March 2000

(m) The Relationship Between Precision-Recall and ROC Curves, Jesse Davis, Mark Goadrich, ICML '06: Proceedings of the 23rd international conference on Machine learning(pages 233-240), June 2006

(n) The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets, Takaya Saito and Marc Rehmsmeier, Guy Brock, University of Louisville, UNITED STATES,March 2015