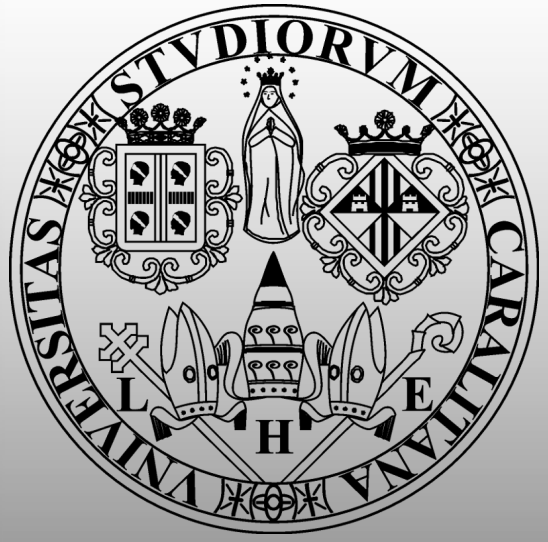



# A Linear-dependence-based Approach to Design Proactive Credit Scoring Models

**Roberto Saia and Salvatore Carta**
*Department of Mathematics and Computer Science*
University of Cagliari, Italy

**KDIR 2016, 8th International Conference on Knowledge Discovery and Information Retrieval**
**9÷11 November 2016 - Porto, Portugal**

# Outline

- **Overview**
  *Introduction ▪ State of the Art ▪ Limits*
- **Intuition**
  *Idea ▪ Usage ▪ Advantages*
- **Implementation**
  *Normalization ▪ ASD ▪ Reliability Band, Classification*
- **Results**
  *Experiments ▪ Conclusions ▪ Future Work*

# OVERVIEW

# Introduction

- The main aim of **credit scoring models** is the classification of the loan applications as *reliable* or *unreliable*

# Introduction

- They are usually **designed** on the basis of the past loan applications
- They represent a **crucial** instrument for the financial operators

# State of the Art

- Many **techniques** have been designed to address the *Credit Scoring* problem, by exploiting several models, such as:
  - *Statistical* and *Data Mining*[1]
  - *Linear Discriminant*[2]
  - *Logistic Regression*[3]
  - *Neural Networks*[4]
  - *Genetic Program*[5]
  - *Decision-tree*[6]
  - … and many more

# State of the Art

- Regardless of the adopted technique, the definition of effective *Credit Scoring* models represents a **hard challenge** for several factors

- The most important of which is the **Imbalanced class distribution**[7]

# Limits

- The **Imbalanced class distribution** exists because the number of *default* cases (*non-refunded loans*) is **much smaller** than that of the *non-default* ones (*refunded loans*)
- This reduces the **effectiveness** of the most widely used approaches[8]

NON-DEFAULT CASES **VS** DEFAULT CASES

# INTUITION

# Idea

- Since the **determinant of a matrix** depends on the linear dependence between the vectors that compose it
- Where **linear dependence** between two vectors means that one of them is a linear combination of the other one

**M1 =**

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |

$det(M1) = 0$

**linear dependence**

**M2 =**

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 2 | 2 |
| 1 | 4 | 6 |

$det(M2) \neq 0$

**linear independence**

We want to exploit this property to evaluate the **similarity** between **a vector** and a series of **other vectors**

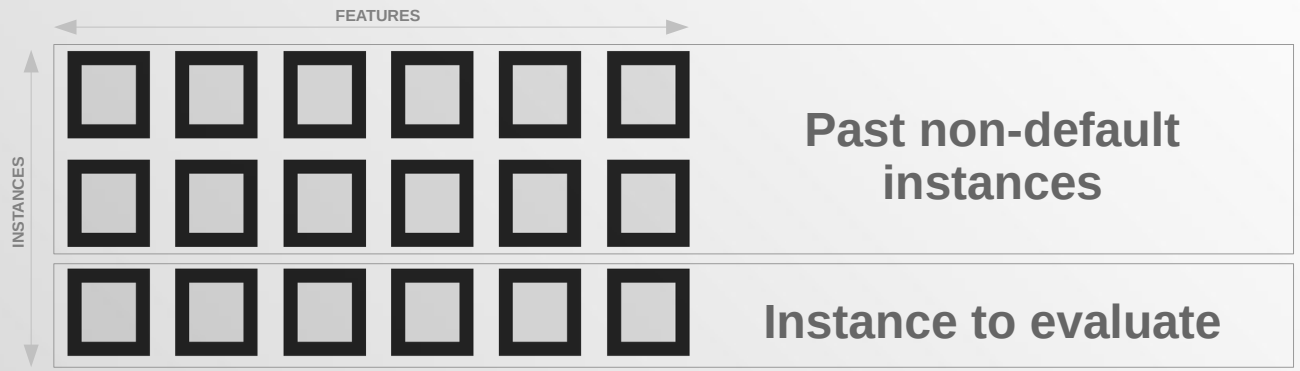
# Usage

- We propose a *Linear Dependence Based* (**LDB**) approach able to evaluate a new loan application in a novel way
- **LDB** performs this operation by exploiting the **linear dependence** concept
- **LDB** evaluates the vector representation of a new loan application (from now on named as **instance***) in a matrix context, in terms of *determinant*

* It contains several information about the applicant, e.g., **gender**, **age**, **marital status**, **income**, **loan amount**, **other loans**, etc.

# Usage

- The **N-1 rows** of the matrix are the vector representation of the past non-default instances
- The **last row N** is the vector representation of the instance to evaluate
- We also define a **method** to deal with the *non-square* matrices (where the *determinant* operation is not defined)



# Usage

The more the vector representation of a new instance is **linearly dependent** than those of the previous non-default instances, the more we consider the instance as reliable


Because such dependence indicates the existence of **similar values** in the features

# Advantages

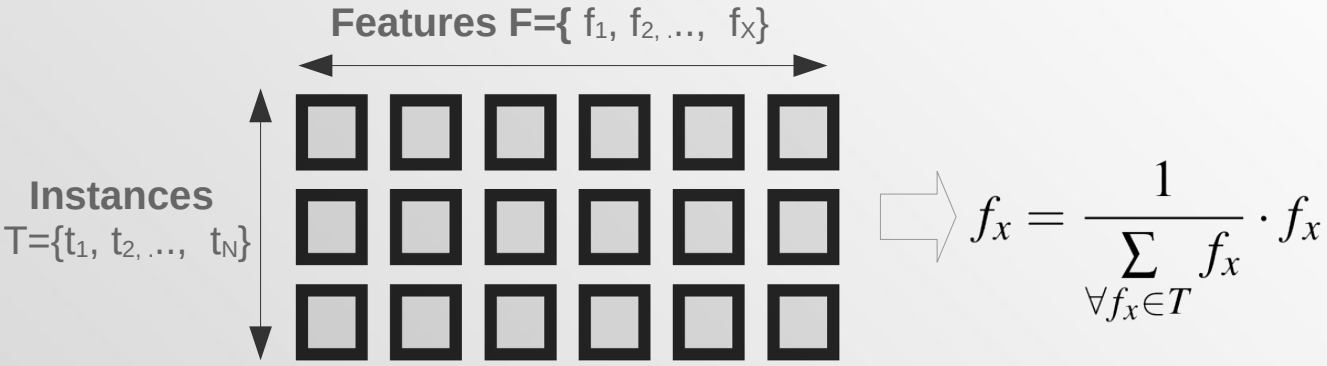
- The adoption of the proposed *Linear Dependence Based* (**LDB**) approach presents several advantages:
  - It **overcomes** the imbalance class distribution issue by using only a class of data (*non-default instances*)
  - The use of only *non-default* cases, allows us to operate in a **proactive** way
  - It reduces the problems related to the **cold-start** issue[9]

# IMPLEMENTATION

# Implementation

## Step 1 of 4 - Data Normalization

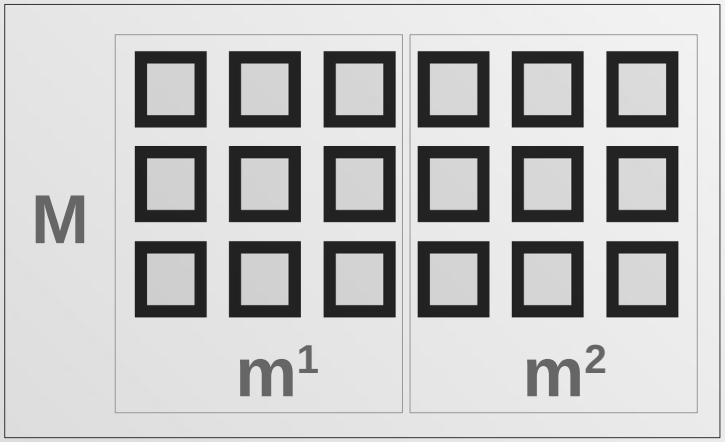
- We **preprocess** the vector representation of the instances in order to normalize all value in a range [0,1]
- This is intended to **prevent** any problem related to the determinant calculation, when it involves very large matrices



# Implementation

## Step 2 of 4 – ASD Definition

- The *Average of Sub-matrices Determinant* (**ASD**) criterion allows us to operate also with the non-square matrices by calculating the average of the determinants of a series of square sub-matrices




$$det(M) = \frac{det(m^1) + det(m^2)}{2}$$

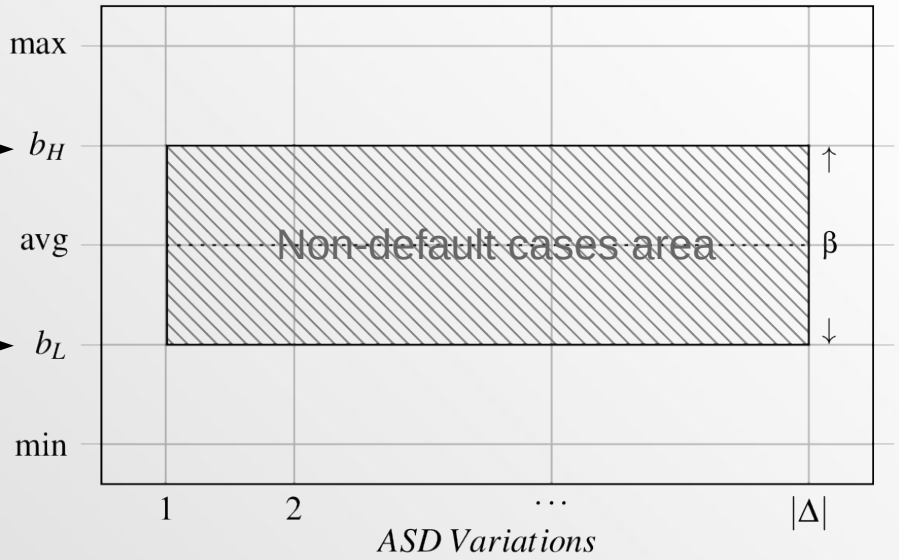
# Implementation

## Step 3 of 4 – Reliability Band

- The **Reliability Band** (β) provides information about the linear variations when the **ASD** process i*nvolves only non-default cases*

$$b_H = \frac{avg + max}{2}$$

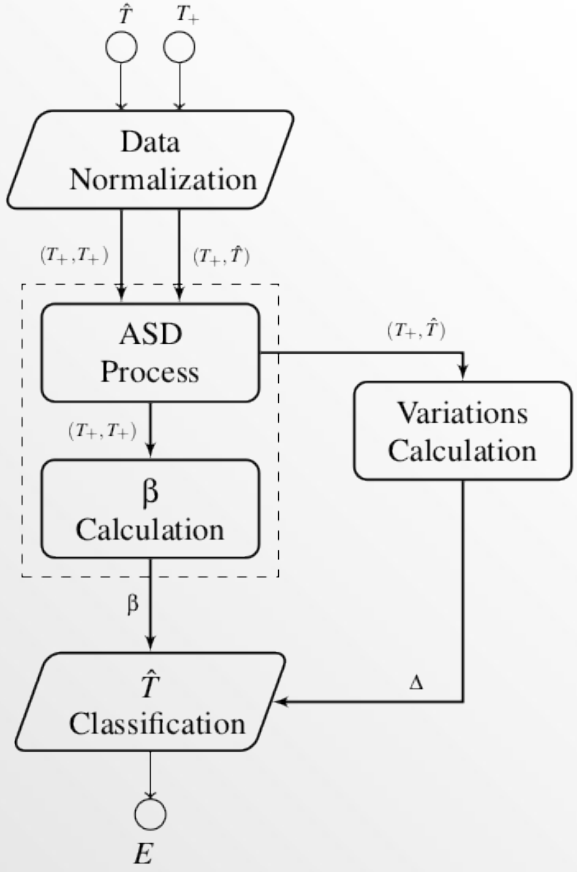
$$b_L = \frac{avg + min}{2}$$



# Implementation

## Step 4 of 4 – Instances Classification

Our approach classifies the **new instances** as *rejected* or *accepted* on the basis of the **ASD** *process* and the *Reliability Band* **β**



# RESULTS

# Experiments

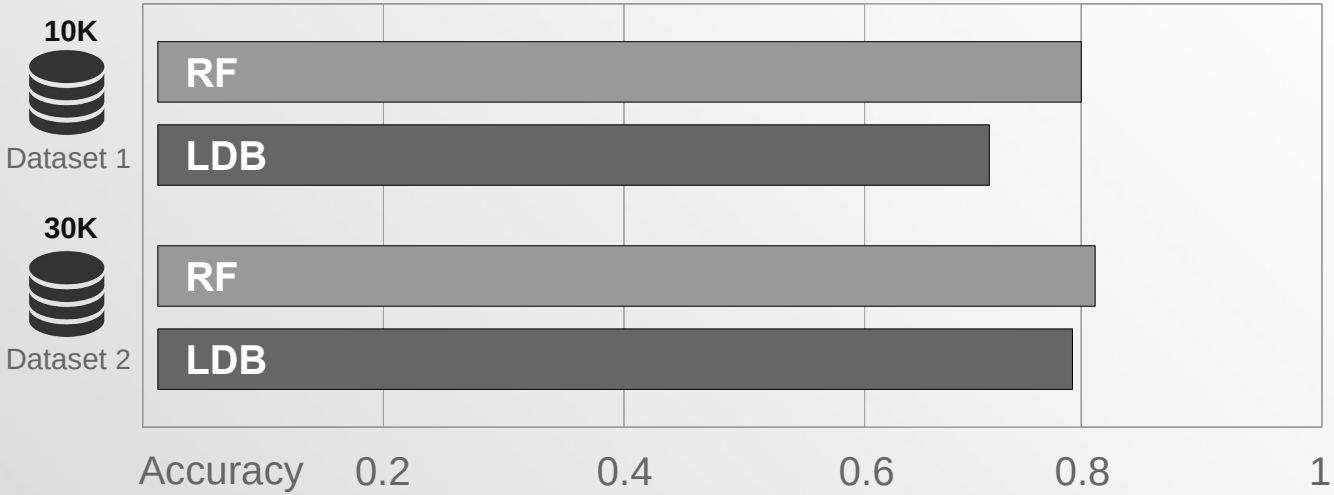
- We conducted the **experiments** by following the *k-fold cross validation* criterion, with *k=10*
- We used two **real-world datasets**[10] that represent a benchmark in this context and present a *strong imbalanced* data distribution
- The selected competitor was **Random Forests**, since it represents one of the most performing state-of-the-art approach in this field[11]

# Experiments

- The *first two* experiments show the *general performance* achieved by our LDB approach and Random Forests, in terms of ***Accuracy*** and ***F-score***
- The *third* experiment is instead aimed to evaluate our predictive model in terms of **AUC**

# Experiments

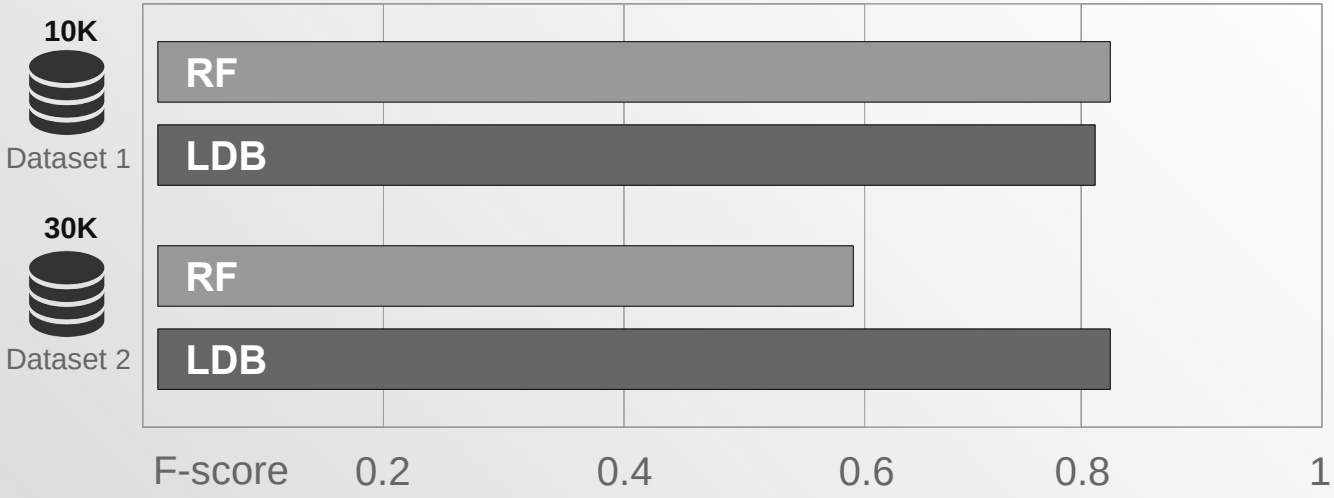
- The general performance of our **LDB** approach in terms of *accuracy* is very close to those of Random Forests (**RF**)




- In spite of the fact that we did **not use** any past default instance to train our model

# Experiments

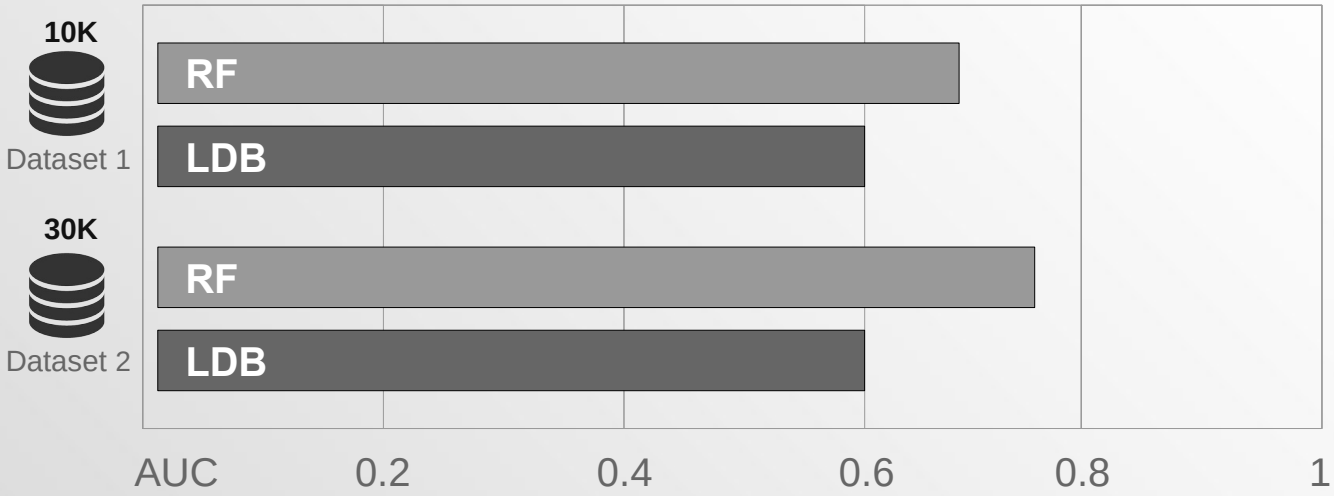
- Also in terms of *F-score*, the general performance of our **LDB** approach is very close to those of Random Forests (**RF**)




- Differently from **RF**, the effectiveness of **LDB** increases with the number of instances

# Experiments

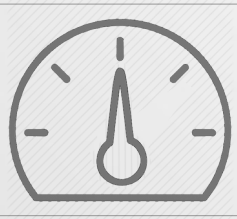
- The last set of experiments measure the performance in terms of **AUC** (*Area Under ROC Curve*)




- Also in this case, the performance of our *predictive model* is very close to those of **RF**

# Conclusions

- We proposed a novel approach of ***Credit Scoring*** that exploits a *Linear Dependence Based* (**LDB**) criterion
- The experiments show **two** main results:

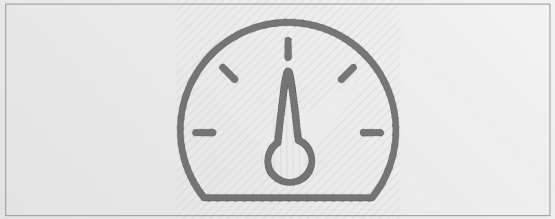


By using a small dataset, LDB performance is **similar** to one of the most performing state-of-the-art approach (*RF*), in spite of the fact that we do not use the past default instances

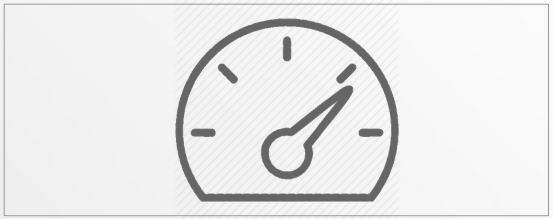
By using a larger dataset, LDB **outperforms** its competitor (*RF*), while continuing to not use past default instances during the training

# Conclusions

- We proposed a novel approach of ***Credit Scoring*** that exploits a *Linear Dependence Based* (**LDB**) criterion
- The experiments show **two** main results:



**Proactivity**



**Improvement**

# Future Work

- A possible follow up of this work could be a new series of experiments performed by using **both** the *past default* and *non-default* cases
- This will allow us to investigate about the advantages and disvantages related to a **non-proactive** approach
- We also consider to test our approach in **heterogeneous** scenarios that involve different types of financial data

# References


1) Chen, S. Y. and Liu, X. (2004). *The contribution of data mining to information science*. Journal of Information Science, 30(6):550–558.

2) Reichert, A. K., Cho, C.-C., and Wagner, G. M. (1983). *An examination of the conceptual issues involved in developing credit-scoring models*. Journal of Business & Economic Statistics, 1(2):101–114.

3) Henley, W. E. (1994). *Statistical aspects of credit scoring*. PhD thesis, Open University.

4) Desai, V. S., Crook, J. N., and Overstreet, G. A. (1996). *A comparison of neural networks and linear scoring models in the credit union environment*. European Journal of Operational Research, 95(1):24–37.

5) Ong, C.-S., Huang, J.-J., and Tzeng, G.-H. (2005). *Building credit scoring models using genetic programming*. Expert Systems with Applications, 29(1):41–47.

6) Davis, R., Edelman, D., and Gammerman, A. (1992). *Machine-learning algorithms for credit-card applications*. IMA Journal of Management Mathematics, 4(1):43–51.

7) He, H. and Garcia, E. A. (2009). *Learning from imbalanced data*. IEEE Trans. Knowl. Data Eng., 21(9):1263–1284.

8) Japkowicz, N. and Stephen, S. (2002). *The class imbalance problem: A systematic study*. Intell. Data Anal., 6(5):429–449.

9) Attenberg, J. and Provost, F. J. (2010). *Inactive learning?: difficulties employing active learning in practice*. SIGKDD Explorations, 12(2):36–41.

*10)* *Default of Credit Card Client* and *German Credit*. UCI Repository of Machine Learning Databases, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/.

11) Lessmann, S., Baesens, B., Seow, H., and Thomas, L. C. (2015). *Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research*. European Journal of Operational Research, 247(1):124–136.

# A Linear-dependence-based Approach to Design Proactive Credit Scoring Models

## THANK YOU FOR YOUR ATTENTION

KDIR 2016, 8th International Conference on Knowledge Discovery and Information Retrieval
9÷11 November 2016 - Porto, Portugal