ENSIAS

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

Data preprocessing project report:

Online News Popularity, Prediction shares

And

Financial Distress Prediction

Project Team Members:

ACHAG Nada

BALLITO Hamza

BEN SAKA Ibtissam

OUAKIL Hajar

Project Mentor:

Mr. LAZAAR Mohamed

University year: 2020-2021

Abstract

In many domains such as computer vision or pattern recognition, solving a problem is based on processing data extracted from a set of real world data acquired by means of sensors or resulting from some data processing.

Data are structured as vectors. The quality of a processing system highly depends on the choice of these vector content. However, in many cases the vectors' high dimensionality makes it almost impossible to use them to solve the problem, both because of the data themselves and of the learning set size. Hence, it is usually recommended, and sometimes required to reduce the vector size in order to make them more usable, even if the reduction might lead to information loss. Sometimes, solving complex problems with large descriptors can also be accomplished using a small set of features selected from initial data set. This can be done if the selected features are relevant with respect to the considered problem. Reducing vector dimensionality is often considered as a pre-processing essential step dedicated to noise and redundant information elimination.

In this project, we will be interested in two dimensionality reduction methods, we will apply it to two different datasets.

List of abbreviations

Abbreviation	Designation
LDA	Linear Discriminant Analysis
MLP	Multi-Layer Perceptron
NN	Neural Network
PCA	Principal Component Analysis

Contents

List of figures

List of tables

General Introduction

In machine learning, the performance of a model only benefits from more features up until a certain point. The more features are fed into a model, the more the dimensionality of the data increases. As the dimensionality increases, overfitting becomes more likely.

There are multiple techniques that can be used to fight overfitting, but dimensionality reduction is one of the most effective techniques.

Dimensional reduction methods are generally classified into two categories:

- A reduction based on a selection of characteristics which consists in selecting the most relevant characteristics from the data set of variables describing the phenomenon under study.
- A reduction based on a transformation of the data also called an extraction of characteristics and which consists in replacing the initial set of data by a new reduced set, constructed from the initial set of characteristics.

This report is divided into three chapters:

The first chapter consists in presenting the dimensionality reduction methods and classifiers used as well as its advantages and disadvantages.

The second chapter concerns the description of the datasets used and the data visualization.

The third chapter is the application part of the model on the two datasets, presenting the tools and the language used as well as the results obtained before and after the dimensionality reduction phases.

1. The goal

The objective of this project is to create a decision support system to compare the score with and without the dimensionality reduction phase. This project aims to apply the dimensionality reduction methods seen in the course on two different datasets in order to observe and show the usefulness of these methods to improve the learning model.

2. The context

This project is part of the part of data preprocessing which is an essential phase in the data processing process.

The pre-treatment phase refers to cleaning operations and transformation that must be applied to the raw data before their processing and analysis.

3. The approach and methodology implemented

In this project we used and implemented various methodology and approaches in order to achieve our goal.

We have researched and compared many methods and classifiers in order to choose the appropriate methods and classifiers for our application.

For the reason that we have large dimensional datasets, we have chosen to work with PCA and LDA as dimensionality reduction methods and MLP as a classifier.

Chapter I:

Dimensionality reduction

Introduction

In this chapter we will first define the different dimensionality reduction methods and the classifiers used, while giving the advantages and disadvantages of the different techniques.

We chose the two methods PCA and LDA because we have large datasets.

1. Presentation of the dimensionality reduction methods used

1.1 PCA

Principal Component Analysis (PCA) is a statistical method that creates new features or characteristics of data by analyzing the characteristics of the dataset. Essentially, the characteristics of the data are summarized or combined together. You can also conceive of Principal Component Analysis as "squishing" data down into just a few dimensions from much higher dimensions space [1].

Following are some of the advantages and disadvantages of PCA [2]:

Advantages of PCA:

- ✓ **Improves Algorithm Performance:** With so many features, the performance of your algorithm will drastically degrade. PCA is a very common way to speed up your Machine Learning algorithm by getting rid of correlated variables which don't contribute in any decision making. The training time of the algorithms reduces significantly with less number of features.
- ✓ **Reduces Overfitting:** Overfitting mainly occurs when there are too many variables in the dataset. So, PCA helps in overcoming the overfitting issue by reducing the number of features.
- ✓ **Improves Visualization:** It is very hard to visualize and understand the data in high dimensions. PCA transforms a high dimensional data to low dimensional data so that it can be visualized easily.

Disadvantages of PCA:

- **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.
- **Information Loss**: Although Principal Components try to cover maximum variance among the features in a dataset, if we don't select the number of Principal Components with care, it may miss some information as compared to the original list of features.

1.2 LDA

Linear Discriminant Analysis or LDA is a dimensionality reduction technique. It is used as a pre-processing step in Machine Learning and applications of pattern classification. The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

The original technique was developed in the year 1936 by Ronald A. Fisher and was named Linear Discriminant or Fisher's Discriminant Analysis. The original Linear Discriminant was described as a two-class technique. The multi-class version was later generalized by C.R Rao as Multiple Discriminant Analysis. They are all simply referred to as the Linear Discriminant Analysis.

LDA is a supervised classification technique that is considered a part of crafting competitive machine learning models. This category of dimensionality reduction is used in areas like image recognition and predictive analysis in marketing [3].

Advantages of LDA:

The major advantage of LDA is that it uses information from the features to create a new axis which in turn minimizes the variance and maximizes the class distance of the variables.

- ✓ Linear decision boundary
- ✓ Fast classification
- ✓ Easy to implement

Disadvantages of LDA:

- Gaussian assumptions
- Training time
- Complex matrix operations.

2. Presentation of the classifiers used

2.1 NN

An artificial neural network learning algorithm, or neural network, or just neural net, is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. The concept of the artificial neural network was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses.

Neural networks are one of many tools and approaches used in machine learning algorithms. The neural network itself may be used as a piece in many different machine learning algorithms to process complex data inputs into a space that computers can understand.

Neural networks are being applied to many real-life problems today, including speech and image recognition, spam email filtering, finance, and medical diagnosis… [4].

2.2 MLP

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable [5].

Structure of a Multi-Layer Perceptron:

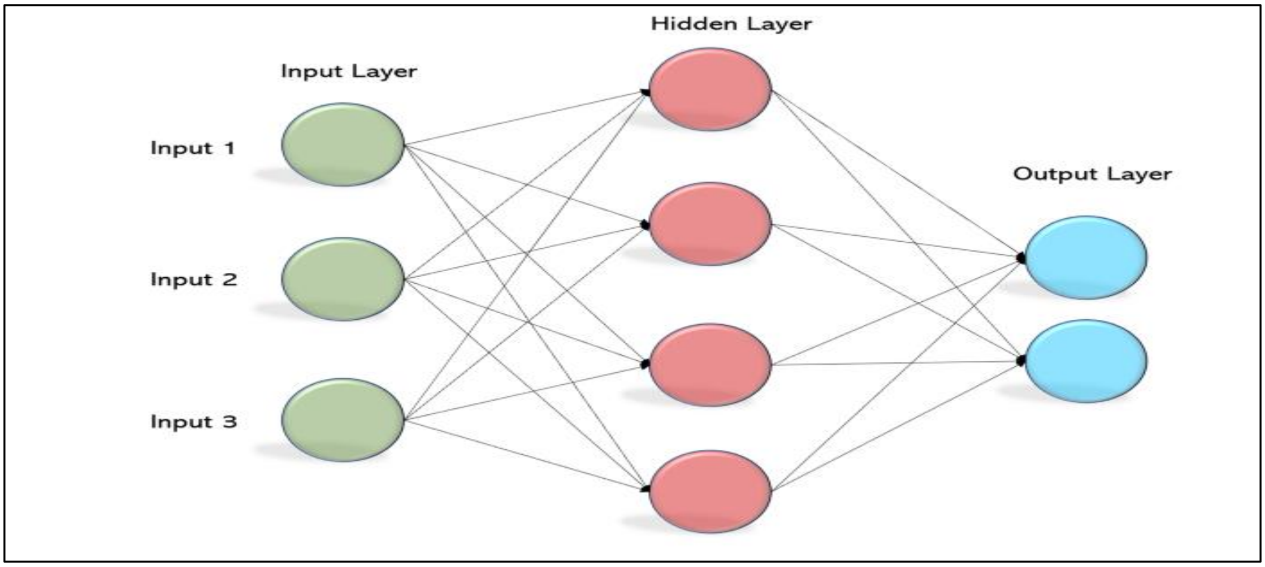


Figure 1: Structure of a Multi-Layer Perceptron

Benefits of MLP:

- ✓ **Connectionist:** used as a metaphor for biological neural networks
- ✓ **Computationally efficient:** Can easily be parallelized
- ✓ **Universal computing machines**

Drawbacks of MLP:

- Convergence can be slow
- Local minima can affect the training process
- Hard to scale

Conclusion

In this chapter, we have presented the dimensionality reduction methods used, namely PCA and LDA as well as its advantages and disadvantages.

Chapter II:

Data set

Introduction

In this chapter we will present the two datasets that we used to evaluate the efficiency of the dimensionality reduction methods and the classifiers used. We chose two different datasets: Online News Popularity, Prediction shares And Financial Distress Prediction.

1. Description of databases

1.1 Financial Distress dataset

This dataset is from https://www.kaggle.com/shebrahimi/financial-distress.

The study involved 422 company on different time periods.

All used features are described in the following table.

Variable	Description
Company	Represents sample companies
Time	Shows different time periods that data belongs to. Time series length varies between 1 to 14 for each company.
Financial Distress	The target variable. If it is greater than -0.50 the company should be considered as healthy (0). Otherwise, it would be regarded as financially distressed (1).
x1 to x83	The features denoted by x1 to x83, are some financial and non-financial characteristics of the sampled companies. These features belong to the previous time period, which should be used to predict whether the company will be financially distressed or not (classification). Feature x80 is a categorical variable.

Table 1: Features of Financial Distress dataset used

1.2 Online news popularity Dataset

This data is from https://www.kaggle.com/btphan/online-news-popularity-dataset.

All used features are described in the following table.

Variable	Description
url	URL of the article.(String)
timedelta	Days between the article publication and the dataset acquisition.(float)
ntokenstitle	Number of words in the title (Integer)
ntokenscontent	Number of words in the content((Integer)
nuniquetokens	Rate of unique words in the content (Integer)
nnonstop_words	Rate of non-stop words in the content (Integer)
nnonstopuniquetokens	Rate of unique non-stop words in the content(Integer)
num_hrefs	Number of links (Integer)
numselfhrefs	Number of links to other articles published by Mashable (Integer)
num_imgs	Number of images (float)
num_videos	Number of videos (float)
averagetokenlength	Average length of the words in the content (float)
num_keywords	Number of keywords in the metadata (float)
datachannelis_lifestyle	Is data channel 'Lifestyle'? (Binary)
datachannelis_entertainment	Is data channel 'Entertainment'? (Binary)
datachannelis_bus	Is data channel 'Business'? (Binary)
datachannelis_socmed	Is data channel 'Social Media'? (Binary)
datachannelis_tech	Is data channel 'Tech'? (Binary)
datachannelis_world	Is data channel 'World'? (Binary)
kwminmin	Worst keyword (min. shares) (float)
kwmaxmin	Worst keyword (max. shares) (float)
kwavgmin	Worst keyword (avg. shares) (float)
kwminmax	Best keyword (min. shares) (float)
kwmaxmax	Best keyword (max. shares) (float)
kwavgmax	Best keyword (avg. shares) (float)
kwminavg	Avg. keyword (min. shares) (float)
kwmaxavg	Avg. keyword (max. shares) (float)
kwavgavg	Avg. keyword (avg. shares) (float)

selfreferencemin_shares	Min. shares of referenced articles in Mashable (float)
selfreferencemax_shares	Max. shares of referenced articles in Mashable (float)
selfreferenceavg_sharess	Avg. shares of referenced articles in Mashable (float)
Weekdayismonday	Was the article published on a Monday? (Binary)
Weekdayistuesday	Was the article published on a Tuesday? (Binary)
Weekdayiswednesday	Was the article published on a Wednesday? (Binary)
weekdayisthursday	Was the article published on a Thursday? (Binary)
weekdayisfriday	Was the article published on a Friday? (Binary)
weekdayissaturday	Was the article published on a Saturday? (Binary)
weekdayissunday	Was the article published on a Sunday? (Binary)
is_weekend	Was the article published on the weekend? (Binary)
LDA_00	Closeness to LDA topic 0 (float)
LDA_01	Closeness to LDA topic 1 (float)
LDA_02	Closeness to LDA topic 2 (float)
LDA_03	Closeness to LDA topic 3 (float)
LDA_04	Closeness to LDA topic 4 (float)
global_subjectivity	Text subjectivity (float)
globalsentimentpolarity	Text sentiment polarity (float)
globalratepositive_words	Rate of positive words in the content (float)
globalratenegative_words	Rate of negative words in the content (float)
ratepositivewords	Rate of positive words among non-neutral tokens (float)
ratenegativewords	Rate of negative words among non-neutral tokens (float)
avgpositivepolarity	Avg. polarity of positive words (float)
minpositivepolarity	Min. polarity of positive words (float)
maxpositivepolarity	Max. polarity of positive words (float)
avgnegativepolarity	Avg. polarity of negative words (float)
minnegativepolarity	Min. polarity of negative words (float)
maxnegativepolarity	Max. polarity of negative words (float)
title_subjectivity	Title subjectivity (float)
titlesentimentpolarity	Title polarity (float)
abstitlesubjectivity	Absolute subjectivity level (float)

abstitlesentiment_polarity	Absolute polarity level (float)
shares	Number of shares (target) (Integer)

Table 2: Features of Online news popularity Dataset used

2. Data visualization

In this part we will visualize the different variables of the both data set.

2.1 Financial Distress dataset

The following figure represents the histogram of each variable in the Financial Distress dataset.

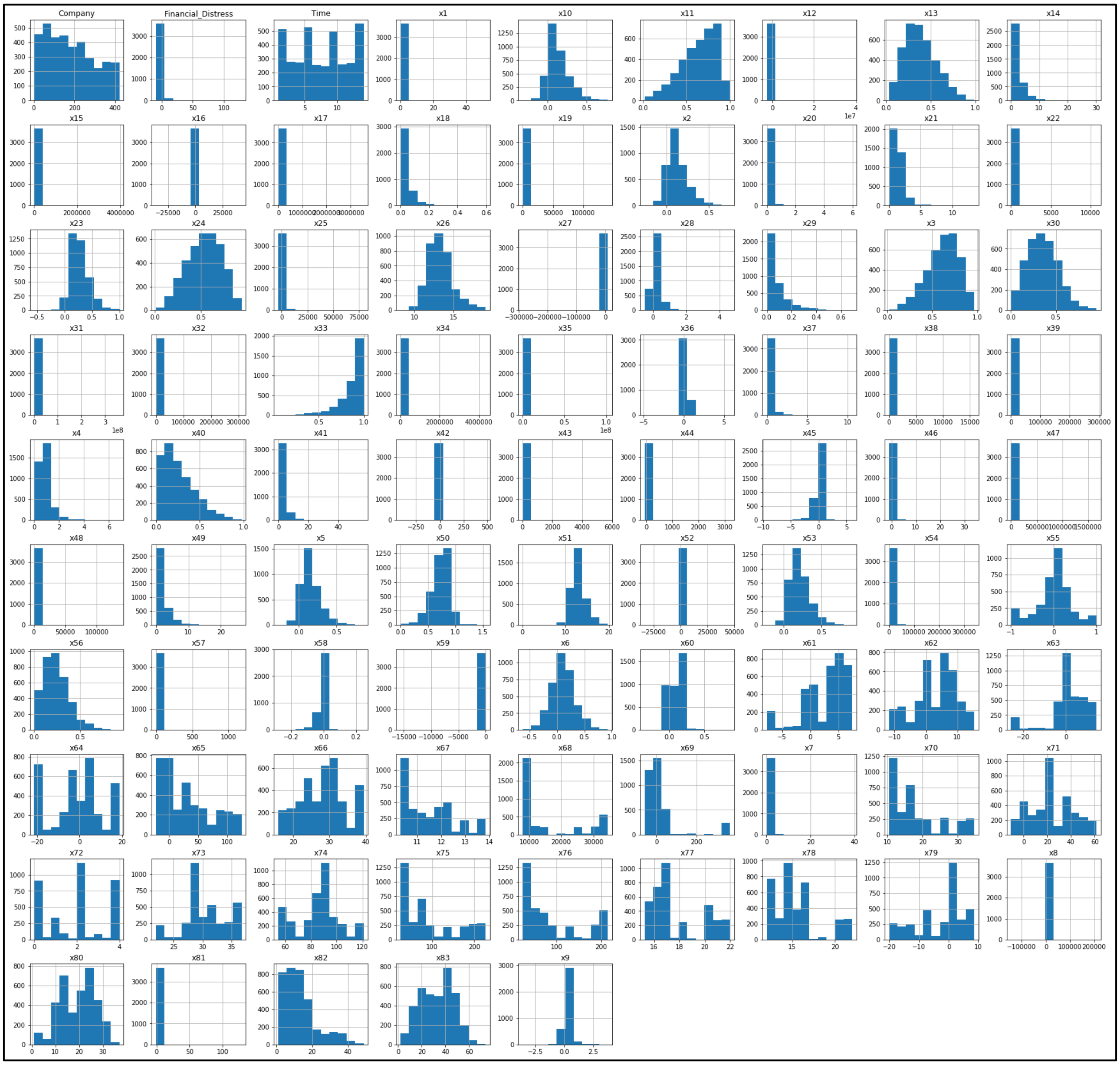


Figure 2: histograms of Financial Distress dataset

2.2 Online news popularity Dataset

The following figure represents the histogram of each variable in the online news popularity dataset.

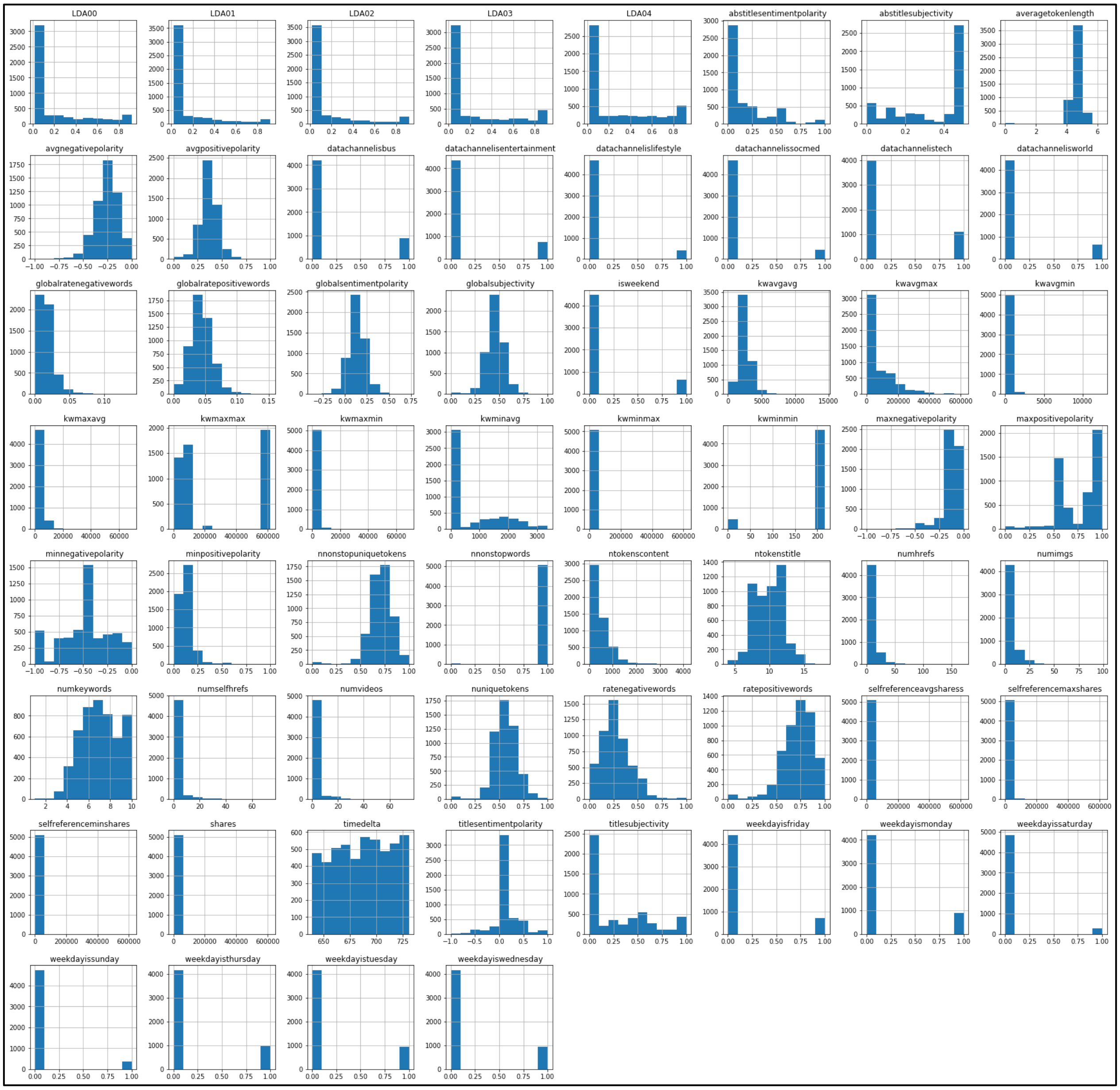


Figure 3: histograms of online news popularity

Conclusion

In this chapter, we have described the two datasets were used in our tests, their characteristics and it sources.

Chapter III:

Experimental results

Introduction

In this chapter we will describe the tools used, and we will present the different results obtained before and after the reduction of dimensionality and we will compare them.

1. Used tools

1.1 The Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use [6].

1.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [7].

2. Results without dimensionality reduction

2.1 Financial Distress dataset

The following figures represent the results obtained before the dimensionality reduction phase.

Figures three below represent loss function after MLP application with 1, 2 and 3 hidden layer.

We have noticed that with the increase in the number of hidden layers, we can quickly find the minimum of the loss function.

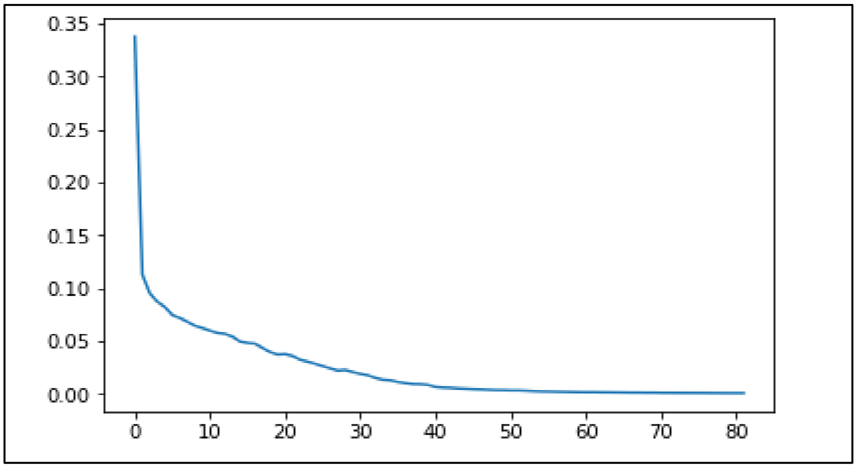


Figure 4: loss function of MLP with 1 hidden layer

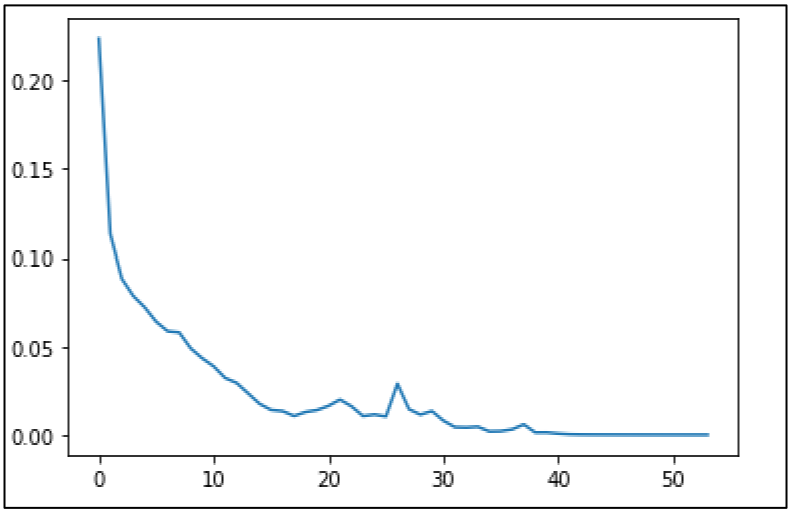


Figure 5: loss function of MLP with 2 hidden layer

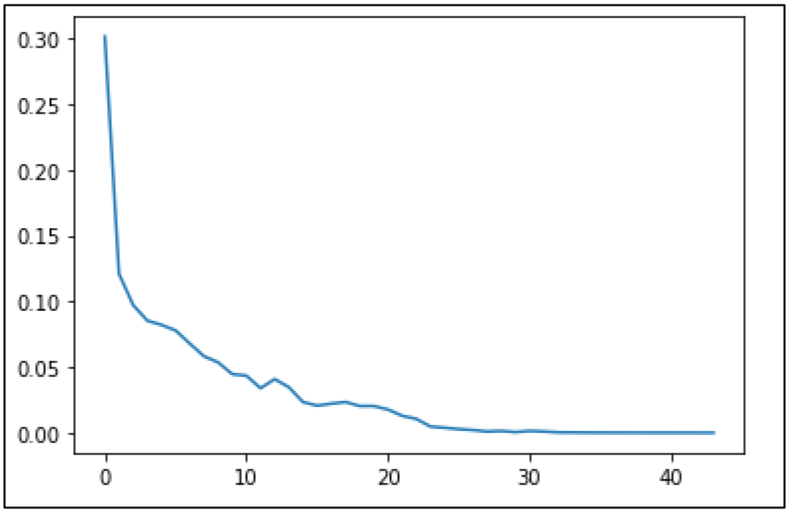
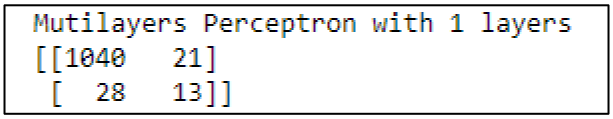


Figure 6: loss function of MLP with 3 hidden layer

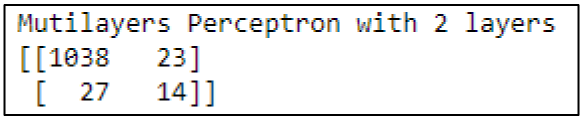
The following figures show the confusion matrix, the classification ratio and the accuracy after applying MLP with 1, 2 and 3 hidden layer.

And we have noticed that the best accuracy (equal to 0.9573) is obtained with an MLP with 3 hidden layer.



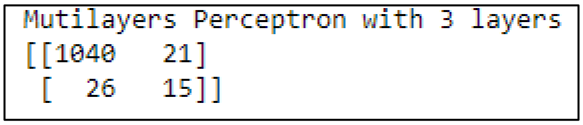
```
Mutilayers Perceptron with 1 layers
[[1040   21]
 [  28   13]]
```

Figure 7: confusion matrix of MLP with 1 hidden layer



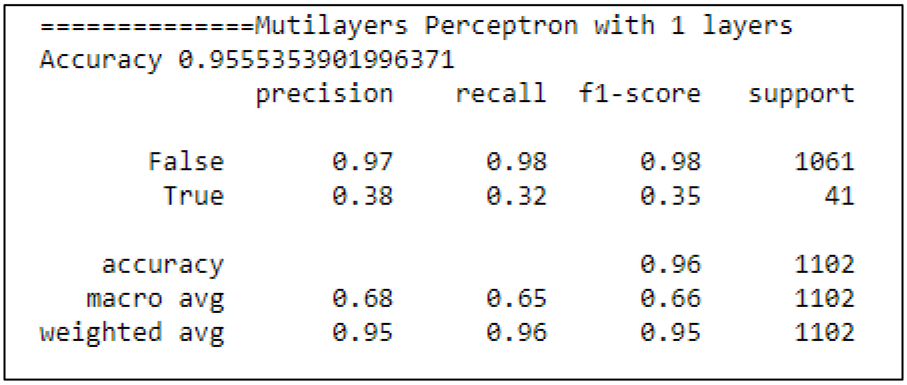
```
Mutilayers Perceptron with 2 layers
[[1038   23]
 [  27   14]]
```

Figure 8: confusion matrix of MLP with 2 hidden layer



```
Mutilayers Perceptron with 3 layers
[[1040   21]
 [  26   15]]
```

Figure 9: confusion matrix of MLP with 3 hidden layer

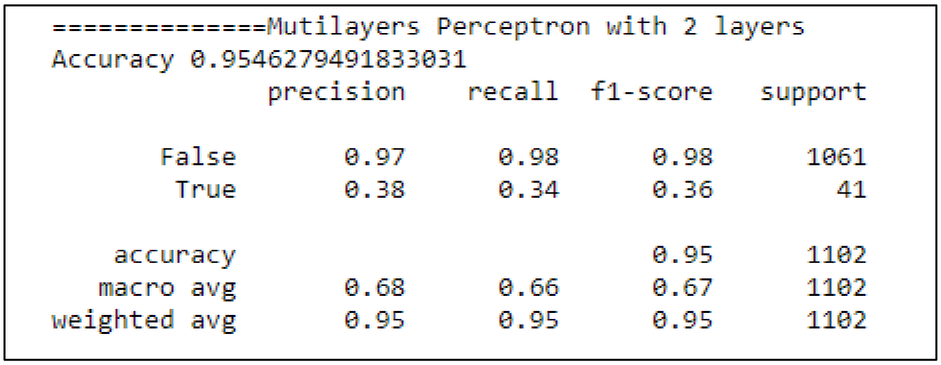


```
==============Mutilayers Perceptron with 1 layers
Accuracy 0.9555353901996371
              precision    recall  f1-score   support

       False       0.97      0.98      0.98      1061
        True       0.38      0.32      0.35        41

    accuracy                           0.96      1102
   macro avg       0.68      0.65      0.66      1102
weighted avg       0.95      0.96      0.95      1102
```

Figure 10: Accuracy and classification report of MLP with 1 hidden layer

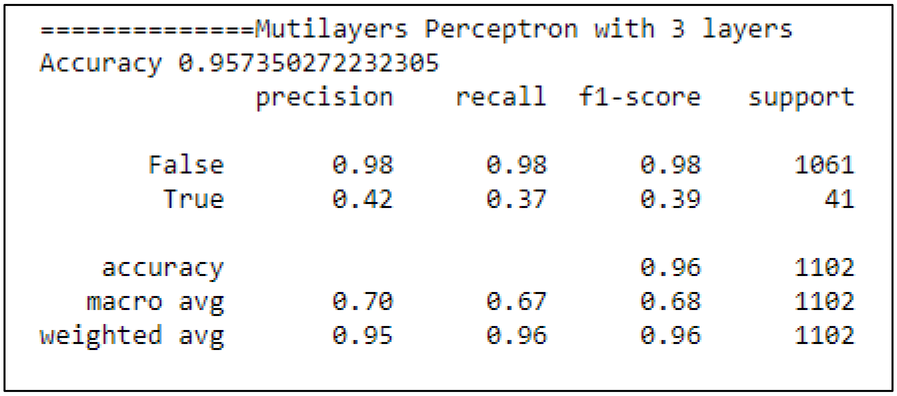


```
==============Mutilayers Perceptron with 2 layers
Accuracy 0.9546279491833031
              precision    recall  f1-score   support

       False       0.97      0.98      0.98      1061
        True       0.38      0.34      0.36        41

    accuracy                           0.95      1102
   macro avg       0.68      0.66      0.67      1102
weighted avg       0.95      0.95      0.95      1102
```

Figure 11: Accuracy and classification report of MLP with 2 hidden layer



```
==============Mutilayers Perceptron with 3 layers
Accuracy 0.957350272232305
              precision    recall  f1-score   support

       False       0.98      0.98      0.98      1061
        True       0.42      0.37      0.39        41

    accuracy                           0.96      1102
   macro avg       0.70      0.67      0.68      1102
weighted avg       0.95      0.96      0.96      1102
```

Figure 12: Accuracy and classification report of MLP with 3 hidden layer

2.2 Online news popularity Dataset

The following figures show loss function and the accuracy after applying MLP with 1, 2 and 3 hidden layer.

We noticed that the MLP with 2 hidden layers, quickly found the minimum of the loss function compared to others.

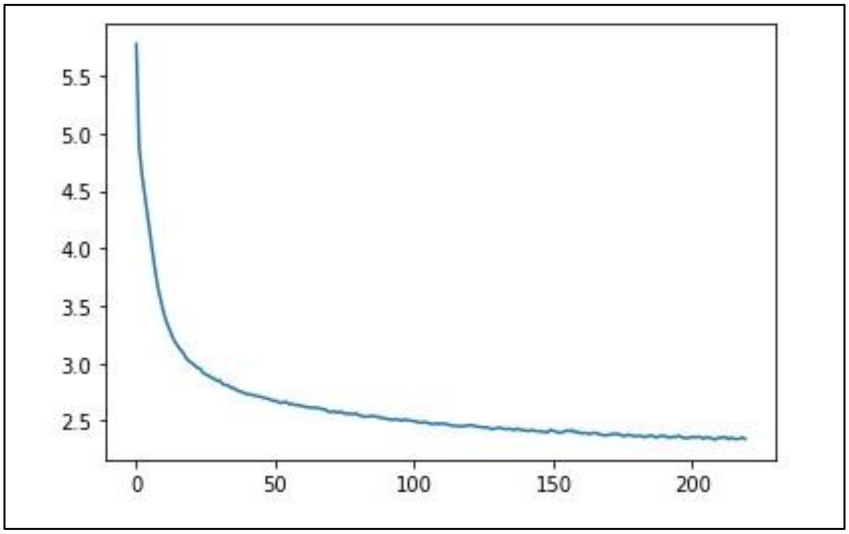


Figure 13: loss function of MLP with 1 hidden layer

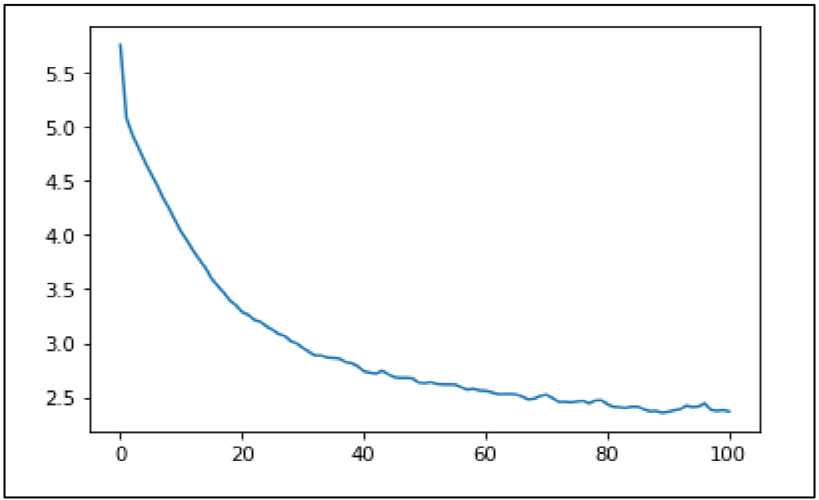


Figure 14: loss function of MLP with 2 hidden layer

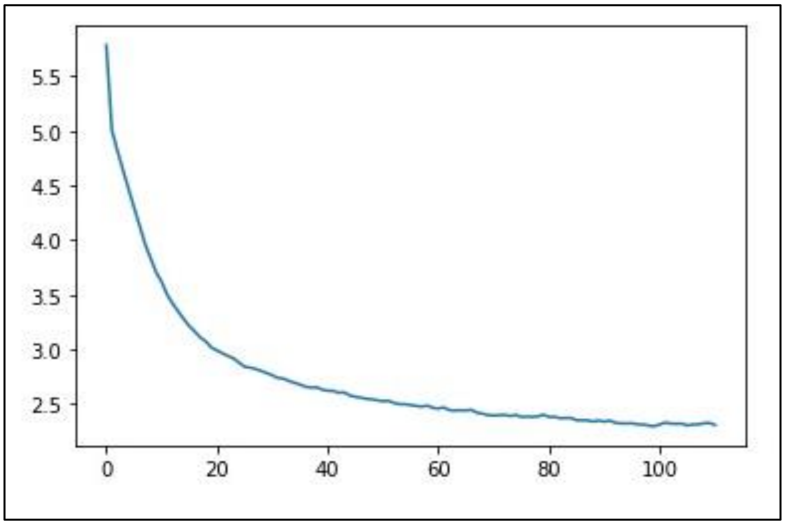
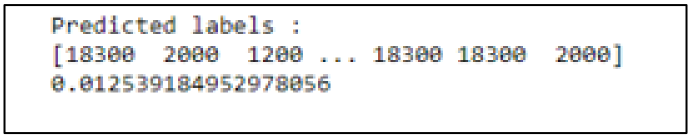


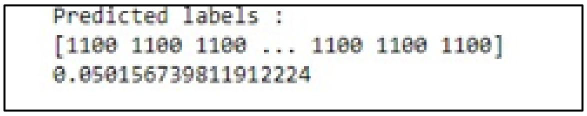
Figure 15: loss function of MLP with 3 hidden layer

We noticed that the use of two layer caches gave us a better score (equal to 5%) compared to the model with 1 and 3 hidden layer.



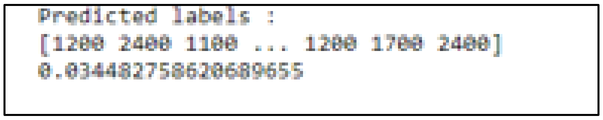
```
Predicted labels :
[18300  2000  1200 ... 18300 18300  2000]
0.012539184952978056
```

Figure 16: accuracy of MLP with 1 hidden layer



```
Predicted labels :
[1100 1100 1100 ... 1100 1100 1100]
0.050156739811912224
```

Figure 17: accuracy of MLP with 2 hidden layer



```
Predicted labels :
[1200 2400 1100 ... 1200 1700 2400]
0.034482758620689655
```

Figure 18: accuracy of MLP with 3 hidden layer

3. Results with dimensionality reduction

3.1 Financial Distress dataset

The following figures represent the results obtained after the dimensionality reduction phase.

3.1.1 PCA

After the application of the PCA we noticed an improvement in the accuracy level of the model (equal to 0.9618).

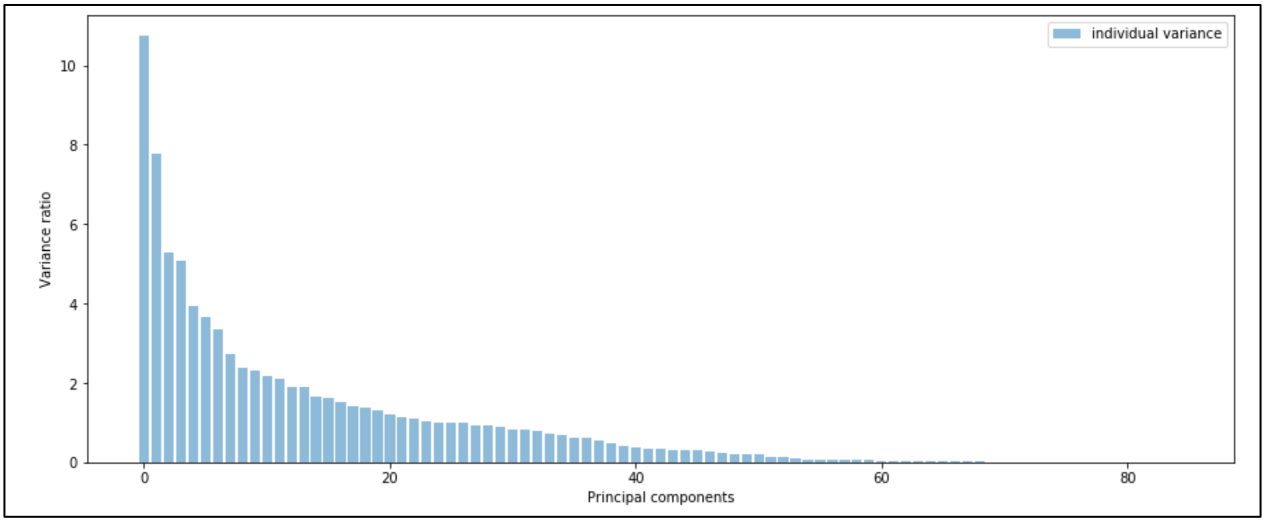
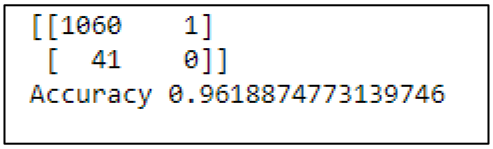


Figure 19: variance obtained by each component

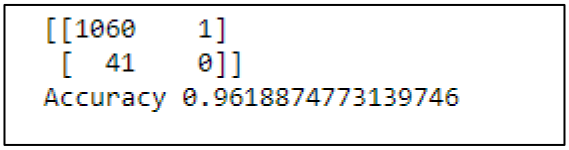


```
[[1060    1]
 [  41    0]]
Accuracy 0.9618874773139746
```

Figure 20: confusion matrix and accuracy of MLP after PCA

3.1.2 LDA

After the application of the LDA we noticed an improvement in the accuracy level of the model (equal to 0.9618).



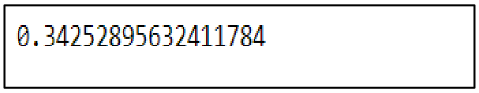
```
[[1060    1]
 [  41    0]]
Accuracy 0.9618874773139746
```

Figure 21: confusion matrix and accuracy of MLP after LDA

3.2 Online news popularity Dataset

3.2.1 PCA

After the application of the PCA we noticed an improvement in the accuracy level of the model (equal to 34.25%).

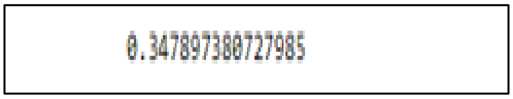


```
0.34252895632411784
```

Figure 22: accuracy of MLP after PCA

3.2.2 LDA

After the application of the PCA we noticed an improvement in the accuracy level of the model (equal to 34.79%).



```
0.347897380727985
```

Figure 23: accuracy of MLP after LDA

4. Synthesis

After comparing the results obtained before and after the reduction of dimensionality, we noticed that there is a very clear difference between the two cases and that the accuracy has increased with a significant level.

We can conclude that the use of dimensionality reduction techniques has provided us with a considerable improvement in data processing, whether in terms of speed, precision or training time.

Also showed us the importance of dimensionality reduction methods in the data processing process.

Conclusion

In this chapter, we have defined the tools and the programming language used and we have presented the results obtained before and after the dimensionality reduction phases.

General Conclusion

The work carried out in this project aims at the realization of a decision support system allowing to compare the score with and without the dimensionality reduction phase, in order to show the usefulness of dimensionality reduction in the data processing process.

We started with the context of this project and the approach and methodology used and implemented.

Then, we present the different dimension reduction techniques and classifiers used while defining the advantages and disadvantages of these techniques. After you have described and visualized the two datasets used.

Finally, we have presented the different results obtained before and after the dimension reduction phase.

This work has allowed us to gain personal and professional experience. It was very beneficial to us because we had the chance to improve our knowledge in the field of data science and that on the theoretical level, but also to discover and acquire new knowledge in terms of development with regard to practicality.

References

[1] https://stackabuse.com/dimensionality-reduction-in-python-with-scikit-learn/

[2] http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of_4.html

[3] https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learning

[4] https://deepai.org/machine-learning-glossary-and-terms/neural-network

[5] https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1

[6] https://realpython.com/jupyter-notebook-introduction/

[7] https://www.python.org/doc/essays/blurb/