Project Scoring M2: Forecasting Bankruptcy with the Linear Discriminant Analysis Method: From Theory to Practice

Théo Bacqueyrisse, Anna Blanpied, Maria Bracque Vendrell

Semester 1 - 2023

Contents

1 Introduction

When a company is in financial trouble and might go bankrupt, it can affect the loans given by banks. This often leads to banks thinking there's more risk due to the uncertainty about whether the company can pay back the money it borrowed, and they become stricter about the terms of the loans. The objective of our project is to predict whether a company is likely to go bankrupt or not. Since there are several components that influence the financial health of a company, it might be difficult to discriminate between the two classes (bankrupt and non-bankrupt companies) just by looking at the original feature space. This is why the use of the LDA method comes in handy.

There are many techniques used for classification problems. Linear Discriminant Analysis (LDA) is one of the most used methods. However, this method is often perceived as a black box and users do not really understand the logic behind it (Tharwat, Gaber, Ibrahim, and Hassanien 2017). The objective of this paper is to clarify the LDA method by looking at the theory behind it and applying it using the example of forecasting bankruptcy.

For this project, we use data coming from the Wharton Research Data Services website concerning quarterly fundamental financial data for mainly North American companies. Our clean dataset contains $340,956$ observations (3212 firms) and 34 variables.

We first implemented the LDA model from scratch on a restricted sample of our data and obtained an accuracy of 0.93, which is an encouraging result. We then applied the LDA model on our full dataset using R packages to have more performant results. We found an accuracy of 0.76, with only 28 false negative predictions out of 3212 firms in our test set.

This project is divided into four different sections. First, there is a literature review that dives into existing research and insights related to the LDA method

and bankruptcy prediction models. Then, there is a theory part that explains in detail the LDA method. Following this, there is the main part of our project, the analysis, where we will apply LDA to predict bankruptcy. Finally, a conclusion will summarize our findings, expose the limits of the project, and give some ideas for future works.

2 Literature Review

The Discriminant Analysis has the objective of separating the data into different classes. In 1936, Fisher first introduced linear discrimination for binary classes through the example of discriminating between two flowers using four different flower measurements. In 1948, Rao generalized the concept for multiple classes in 1948.

In a nutshell, Linear Discriminant Analysis (LDA) is a supervised technique that seeks to predict the class of a dependent variable using a linear combination of independent variables[1]. The objective is to find the linear combination of ratios that best discriminates between the two classes that are being classified (Deakin 1972). To do so, LDA first calculates the mean and the covariance matrix for each class in the data. Then, it computes the between-class variance which is the separability between the mean of different classes, and the within-class variance which is the separability between the mean and sample of each class. And finally, it computes the eigenvectors and eigenvalues of the scatter matrices to obtain the optimal linear discriminants.

Contemporary research studies are trying to integrate Deep Learning techniques and Discriminant Analysis. For example, Wu, Chunhua Shen, and Hengel 2017 introduced a hybrid deep architecture for person re-identification that combines Fisher vectors and deep neural networks. The objective of this approach is the same as the general LDA but with the added feature that it allows the training of the network with Stochastic Gradient Descent and back-propagate LDA gradients.

Finally, the LDA method is widely used in various fields such as biology, finance, text analysis, and image processing. In our study, we are going to focus on bankruptcy prediction.

1. Srinidhi Devan, "Discriminant Analysis - A Conceptual Understanding" Medium, January 28, 2021, https://medium.com/analytics-vidhya/discriminant-analysis-a-conceptual-understanding-c2ccc0dd2906

In the context of bankruptcy prediction, LDA was the initial statistical approach used to systematically differentiate between firms that went bankrupt and those that survived (Altman 1968). Altman (1968) used a sample of 33 bankrupt companies and 33 non-bankrupt ones and calculated a bankruptcy probability model using the discriminant analysis. He introduced five ratios that are still currently used in other bankruptcy prediction models: the Working Capital/Total Assets ratio, the Retained Earnings/Total Assets ratio, the Earnings Before Interest and Tax/Total Assets ratio, the Market Value of Equity/Total Liabilities ratio, and the Total Sales/Total Assets ratio.

Later, in a subsequent study, Daekin (1972) tried to replicate the paper by Altman (1968) by performing a Multiple Discriminant Analysis (MDA). He concluded that the MDA could show accurate results of failure prediction up to three years before bankruptcy.

One important instrument to assess the probability of default of companies is the credit score. A credit score is a numerical representation of an individual's or entity's creditworthiness. In other words, it is a prediction of how likely an individual or a company is to pay a loan back based on their financial background information. The commercial credit rating was first introduced by Bradstreet in 1857. However, it would not be until the mid-1950s that this commercial credit rating would go from commercial- to consumer-based credit evaluation[2]. During these years, the approach used was the 5C's approach (Character, Capital, Collateral, Capacity, and Condition). Yet, this technique was mainly done by hand. The development of Credit score models grew rapidly between the mid-1950s and the mid-1970s. Nowadays, the two most used credit score models are the FICO and the VantageScore. The FICO model was introduced in the 1960s by engineers William R. Fair and Earl J. Isaac. This scoring system used a statistical approach to assess credit risk and is considered a major breakthrough in the field of credit scoring.

2. Sean Trainor, "The Long, Twisted History of Your Credit Score" Time, July 22, 2015, https://time.com/3961676/history-credit-scores/

The VantageScore was introduced in 2006 by VantageScore Solutions, a company created by the three major credit reporting agencies in the United States: Equifax, Experian, and TransUnion.

During the 1980s and 1990s, there was an increase in the use of Machine Learning (ML) algorithms to try to increase the accuracy of the bankruptcy forecast models. In 1980, Ohlson was the first one to use logistic regression. In 1984, Zmijewski introduced the first probit model. He tested different potential biases caused by sample selection or data collection procedures. In 2001, Shumway developed a dynamic logit or hazard model for bankruptcy forecasting. One advantage of these ML algorithms is that no initial assumptions have to be made regarding prior probabilities of bankruptcy and the distribution of predictors. In addition, these models offered greater flexibility, allowing for the incorporation of non-financial variables and the assessment of macroeconomic indicators. Moreover, after the 2007 financial crisis, the interest in developing such models peaked (Radovanovic and Haas 2023). Simultaneously, there was an increase in the performance of ML techniques and neural networks-based techniques for financial data (Giordani, Jacobson, Schedvin, and Villani 2014).

In this section, we have summarized the main findings of the papers on Linear Discriminant Analysis and on the different bankruptcy prediction models. Next, we will go into the most theoretical part of the project by demystifying the theoretical part of the LDA.

3 Theory

The Linear Discriminant Analysis (LDA) method aims to find the linear combination of features that maximizes the separation between the two classes, also called the discriminant function. For LDA to be valid, some fundamental assumptions have to be verified. First, the sample measurements have to be independent of each other. Second, the distributions of data within each class must follow a multivariate normal (Gaussian) distribution. This assumption determines the probability density function estimation crucial to LDA's classification decision. Third, there must be homoskedasticity over the classes, meaning that the covariances are identical across classes. These assumptions make the LDA method a powerful tool when they hold, but they potentially lead to non-optimal results when they are violated.

A discriminant function for two classes is a mathematical model that helps classify data points into one of two categories or classes. It's a fundamental concept in pattern recognition, machine learning, and statistics. The goal of a discriminant function is to find a decision boundary that separates the two classes as effectively as possible. The discriminant function takes an input vector, denoted as x, and assigns it to one of the K classes, represented as C_k. In the context of LDA, we focus on linear discriminants. For our specific case of bankruptcy forecast, we have to deal with a binary classification problem where we aim to predict the default status of a firm. This binary classification involves two classes, 0 denoting 'no default' and 1 representing 'default'. The discriminant function is as follows:

$$f(x) = w^T x + w_0$$

The main objective during the training process is to determine the optimal set of parameters, usually represented as 'w'. The *weight vector* (w) defines the decision boundary separating the classes.

Once obtained, the discriminant function calculates a score or value for each input

data point to assign it to an input space as a particular class. Since we have 2 classes, the sign of this score determines the predicted class. For example, if $f(x) > 0$, the point is predicted to belong to one class, and if $f(x) < 0$, it's predicted to belong to the other class. The corresponding decision boundary is therefore defined by the relation $f(x) = 0$. This gives us the following hyper-plan:

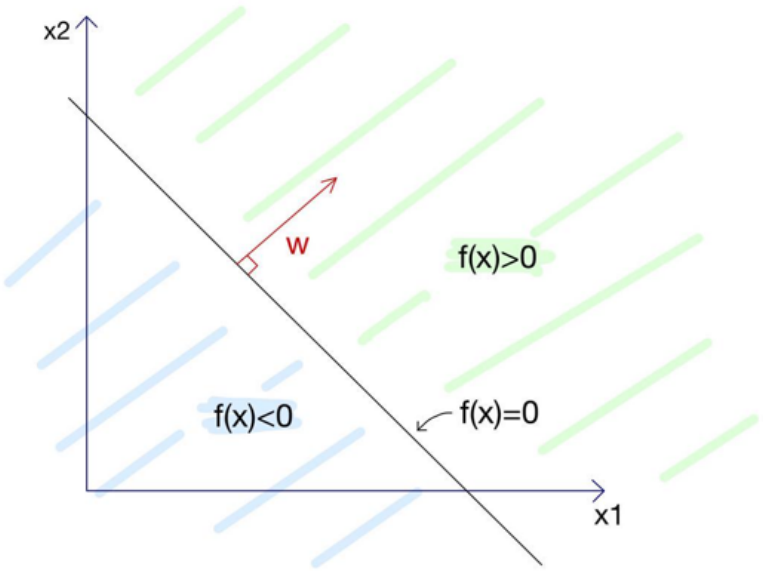

Figure 1: Two-class linear discriminant $f(x) = w_T x + w_0$

During the training phase, an optimization algorithm (e.g., gradient descent) is used to find the best set of weights that minimizes a cost function. The cost function measures the difference between the predicted class and the true class for the training data. The goal is to adjust the weights to make the discriminant function assign the correct classes to the training data.
Once the discriminant function is trained, we can use the obtained result to predict the classes of our test set.

The LDA method has some pros and cons. Let us first enumerate the pros:

- First LDA allows us to reduce the dimensionality of the data while maintaining class separation which is very useful in our case since we are

working with binary data.

- It produces a linear decision boundary that is easy to interpret.
- It tends to perform well in situations where there is an imbalance number of samples for each class, which is our case here.

And here are the cons:

- Since LDA assumes that the data can be separated by a linear decision boundary, if the relationship between the features and classes is highly nonlinear, LDA may not perform well.
- If the assumptions of homoskedasticity and multivariate normal distribution are not met, it can lead to non-optimal results.
- It can be sensitive to outliers since it relies on the calculation of means and covariances.
- It is sensitive to differences in feature scales.

To sum up this section, we have seen the fundamental idea of the Linear Discriminant Analysis. In the next sections, we are going to shift our focus to the practical part of the project. Thus, we will first introduce the data used to predict bankruptcy and how we selected the variables, and then, we will implement a Linear Discriminant Analysis from scratch.

4 Data

The Wharton Research Data Services (WRDS), is a restricted access data platform that provides access to a diverse and extensive collection of financial, economic, and business data for academic and business research. It primarily serves researchers, students, and professionals in the fields of finance, accounting, economics, and related disciplines.

We were interested in the quarterly fundamental financial data for North American companies from the Compustat database in order to retrieve detailed financial information about companies listed in North America on a quarterly basis. We chose data from 1964 to today and we used the GVKEY in order to identify the firms. GVKEY is an abbreviation commonly used in finance and research to refer to the Global Identifier Key. It is a unique alphanumeric code assigned to individual companies and entities to identify them in financial and economic databases. GVKEYs are often used to link various datasets and information about a particular company across different sources. We decided to search through the entire database to obtain as many firms as possible.

In our WRDS query, we included variables related to company assets, liabilities, debts, revenue, and stocks. We wanted to include variables that previous studies showed to be relevant, and also other variables that we thought could play a role in a company bankruptcy. We obtained a raw dataset with 1,970,374 rows from 41,713 different companies.

Compustat already includes a variable related to bankruptcy, called *dlrsn*, but previous studies showed that this variable was not completely relatable and that other sources of bankruptcy data were more accurate. This is why we decided to use the LoPucki default data which includes detailed data on companies that went out of business, especially for companies that went Chapter 7 or Chapter 11. Chapter 7 occurs when a company's assets are simply liquidated, and Chapter 11 is when a

company can continue to operate under a different organization. These two cases are widely used as bankruptcy cases, while other cases could be related to other reasons for a company to go out of business, like a merger for example.

We then merged the two datasets on the *gvkey* variable, and we created the *default* variable, taking value 1 if the *gvkey* is present in both datasets and 0 otherwise. We obtained 40,657 different companies not having defaulted, and 1,056 companies having defaulted. However, we had a lot of *NaN*. In particular, some columns were composed mainly of *NaN* values, so after trying different combinations, we decided to remove columns that contained more than 1,200,000 *NaN*. From there, we dropped any line that still contained a *NaN* value, and obtained a clean dataset of 340,956 rows, containing 15,532 companies not having defaulted, and 705 companies having defaulted. A list of the variables can be found in Appendix 1. Based on Ohlson (1980) and Barboza, Kumura, and Altman (2017), we were able to create the following new variables with the variables available:

Variable	**Description**
size	log(total assets)
TLTA	total liabilities / total assets
CLCA	total current liabilities / total current
OENEG	1 if total liabilities > total assets
NITA	net income / total assets
CHIN	change in net income (in %)
INTWO	1 if net income < 0 during the last two years
GA	change in total assets (in %)

Table 1: Created variables based on the literature

5 Analysis - LDA

5.1 Exploratory Data Analysis

First, we had a look at the summary statistics of companies that have defaulted and companies that have not defaulted yet. The first interesting thing that we notice is that defaulting companies have a Mean Total Assets value of 1,213, compared to 491 for non-defaulting companies. In addition, defaulting companies have a Total Liabilities mean value of 958.7, whereas non-defaulting companies have a mean of 284.69.

Second, let us have a look at the correlation matrix to see strongly interacting variables, and the correlation coefficients related to the variable *default*. Here are these values:

Variable	Correlation	Variable	Correlation
company_id	-0.02	tot_income_taxes...26	0.00
fiscal_year	0.00	long_term_debt_issuance	0.07
fiscal_quarter	0.00	long_term_debt_reduction	0.06
tot_current_assets	0.08	equity_net_loss_earnings	0.00
tot_assets	0.05	income_before_extra_items	-0.03
comm_ord_equity	0.00	tot_revenue...31	0.08
cash_shortterm_inv	0.03	tot_income_taxes...32	0.00
debt_curr_liabilities	0.06	deletion_reason	0.09
tot_long_term_debt	0.08	default	1.00
tot_curr_liabilities_other	0.06	size	0.23
tot_curr_liabilities	0.08	TLTA	-0.01
tot_long_term_liabilities	0.08	CLCA	-0.00
tot_liabilities_stholder_equity	0.05	OENEG	0.04
tot_liabilities	0.08	NITA	0.00
net_income	-0.02	CHIN	-0.01
tot_revenue...24	0.09	GA	0.00
payable_income_taxes	0.02	INTWO	-0.01

We see for example an interesting correlation of 0.23 between the logarithm of *tot_assets* and the variable *default*. We also see that variables related to equities of the company seem uncorrelated to *default*.

We can also visualize the interactions between variables with a Heatmap of the

Correlation Matrix:

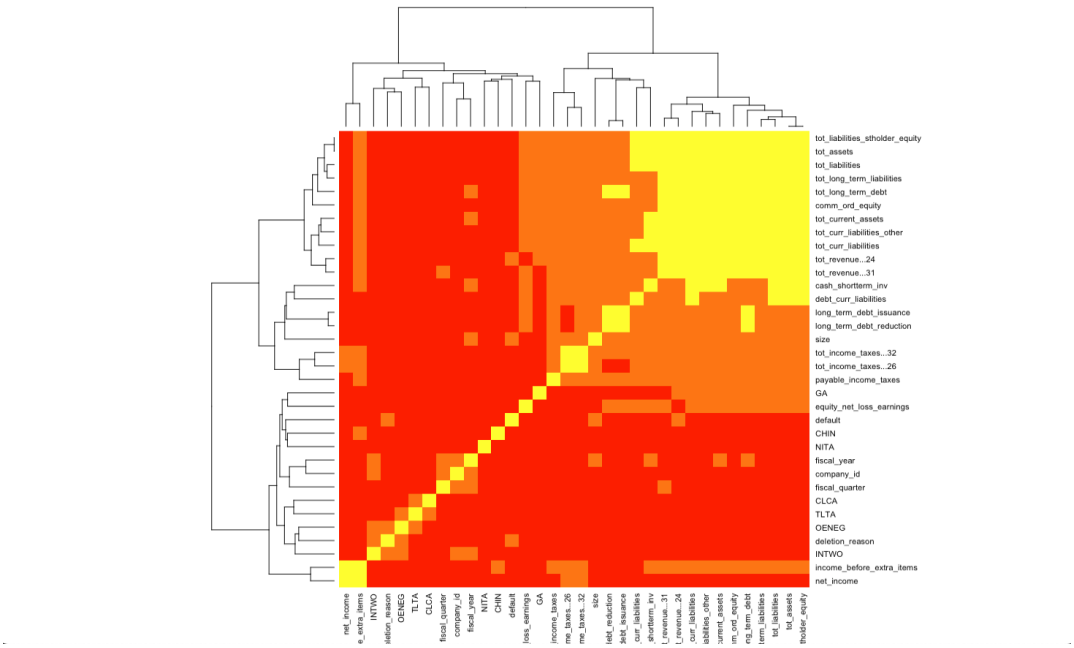

Figure 2: Heatmap of the Correlations of our variables

Now, let us have a look at the evolution of Total Liabilities over time for a given Defaulting Company. Just below, we can find the evolution of Total Liabilities for ACETO CORP, which defaulted in 2020:

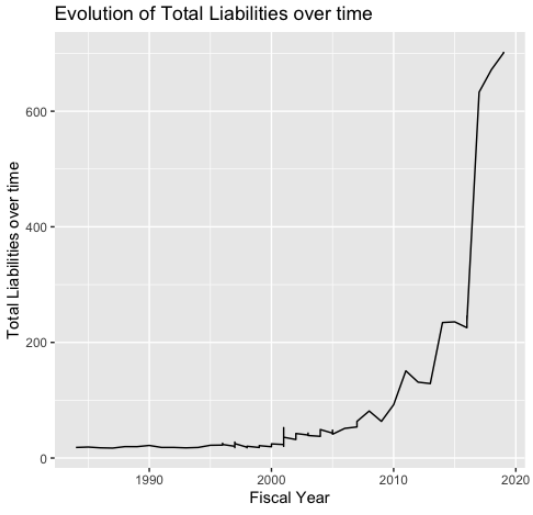

Figure 3: Evolution of ACETO CORP Total Liabilities over time

What we can see on this graph is that for this defaulting company, the Total Liabilities increased significantly in the final years of their existence. This suggests

that the total liabilities of a company could be a good indicator of a possible future default.

Next, let us look into the relation between the *tot_assets* and *tot_liabilities* variables, and its differentiation according to the value of default. We can find below a scatterplot giving the relation between these variables, colored by the value of default for the fiscal year 2009.

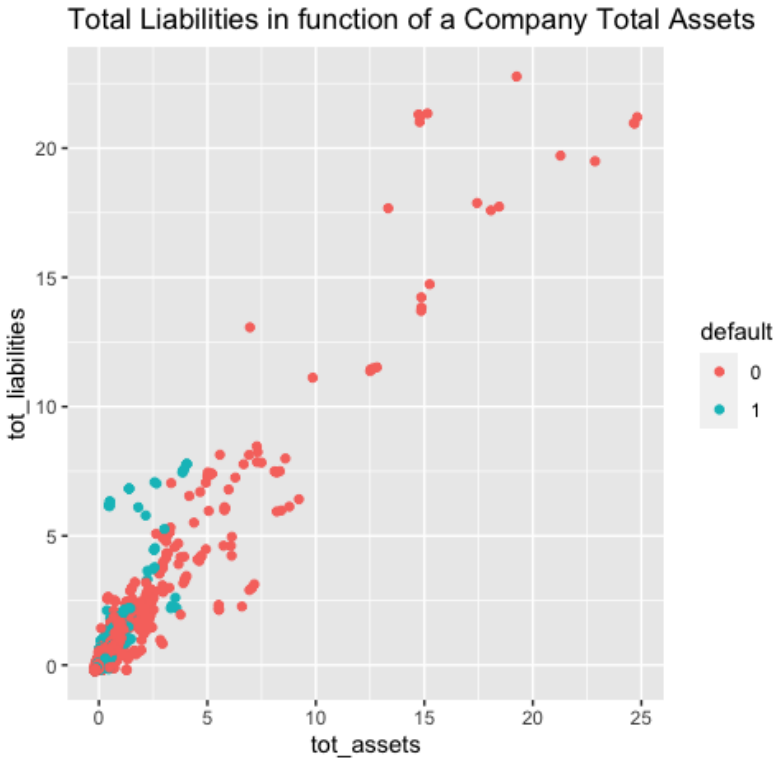

Figure 4: Total Liabilities according to a company Total Assets

We can see that defaulting companies may tend to have higher Total Liabilities given their Total Assets compared to Non defaulting companies, which indicates the role that can play these variables in our future predictions, and especially the ratio of both variables that was used in the literature by Ohlson (1980).

Let us now plot the distribution of the Year on which defaulting companies have defaulted. This is useful to better understand the data, and we can use it later to split our data into training and testing sets.

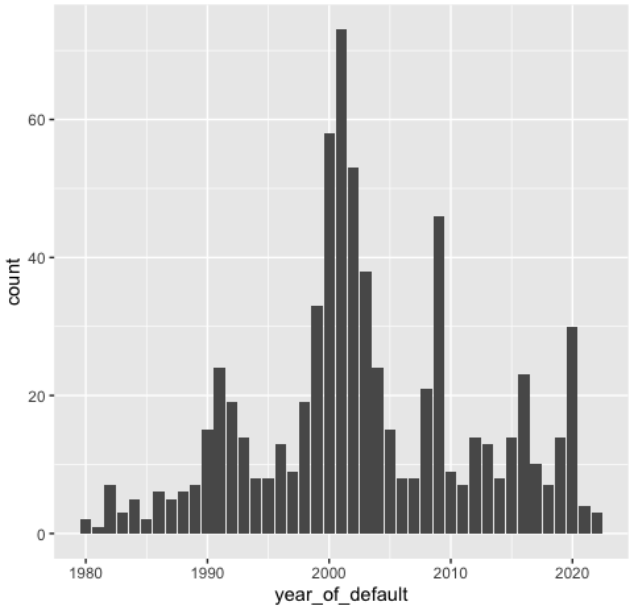

Figure 5: Histogram of the Year on which defaulting companies have defaulted

We see here that many companies defaulted between 1995 and 2005, while few companies defaulted before 1980 and after 2010. However, peaks in the number of defaulting companies around 2010 and 2020 may be highly caused by the financial and the Covid crisis.

Finally, we can look at the variables we created from the literature. For example, we can see the relation between the variable *size* and the variable *default* in the following plot:

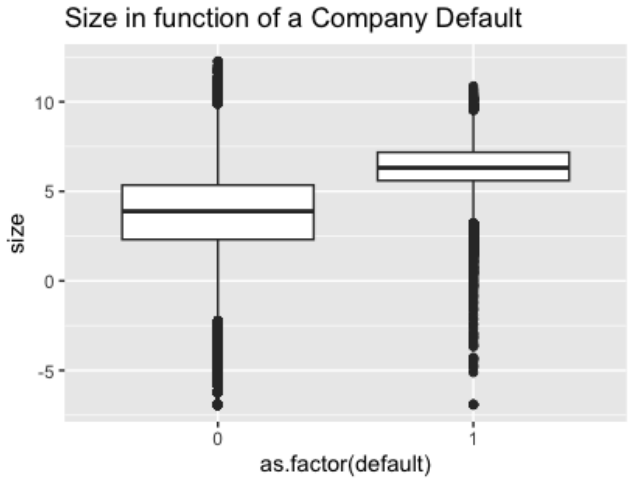

Figure 6: Boxplots of the size variable separated by default variable

We indeed see that Defaulting Companies tend to have much higher *log(tot_assets)* values than non-default companies, since all the values greater than the 1st quartile for $default = 1$ are above all the values below the 3rd quartile for $default = 0$, which indicates that this variable could play a major role in the prediction of default.

5.2 Data Preprocessing

Feature Selection

Given the variables at our disposal, we decided to only work with the numerical variables. Indeed, the other variables were not relevant to our study since they contained information such as the currency used by the firm for example.

Dropping the Outliers

We noticed that the dataset contained a significant number of outliers for many variables. In order to help us in the decision of which observations to drop, we computed the z_scores of all the variables in the data using this formula:

$$z\_score = \frac{x-\mu}{\sigma}$$

Thus, we dropped the observations that had an absolute value of *z_score* higher than 3.

Dropping the Nearly Colinear Variables

The next step was to remove the variables that appeared to be too strongly correlated. So using our correlation matrix, we removed one of two variables that had a correlation coefficient higher than 0.8.

We have ended up with a dataset made of 327,681 rows and 27 columns. We actually took only the first 26 columns, removing the date of default in order to keep our prediction reliable, considering that non-default companies do not have a date of default.

5.3 Logistic Regression as Benchmark

We first implement a Logistic Regression model to our preprocessed data in order to have a benchmark of results for predicting default.
In order for the results to be comparable, we will first use the same parameters for both models, so that difference in performance will only be explained by the fact that a model would be more adapted to solve our problem than the other one. This is why we will give a 60 % class importance to class 1 for both models.
Then, we will try to obtain the best results we can with a Logistic Regression.

For the model with parameter of 0.6 we obtained the following confusion matrix :

	$Class0(True)$	$Class1(True)$
$Class0(Prediction)$	$TN = 3000$	$FN = 121$
$Class1(Prediction)$	$FP = 71$	$TP = 20$

and the following metrics :

Metrics	**Value**
Accuracy	0.94
Precision	0.21
Recall	0.14
F1-Score	0.17
AUC	0.559

This model gives a good accuracy, which is mainly explained by the high rate of well predicted class 0. However, we observe that the other metrics are very low, and more importantly, the rate of well predicted class 1 (*Recall*) is only 14% which is not encouraging since this metric is the one that we want to maximise.
In order to improve this, we then run the model with parameters 0.95 / 0.05, meaning that we will give a 95 % class importance to class 1, and obtained the

following confusion matrix :

	$Class0(True)$	$Class1(True)$
$Class0(Prediction)$	$TN = 1520$	$FN = 19$
$Class1(Prediction)$	$FP = 1551$	$TP = 122$

and the following model metrics :

Metrics	**Value**
Accuracy	0.51
Precision	0.07
Recall	0.86
F1-Score	0.13
AUC	0.68

With this new model, we succeeded to improve the *Recall* and AUC statistics. However all the other metrics are worst off. We will use these two models as benchmark in order to analyse the results of our LDA model.

5.4 Linear Discriminant Analysis implementation

5.4.1 Implementation "by hand"

In order to implement the LDA method, we first needed to divide the dataset into two subsets - a training set and a test set. To do so, we used a 70-30 ratio, 70% of the data used for training and 30% for testing the predictive model. We also had to isolate the numeric columns of our dataset, to focus on only the numerical predictors for analysis.

```
data_nmc <- mydata %>% select_if(is.numeric)
set.seed(42)
train_index <- sample(1:nrow(data_nmc), 0.7 * nrow(data_nmc))
train_data <- data_nmc[train_index, ]
test_data <- data_nmc[-train_index, ]
```

Figure 7: Splitting test and train set

Then, we computed the *within-class* scatter matrix and the *between-class* scatter matrix. Indeed, as mentioned in the literature review section, these two matrices are important to help find a linear transformation of the data that maximizes the separation between different classes. We also computed the class means for both classes 'default = 0' and 'default = 1' based on the training data.

```
# Calculating within-class and between-class scatter matrices
within_class_scatter <- matrix(0, ncol(train_features), ncol(train_features))
between_class_scatter <- matrix(0, ncol(train_features), ncol(train_features))

# Calculate class means
class_means <- tapply(train_features, train_labels, colMeans)


# Calculate scatter matrices
for (class in unique(train_labels)) {
  class_data <- train_features[train_labels == class, ]
  class_mean_diff <- class_data - class_means[class + 1]
  within_class_scatter <- within_class_scatter + as.matrix(t(class_mean_diff))
  %*% as.matrix(class_mean_diff)
}

overall_mean <- colMeans(train_features)
for (class in unique(train_labels)) {
  n_class <- sum(train_labels == class)
  class_mean_diff <- t(data.frame(class_means[class+1]))- overall_mean
  between_class_scatter <- between_class_scatter + n_class * as.matrix(t(class_mean_diff))
  %*% as.matrix(class_mean_diff)
}
```

Figure 8: Computation of within and between classes

After, we calculated the eigenvectors and eigenvalues of the regularized within-class scatter matrix multiplied by the between-class scatter matrix, and we selected the top eigenvectors corresponding to the highest eigenvalues to transform the training data into a new space.

```
# Regularize within-class scatter matrix
within_class_scatter_reg <- within_class_scatter + 0.01 * diag(ncol(within_class_scatter))

# Calculate eigenvalues and eigenvectors
eigen_result <- eigen(solve(within_class_scatter_reg) %*% between_class_scatter)
eigenvalues <- eigen_result$values
eigenvectors <- eigen_result$vectors

# Select top eigenvectors
k <- 1  # Choisissez le nombre de vecteurs propres à utiliser
top_eigenvectors <- eigenvectors[, 1:k]
```

Figure 9: Computation of eigenvectors

In order to train the model, we determined the model coefficients (weights) for the LDA by computing the matrix inverse of the regularized within-class scatter matrix and multiplying it by the difference between the class means. We also computed the intercept by using the calculated coefficients and class means.

```r
# Transform training data into the new space
train_lda <- as.matrix(train_features) %*% top_eigenvectors

# Train the LDA model - calculate coefficients and intercept
mat <- as.matrix(data.frame(class_means[2])-data.frame(class_means[1]))
coefficients <- as.matrix(solve(within_class_scatter_reg)) %*% mat
intercept <- -0.5 * (as.matrix(t(data.frame(class_means[1])))+
                       as.matrix(t(data.frame(class_means[2])))) %*% coefficients
```

Figure 10: Computation of the model coefficient

Finally, we used the trained LDA model to predict the default status for the test set, and we evaluated these predictions using a confusion matrix, the accuracy, and the precision statistics.

```r
# Prediction on the test set
test_features <-  test_data[, -1]
adjusted_intercept <- rep(intercept, nrow(test_features))
predicted <- as.numeric(as.matrix(test_features) %*% as.matrix(coefficients)
                        + adjusted_intercept > 0)
```

Figure 11: Prediction

We decided to apply the model to a sample of the dataset containing 748 observations since the goal was just to understand how it works. Using the model created before, we obtained the following confusion matrix:

	$Class0(True)$	$Class1(True)$
$Class0(Prediction)$	$TN = 141$	$FN = 4$
$Class1(Prediction)$	$FP = 7$	$TP = 9$

And the following metrics:

Metrics	**Value**
Accuracy	0.93
Precision	0.56
Recall	0.69
F1-Score	0.62

We can observe a high level of accuracy here, indicating the model's strong predictive capabilities. Moreover, the notably low count of false negatives is promising, as minimizing this value is crucial in our analysis. Indeed, a type-I error of predicting default when a company does not default is in our eyes less of a problem than

predicting a company will not default when it actually will. Hence the importance of a high Recall value.

5.4.2 Implementation using R package

First, we separated our dataset into a train and a test set. We performed this train-test split on the *company_id* variable. We selected randomly 80% of the company IDs from default and 80% of the company IDs from non-default companies to go into our train set, and the remaining 20% to go into our test set. This ensures that train and test sets contain the same proportion of default and non-default companies to avoid any bias. We verified the process by checking that the proportion of default companies in both samples is around 5.5%.

We then created our LDA model using the *lda()* function of the package *MASS* of R. We chose to set the parameter *prior* to *c(0.4, 0.6)*. This was done to tackle the strong imbalance of classes on the data, and it will make the model give more relative importance to the prediction of class 1 of *default*.

Finally, we evaluated our model predictions on our test set to see how well default companies were classified. To this end, we added the predictions to the test set as a new variable, and then grouped the observations by *company_id*, to see the predictions company-wise instead of observation-wise which would be misleading to interpret.

Here is the confusion matrix obtained by our model on the Test set:

	$Class0(True)$	$Class1(True)$
$Class0(Prediction)$	$TN = 2354$	$FN = 28$
$Class1(Prediction)$	$FP = 717$	$TP = 113$

From this confusion matrix, let us show some metrics of our results we can get the following metrics:

From these results, we can say that the poor values for precision and *F1-Score* are

Metrics	Value
Accuracy	0.76
Precision	0.13
Recall	0.80
F1-Score	0.23
AUC	0.784

mainly due to the imbalance of classes. Indeed, since $Precision = \frac{TP}{TP+FP}$, and $Accuracy = \frac{2*Precision*Recall}{Precision+Recall}$, the high value of false positive compared to true positive cause bad values for these metrics. However, the value of *Recall* ($= \frac{TP}{TP+FN}$) being 0.80 suggests that we are able to predict correctly 80% of the Default Companies, which can be valuable for banks for example. Indeed, Recall measures how well Positives are predicted, which was the main goal of this project. Indeed, as mentioned before, a type-I error of predicting default when a company does not default is in our eyes less of a problem than predicting a company will not default when it actually will. Hence the importance of a high *Recall* value.

Looking at our confusion matrix, we see that we have approximately $\frac{1}{3}$ or errors for the 0 class, and $\frac{1}{4}$ of errors for the 1 class, suggesting that we produce slightly better results for the class 1, and this is mainly due to the parameter *prior* we chose when training the model. These results are better than the ones we obtained with the Logistic Regression model, which leads us to prefer the LDA model over the Logistics Regression one.

Here is the ROC curve we obtained out of predictions grouped by Companies:

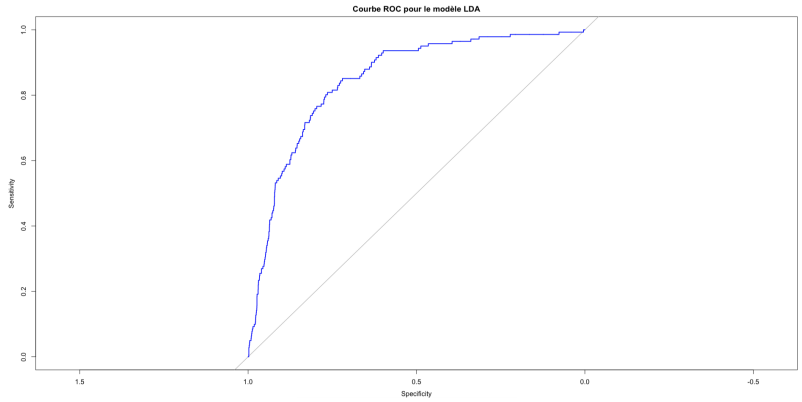

Figure 12: ROC Curve of the LDA model predictions grouped by companies

Finally, the most interesting plot is to visualize how well our model separated Default and non-default companies. To this end, we can plot the value of the LDA component for each of the companies in our test set. We know that a company is classified as 1 if the component is positive, and as 0 if the component is negative. Here is the obtained plot:

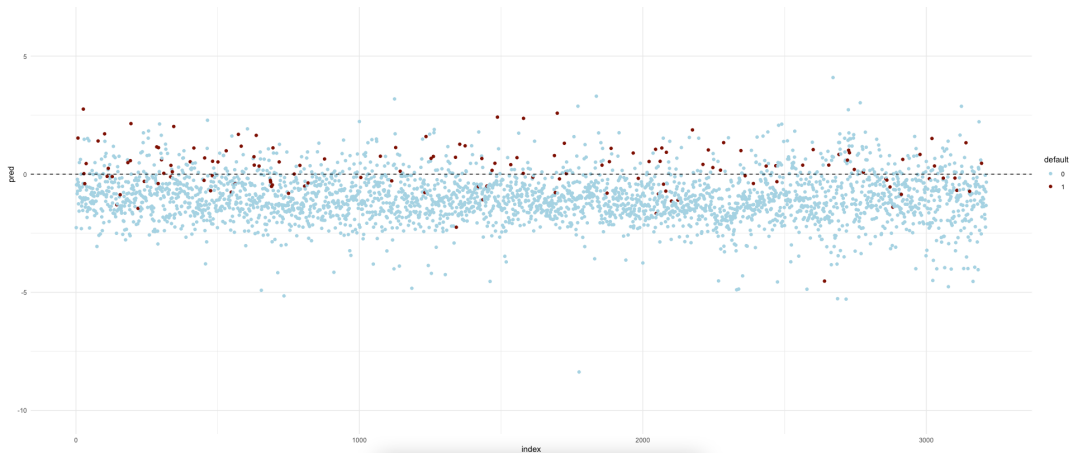

Figure 13: Scatterplot of the results obtained by LDA

6 Conclusion

In conclusion, this project looked at how linear discriminant analysis works, particularly in the context of bankruptcy prediction. We started by understanding the fundamental principles of Linear Discriminant Analysis (LDA) and its basic components.

To perform the analysis we used quarterly fundamental financial data from 1964 to today from the WRDS platform, and obtained a dataset containing $340,956$ rows and 34 variables. The objective was to be able to predict if a firm would default according to these variables using an LDA model.

First, we implemented a Linear Discriminant Analysis (LDA) by hand using a small dataset of 748 observations. The results we have are very promising results taking into account the small size of the sample. We achieved an accuracy of 0.93, a precision of 0.56, a recall of 0.69, and an F1-score of 0.62. Thus, the model exhibits high accuracy, but there is room for improvement in precision. The trade-off between precision and recall is reflected in the F1-Score, which suggests a moderate overall performance. Furthermore, the notably low count of false negatives (4) is encouraging, as minimizing this value is crucial in our analysis.

Then, we used an R package to create a prediction model. In particular, we used the lda() function of the package MASS. We achieved an accuracy of 0.76, a precision of 0.13, a recall of 0.80, an F1-score of 0.23, and an AUC of 0.784. The model shows a strong ability to identify a large portion of actual positive instances, as indicated by the high recall rate. However, there is room for improvement in finding a better balance between precision and recall, as suggested by the lower precision and F1-Score. The AUC value indicates that the model has good discriminatory power in distinguishing between positive and negative instances. To refine the model further, it's important to consider the specific goals and requirements of the analysis.

During the realization of this project, we faced several limitations. First, the

feature selection of the variables was complicated due to the huge amount of variables available and the difficulty of retrieving them. Thus, we might have omitted important variables that could have had a great impact on bankruptcy prediction. Second, the quality of our dataset was not ideal. Indeed, we had variables that have been used in almost all the previous papers such as EBITA, but we had to delete them due to the important amount of missing values. Third, we did not take into account the temporal aspect of the bankruptcy. Thus our project does not predict if a company will default in the next years, but rather if a company is likely to default. Finally, we did not verify all the assumptions of the LDA and it might have affected the performance of our model.

To expand our project, we could do further research on how to integrate LDA with Neural network structures to enhance predictive models. Indeed, investigating how these techniques can complement each other may contribute to the development of more robust and accurate predictive models, especially in scenarios where intricate patterns and relationships in the data may be effectively captured by neural networks. Notably, Alahmadi et al. (2022) demonstrated the success of this integration in periocular recognition, surpassing state-of-the-art methods in unconstrained environments. They introduced a Linear Discriminant Analysis Convolution Neural Network (LDA-CNN) to deal with the degradation of the discriminant power of the features extracted from the periocular trait caused by non-ideal conditions in real-world image capture. Expanding on this success, it would be interesting to explore the application of similar integrated approaches in other fields, such as bankruptcy predictions.

Bibliography

Alahmadi, Amani, Muhammad Hussain, and Hatim Aboalsamh. 2022. "LDA-CNN: Linear Discriminant Analysis Convolution Neural Network for Periocular Recognition in the Wild." *Mathematics* 10 (23). ISSN: 2227-7390. https://doi.org/10.3390/math10234604. https://www.mdpi.com/2227-7390/10/23/4604.

Altman, Edward I. 1968. "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy." *The Journal of Finance* 23 (4): 589–609. ISSN: 00221082, 15406261, accessed October 24, 2023. http://www.jstor.org/stable/2978933.

Barboza, Flavio, Herbert Kimura, and Edward Altman. 2017. "Machine learning models and bankruptcy prediction." *Expert Systems with Applications* 83:405–417. ISSN: 0957-4174. https://doi.org/https://doi.org/10.1016/j.eswa.2017.04.006. https://www.sciencedirect.com/science/article/pii/S0957417417302415.

Bardos, Mireille. 1998. "Detecting the risk of company failure at the Banque de France." *Journal of Banking & Finance,* https://doi.org/https://doi.org/10.1016/S0378-4266(98)00062-4.

———. 2007. "What is at stake in the construction and use of credit scores?" *Comput Econ 29, 159–172,* https://doi.org/https://doi.org/10.1007/s10614-006-9083-x.

Deakin, Edward B. 1972. "A Discriminant Analysis of Predictors of Business Failure." *Journal of Accounting Research* 10 (1): 167–179. ISSN: 00218456, 1475679X, accessed October 25, 2023. http://www.jstor.org/stable/2490225.

Efron, Bradley. 1975. "The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis." *Journal of the American Statistical Association, Vol. 70, No. 352,* https://doi.org/https://doi.org/10.2307/2285453.

Fisher, Ronald A. 1936. "The use of multiple measurements in taxonomic problems." *Annals of eugenics 7.2,* https://doi.org/https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Giordani, Paolo, Tor Jacobson, Erik von Schedvin, and Mattias Villani. 2014. "Taking the Twists into Account: Predicting Firm Bankruptcy Risk with Splines of Financial Ratios." *The Journal of Financial and Quantitative Analysis* 49 (4): 1071–1099. ISSN: 00221090, 17566916, accessed October 25, 2023. http://www.jstor.org/stable/43303979.

Nur, Triasesiarta, and Rosinta Panggabean. 2020. "Accuracy of Financial Distress Model Prediction: The Implementation of Artificial Neural Network, Logistic Regression, and Discriminant Analysis" (May). https://doi.org/10.2991/assehr.k.200529.084.

Ohlson, James A. 1980. "Financial Ratios and the Probabilistic Prediction of Bankruptcy." *Journal of Accounting Research* 18 (1): 109–131. ISSN: 00218456, 1475679X, accessed November 1, 2023. http://www.jstor.org/stable/2490395.

Radovanovic, Jelena, and Christian Haas. 2023. "The evaluation of bankruptcy prediction models based on socio-economic costs." *Expert Systems with Applications* 227 (April): 120275. https://doi.org/10.1016/j.eswa.2023.120275.

Rao, C. Radhakrishna. 1948. "The Utilization of Multiple Measurements in Problems of Biological Classification." *Journal of the Royal Statistical Society. Series B (Methodological)* 10 (2): 159–203. http://www.jstor.org/stable/2983775.

Sumitkrsharma. *What is Linear Discriminant Analysis ?—Assumptions of Linear Discriminant Analysis — How LDA makes predictions ?—Advantages and Disadvantages of LDA*. Accessed: 2023-10-24. https://sumit-kr-sharma.medium.com/what-is-linear-discriminant-analysis-assumptions-c940f366a463.

Tharwat, Alaa, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. 2017. "Linear discriminant analysis: A detailed tutorial." *Ai Communications* 30. https://doi.org/10.3233/AIC-170729.

Trainor. *The Long, Twisted History of Your Credit Score.* Accessed: 2023-11-01. https://time.com/3961676/history-credit-scores/.

Wu, Lin, Chunhua Shen, and Anton van den Hengel. 2017. "Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification." *Pattern Recognition* 65:238–250. ISSN: 0031-3203. https://doi.org/https://doi.org/10.1016/j.patcog.2016.12.022. https://www.sciencedirect.com/science/article/pii/S0031320316304447.

Zmijewski, Mark E. 1984. "Methodological Issues Related to the Estimation of Financial Distress Prediction Models." *Journal of Accounting Research* 22:59–82. ISSN: 00218456, 1475679X, accessed November 1, 2023. http://www.jstor.org/stable/2490859.

Appendices

Appendix 1: Variables

Variable	Description
company_id	Company identifier
date	Date of financial data
fiscal_year	Fiscal year of the data
fiscal_quarter	Fiscal quarter of the data
industry_format	Format of industry classification
conso_level	Consolidation level
pop_source	Source of population data
data_format	Format of financial data
cusip_id	CUSIP identifier
company_name	Company name
currency	Currency used
tot_current_assets	Total current assets
tot_assets	Total assets
comm_ord_equity	Common ordinary equity
cash_shortterm_inv	Cash and short-term investments
debt_curr_liabilities	Debt on current liabilities
tot_long_term_debt	Total long-term debt
tot_curr_liabilities_other	Total current liabilities (other)
tot_curr_liabilities	Total current liabilities
tot_long_term_liabilities	Total long-term liabilities
tot_liabilities_stholder_equity	Total liabilities and shareholders' equity
tot_liabilities	Total liabilities
net_income	Net income
tot_revenue	Total revenue
payable_income_taxes	Payable income taxes
tot_income_taxes	Total income taxes
long_term_debt_issuance	Long-term debt issuance
long_term_debt_reduction	Long-term debt reduction
equity_net_loss_earnings	Equity net loss or earnings
income_before_extra_items	Income before extraordinary items
company_status	Company status
business_desc	Business description
city	City of the company
deletion_reason	Reason for deletion
location	Location of the company
state	State of the company
deletion_date	Date of deletion
Date_of_Default	Date of default
default	Default status

Table 2: Variables