The impact of the COVID-19 crisis on the credit risk of the accommodation services sector in Portugal: a contingent claims approach

Catalina Serrano Rosero

Dissertation written under the supervision of Professor Diana Bonfim, with the collaboration of Industry Expert Nuno Silva

Dissertation submitted in partial fulfilment of requirements for the MSc in Finance, at the Universidade Católica Portuguesa, June 2021.

Abstract

Title: "The impact of the COVID-19 crisis on the credit risk of the accommodation services sector in Portugal: a contingent claims approach"

Author: Catalina Serrano Rosero

Keywords: COVID-19, Structural Models, Machine Learning, Credit Risk, Accommodation Sector, Private Firms, Portugal.

This dissertation aims to study the impact of the COVID-19 crisis on the credit risk of the accommodation services sector in Portugal using a structural model based on the one developed by Eisdorfer, Goyal, & Zhdanov (2019). Three major changes were implemented in the model: i) it was adapted to assess the impact of the pandemic crisis, ii) it was considered that firms have a cash account, and iii) a different decomposition of operating costs was made. Using the Central Balance Sheet - Harmonized Panel (CBHP) and the Fast and Exceptional Enterprise Survey (COVID-IREE) databases it was possible to implement the model to private Portuguese firms. The study is performed on four representative firms, that were constructed using a hybrid machine learning approach. The clustering variables - Adjusted Return on Assets, Fixed-Asset Turnover, Current Ratio and Long-Term Debt - were selected aiming to contemplate a different aspect regarding the firms' pre-COVID-19 financial situation. While the results obtained indicate that all of the representative firms would see a decrease in their distance to default due to the pandemic shock, the dimension of the effect is heterogeneous on the four studied firms. The results also seem to confirm the intuition that firms with a better cash position before the shock suffer a less negative impact on their levels of credit risk. Additionally, less fixed obligations seem to influence firm's chances of survival.

Resumo

Título: "O impacto da crise de COVID-19 no risco de crédito do setor de alojamento em Portugal: uma abordagem de reivindicações contingentes"

Autora: Catalina Serrano Rosero

Palavras-Chave: COVID-19, Modelos Estruturais, Machine Learning, Risco de Crédito, Setor de Alojamento, Empresas Privadas, Portugal.

Esta dissertação tem o objetivo de estudar os impactos da crise de COVID-19 no risco de crédito no setor de alojamento em Portugal utilizando um modelo estrutural baseado naquele desenvolvido por Eisdorfer, Goyal, & Zhdanov (2019). Três alterações principais foram implementadas no modelo: i) ele foi adaptado para medir o impacto da crise da pandemia, ii) foi considerado que as empresas possuem uma conta de caixa, e iii) uma decomposição diferente para os custos operacionais foi feita. As bases de dados Central de Balanços (CBHP) e Inquérito Rápido e Excecional às Empresas (COVID-IREE) foram utilizadas para implementar o modelo para empresas privadas portuguesas. O estudo foi realizado em quatro empresas representativas, que foram contruídas utilizando um método híbrido de agrupamento com machine learning. As variáveis de agrupamento escolhidas – Retorno de Ativos Ajustado, Turnover de Ativos Fixos, Rácio de Liquidez, e Dívida de Longo Prazo – foram selecionadas com o objetivo de contemplar diferentes aspetos referentes à situação financeira das empresas pré-COVID-19. Apesar de todas as empresas representativas registarem uma redução na sua distância ao incumprimento em resultado do choque pandêmico, a dimensão do efeito é heterogêneo nas quatro empresas estudadas. Os resultados também parecem confirmar a intuição de que empresas com uma posição de caixa melhor antes do choque sofrem um impacto menos negativo nos seus níveis de risco de crédito. Além disso, ter obrigações fixas mais baixas parece influenciar as chances da empresa de sobrevivência.

Acknowledgements

My most sincere and appreciative thank you to those that have been a part of my formative years, that shaped me into the person I am today.

First of all, I am extremely grateful for the professors and mentors that made me ask questions and pushed me to go the extra mile. I also thank all of those that worked behind the scenes in allowing me to have a good environment to study and work.

Thank you to dear the friends I’ve made throughout my life - I am lucky to have stumbled upon you along the way and I hope that I could have made a positive impact in your life like you have made in mine.

The biggest thank you of all is to my family. I thank you with all my heart not only for allowing me to have the opportunities I have had, but also for teaching me the value of education and hard work. I love you more than words can describe.

Finally, a very special thank you to Professor Nuno Silva, without whom it wouldn’t have been possible to deliver this dissertation in the first place. I appreciate all of your patience and willingness to share your knowledge. Last but not least, thank you to Professor Diana Bonfim for allowing me to explore a bit more of the world of credit risk.

Table of Contents

List of Equations

List of Tables

List of Figures

1. Introduction

1.1. Context and motivation

The year of 2020 was marked by what many called a “new normal”: the COVID-19 pandemic forced people around the world to change their daily routines and habits – impacting businesses (and economies) along the way. In Portugal, with two lockdowns and restrictive measures of various degrees implemented since March of 2020, the Gross Domestic Product (GDP) was estimated to tumble 7.8% on the yearly comparison (Statistics Portugal, 2020).

In 2019, Tourism represented 15.4% of Portugal’s GDP, out of which restaurant and accommodation services were estimated to represent about 50% – having the largest impact on the sector’s wealth creation (Statistics Portugal, 2020). Additionally, according to Portugal’s Tourism Satellite Accounts, the sector was more dynamic than the rest of the economy during that year as well. Needless to say, there was a big contrast between 2019 and 2020. As travel restrictions were implemented, accommodation services were heavily impacted by the crisis triggered by the pandemic. While the borders were not fully closed throughout the entire year, the sector expected a significant decrease in turnover compared to previous years.

Using the Central Balance Sheet - Harmonized Panel (CBHP) and Fast and Exceptional Enterprise Survey (COVID-IREE) data, it is possible to study Portuguese private firms using information that is not publicly available. Generally, traditional statistical models such as the Altman Z-Score are applied to assess the credit risk of this type of firms, however, an alternative that allows to consider explicitly the pandemic shock are structural models.

Having a precise risk assessment would be ideal at any period but considering that the effects of the COVID-19 crisis are still unfolding, it is especially important now. Having a clear notion of patterns, along with their causes and catalysts, can be an important tool for managers on the private and public spheres.

1.2. Goals and structure

The main goal of this dissertation is to explore the impact of the COVID-19 crisis on the credit risk of firms in the Portuguese accommodation sector from an aggregate perspective, rather than at the individual level. This was done as an attempt to identify patterns among firms and obtain a better understanding of the underlying factors that could influence their chances of survival after the shock. The recognition of patterns is important not only to help firms prepare for future shocks, but also to assist decision makers when there is plentiful information available.

This dissertation is structured as follows. Chapter 2 gives an overview of the importance of credit risk measurements, reviews briefly the approaches that can be used to do so (with an emphasis on structural models) and discusses the methods that can be used to overcome the challenges in estimating the credit risk for private firms. Chapter 3 explains the model used by Eisdorfer, Goyal, & Zhdanov (2019) and the modifications that were made for it to be implemented in this dissertation. Chapter 4 starts by describing the original data and its cleaning process, then it explains the method used to build the representative firms, and finally details how the inputs of the model were calibrated. Results are reported on Chapter 5, while Chapter 6 contains conclusions and possible topics for further research.

2. Literature Review

2.1. On the importance of credit risk measurement

The assessment of a firm's credit risk is of great importance for banks, investors, asset managers, rating agencies and firms in general (Altman, Iwanicz-Drozdowska, Laitinen, & Suvas, 2017). Over the past 50 years, credit risk measurement has seen immense developments as financial institutions have moved away from subjective systems, mostly based on the analysis of several borrower characteristics, such as reputation, leverage, capacity and collateral (i.e., the so-called 4 "Cs" of credit), towards more objective ones (Altman & Saunders, 1998).

The trend of credit risk measurement steering towards more objective methods makes sense, considering that it is in the interest of banks to have continuous improvement of their assessments, as lending and borrowing represent a key portion of their operations (Freixas & Rochet, 1997). Added to that, accurate credit risk measures are increasingly important for banks to be able to comply with regulations such as the Basel II Internal Rating Based (IRB) approach.

As mentioned above, not only banks are interested in a correct risk assessment. For firms this is also relevant, due to the importance of trade credit and the impact of credit risk on their funding costs. The assessment of firms' default risk is reflected in the default spread included in interest rates, and those default-adjusted interest rates represent the cost of debt for firms (Damodaran, 2012). Intuitively, the higher the interest rates, the costlier it becomes for a firm to service its debt, and the higher the likelihood that there might be an event of default.

2.2. Approaches to measure credit risk

The literature on credit risk measurement is extensive. New approaches are constantly developed, and it is possible to divide them into four main categories: i) traditional statistical models; ii) machine learning methods; iii) reduced form or intensity models; and iv) structural form models. Below each of them is reviewed according to the relevant literature.

The first category, the so-called traditional statistical models, refers to the first methods proposed to assess credit risk (Lando, 2004). These methods are still widely used, arguably due to their simplicity in comparison to others. Some of those methods include discriminant analysis and logistic regressions.

Among the most cited papers using statistical methods to predict default are the ones by Beaver (1966) and Altman (1968). While Beaver's model is based on a univariate discriminant analysis, Altman used a multiple discriminant analysis. In both, the main assumption is that there are two populations with normal distributions, but with different means. One of those groups survives during the relevant period of time, while the other defaults. Altman's Z-Score is still commonly used both in research and in practice, and Altman, et al. (2017) argue that this is an indication of the model's reasonability, simplicity and consistency as a measure of distress. Despite that, Balcaen & Ooghe (2006) consider that multiple discriminant analysis merely assigns firms to the groups they resemble the most, and in a strict sense, a classification statement built on that cannot be considered as a prediction.

A logistic regression, or simply a logit, is a type of conditional probability model. Underlying this approach, the probability of default is seen as dependent on a set of predictors, which typically include macroeconomic indicators and firm characteristics. This model allows to determine whether a firm will default or not based on its attributes by creating a multivariate probablity score (Balcaen & Ooghe, 2006). Ohlson (1980) argues that using a conditional logit analysis avoids a number of issues associated with multivariate discriminant analysis, such as the statistical requirements imposed on the distributional properties of the predictors. While the application of a logistic regression discards the need for assumptions regarding prior probabilities of bankruptcy or the distribution of predictors, this type of model has been found to be very sensitive to multicollinearity (Doumpos & Zopounidis, 1999), outliers and missing values (Joos, Vanhoof, Ooghe, & Sierens, 1998).

The second category of methods that can be used to assess credit risk is machine learning. These techniques vary in terms of sophistication and are designed to recognize patterns/groups with similar characteristics. The pattern recognition ability allows the use of these algorithms to classify the level of creditworthiness of counterparties (Barboza, Kimura, & Altman, 2017). Machine learning methods are usually divided between supervised and unsupervised models. Supervised learning models assume the availability of a teacher/supervisor to classify the training examples into classes, while unsupervised learning models identify the pattern class information independently (Sathya & Abraham, 2013). Logit models are many times considered a form of supervised learning models. Other examples of supervised learning models include support vector machines and random forest models. Among the unsupervised learning models it is relevant to highlight clustering models. Finally, artificial neural networks can be classified as either supervised or unsupervised learning, depending on the algorithm.

The major difference between the referred machine learning techniques and the more traditional statistical methods relies on the parameter structure imposed on the data. While traditional models tend to impose more rigid structures to the data and then choose the parameter values that fit the data best, machine learning techniques often extract knowledge from the data and build different model representations to explain the data itself (Huang, Chen, Hsu, Chen, & Wu, 2004). In result, machine learning models tend to fit the data better. The other side of the coin is that traditional statistical methods are simpler to understand and judge their reliability, while several machine learning methods are often seen as leading to black-boxes.

The third type of model that is widely studied in literature is the reduced form or intensity models. These were initially developed by Jarrow & Turnbull (1995), as an attempt to present a new approach for pricing derivative securities using credit risk and build an alternative to structural models. This model is ideal for risk management purposes (i.e., pricing and hedging), according to Jarrow & Protter (2004), as it is assumed that the modeler's information set is the one observed by the market. In their seminal paper, Jarrow & Turnbull (1995) assume that capital structure does not have a direct relation with bankruptcy and that bankruptcy can occur at any time. Reduced form models consider that the firm's default time and the recovery rate process are exogenous (Jarrow, 2009). The core idea behind this model is that there is no underlying theory explaining default, meaning that everyone is exposed to it. Default can occur at any time due to the assumption of a jump process (Cooper & Martin, 1996).

Finally, structural form models are characterized as those that stem from Black & Scholes (1973) and Merton (1974). Structural models are built under the idea that default occurs whenever the market value of the firms falls below a certain default barrier, that can be determined by looking into equity data. Arguably, structural form models develop at a slower pace due to the issues stemming from its dependence on market values (Liao & Chen, 2005).

While Merton's insight was revolutionary, later other authors introduced different considerations, resulting in ramifications from Merton's original model. Much of the subsequent research following the Merton model consisted of relaxing some of the assumptions made, as an attempt to create more realistic models (Cooper & Martin, 1996). For the discussion in this dissertation, the main distinction that will be explored is between traditional structural models, that are built based on the stock of the firm, and recent ones, that assume that the assets of the firm are directly linked to its fundamentals, such as a firm's revenue.

Traditional structural models include those developed by Merton (1974), Black & Cox (1976) and Leland (1994). According to Sundaresan (2013), those three papers are responsible for providing the basis for studying topics ranging from capital structure to credit spreads. In the specific case of Merton's model, the firm's equity is seen as a European call option on the firm's assets, with a strike equal to the face value of debt (that consists of a single zero-coupon bond). This is so due to the decision factor embedded in the process of bankruptcy: the decision to do so is very similar to the decision of exercising a call option. If at maturity the market value of the assets exceeds the nominal value of debt, shareholders will decide to pay off the debt. On the other hand, if the value of debt is higher, the firm declares bankruptcy and the assets are liquidated to pay creditors.

In Merton's model, debt can be characterized in two ways: as the difference of the firm value and the payoff of the European call option discussed above, or as the difference of the face value of the debt and the payoff of a European put option on the firm's value. From the second representation it is clear that debtholders are providing insurance on the firm's assets to shareholders, and that the interest earned on the debt should compensate for the risk undertaken (and therefore be higher than the risk-free rate) (Jarrow, 2009).

In their 1976 paper, Black & Cox took the first step into trying to create more realistic models, based on Merton's previous work. They introduce the possibility of the firm defaulting before the debt maturity date (first passage time model) by including safety covenants. Allowing the firm to default before the maturity date addresses one of the main criticisms to the Merton model, in which only time T matters. The addition of safety covenants gives debt holders the right to force the firm into bankruptcy, in case it is not meeting a determined standard. Including this new set of rights for debt holders leads to the valuation of equity as a down-and-out call option (rather than a European call option) on the firm's assets.

Leland (1994) gave continuity to Merton's and Black & Cox's work by including taxes, bankruptcy costs and protective covenants in his model. His paper is focused on corporate finance, studying corporate debt values, but also an optimal capital structure, as he argues that the capital structure cannot be optimized without knowledge of the effect of leverage on the value of debt. Leland considers two possible bankruptcy triggers: an endogenous barrier, that can be met when the firm is unable to raise enough equity capital to meet its debt obligations, and the principal value of debt.

The other type of structural model includes those that assume that the firm's assets are a non-traded security, whose value is linked to the firm's earnings generation capacity. Those models can be exemplified by the ones that were developed by Goldstein, Ju & Leland (2001) (GJL) and Eisdorfer, Goyal, & Zhdanov (2019) (EGZ). A major difference between these models and all structural models previously referred to is that, as in these models the asset value is not traded, the value of all contingent claims depends on risk pricing parameters. The assumption that the asset is not tradeable is key, as now the asset prices will be dependent on preferences.

GJL (2001) can be considered an extension of Leland's 1994 paper. GJL use Earnings Before Interest and Taxes (EBIT) as the state variable, rather than the unlevered firm value that was previously used in other models, linking the asset value to corporate earnings. They argue that EBIT is the ideal state variable because it should be insensitive to capital structure changes, while the unlevered firm ceases to exist once the capital structure is changed. By assuming that EBIT will not be affected by the capital structure, GJL argue that the EBIT-generating machine will continue to run independently of how the flow it creates is distributed among its stakeholders. They treat all of the contingent claims to EBIT (i.e., shareholders, debtholders and government) in a consistent fashion.

EGZ's 2019 model is aimed at equity valuation, as they investigate whether equity misvaluation is in part driven by investor's inability to fully recognize and price the optionality in equity. As in GJL, the model built by EGZ also connects the firm's asset value to its earnings. However, instead of using EBIT as the state variable, they opt for using the firm's gross margin (defined as the difference between sales and cost of goods sold). This model is better presented in section 3 of this dissertation.

2.3. Credit risk measurement for private firms

Having a correct risk assessment gains even more relevance for private firms, that opposed to large, quoted ones, do not have other financing alternatives easily at their disposal. Rikkers & Thibeault (2009) state that, for private firms that do not have any public market information available, the most common approach for credit scoring are the traditional accounting-based statistical models.

Generally, models such as the Altman's Z-Score (1968) are the most used. As discussed by Chen, Hu, & Pan (2006), the issue with those statistical models is that they essentially ignore the information contained in volatility, something which is at the center of structural models.

An alternative to the traditional statistical models would be structural models. Among the benefits of using structural models to assess credit risk one can list the following: i) their stochastic nature considers risk explicitly, ii) they incorporate shareholder's strategic decision with respect to making the firm default or not, and iii) they have the capacity of distinguishing temporary shocks from permanent ones. Despite those advantages, it is relevant to keep in mind that it is hard to build a model that includes all the variables that are known to influence the probabilities of default (Lando, 2004).

While a lot of research has been done on structural models overall, the literature still lacks research focused on credit risk models for private companies, as, traditionally these models are built based on the stock value of publicly traded firms. This is particularly relevant for private firms, given that for them there is no information on the market value of contingent claims, notably equity. Since most of the structural models rely on market data, they tend to be limited for use in large public firms (Rikkers & Thibeault, 2009).

A solution that can be employed to overcome the shortcoming of lack of public data is to estimate the private firm's free cash flows using an alternative valuation model. Some types of structural models can be applied as an alternative to traditional valuation models, more specifically those that link the firm's asset value to its earning generating capacity. This is especially the case of the EGZ (2019) model, a model initially built for equity valuation purposes but that can be also applied for credit risk assessment. The following section discusses the model in further detail.

3. Model

The model used in this dissertation is based on the one developed by EGZ (2019), with some modifications made in order to adjust their equity valuation model for this dissertation's credit risk purposes. The first section of this chapter describes the original model, while the second discusses the modifications made. Finally, the third section explains how an equity valuation model can be used for credit risk.

3.1. Eisdorfer, Goyal and Zhdanov's 2019 model

EGZ's (2019) equity valuation model differs from traditional discounted cash flows methods by explicitly considering the default option held by shareholders. The key variable in the model is the capacity of the firm to generate gross profits, which is denoted by:

$$x_{it} = Sales_{it} - COGS_{it}, \tag{1}$$

where $Sales_{it}$ represents the firm's annual sales and $COGS_{it}$ the cost of goods sold. The evolution of x_{it} reflects the stochastic demand for the firm's products.

EGZ assume that the state variable x_{it} follows a geometric Brownian motion under the physical measure P with a drift parameter $\mu_{i,P}$ and volatility σ_i:

$$\frac{dx_{it}}{x_{it}} = \mu_{i,P}dt + \sigma_i dW_t^P. \tag{2}$$

As long as there are no arbitrage opportunities, this process can be written under the risk-neutral measure (measure Q). This can be done subtracting $\overline{m}\sigma$ from the drift under the physical measure (measure P), where $\overline{m}$ represents the market price of risk, resulting in $\mu_{i,Q} = \mu_{i,P} - \overline{m}\sigma$.

The estimation of the free cash flow to equity (FCFE) is at the core of EGZ's model, as the equity value at time zero is given by the present value of the expected future free cash flows to equity. As shown in equation 3, the FCFE is discounted by the risk-free rate *r* under the risk-neutral measure[1].

[1] The use of the risk-free rate does not imply that the agents in the model are indifferent to risk, as risk aversion is included in the model by changing the probability measure.

$$E_{i0}(x_0) = \sup_{T_{x_d(t)}} \mathbf{E}_{x_0}^{Q} \int_0^{T_{x_d(t)}} e^{-rt}\, FCFE_{it} dt, \tag{3}$$

where $x_d(t)$ is the optimal default boundary and $T_{x_d(t)}$ is a first time passage to the boundary $x_d(t)$ of the process.

The free cash flow to equity is defined as how much cash is available to be paid out to stockholders after meeting all of the firm's financial obligations (Damodaran, 2012). The equation below is typically shown in finance textbooks to represent that definition.[2]

$$\begin{aligned} &\text{FCFE} \\ &= \text{Net income} - (\text{Capital expenditures} - \text{Depreciation}) \\ &- \text{Change in noncash working capital} \\ &+ \text{New debt issued } - \text{Debt repayments.} \end{aligned} \tag{4}$$

EGZ follow an approach that resembles the one presented above. Their FCFE calculation is based on six key elements: i) gross margin, ii) fixed operating costs, iii) interest expense, iv) capital expenditures and depreciation, v) additional distress costs, and vi) principal debt. Each of those elements can be linked back to the "traditional" equation presented above.

First, instead of basing the estimation on net income, the base of their estimations is the gross margin. Net income is the result of the subtraction of costs and expenses from sales. Since only the cost of goods sold are subtracted from sales to obtain the gross margin, expenses have to be introduced in the model in another form. This is done by including explicitly in the equation fixed operating costs (F_i) and interest expenses (I_i), that are assumed to be constants. With that, EGZ assume that the operating and financial costs involved in selling and delivering the firm's products and/or services are fixed[3].

For capital expenditures and depreciation, the authors find a way to link them to the state variable, allowing the model to have a single risk factor and simplifying calculations. In order to do that, first they compute the average $Capex/Sales$ and $Depreciation/Sales$ ratios for the two-digit SIC

[2] Equation 4 was extracted from Damodaran's Investment Valuation: Tools and Techniques for Determining the Value of Any Asset, University Edition (2012).
[3] EGZ opt to proxy for the fixed operating costs by using selling, general and administrative costs (variable XGSA in Compustat). By doing that, they are assuming that costs such as salesperson's salaries and commissions, advertising and promotion, travel and entertainment, office payroll and expenses, and executive salaries are fixed.

industry over the last three years, $\overline{CSR}$ and $\overline{DSR}$. The use of 3-year averages helps avoid the noise generated by the short-term variation in capital expenditures and depreciation. Then, they assume that capex and depreciations are a multiple of sales:

$$Capex_{it} = Sales_{it} \times \overline{CSR}, \tag{5}$$

$$Dep_{it} = Sales_{it} \times \overline{DSR}. \tag{6}$$

They also calculate a gross margin ratio, GM_{it}, and assume that this stays constant in the future, linking future sales proportionally to the state variable x_{it} (this relationship can be seen in equation 7).

$$GM_{it} = \frac{Sales_{it} - COGS_{it}}{Sales_{it}} = \frac{x_{it}}{Sales_{it}}, \tag{7}$$

One can express the assumption that GM_{it} will remain constant in the following way as well:

$$Sales_{is} = x_{is}/GM_{it}, \tag{8}$$

for $s \geq t$. Substituting $Sales_{it}$ by expression 8 allows to write capex and depreciation as a linear function of x_{it}, as shown below.

$$Capex_{is} = Sales_{is} \times \overline{CSR} = \frac{x_{is} \times \overline{CSR}}{GM_{it}}, \tag{9}$$

$$Dep_{is} = Sales_{is} \times \overline{DSR} = \frac{x_{is} \times \overline{DSR}}{GM_{it}}. \tag{10}$$

The fifth key element in EGZ's model are additional distress costs (η). These are costs that a firm would not face in a normal situation. The authors suggest that they can represent the increased cost in maintaining a healthy relationship with suppliers, retaining the customer base, addressing increased agency costs, or even raising new funds. In sum, they reflect the cost that the firm incurs in keeping its operations running when it is already in distress. Those additional distress costs could represent a change of non-cash working capital, for example, in a scenario when the firm has to negotiate the terms of its trade credit with suppliers.

The firm will incur in the additional distress costs only when its FCFE is negative. Based on that, EGZ set a distress barrier (x^*) [4] that determines whether the firm will have those additional distress costs or not. The value of equity will depend on whether x_{it} is above or below x^*. This barrier can be calculated per the equation below.

$$x^* = \frac{(I_{it} + F_i)(1-\tau)}{1 - \tau + (\tau\overline{DSR} - \overline{CSR})/GM_{it}}. \tag{11}$$

The last element of the FCFE equation is net debt issuance. Their model consists of two stages: on the first stage, the firm issues short-term (one year) and long-term (five year) interest-paying debt; on the second stage, if solvent, the firm shareholders pay all the firms' debt. The firm then issues new perpetual debt, whose proceeds are used to pay an extra dividend to the shareholders. The different assumptions for debt in each stage lead to two equations of the estimation of the free cash flow to equity, as the equation for the first stage must include a debt repayment term (D_{it}).

Based on all the assumptions presented, the resulting free cash flow to equity equation used by EGZ are shown below.

$$\begin{aligned} &FCFE_{it} \\ &= \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right] x_{it} - (1-\tau)(I_{it} + F_i) \\ &\times \left[1 + \eta \mathbf{1}_{x_{it}<x^*}\right] - D_{it}, \end{aligned} \tag{12}$$

$$\begin{aligned} &FCFE_{it} \\ &= \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right] x_{it} - (1-\tau)(I_i + F_i) \\ &\times \left[1 + \eta \mathbf{1}_{x_{it}<x^*}\right] d_t, \end{aligned} \tag{13}$$

where $\mathbf{1}_{(.)}$ is an indicator variable.

Regarding the expected growth of gross profits, EGZ assume that it is directly related to the capital expenditures made by the firm. In particular, $Capex_{it}$ invested over a time period of $[t, t+dt]$ is assumed to result in an expected (under *P*) increase in operating cash flow of $Capex_{it} R_A dt$ over

---

[4]Note that the distress barrier (x^*) is not the same as the default boundary (x_d). The first one determines if the firm will incur in additional distress costs, while the later indicates whether the shareholders will opt to exercise their option to default or not.

the next time interval dt, where R_A represents the after-tax return on the firm project. Under this hypothesis, it is possible to show that the expected growth rate in operating cash flows (under P) is given by:

$$\mu_{i,P} = \frac{\mathbb{E}_t^P(dx_{it})}{x_{it}dt} = \frac{\overline{CSR} \times R_A}{(1-\tau)GM_{it} + \tau\overline{DSR}}, \tag{14}$$

where $\mathbb{E}_t^P$ is the conditional expectation under the physical measure P at time t.

Differently from other structural models, in EGZ (2019) the default barrier is time dependent. This occurs due to the existence of the two stages. Notice that during the first stage (including the transition from the first to the second stage), it is assumed that the capital outflows are not constant because the coupon and principal payments are allowed to vary over time. On the other hand, as mentioned previously, it is assumed that on the second stage debt is perpetual, and therefore the value of the coupons paid is constant and there are no more principal repayments. The consequence of the perpetuity of debt is a constant default barrier at that stage.

Given that the default boundary is constant in the second stage, EGZ decided to solve the model following a backwards strategy. They do this in four steps. The first step is to determine the value of equity immediately after the perpetual debt is issued in year five. Using standard arguments, they show that the terminal value of the firm after refinancing is given by the system of equations below.

$$E(x_{it}) = \begin{cases} Ax_{it}^{\beta_1} + Bx_{it}^{\beta_2} + \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right] \\ \frac{x_{it}}{r-\mu} - (1-\tau)\frac{I_i + F_i}{r} & \text{if } x_{it} \geq x^* \\ Cx_{it}^{\beta_1} + Dx_{it}^{\beta_2} + (1+\eta)\left\{\left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\right. \\ \left.\frac{x_{it}}{r-\mu} - (1-\tau)\frac{I_i + F_i}{r}\right\} & \text{otherwise,} \end{cases} \tag{15}$$

where β_1 and β_2 are the roots of the quadratic function $\frac{1}{2}\sigma^2\beta(\beta-1) + \mu_Q\beta - r$. A, B, C and D are constants, which can be found by solving the below system of five equations:

$$A = 0,$$

$$Bx^{*\beta_2} + \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\frac{x^*}{r-\mu_Q} - (1-\tau)\frac{I_i + F_i}{r}$$

$$= Cx^{*\beta_1} + Dx^{*\beta_2} + (1+\eta)\left\{\left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\frac{x^*}{r-\mu_Q} - (1-\tau)\frac{I_i + F_i}{r}\right\},$$

$$\beta_2 Bx^{*\beta_2 - 1} + \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\frac{1}{r-\mu_Q}$$

$$= \beta_1 Cx^{*\beta_1 - 1} + \beta_2 Dx^{*\beta_2 - 1} + (1+\eta)\left\{(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right\}\frac{1}{r-\mu_Q},$$

$$Cx_d^{\beta_1} + Dx_d^{\beta 2} + (1+\eta)\left\{\left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\frac{x_d}{r-\mu_Q} - (1-\tau)\frac{I_i + F_i}{r}\right\} = 0,$$

$$\beta_1 Cx_d^{\beta_1 - 1} + \beta_2 Dx_d^{\beta_2 - 1} + (1+\eta)\left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}\right]\frac{1}{r-\mu_Q} = 0. \qquad (16)$$

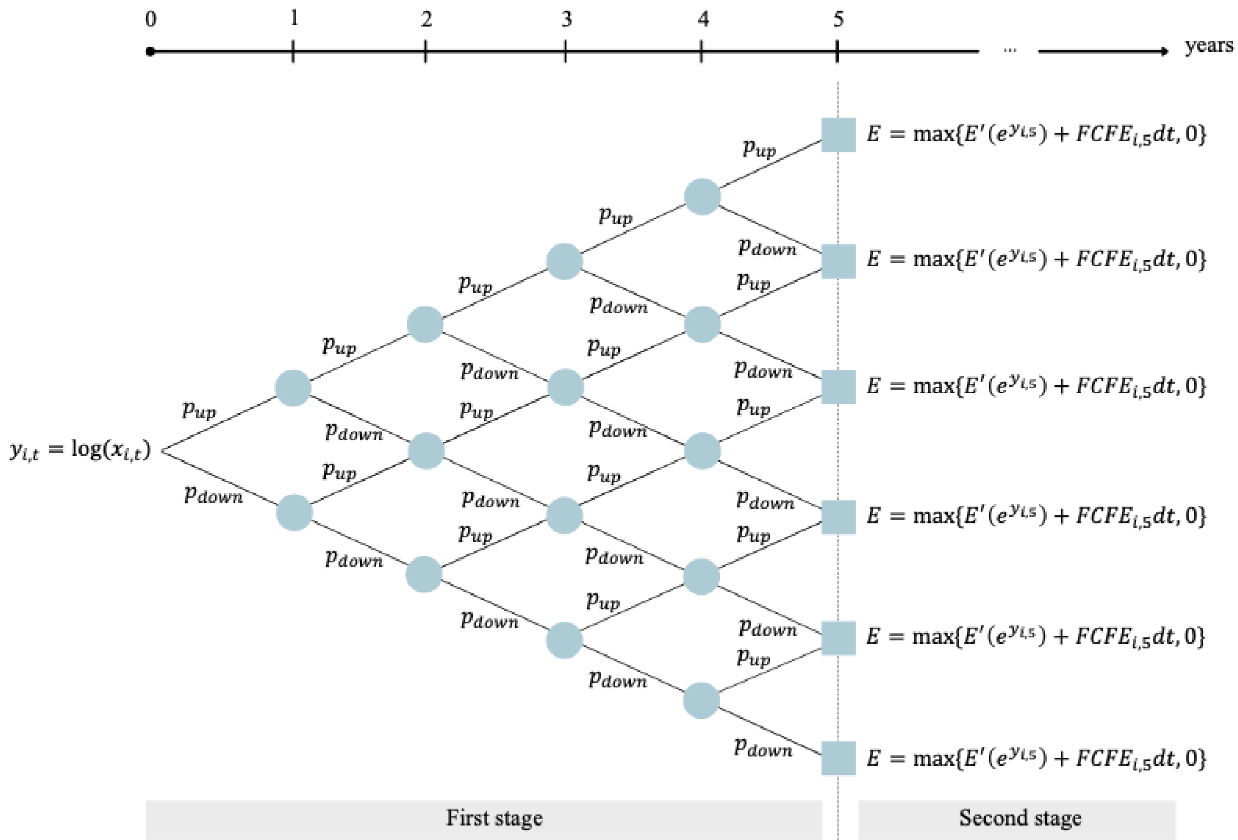
The second step is to find the value of the perpetual debt. Having the value of the perpetual debt along with the post-refinancing equity value, it is possible to move on to the third step, which consists of calculating the pre-refinancing equity value. Finally, using a binomial approximation of the Brownian motion to perform a backwards iteration of the values, the equity value in year zero is estimated.

Figure A illustrates the model, where each node of the binomial tree represents the decision shareholders have of exercising their option to default.

Figure A

Illustration of the Two-stage EGZ 2019 Model

The figure is an illustration of the setup of the EGZ 2019 model, indicating the underlying equations and processes used in the construction of each stage.



3.2. Eisdorfer, Goyal and Zhdanov's adjusted model

This dissertation studies the impact of the pandemic crisis using a modified version of the EGZ model. Three major changes are implemented: i) the model was adapted to assess the impact of the pandemic crisis, ii) it was considered that firms have a cash account, and iii) a different decomposition of operating costs was made. Each of the modifications is explained in further detail below.

First, the first stage is set to take four years instead of five as in the original model. The period of four years was chosen based on the expectation that the travel industry will return to 2019 spending levels between 2023 and 2024 (Skift Research in Partnership with McKinsey & Company, 2020).

Additionally, the report by Skift Research and McKinsey also suggests that the hotel sector demand may not reach pre-pandemic levels before 2023, while the revenue per available room might not recover until 2024. Following these insights, it is assumed that firms face a strong reduction in their annualized gross profit in March 2020. The gross profit expected to return to pre-pandemic levels by the end of 2024.

The second alteration is the introduction of cash in the first stage of the model. Cash buys firms time to react to short-term crises (Brealey, Myers, & Allen, 2014), and in practice one can see that indeed many firms hold onto it as a precautionary measure. Having a cash cushion when shocks such as the one observed in 2020 hit, can be the determinant factor of a company's survival. It is relevant to include cash into the model when analyzing the COVID-19 shock because, despite some governmental aid received, firms with a negative FCFE, hardships to borrow, and low cash reserves may not survive.

During the first stage of the model, regardless of the firm's revenue and the amount of cash it has, no dividend is paid out. In case the cash reserves fall below zero, the shareholders will have to inject capital. The equation below describes the cash dynamics assumed for any t during the four-year shock period.

$$Cash_{t+dt} = \max\{0, Cash_{t-1} + FCFE_t dt\} \tag{17}$$

At the end of the first stage, it is considered that the firm must hold a certain amount of cash. This amount of cash is interpreted as a pledge that the shareholders must keep in the firm in order for it to keep its activity. The shareholder will never receive this pledged cash back. In the transition between stages, if the firm's cash account is above that barrier, the excess cash is distributed as a dividend to shareholders. Alternatively, in case it is below the cash barrier, the account can be replenished with a capital injection from the shareholders.

The addition of cash into the model impacts the level of the default barrier x_d during the first stage of the model. The default barrier now depends on the initial values of cash at the start of the pandemic: the bigger the cash cushion the firm has initially, the higher the likelihood that there will be cash at the end of the period and thus the higher the expectation shareholders have of receiving an extraordinary dividend. When the firm's cash account starts deteriorating, the shareholder's expectation of receiving cash at the end of stage one decreases and therefore the default barrier increases. In essence, cash acts as an additional buffer. Whenever this falls to zero and the free cash flow is negative, we end up in the case of the EGZ model: shareholders have to decide between injecting capital into the firm or declaring the firm bankrupt.

The third modification in EGZ's model is that a different decomposition of costs was made. Besides a constant portion of fixed costs (F_i), there is also another portion, called other operating costs (Ooc_{it}) that is directly linked to the firm's sales. The goal behind this decomposition is to account for the dynamics that affect each operating cost differently, where for example, office payroll expenses are expected to behave differently from advertising and promotion expenses.[5] While fixed costs F_i are assumed to stay constant forever similar to EGZ, other operating costs Ooc_{it} are considered to move with sales. The equation below shows the relationship between other operating costs and sales. Note that it is very similar to equations 5 and 6, for capex and depreciation.

$$Ooc_{it} = Sales_{it} \times \overline{OSR}, \tag{18}$$

where $\overline{OSR}$ represents the ratio of other operating costs over sales over the past three years for the industry. Similarly to capex and depreciation, equation 8 leads to:

$$Ooc_{is} = Sales_{is} \times \overline{OSR} = \frac{x_{is} \times \overline{OSR}}{GM_{it}}. \tag{19}$$

The addition of other operating costs modifies the free cash flow to equity equations presented by EGZ (12 and 13), and consequently the system of equations (15) and boundary conditions (16). The term 20 shown below is substituted by term 21 in those equations mentioned. The modified equations can be found in Appendix A.

$$(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}}. \tag{20}$$

$$(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}. \tag{21}$$

[5] Under the example given, the expectations were formed based on the tendency of wages being sticky in Europe, and on the decrease seen in advertising budgets (World Federation of Advertisers, 2020).

3.3. Using Eisdorfer, Goyal and Zhdanov's 2019 model for credit risk

EGZ's model objective is to do equity valuation, however, the model can also be used for credit risk. Instead of focusing on obtaining the value of equity at time zero using the binomial tree presented in figure A, the goal is to obtain the default barriers over those four initial years analyzed in the model.

In structural models such as Merton's (1974), where only time T matters, distance to default (DD) is used as the standard credit risk measurement. The DD indicates the distance in standard deviations between the value of the project and the default barrier at a debt maturity, normalized by the project's risk. The probability of default (PD) represents the probability of the shareholders not exercising the call option they have on the firms' assets. It can be estimated merely by doing the inverse normal of the DD.

For structural models such as EZG (2019), in which default is triggered whenever the state variable reaches the default barrier, DD is an incomplete credit risk measure. The reason for that is simply that the firm's exposure to default is not limited to the moment when debt matures, and DD cannot account for the possibility of it occurring at any time.

In order to assess credit risk more accurately in this dissertation, the solution found is to estimate a so-called "first passage time adjusted" distance to default (DD'). The first step is to estimate the default barrier for each month. Then, it is necessary to use Monte Carlo simulations to model the stochastic process of the state variable. Once the simulations are done, one must count how many times the simulated values fall below the default barrier threshold for each trajectory. If for a determined trajectory the values are lower than the default boundary at a determined point in time, the firm is considered as if it had entered default from that moment on. The proportion between the values that fall below the barrier and all the simulations gives the PD. The inverse normal of the PD gives DD'.

4. Data

This chapter has four objectives: i) explain the original databases, ii) describe the data cleaning process, iii) describe the representative firms used to study the accommodation sector, and iv) show how the model was calibrated. Each of those objectives is tackled in one of the sections below.

4.1. Original databases

This dissertation uses CBHP and the COVID-IREE databases. Both classify firms according to the Portuguese Classification of Economic Activities (CAE-Rev.3).

The CBHP data contains detailed information on Portuguese non-financial firms' balance sheets and income statements, spanning over the period between 2006 and 2018. It is a collection of data obtained from the mandatory financial information firms report under the Simplified Corporate Information. It includes information for over 700,000 different companies, distributed in 87 different economic sectors.

The IREE database was built using firm responses to the COVID-19 related survey, an initiative led by Statistics Portugal and the Bank of Portugal. The objective of collecting such data was to identify the main effects of the pandemic on key aspects of enterprise activity. A sample of firms was used to generate the data, designed to represent approximately 80% of total turnover within each economic activity. The survey was launched in the week of 6-10 April 2020 and contains ten editions. The editions used were 15, 21 and 23, with the reference periods of 6-10 April 2020, 1st fortnight of June, and 1st fortnight of July, respectively.

CBHP and IREE data are qualitatively very different. While CBHP consists of hard information, IREE contains predictions made by a sample of firms' management about the impact of the pandemic in their 2020 financial results. Those predictions were used to estimate the shock of the COVID-19 crisis for the firms studied, as official data for the period will most likely be available only in the second semester of 2021.

4.2. Data cleaning

Two main steps were performed when cleaning the CBHP data: i) filtering the data according to the firms' economic sector, and ii) overcoming accounting related issues.

The initial step in cleaning the data was to drop the firms that were not classified as in the accommodation sector. Following the CAE-Rev.3 segmentation, this economic activity consists of hotels and similar establishments, holiday and other short-stay accommodations, camping sites (including caravan sites), and other accommodations. All of those four groups were used.

The second step is referent to the firms' accounting variables. The goal was to eliminate possible accounting errors that could bias the results. In order to do so, it was opted to establish a "zero floor" to COGS (D025), liabilities (B080), and account receivables (B041). Additionally, firms that had a value of zero for total assets (B001), fixed tangible assets (B012), current assets (B029) and current liabilities (B089) were dropped. Finally, the last step was to drop all companies that had turnover (D001), services (DL017), gross margin, or gross margin ratio equal or below to zero.

The cleaning process for IREE was simpler. First, it was ensured that the firms were the same in the three editions used, for consistency. Then, firms that were not in the accommodation sector were dropped.

Table I

Data Cleaning Description

The table shows the changes in the sample throughout the cleaning process for each CBHP (Panel A) and IREE (Panel B). *n* obs. indicates the number of observations in the dataframe, *n* firms the unique number of anonymous IDs, *n* groups the unique number of groups according to CAE-Rev.3, and *n* divisions the unique number of divisions according to CAE-Rev.3. For CBHP, note that despite the sample having 7,010 firms throughout the period, it does not mean that there will be 7,010 firms in every year due to a fluctuation in the number of firms every year.

	Panel A: CBHP				Panel B: IREE	
	Original	2014-2018	Division 55	Accounting	Original	Division 55
n obs.	5,028,327	2,067,139	38,587	23,445	3,264	49
n firms	727,944	536,636	11,499	7,010	3,264	49
n groups	265	263	4	4	209	3
n divisions	87	87	1	1	78	1

4.3. The representative firms: construction and description

Instead of applying the EGZ model to each firm individually, the route taken in this dissertation was to form groups of companies with similar characteristics and analyze each group through a representative firm. Using representative firms allows to have a more aggregate vision of the firms and study the impact of certain characteristics in their survival probability.

The parameters of the representative firms are simply set as the median of those observed for the firms within each group. The groups were built using cluster analysis, a technique that arranges objects (in this case, companies) into groups (i.e., clusters) according to their level of similarity under a specific set of user selected characteristics. A successful cluster analysis should form groups with high internal homogeneity and high external heterogeneity (Hair, Black, Babin, & Anderson, 2010). By creating these types of segments in the data, it is possible to study the observations more concisely while ensuring a minimal loss of information.

<u>4.3.1. Cluster Construction</u>

There are four main steps involved in performing a cluster analysis, based in the book Multivariate Data Analysis (2010). They are the following: i) variable determination, ii) eliminate possible sources of bias, iii) verify underlying assumptions, and iv) perform partitioning procedures. Each of them is described below, besides giving detail of how each step was performed to build the clusters of the firms in this dissertation.

The first step in a cluster analysis is to define which variables will be used. The variables used must be relevant characteristics of the objects being clustered, but also relate specifically to the objective of the clustering analysis itself. Including variables that are not ideal for the analysis will have a direct impact on the results. The variables chosen were the following financial ratios: Adjusted Return on Assets (AdjROA)[6], Fixed-Asset Turnover (FATO), Current Ratio (CUR) and Long-Term Debt (LTD). Each ratio was selected aiming to contemplate a different aspect regarding the firms' financial situations.

[6]Adjusted ROA is used instead of ROA because by adding back the interest expense adjusted for its tax effects, it takes into consideration the firm's financing decisions (Weil, Schipper, & Francis, 2014).

According to Weil et al., (2014), ratios summarize data in a more digestible form, and therefore aid in financial statement analysis. Financial ratios are a useful tool to explore firms' profitability and risk. On the profitability side, AdjROA and FATO were used. AdjROA measures the firm's capability of net income generation with its assets, independently of its financing source. While still measuring levels of profitability, instead of looking at net income, FATO measures the relation between the firm's investment in fixed assets and its sales. With respect to risk analysis, an important assessment is of the firm's ability to meet its obligations in a timely manner (i.e., liquidity) (Weil, Schipper, & Francis, 2014). CUR indicates the firm's ability to meet its short-term obligations and LTD measures the firm's long-term liquidity risk by measuring the portion of assets financed with long-term debt.

Table II

Clustering Ratios Details

Formulas and CBHP variables used to calculate the ratios used in the cluster analysis. All of the ratios were calculated using 2019 data (assuming that the accounting data for 2019 was the same as for 2018), to represent the firms' situation pre-pandemic. Tax (τ) was estimated by calculating the average effective tax rate paid by the sample firms in that year.

Ratio	Formula	Numerator (CBHP)	Denominator (CBHP)
AdjROA	$\frac{Net\ Income + (1-\tau) \times Int.Expense}{Total\ Assets}$	D087, D053	B001
FATO	$\frac{Turnover}{Fixed\ Tangible\ Assets}$	D001	B012
CUR	$\frac{Current\ Assets}{Current\ Liabilities}$	B029	B089
LTD	$\frac{LT\ Debt}{Total\ Assets}$	B085	B001

The second step is referent to eliminating possible sources of bias, where one must look into aspects such as analyzing outliers and standardizing the data. When performing a cluster analysis, it is essential to treat the outliers appropriately. One must try to maintain a structure where even small groups are represented, while deleting outliers that would distort the results. It was opted to follow the same approach as Campbell et al., (2008): winsorize ratio values at the 5th and 95th percentiles.

Once outliers were removed, the variables used in the cluster analysis were standardized employing the Scikit-Learn StandardScaler method, which transforms the data into Z-scores. Standardizing the data eliminates possible bias introduced by the different scales of the variables used in the analysis (Hair, Black, Babin, & Anderson, 2010).

Table III
Descriptive Statistics of the Clustering Ratios

Descriptive statistics for the non-standardized ratios after winsorizing the sample at the 5th and 95th percentiles.

	AdjROA	FATO	CUR	LTD
Mean	0.02	5.87	5.02	0.26
StDev.	0.17	11.65	8.66	0.33
Min.	-0.43	0.03	0.07	0.00
25%	-0.03	0.20	0.41	0.00
50% (Mdn)	0.02	0.79	1.31	0.08
75%	0.09	4.44	4.65	0.46
Máx.	0.33	45.64	34.28	1.01

Next, it is necessary to verify whether the assumptions of a cluster analysis hold. Contrary to other statistical methods, cluster analysis has no requirements such as normal linearity. Instead, it requires that the sample is representative of the population and that there is no multicollinearity between the variables. According to Hair et al., (2010), a cluster analysis is only as good as the level of representativeness of the sample. In this dissertation, since the sample used is practically the same as the population, there is no need for concern. Having no multicollinearity among variables is the main assumption, as it is an implicit form of adding weight to certain variables. The Variance Inflation Factor score (VIF) was used to check for multicollinearity within the ratios, as it assesses how well a variable can be explained by the others.

Table IV

Variance Inflation Factors (VIFs) for the Clustering Ratios

The table indicates the VIF scores for the ratios used as variables in the cluster analysis. Note that the scores start at 1 and have no upper limit – the higher the value, the higher the level of correlation between variables. If a determined variable has a high score, it is recommended to drop it. None of the variables used seem to have multicollinearity issues, as all of the VIF scores are extremely low.

Variable	VIF
AdjROA	1.02
FATO	1.02
CUR	1.00

The following step is to define which method will be used in building the clusters. Hair et al. (2010) describe that there are two main types of partitioning procedures that can be used in cluster analysis: hierarchical and non-hierarchical methods (also known as k-means). An alternative between either of those methods is to combine them and use a so-called hybrid approach.

Hierarchical procedures consist of combining observations according to their distance into a treelike structure, known as a dendrogram. A dendrogram can be used to determine the ideal number of clusters (as a rule of thumb, uninterrupted long lines are a good indicator of what should be the cutoff distance, since they indicate that the clusters are very different from each other). There are two basic types of hierarchical methods: agglomerative and divisive. The difference between those methods is the direction in which each of them works through the dendrogram to achieve the final number of clusters. The first starts grouping individual observations until they all form one final cluster, while the later goes in the opposite direction, breaking a big cluster into smaller ones.

Contrary to hierarchical clustering methods, k-means methods do not depend on a treelike structure. Usually, the algorithm starts by determining the cluster seeds, and based on that, they generate the clusters by grouping the observations with the seed they are the most similar to. The position of the centroids is changed until an optimal solution is found. The evaluation of the quality of the clusters is made by computing the sum of the squared error (SSE), a measure of error. The objective is to minimize the SSE when searching for the optimal solution. Despite the iteration assisting in the pursuit for the best cluster formation, the initialization of the centroids is an important step that can impact the final results.

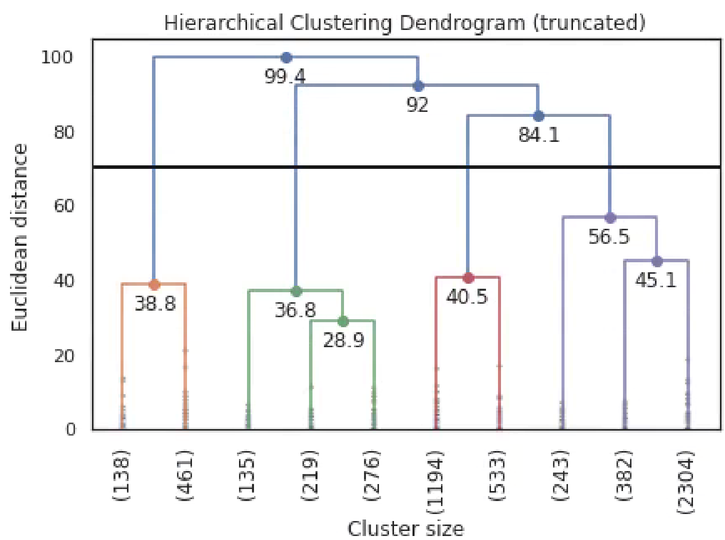
In this dissertation a hybrid approach was used, similar to Bassetto & Kalatzis (2011). First, a dendrogram was built using SciPy's Hierarchical clustering was used to determine the optimal number of clusters. Second, the k-means algorithm was used to set the clusters.

In the hierarchical clustering, Ward's method was used to measure similarity. Instead of using a single measure of similarity, this method uses the within-cluster sum of squares. As shown in Figure B, it would be ideal to have four clusters for the data. The heights of the horizontal lines show the distances between the clusters, and therefore, the bigger the distances, the more different the groups are from each other.

Figure B

Hierarchical Clustering Dendrogram (truncated)

The truncated dendrogram below shows the last ten merges performed, for visualization purposes. The horizontal axis indicates the number of observations within each cluster in parenthesis. The vertical axis shows the distances estimated using Ward's method.



With the number of clusters defined, the SciKit-Learn KMeans algorithm was used to find a final solution.[7] The centroids of the clusters obtained in the agglomerative clustering were used to initiate the algorithm. According to the SciKit-Learn User Guide, this algorithm clusters data by attempting to separate the sample into *n* groups of equal variances. It has the objective to minimize

[7] Note that the KMeans algorithm requires that the number of clusters is specified beforehand.

inertia: the measure of internal cohesiveness within clusters (using the within-cluster sum-of-squares criterion).

Once the clusters were constructed, the performance of the cluster analysis was analyzed. The hybrid method was compared with SciKit-Learn's Agglomerative Clustering algorithm. The same number of clusters were constructed in each method. The Silhouette Coefficient was used to assess which procedure would yield the best results. The score varies from -1.0 to +1.0 and is an indication of whether the clusters are dense and well separated (the closer to +1.0, the better the clustering). The Silhouette Coefficient for the hybrid method was 0.40, while for the agglomerative procedure it was 0.34.

4.3.2. Clusters Description

Each of the four groups constructed has its own set of characteristics. The representative firms are based on those characteristics and aim to give a better understanding of the bigger picture when running the modified-EGZ model.

While there are many dimensions of the clusters that can be analyzed, Table V is a compilation of the characteristics of each cluster: number of observations per cluster (Panel A), average value of the financial ratios (Panel B), and size of the companies, based on the average number of paid employees (Panel C). All of those characteristics are important, but the main focus is on the financial ratios used to build the clusters themselves. Figure C is helpful to visualize the distribution of the value of the ratios within each cluster.

Analyzing Table V, it is possible to identify two types of clusters: the ones that represent a larger portion of firms (A and B), and the ones that represent some firms that are in a more specific situation (C and D). Note in Figure C that, for the first type of cluster the observations are extremely concentrated, while for the second the observations seem to be sparser.

Just based on the number of observations, one can infer that A and B are the most representative of the sector, as they are formed by around 82% of the firms in the sample. Cluster A is by far the one with the most observations, and the average of the financial ratios for the firms within it are worse than the average of the full sample. Firms in cluster B are the ones with the most precarious ratios, which could be an indication of their worsened financial health.

Clusters C and D are attention grabbers due to the high values they have in certain ratios. It is relevant to highlight that those abnormal values can be partially explained by the high standard deviation that accompanies them. Cluster C appears to be using its fixed assets much more efficiently than the other firms in the sector to generate sales. It is also the cluster with the highest number of employees on average (nearly 15). Finally, cluster D stands out due to its CUR values. Additionally, it is also interesting to observe that it is the cluster with the lowest number of paid employees on average.

Factors such as the geographical location of the firm or the CAE-Rev.3 group classification of the firm's activity don't seem to be relevant, as a similar pattern was observed in the clusters – in all of them, Lisbon is the most frequent district and the split between the four groups is proportionate to the size of the cluster.[8]

Table V

Descriptive Statistics of the Clusters

Panel A indicates the number of observations in each of the clusters, A, B, C and D. Panel B has the mean and median for the ratios in each cluster. Panel C has the average number of paid employees per cluster.

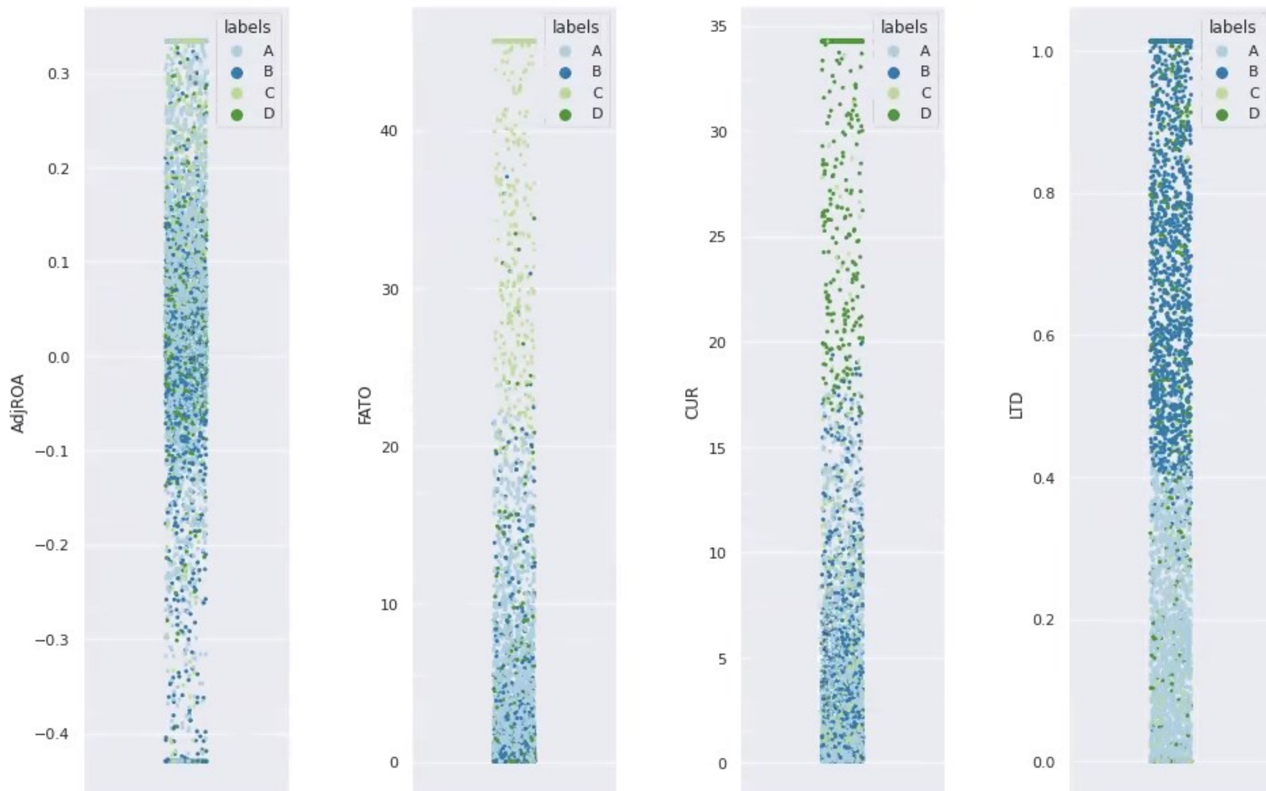
Panel A: Number of observations										
	A		B		C		D		All sample	
n obs.	3,430		1,429		537		489		5,885	
Panel B: Mean and Median for the non-standardized clustering ratios										
	A		B		C		D		All sample	
	Mean	Mdn	Mean	Mdn	Mean	Mdn	Mean	Mdn	Mean	Mdn
AdjROA	0.03	0.02	-0.03	0.00	0.08	0.11	0.01	0.01	0.02	0.02
FATO	2.84	0.82	1.66	0.35	39.18	45.64	2.87	0.45	5.87	0.79
CUR	2.21	0.87	3.24	1.60	4.70	2.16	30.25	34.28	5.02	1.31
LTD	0.08	0.00	0.72	0.70	0.10	0.00	0.34	0.17	0.26	0.08
Panel C: Average firm characteristics										
	A		B		C		D		All sample	
n employ.	12.58		11.55		14.37		1.99		11.62	

[8] For more information on the frequencies of districts and group divisions per clusters, refer to Appendix B.

Figure C

Clustering Ratios Strip Plots

The strip plots constructed for the financial ratios AdjROA, FATO, CUR and LTD indicate the distribution of the observations of each cluster.



4.4. Model calibration

Each of the sub-sections below aims to explain how the inputs for the model were obtained.

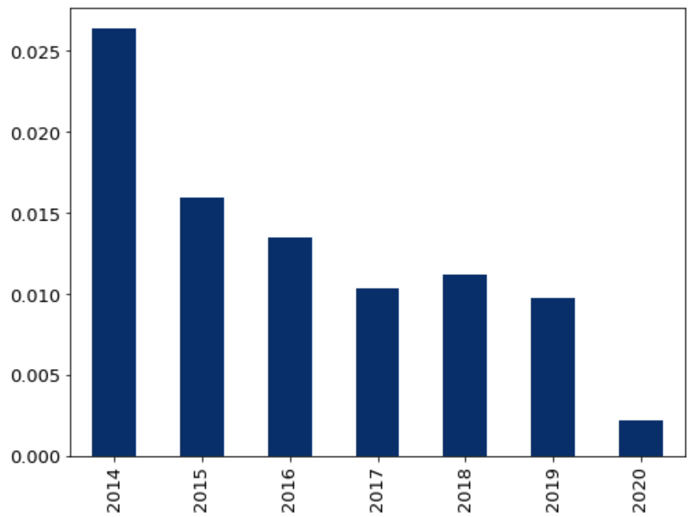
<u>4.3.1. Risk-free rate</u>

As EGZ (2019) focuses on US companies, they opt to use a rate that reflects the risk referent to that market (they use the average between the 3-month T-Bill and the 10-year Treasury Note issued by the US Government). As the companies studied in this dissertation are based in Europe, the risk-free rate used is the 30y German Bond (variable code *TRBD30T*, on DataStream). The average rate in the period between 2014 and 2020 was of 0.0128, with a standard deviation of 0.0074 and median of 0.0115. Due to the high sensitivity of the model to low interest rates, it was opted to use the average between 2014-2020.

Figure D

30y German Bond Rate

This figure shows the evolution of the 30y German Bond rate in the period between 2014-2020.



4.3.2. Corporate tax rate

A corporate tax rate of 35% is used by EGZ (2019). In this dissertation, it was opted to use the 21% statutory tax rate established by the IRC for Portuguese enterprises.

4.3.3. Additional distress costs

EGZ (2019) assume the additional distress costs (η) to be a fixed value of 15%. The same assumption is held for both stages of the model in this dissertation.

4.3.4. Capex-to-Sales and Depreciation-to-Sales ratios

The ratios Capex-to-Sales ($\overline{CSR}$) and Depreciation-to-Sales ($\overline{DSR}$) are estimated similarly as in EGZ (2019) by estimating three-year industry ratios. Due to the unavailability of 2019 data when this dissertation started being written, the three years considered when doing the industry averages were 2016, 2017 and 2018. Recall that it was assumed that the financial statements for 2019 would be the same as 2018.

The value used for $\overline{CSR}$ and $\overline{DSR}$ is 0.0462. As the sector in Portugal saw intensive levels of growth in the period in which the ratios were calculated, using the value of $\overline{CSR}$ originally estimated would lead to unsustainable assumptions in the long run.

The numerator used in $\overline{DSR}$ is D041 (*Expenses/reversals of depreciations and amortizations*). It is winsorized at the 10^{th} and 90^{th} percentiles to avoid the effects of outliers. The denominator used is D001 (*Turnover*). By using Turnover, it was possible to consider the revenue generated by both, Sales and Services. As for the accommodation sector most of the revenue is originated by Services, it made sense to opt for this variable.

4.3.5. Cash and Cash Pledge

The value of Cash for the start of the first stage of the model used is the 2019 value of the variable B049 (*Cash and bank deposits*) from CBHP observed for each representative firm, normalized by the gross margin. Table VI below indicates the values used those inputs in the model. Note that D's high values, in comparison to its peers, are in line with its high CUR ratio. The pledged Cash was set 0.071, the median cash/gross margin value for companies that enter into default one year before they do so.

Table VI
Cash Account

This table has the values used for the starting point of the Cash account for each representative firm on the first stage of the model, normalized by the gross margin.

	A	B	C	D
Cash	0.2533	0.2001	0.1554	0.9602

4.3.6. Fixed operating costs

The estimation of fixed operating costs (F_i) is based on Félix, Moreira, & Silva (2021), that studied the flexibility of Portuguese firms by estimating their level of fixed operational costs to revenue. By defining fixed operational costs as those that the firm would incur even if its revenue were zero, the authors follow a different direction from the traditional definition of fixed costs.

Félix, Moreira, & Silva (2021) follow a similar econometric approach as Gu, Hackbarth, & Johnson (2017) by using a regression to estimate the fixed operational costs. They separate the impact of lagged sales from the effect of current sales on operational costs. Using the equation below, it is possible to obtain an estimate of what the fixed operating costs would be in case the current year's sales were zero.

$$QFC_{i,t} = a_i + b_j OpCosts_{i,t-1} + d_j Sales_{i,t-1}, \tag{22}$$

where $QFC_{i,t}$ are the firm's fixed operational costs in year *t*, $OpCosts_{i,t-1}$ the firm's operational costs on the previous term, and $Sales_{i,t-1}$ the firm's sales on the previous period.

They find that firms in the services sector have a less flexible cost structure, meaning that they face higher levels of fixed operational costs to revenue. While fixed operational costs to revenue for Portuguese firms is on average 0.15, for the accommodation and catering sectors this ratio amounts to 0.31, the highest amongst all sectors.

In this dissertation, F_i is assumed to be a fixed value throughout the model and it is calculated by applying the ratio of 0.31 to the variable D001 (*Turnover*) of 2019 for each firm.

4.3.7. Other Operating Costs to Sales ratio

The Other Operating Costs to Sales ratio ($\overline{OSR}$) is calculated similarly to $\overline{DSR}$. The ratio is estimated using industry data from 2016, 2017 and 2018. First, the variables D026 (*Supplies and external services*) and D029 (*Employee expenses*) were summed. Next, for each firm, fixed operating costs were subtracted from that sum to obtain other operating costs.

The numerator used in $\overline{OSR}$ is other operating costs, and it is winsorized at the 10th and 90th percentiles to avoid the effects of outliers. The denominator used is D001 (*Turnover*). The value used is of 0.3773.

4.3.8. Gross margin ratio

The gross margin ratio was calculated simply by dividing the gross margin by the variable D001 (*Turnover*). Per Table VII below, the representative firms have a high gross margin ratio for the period. This is a direct consequence of the specific characteristics of the accommodation sector: since it is in the services industry the COGS tends to be not as high as for other sectors.

Table VII
Gross Margin Ratio Descriptive Statistics

This table consists of the mean, standard deviation, and median of the gross margin ratio for the representative firms for the period of 2014-2019.

	A	B	C	D
Mean	0.8815	0.8819	0.8835	0.9028
StDev.	0.0097	0.0077	0.0063	0.0263
Mdn.	0.8816	0.8851	0.8833	0.8987

4.3.9. Volatility

In EGZ, the proxy used for σ is the annualized quarterly volatility over the last eight quarters. An issue faced when estimating volatility for the representative firms in this dissertation was the limited number of observations available to do a proper estimation. The solution found was to do the estimation considering the firms from the sector as a whole and use a single measure of volatility for the four representative firms.

Using annual data from 2008 to 2018, the approach used was the following: i) calculate the log change for the gross margin in every year for every firm in the industry, ii) demean the annual log change for each company; iii) aggregate all demeaned log-changes and compute their standard deviation. This process yielded a proxy of volatility equal to 0.3354.

4.3.10. Return on Assets

The Return on Assets (R_A) was calculated using the Weighted Average Cost of Capital (WACC) formula shown in equation 23 below. Since the cost of capital is being estimated for private firms, it was necessary to make some assumptions for some inputs (such as for $\frac{E}{V}$), due to the lack of market information available.

$$WACC = k_e \times \frac{E}{V} + k_d \times \left(1 - \frac{E}{V}\right) \times (1 - \tau)\,, \tag{23}$$

where k_e is the cost of equity, $\frac{E}{V}$ the proportion of equity to value, k_d the cost of debt and $\frac{D}{V}$ the proportion of debt to value.

The cost of capital for the representative firms was estimated at each stage of the model. For the first stage, it was of 0.1694 for all representative firms. On the second stage, the cost of capital is of 0.1684, 0.1775, 0.1688 and 0.1695 for firms A, B, C and D, respectively. Table VIII summarizes inputs used in the calculations for each.

4.3.11. Interest Expenses

The interest expenses (I_i) are kept constant for each firm throughout the model. They are calculated by multiplying the cost of debt (k_d) of the first stage of the model by the CBHP variable B089 (*Obtained Funding*) in 2019. The values used for A, B, C and D, normalized by their 2019 gross margin, are 0.0115, 0.0900, 0.0007 and 0.0806, respectively.

Table VIII

Return on Assets Input Description

This table summarizes the estimation methods and value of the inputs used in the calculation of the Return on Assets (R_A) for each representative firm. Panel A refers to the calculation of the Cost of Equity (k_e) and Panel B to the Costs of Debt (k_d) for each stage of the model. Damodaran's data was retrieved from his website (http://pages.stern.nyu.edu/~adamodar/New_Home_Page/home.htm, accessed in March 2021).

Panel A: Cost of Equity (k_e)		
	Estimation method	Value
Risk-free	2014-2020 30y German Bond rate average.	0.0127
Equity Risk Premium (ERP)	Damodaran's Equity Risk Premium for Portugal.	0.0685
Industry Average D/E	(1-Industry Average E/V)/Industry Average E/V, using Damodaran's Industry Average E/V for the Hotel/Gaming Industry.	0.4731
$\beta_{unlevered}$	Damodaran's Total Unlevered Beta for the Hotel/Gaming Industry.	2.4251
$\beta_{private}$	$\beta_{unlevered}[1 + (1 - \text{Tax rate}) \times$ Industry Average D/E.	3.3314
k_e	Risk-free + ERP × $\beta_{private}$.	0.2410

Panel B: Cost of Debt (k_d)		
	Estimation method	Value
	Stage 1	
Risk-free	2014-2020 30y German Bond rate average.	0.0127
Spread	Spread for 1 year credit operations[9]	0.0100
k_d	Risk-free + Spread.	0.0227
	Stage 2	
Risk-free	2014-2020 30y German Bond rate average.	0.0127
Spreads	Credit Rating based on the interest coverage ratio for all emerging market firms and developed market firms with market cap < $5 billion. (A: AAA; B: B; C: AA; D: A)	A: 0.0063 B: 0.0421 C: 0.0078 D: 0.0108
k_d	Risk-free + Spread.	A: 0.0190 B: 0.0548 C: 0.0205 D: 0.0235

[9] Covid-19 Webinar: "Apoio às Empresas na Área do Financiamento" (PwC Portugal, 2020).

4.3.12. Physical and risk-neutral measures

The value of growth under the physical measure (μ_P) used in the first stage is of 0.1875. It was assumed to be the estimated shock of 75% decrease in turnover divided by the number of years that it is expected for the sector to recover.[10] For the second stage, the value of μ_P used is of 0.0112. It is the average of the physical measures obtained for each representative firm using equation 14.

The risk-neutral measure (μ_Q) was estimated using the equation below.

$$\mu_{i,Q} = \mu_{i,P} - R_A + r. \tag{24}$$

For the first stage, the risk-neutral drift is 0.0309, and for the second -0.1473. It is relevant to note that since the two stages represent such different periods of the studied firms, the expected growth rates should reflect that.

[10] The 75% estimated decrease is based on the most frequent answer given to the IREE question V2110 (*"Indicate the best estimate for the reduction in the enterprise's turnover in the reference period."*) in the editions 15, 21 and 23. In all three, the most common answer was *"5 More than 75%"*.

5. Results

This chapter discusses the results obtained by implementing the EGZ' model described in Chapter 3 for the representative firms A, B, C and D constructed in Chapter 4. The first section discusses the effect of the COVID-19 pandemic crisis on the default barriers of the sector, while the second section looks into the impact on the adjusted distances to default on the first year after the shock. The Python code used to obtain the results is in the Appendix D.

As the accommodation sector was already expected to be hit by the COVID-19 crisis, the results obtained in this dissertation are not surprising. The model was applied to a base case (referred to as "Normal with Cash"), and to a "Pandemic with Cash" case. In the first one, it is assumed that the trend of stable growth seen in the sector would continue to be observed, while in the second the COVID-19 estimated shock was applied in time 0, and for the remaining four years a strong recovery was applied with a high growth rate. The comparison of those two cases allows to isolate the deterioration of credit risk generated by the crisis studied. An additional case is studied, the "Pandemic without Cash and Cash pledge", which is used to gain a better understanding of the role of the cash cushion for firms in stressful situations.

As mentioned previously, when the representative firms were constructed, the clusters that originated firms A and B represented about 82% of the sample, with A being composed with the most companies (3,430 out of 5,885). It is relevant to keep this in mind when analyzing the results obtained from the EGZ' model because one could easily lose hindsight of the proportion underlying the construction of the representative firms studied. The results obtained for A and B can be generalized to a certain extent to most of the sector, and even though firms C and D represent a smaller portion of the sample, they continue being interesting cases to be studied.

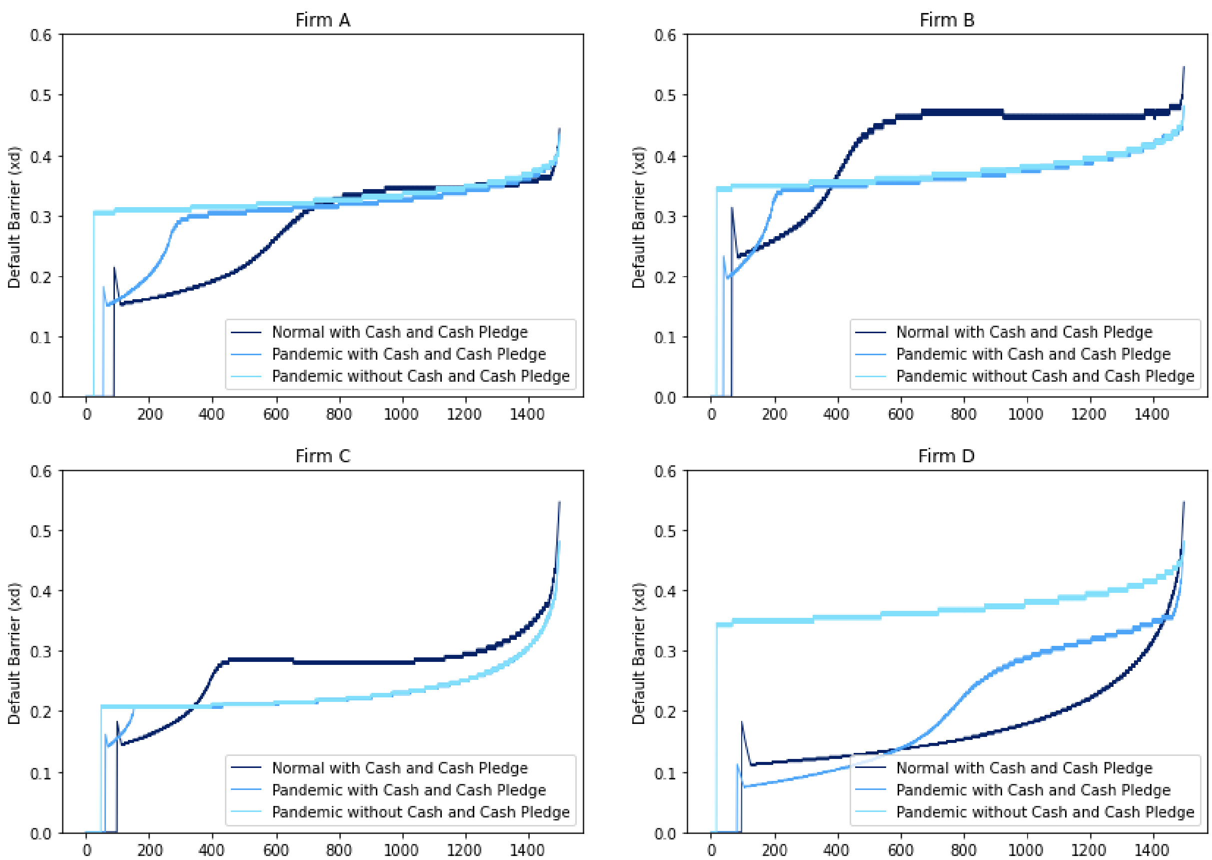
5.1. Default Barriers

The default barriers (x_d) were the first credit risk output obtained from the model. As explained previously, they determine whether the shareholder will exercise its option to default or not, and during the first stage of the model they vary over time. More specifically, they represent the minimum level of gross margin that they would be willing to accept at each point in time in order to keep the company alive. The higher the default barrier, the more shareholders are demanding not to exercise their option. As a consequence, the probability of default will increase and the distance to default decrease. In this dissertation, all of the inputs were normalized over the 2019 gross margins. For that reason, the barriers for different firms are comparable in terms of scale.

Figure E

Default Barrier Evolution

The figure shows the default barriers (x_d) at each point in time at the first stage of the model for the representative firms, for each of the cases studied ("Normal with Cash", "Pandemic with Cash", and "Pandemic without Cash and without Cash Pledge").



Analyzing the default barriers above, it is possible to identify a clear pattern observed in all firms: even though in the three analyzed cases the barriers start at different levels, the tendency is that over the four-year shock studied, they converge towards a similar value for the fixed default barrier for the firm in the second stage of the model. Despite this general pattern observed, there are still two puzzling aspect observed in the graph. Namely, an explanation for the initial jumps observed in the “Normal with Cash” and “Pandemic with Cash” cases, and for the “Normal with Cash” curve going above the pandemic ones for firms B and C couldn’t be found. Below, the default boundaries for every company are analyzed.

The initial default barriers for firm A’s cases that include cash are low (0.2140 for “Normal with Cash”, and 0.1822 for “Pandemic with Cash”). Despite having similar starting points, the rate at which they converge to A’s x_d for the second stage is different. In the “Pandemic with Cash” scenario the boundary increases at a higher slope and earlier than when no shock is applied. Note that once Cash and Cash Pledge are removed from the model dynamics, the default barrier starts at a higher point and the conversion slope to the perpetual x_d is lower. It is interesting to highlight that around t = 290 the barriers for the pandemic cases become alike. This indicates that there was an erosion in the cash account that eliminated the expectations the shareholders had of receiving the additional dividend at the end of the first stage of the model, in case the firm survived.

In firm B, the default barriers have some similarities to the ones in firm A. First, the starting point for the “Normal with Cash” and “Pandemic with Cash” scenarios are similar (especially when ignoring the initial jump) and they converge to their second stage x_d at different rates. Second, the barrier for the “Pandemic with Cash” case follows the same pattern as it did for firm A: it starts at a higher level than before and has a gentle slope. Note that firm B has a slightly lower level of cash at the end of 2019 in comparison to A, and therefore its “Pandemic with Cash” barrier moves towards the “Pandemic without Cash and Cash Pledge” curve faster.

The curves for the default barriers for firm C are comparable to those of firm B – the main difference is in the scale, as for this representative firm all of the initial points are lower (per Table IX). Additionally, the deterioration of the cash account in the “Pandemic with Cash” scenario is faster than in firms A and B. The reason behind this is solely because the starting cash cushion for this company was the lowest among all (0.1554, normalized by 2019’s gross margin).

Firm D, the one with the highest cash position at the end of 2019 (0.9602, normalized by the gross margin), seems to be the most impacted one by the removal of Cash and Cash Pledge from the model. Per Figure E, it is the firm to see the most dramatic change from its default boundaries obtained in cases that include cash, compared to “Pandemic without Cash and Cash Pledge”. The high levels of cash also generate an effect in the “Pandemic with Cash” default barrier curve. Since the cushion is so much bigger than the pledge, the conversion of the pandemic boundaries is seen only at the end of the studied period, contrary to what was seen on the previous firms.

Table IX

Initial First Stage Default Barriers

The table has the values of the first default barriers for the representative firms in the first stage of the model for the three studied cases.

	A	B	C	D
Normal with Cash	0.2140	0.3133	0.1831	0.1831
Pandemic with Cash	0.1822	0.2322	0.1614	0.1122
Pandemic without Cash and Cash Pledge	0.3063	0.3458	0.2093	0.3458

5.2. Adjusted Distances to Default

The other credit risk output from the model, which is obtained from the default barriers (as explained in Chapter 3), is the adjusted distance to default DD'. This is a distance measure that indicates the number of asset value standard deviations that the firm is away from default. Contrary to the default barrier, a low DD indicates that the company is close to default.

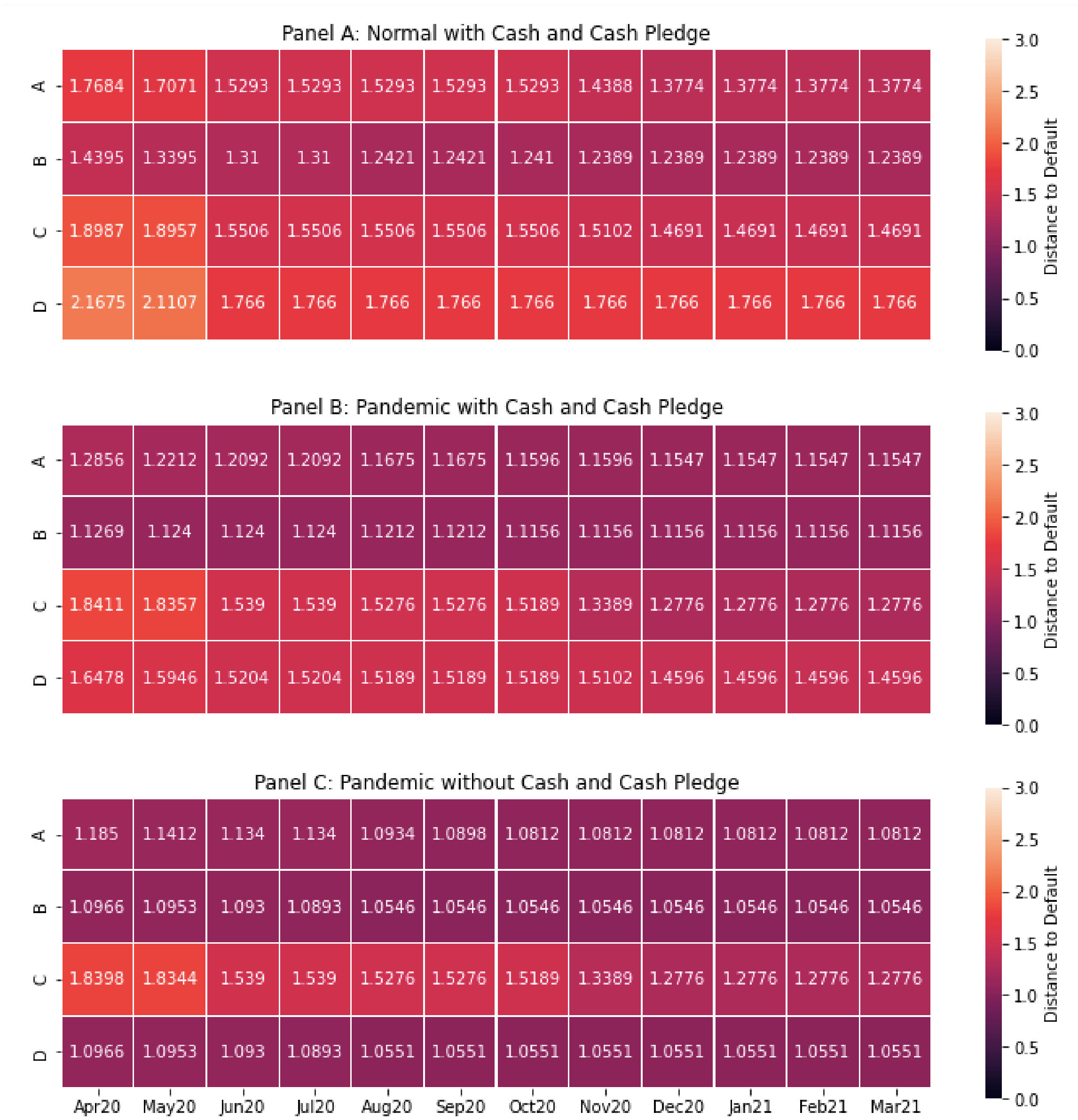
All of the firms saw a decrease in their DD' once the shock was applied to the model. Nevertheless, while the result sector-wise was expected, it is interesting to observe how each representative firm was affected differently. Figure F summarizes the impact of the crisis over the four representative firms. Comparing April of 2020 from the "Normal with Cash" and the "Pandemic with Cash" cases, the firms A, B, C and D saw an estimated decrease of 0.4828, 0.3126, 0.0575, and 0.5196 in their distances to default, respectively. These figures show that the effect of the COVID-19 shock was very heterogeneous across representative firms.

While it is important to look at the first impact of the crisis on the credit risk of the firms, it is also relevant to study the impact over a longer time horizon. In this dissertation it was opted to look into the effects over a one-year time frame, and the estimated evolution of the DD'. It is relevant to note that, per Figure F, all of the distances to default for the representative firms decrease over the 1-year period. The reason behind this is that all of the firms have an increasing default barrier, and this effect seems to dominate the increase seen in the Monte Carlo simulations (i.e., as time passes, it becomes harder for the simulated values to cross the estimated boundaries).

Figure F

Adjusted Distances to Default Heatmap

The figure shows the values of the estimated monthly 1-year Adjusted Distances to Default (DD') for the representative firms for the period between April 2020 and March 2021. Panel A indicates the estimated monthly DDs' for the base case. Panel B includes the monthly DDs' for the "Pandemic with Cash" scenario. Finally, Panel C has the DDs' for "Pandemic without Cash and Cash Pledge".



It should be noted that, since the DD' is directly connected to the default barrier, the rate of the decrease observed in the first depends on the rate at which the slope increases in the later. In Figure F above, Panel C indicates that all the representative firms with exception of C did not see many changes in their DD' throughout that first post-pandemic year. When referring back to Figure E, one can observe that the default barriers for firms A, B and D had a very gentle slope initially in the ”Pandemic without Cash and Cash Pledge” case.

It also seems that the low levels of interest expenses paid by firm C played a role in the firm's survival and keeping its DDs' barely unaffected despite the shock and the small cash cushion at the start of stage one.

6. Conclusions and Further Research

The main goal of this dissertation was to study the effects of the COVID-19 pandemic on the Portuguese accommodation sector using a structural model. By creating representative firms, based on clusters generated by a hybrid machine learning algorithm, it was possible to look at the sector while analyzing four representative firms.

Even though the COVID-19 pandemic crisis affected the accommodation sector in Portugal as a whole, the dimension of the impact for the firms was heterogeneous. This dissertation makes that clear by showing to which extent each of the representative firms A, B, C and D was affected in its credit risk measures by studying their default barriers and adjusted distance to default.

The level of cash the firm has before a negative shock directly impacts its chances of survival. If the firm has a cash balance higher than the minimum cash required for it to function, it will help delay for a determined amount of time the default barrier trajectory the firm were ought to have without considering cash nor a cash pledge. Other factors such as the level of fixed obligations a firm has (e.g., interest expenses) also influence a firm's chances of survival.

More specifically in line with the discussion of this dissertation, further research could include investigating the causes for the puzzling effects obtained for some of the default boundary curves. Other general topics could include: i) verifying the results yielded by the model using the actual financial statements for 2019 and 2020, ii) testing different clustering techniques (e.g., using different clustering variables, changing the algorithm implemented, or using another number of clusters), and iii) applying the model for other geographies and verifying the impact of COVID-19 and public policies on the accommodation sector around the world.

References

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance*, 589-609.

Altman, E. I., & Saunders, A. (1998). Credit risk measurement: Developments over the last 20 years. *Journal of Banking & Finance*, 1721±1742.

Altman, E. I., Iwanicz-Drozdowska, M., Laitinen, E. K., & Suvas, A. (2017). Financial Distress Prediction in an International Context: A Review and Empirical Analysis of Altman's Z-Score Model. *Journal of International Financial Management & Accounting*, 131-171.

Altman, E., & Sabato, G. (2007). Modelling Credit Risk for SMEs: Evidence from the U.S. Market. *ABACUS*, 332-357.

Balcaen, S., & Ooghe, H. (2006). 35 years of studies on business failure: an overview of the classic statistical methodologies and their related problems. *The British Accounting Review*, 63-93.

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems With Applications*, 405-417.

Beaver, W. H. (1966). Financial Ratios As Predictors of Failure. *Journal of Accounting Research*, 71-111.

Black, F., & Cox, J. C. (1976). Valuing Corporate Securities: Some Effects of Bond Indenture Provisions. *Journal of Finance*, 351-367.

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 637-654.

Brealey, R. A., Myers, S. C., & Allen, F. (2014). *Principles of Corporate Finance.* New York: McGraw-Hill Irwin.

Chen, R.-R., Hu, S.-y., & Pan, G.-G. (2006). Default prediction of various structural models. *Rutgers University, National Taiwan University, and National Ping-Tung University of Sciences and Technologies*.

Cooper, I., & Martin, M. (1996). Default risk and derivative products. *Applied Mathematical Finance*, 53-74.

Damodaran, A. (2012). *Investment Valuation: Tools and Techniques for Determining the Value of Any Asset, University Edition.* John Wiley & Sons.

Degryse, H., & Van Cayseele, P. (2000). Relationship Lending within a Bank-Based System: Evidence from European Small Business Data. *Journal of Financial Intermediation*, 90-109.

Doumpos, M., & Zopounidis, C. (1999). A Multicriteria Discrimination Method for the Prediction of Financial Distress: The Case of Greece. *Multinational Finance Journal*, 71-101.

Eisdorfer, A., Goyal, A., & Zhdanov, A. (2019). Equity Misvaluation and Default Options. *Journal of Finance*, 845-898.

Freixas, X., & Rochet, J.-C. (1997). *Microeconomics of Banking.* MIT Press Books.

Goldstein, R. S., Ju, N., & Leland, H. E. (2001). An Ebit-Based Model of Dynamic Capital Structure . *Journal of Business*.

Gu, L., Hackbarth, D., & Johnson, T. (2017). Inflexibility and Stock Returns. *Review of Financial Studies*, 278-321.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neuralnetworks: a market comparative study. *Decision Support Systems*, 543-558.

Hull, J. C. (2015). *Risk Management and Financial Institutions.* New Jersey: John Wiley & Sons, Inc.

Jarrow, R. A. (2009). Credit Risk Models. *Annual Review of Financial Economics*, 37-68.

Jarrow, R. A., & Protter, P. (2004). Structural Versus Reduced Form Models: A New Information Based Perspective. *Journal of Investment Management*, 1-10.

Jarrow, R. A., & Turnbull, S. M. (1995). Pricing Derivatives on Financial Securities Subject to Credit Risk. *Journal of Finance*, 53-85.

Joos, P., Vanhoof, K., Ooghe, H., & Sierens, N. (1998). *Credit classification: A comparison of logit models and decision trees.* RUG.

Lando, D. (2004). *Credit Risk Modeling: Theory and Applications.* New Jersey: Princeton University Press.

Leland, H. E. (1994). Corporate debt value, bond covenants, and optimal capital structure. *Journal of Finance*, 1213–1252.

Liao, H.-h., & Chen, T.-k. (2005). A Multi-period Corporate Credit Model---An Intrinsic Valuation Approach.

Merton, R. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *Journal of Finance*, 449-470.

National Statistics Institute (INE). (2020). *Fast and Exceptional Enterprise Survey – COVID-19 – 1st fortnight of July 2020.*

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, 109-131.

Rikkers, F., & Thibeault, A. E. (2009). A Structural form Default Prediction Model for SMEs, Evidence from the Dutch Market . *Multinational Finance Journal*, 229-264.

Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 34-38.

Skift Research in Partnership With McKinsey & Company. (2020, September). The Travel Industry Turned Upside Down: Insights, Analysis and Actions for Travel Executives.

Sundaresan, S. (2013). A Review of Merton's Model of the Firm's Capital Structure with its Wide Applications. *Annual Review of Financial Economics* , 21-41.

World Federation of Advertisers. (2020). *Covid-19 Response: Wave II Survey Results.*

Appendix A: Modified Formulas

Appendix A.1: Modified Free Cash Flow to Equity Equations.

$$FCFE_{it} = \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}\right]x_{it} - (1-\tau)(I_{it} + F_i) \times \left[1 + \eta \mathbf{1}_{x_{it}<x^*}\right] - D_{it} \quad 25$$

$$FCFE_{it} = \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}\right]x_{it} - (1-\tau)(I_i + F_i) \times \left[1 + \eta \mathbf{1}_{x_{it}<x^*}\right]d_t \quad 26$$

Appendix A.2.: Modified Additional Distress Costs Barrier.

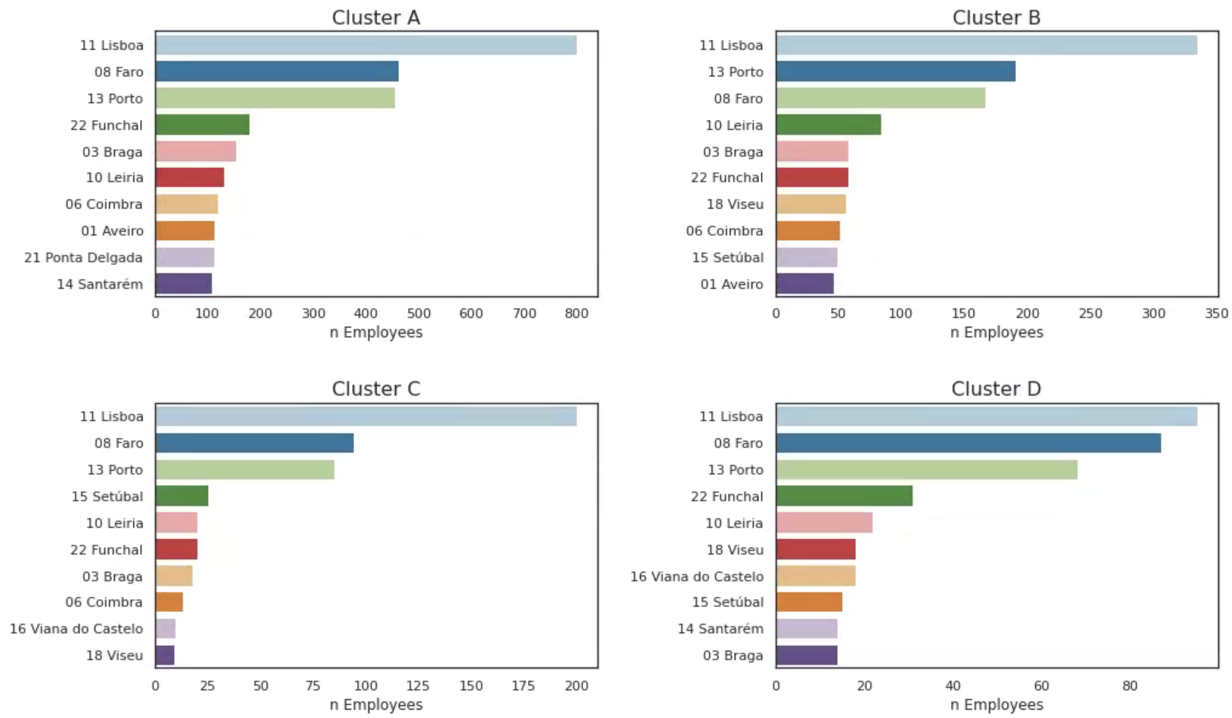
$$x^* = \frac{(I_{it} + F_i)(1-\tau)}{(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}} \quad 27$$

Appendix A.3.: Modified Equity Value.

$$E(x_{it}) = \begin{cases} Ax_{it}^{\beta_1} + Bx_{it}^{\beta_2} + \left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}\right] \frac{x_{it}}{r-\mu} - (1-\tau)\frac{I_i + F_i}{r} & \text{if } x_{it} \geq x^* \\ Cx_{it}^{\beta_1} + Dx_{it}^{\beta_2} + (1+\eta)\left\{\left[(1-\tau) + \frac{\tau\overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau)\frac{\overline{OSR}}{GM_{it}}\right] \frac{x_{it}}{r-\mu} - (1-\tau)\frac{I_i + F_i}{r}\right\} & \text{otherwise.} \end{cases} \quad 28$$

Appendix B: Cluster-specific information

Appendix B.1: Distribution of top 10 districts per cluster



Appendix B.2: Proportion of CAE-Rev.3 group classification per cluster

	A	B	C	D	All Sample
Hotels and similar	1,949	847	301	347	3,444
Short stay accomm.	1,374	547	133	133	2,279
Camping sites	65	19	6	6	92
Other	42	16	3	3	70
Total	3,430	1,429	537	489	5,885

Appendix C: Clustering Code

```
#0.Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
from sklearn.neighbors import NearestCentroid
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, adjusted_rand_score

#1.Build variables that will be used to cluster the data (ratios)
ratios_cols = ['AdjROA', 'FATO', 'CUR', 'LTD']
cluster_ratios = pd.DataFrame(columns = ratios_cols)

##RETURN ON ASSETS (Adjusted for financing)
###(Net Income + Interest Expense net of tax)/Total Assets
cluster_ratios['AdjROA'] = (CBHP_2019['D087'] + (1 - EffTR)*CBHP_2019['D053'])/CBHP_2019['B001']

##FIXED-ASSET TURNOVER RATIO
###Sales/Fixed Assets #obs: I used Turnover and fixed tangible assets
cluster_ratios['FATO'] = CBHP_2019['D001']/CBHP_2019['B012']

##CURRENT RATIO
###Current Assets/Current Liabilities
cluster_ratios['CUR'] = CBHP_2019['B029']/CBHP_2019['B089']

##LONG-TERM DEBT RATIO
###Long-term Debt/Assets
cluster_ratios['LTD'] = CBHP_2019['B085']/CBHP_2019['B001']


#2.Research design
#2.1. Check if there are outliers and decide whether they need to be eliminated
##Winsorize ratios
cluster_ratios['AdjROA'] = stats.mstats.winsorize(cluster_ratios['AdjROA'], limits=0.05)
cluster_ratios['FATO'] = stats.mstats.winsorize(cluster_ratios['FATO'], limits=0.05)
cluster_ratios['CUR'] = stats.mstats.winsorize(cluster_ratios['CUR'], limits=0.05)
cluster_ratios['LTD'] = stats.mstats.winsorize(cluster_ratios['LTD'], limits=0.05)
cluster_index = cluster_ratios.index

#2.2.Standardize variables (z-score)
##calculate zscores
scaler = StandardScaler()
scaler.fit(cluster_ratios)
print(scaler.mean_)
standard = scaler.transform(cluster_ratios)
standard_ratios = pd.DataFrame(standard, index = cluster_index, columns = ratios_cols)


#3.Check if there is multicollinearity
def calc_vif(X):
    vif = pd.DataFrame()
    vif['variables'] = X.columns
    vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
    return vif

X = standard_ratios.iloc[:, :-1]
calc_vif(X)
```

```
#4.Partitioning procedure
#4.1.Find number of clusters with hierarchical agglomerative method (dendrogram)
Z = linkage(standard_ratios, 'ward')

def fancy_dendrogram(*args, **kwargs):
    max_d = kwargs.pop('max_d', None)
    if max_d and 'color_threshold' not in kwargs:
        kwargs['color_threshold'] = max_d
    annotate_above = kwargs.pop('annotate_above', 0)

    ddata = dendrogram(*args, **kwargs)

    if not kwargs.get('no_plot', False):
        plt.title('Hierarchical Clustering Dendrogram (truncated)')
        plt.xlabel('Cluster size')
        plt.ylabel('Euclidean distance')
        for i, d, c in zip(ddata['icoord'], ddata['dcoord'], ddata['color_list']):
            x = 0.5*sum(i[1:3])
            y = d[1]
            if y > annotate_above:
                plt.plot(x, y, 'o', c=c)
                plt.annotate('%.3g'%y, (x,y), xytext=(0, -5), textcoords='offset points',
                             va='top', ha='center')
            if max_d:
                plt.axhline(y=max_d, c='k')
    return ddata

max_d = 70

fancy_dendrogram(Z, truncate_mode='lastp',
                p=10, leaf_rotation=90.,
                leaf_font_size=12.,
                show_contracted=True,
                annotate_above=10,
                max_d=max_d)

plt.show()
```

```
#4.2.Find centroids for Agglomerative approach
##Run Agglomerative
hc = AgglomerativeClustering(n_clusters = 4, affinity = 'euclidean', linkage = 'ward')
y_hc = hc.fit_predict(standard_ratios)
labels2 = pd.DataFrame(hc.labels_, index = cluster_index)
labeledfirms2 = pd.concat((cluster_ratios, labels2), axis = 1)
labeledfirms2 = labeledfirms2.rename({0:'labels2'}, axis = 1)

##Find centroids
clf = NearestCentroid()
clf.fit(standard_ratios, y_hc)
hc_centroids = clf.centroids_
print(hc_centroids)

#4.3.Initiate k-means with HYBRID approach
##Run KMeans
kmeans = KMeans(init = hc_centroids,
                n_clusters = 4,
                max_iter = 500)

kmeans.fit(standard_ratios)

labels = pd.DataFrame(kmeans.labels_, index = cluster_index)
labeledfirms = pd.concat((cluster_ratios, labels), axis = 1)
labeledfirms = labeledfirms.rename({0:'labels'}, axis = 1)
labeledfirms['labels'].replace({0:'A', 1:'B', 2:'D', 3:'C'}, inplace = True)
labeledfirms['Constant'] = 'Data'

##Strip Plots
f, axes = plt.subplots(1, 4, figsize = (14, 10))
f.subplots_adjust(hspace = 1, wspace = .6)
```

```python
ax = sns.stripplot(x = labeledfirms['Constant'],
                   y = labeledfirms['AdjROA'],
                   hue = labeledfirms['labels'],
                   hue_order = ['A', 'B', 'C', 'D'],
                   palette = 'Paired',
                   jitter = True,
                   ax = axes[0], s=3)

ax.set(xlabel=None)
ax.set(xticklabels=[])

ax = sns.stripplot(x = labeledfirms['Constant'],
                   y = labeledfirms['FATO'],
                   hue = labeledfirms['labels'],
                   hue_order = ['A', 'B', 'C', 'D'],
                   palette = 'Paired',
                   jitter = True,
                   ax = axes[1], s=3)

ax.set(xlabel=None)
ax.set(xticklabels=[])

ax = sns.stripplot(x = labeledfirms['Constant'],
                   y = labeledfirms['CUR'],
                   hue = labeledfirms['labels'],
                   hue_order = ['A', 'B', 'C', 'D'],
                   palette = 'Paired',
                   jitter = True,
                   ax = axes[2], s=3)

ax.set(xlabel=None)
ax.set(xticklabels=[])

ax = sns.stripplot(x = labeledfirms['Constant'],
                   y = labeledfirms['LTD'],
                   hue = labeledfirms['labels'],
                   hue_order = ['A', 'B', 'C', 'D'],
                   palette = 'Paired',
                   jitter = True,
                   ax = axes[3], s=3)

ax.set(xlabel=None)
ax.set(xticklabels=[])

#4.3.Compare hierarchical with k-means
##Calculate Silhouette Score
kmeans_silhouette = silhouette_score(standard_ratios, kmeans.labels_).round(2)
hc_silhouette = silhouette_score(standard_ratios, hc.labels_).round(2)

print(kmeans_silhouette)
print(hc_silhouette)
```

Appendix D: Solving the EGZ-model

Appendix D.1.: Initial guesses for the system of equations

The initial guesses for the unknowns on system of equations were set as follows:

- A and C were set to zero. Keep in mind that A will always be zero.
- Initially, B and D estimated using the formula for B_{proxy} below. In case there was no conversion, other guesses for B and D are used until conversion is reached, that are specified in the Python code below.
- The distress barrier is based on the x^* equation (27).

$$B_{proxy} = 0 - \left\{ \left[(1-\tau) + \frac{\tau \overline{DSR} - \overline{CSR}}{GM_{it}} - (1-\tau) \frac{\overline{OSR}}{GM_{it}} \right] \frac{x_{it}}{r-\mu} - (1-\tau) \frac{(I_{it} + F_i)}{r} \right\} \tag{29}$$

Appendix D.2.: Python code

```
#0.Import libraries
import pandas as pd
import numpy as np
import math
from scipy import stats
from scipy.stats import norm
from scipy.optimize import fsolve
import statsmodels.api as sm
from statsmodels.regression.rolling import RollingOLS
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import random


#1.Prepare Model
#1.1.Create base functions
def base1(Tax, DSR, CSR, GMR, OSR):
    """One of the functions that repeats itself all the time.
    Inputs: Tax, DSR, CSR, GMR, and OSR."""
    return ((1-Tax) + ((Tax*DSR-CSR)/GMR) - ((1-Tax)*OSR/GMR))

def base2(x, rf, Q):
    """One of the functions that repeats itself all the time.
    Inputs: x (gross margin, normalized), rf and Q."""
    return x/(rf-Q)

def base3(Tax, coup, f, rf):
    """One of the functions that repeats itself all the time.
    Inputs: Tax, coup, f, rf"""
    return (1-Tax)*(coup+f)/rf

def x_star(Tax, coup, f, base1):
    """Determines if the firm has to pay distress costs or not.
    Inputs: Tax, coup, f, base1"""
    return ((coup+f)*(1-Tax))/base1
```

```python
def beta1(Vol, rf, Q):
    """One of the roots for the quadratic function.
    Inputs: Vol, rf, Q"""
    return (-1/Vol**2*
            (Q-0.5*Vol**2 -
             ((Q-0.5*Vol**2)**2 + 2*rf*Vol**2)**0.5))

def beta2(Vol, rf, Q):
    """One of the roots for the quadratic function.
    Inputs: Vol, rf, Q"""
    return (-1/Vol**2*
            (Q-0.5*Vol**2 +
             ((Q-0.5*Vol**2)**2 + 2*rf*Vol**2)**0.5))

#1.2.Find values that don't depend on x
##X_STAR
x_starA = x_star(TR, cA, fc, base1(TR, DSR, CSR, GMA, OSR))
x_starB = x_star(TR, cB, fc, base1(TR, DSR, CSR, GMB, OSR))
x_starC = x_star(TR, cC, fc, base1(TR, DSR, CSR, GMC, OSR))
x_starD = x_star(TR, cD, fc, base1(TR, DSR, CSR, GMD, OSR))

##BETA1 AND BETA2
b1 = beta1(Vol, rf, Q2)
b2 = beta2(Vol, rf, Q2)

#1.3.Create a system that joins all boundary conditions
def system(params, *data):
    """System with all boundary conditions to find B, C, D and xd in year 4."""
    A = params[0]
    B = params[1]
    C = params[2]
    D = params[3]
    xd = params[4]

    x_star, beta1, beta2, base1, base3, rf, Q, eta = data


    cond1 = A

    cond2 = (B*x_star**beta2 + base1*x_star/(rf-Q)-base3 - (
        C*x_star**beta1 + D*x_star**beta2 + (1+eta)*(base1*x_star/(rf-Q)-base3)))

    cond3 = (beta2*B*x_star**(beta2-1) + base1*1/(rf-Q) - (
        beta1*C*x_star**(beta1-1) + beta2*D*x_star**(beta2-1) +
        (1+eta)*base1*1/(rf-Q)))

    cond4 = C*xd**beta1 + D*xd**beta2 + (1+eta)*(base1*xd/(rf-Q)-base3)

    cond5 = beta1*C*xd**(beta1-1) + beta2*D*xd**(beta2-1) + (1+eta)*base1*1/(rf-Q)


    return np.array([cond1, cond2, cond3, cond4, cond5])

#1.4.Post-refinancing Equity Function
def post_refE(x, x_star, beta1, beta2, base1, base2, base3, rf, eta, B, C, D):
    """Returns post-refinancing Equity Value."""
    if x >= x_star:
        return (B*x**beta2 + base1*base2-base3)
    else:
        return (C*x**beta1 + D*x**beta2 + (1+eta)*(base1*base2-base3))


###EXAMPLE FOR NORMAL WITH CASH###
#2.Solve Model
#2.1.Prepare Binomial Trees
N = (1500 + 1) #REMEMBER: N + 1
y = 4
dt = y/(N-1)
dy = Vol*np.sqrt(dt)

u = np.exp(dy)
d = 1/u

p_u = (np.exp(Q2*dt) - d)/(u-d)
p_d = 1 - p_u
```

```python
#2.2.Build Binomial Trees
A_xtree = np.zeros([N,N])
A_cashtree = np.zeros([N,N])
A_fcfetree = np.zeros([N,N])
A_injtree = np.zeros([N,N])
A_eqtree = np.zeros([N,N])

B_xtree = np.zeros([N,N])
B_cashtree = np.zeros([N,N])
B_fcfetree = np.zeros([N,N])
B_injtree = np.zeros([N,N])
B_eqtree = np.zeros([N,N])

C_xtree = np.zeros([N,N])
C_cashtree = np.zeros([N,N])
C_fcfetree = np.zeros([N,N])
C_injtree = np.zeros([N,N])
C_eqtree = np.zeros([N,N])

D_xtree = np.zeros([N,N])
D_cashtree = np.zeros([N,N])
D_fcfetree = np.zeros([N,N])
D_injtree = np.zeros([N,N])
D_eqtree = np.zeros([N,N])

#2.3.Populate trees
##A.GROSS MARGIN
A_xtree[0,0] = 1
B_xtree[0,0] = 1
C_xtree[0,0] = 1
D_xtree[0,0] = 1

for i in range(1,N):
    M = i+1
    A_xtree[i,0] = d*A_xtree[i-1,0]
    B_xtree[i,0] = d*B_xtree[i-1,0]
    C_xtree[i,0] = d*C_xtree[i-1,0]
    D_xtree[i,0] = d*D_xtree[i-1,0]
    for j in range(1,M):
        A_xtree[i,j] = u*A_xtree[i-1, j-1]
        B_xtree[i,j] = u*B_xtree[i-1, j-1]
        C_xtree[i,j] = u*C_xtree[i-1, j-1]
        D_xtree[i,j] = u*D_xtree[i-1, j-1]

###Plot###
fig, axs = plt.subplots(1, 4, figsize = (20, 5))
ax1 = axs[0]
ax2 = axs[1]
ax3 = axs[2]
ax4 = axs[3]

ax1.spy(A_xtree, precision=0.0001, markersize=5)
ax2.spy(B_xtree, precision=0.0001, markersize=5)
ax3.spy(C_xtree, precision=0.0001, markersize=5)
ax4.spy(D_xtree, precision=0.0001, markersize=5)
plt.show()

#B.FCFE
def fcfe(x, x_star, Tax, base1, coup, f, eta):
    if x >= x_star:
        return (base1*x-(1-Tax)*(coup+f))
    else:
        return (base1*x-(1-Tax)*(coup+f)*(1+eta))

A_fcfetree[0,0] = fcfe(A_xtree[0,0], x_starA, TR, base1(TR, DSR, CSR, GMA, OSR),
                       cA, fc, ADC)*dt
B_fcfetree[0,0] = fcfe(B_xtree[0,0], x_starB, TR, base1(TR, DSR, CSR, GMB, OSR),
                       cB, fc, ADC)*dt
C_fcfetree[0,0] = fcfe(C_xtree[0,0], x_starC, TR, base1(TR, DSR, CSR, GMC, OSR),
                       cC, fc, ADC)*dt
D_fcfetree[0,0] = fcfe(D_xtree[0,0], x_starD, TR, base1(TR, DSR, CSR, GMD, OSR),
                       cD, fc, ADC)*dt
```

```python
for i in range(1,N):
    for j in range(i+1):
        A_fcfetree[i,j] = fcfe(A_xtree[i,j], x_starA, TR, base1(TR, DSR, CSR, GMA, OSR),
                               cA, fc, ADC)*dt
        B_fcfetree[i,j] = fcfe(B_xtree[i,j], x_starB, TR, base1(TR, DSR, CSR, GMB, OSR),
                               cB, fc, ADC)*dt
        C_fcfetree[i,j] = fcfe(C_xtree[i,j], x_starC, TR, base1(TR, DSR, CSR, GMC, OSR),
                               cC, fc, ADC)*dt
        D_fcfetree[i,j] = fcfe(D_xtree[i,j], x_starD, TR, base1(TR, DSR, CSR, GMD, OSR),
                               cD, fc, ADC)*dt

###Plot###
fig, axs = plt.subplots(1, 4, figsize = (20, 5))
ax1 = axs[0]
ax2 = axs[1]
ax3 = axs[2]
ax4 = axs[3]

ax1.spy(A_fcfetree, precision=0.0001, markersize=5)
ax2.spy(B_fcfetree, precision=0.0001, markersize=5)
ax3.spy(C_fcfetree, precision=0.0001, markersize=5)
ax4.spy(D_fcfetree, precision=0.0001, markersize=5)
plt.show()

#C.CASH
A_cashtree[0,0] = cashA
B_cashtree[0,0] = cashB
C_cashtree[0,0] = cashC
D_cashtree[0,0] = cashD

for i in range(1,N):
    M = i + 1
    for j in range(1,M):
        A_cashtree[i,j] = max(0,((A_cashtree[i-1,j-1]+A_cashtree[i-1,j])/2)+A_fcfetree[i,j])
        B_cashtree[i,j] = max(0,((B_cashtree[i-1,j-1]+B_cashtree[i-1,j])/2)+B_fcfetree[i,j])
        C_cashtree[i,j] = max(0,((C_cashtree[i-1,j-1]+C_cashtree[i-1,j])/2)+C_fcfetree[i,j])
        D_cashtree[i,j] = max(0,((D_cashtree[i-1,j-1]+D_cashtree[i-1,j])/2)+D_fcfetree[i,j])

        A_cashtree[i,0] = max(0, A_cashtree[i-1,0]+A_fcfetree[i,0])
        B_cashtree[i,0] = max(0, B_cashtree[i-1,0]+B_fcfetree[i,0])
        C_cashtree[i,0] = max(0, C_cashtree[i-1,0]+C_fcfetree[i,0])
        D_cashtree[i,0] = max(0, D_cashtree[i-1,0]+D_fcfetree[i,0])

        A_cashtree[i,i] = max(0, A_cashtree[i-1,i-1]+A_fcfetree[i,i])
        B_cashtree[i,i] = max(0, B_cashtree[i-1,i-1]+B_fcfetree[i,i])
        C_cashtree[i,i] = max(0, C_cashtree[i-1,i-1]+C_fcfetree[i,i])
        D_cashtree[i,i] = max(0, D_cashtree[i-1,i-1]+D_fcfetree[i,i])

###Plot###
fig, axs = plt.subplots(1, 4, figsize = (20, 5))
ax1 = axs[0]
ax2 = axs[1]
ax3 = axs[2]
ax4 = axs[3]

ax1.spy(A_cashtree, precision=0.0001, markersize=5)
ax2.spy(B_cashtree, precision=0.0001, markersize=5)
ax3.spy(C_cashtree, precision=0.0001, markersize=5)
ax4.spy(D_cashtree, precision=0.0001, markersize=5)
plt.show()

#D.CASH INJECTION
def injection(curr_cash, fcfe):
    'Dynamics of the SH injection throughout the tree.'
    if curr_cash > 0:
        return 0
    elif (fcfe < 0) & (curr_cash == 0):
        return fcfe
    else:
        return 0

A_injtree[0,0] = injection(cashA, A_fcfetree[0,0])
B_injtree[0,0] = injection(cashB, B_fcfetree[0,0])
C_injtree[0,0] = injection(cashC, C_fcfetree[0,0])
D_injtree[0,0] = injection(cashD, D_fcfetree[0,0])
```

```python
for i in range(1,N):
    A_injtree[i,0] = injection(A_cashtree[i-1,0], A_fcfetree[i,0])
    B_injtree[i,0] = injection(B_cashtree[i-1,0], B_fcfetree[i,0])
    C_injtree[i,0] = injection(C_cashtree[i-1,0], C_fcfetree[i,0])
    D_injtree[i,0] = injection(D_cashtree[i-1,0], D_fcfetree[i,0])
    for j in range(i+1):
        A_injtree[i,j] = injection(A_cashtree[i-1,j-1], A_fcfetree[i,j])
        B_injtree[i,j] = injection(B_cashtree[i-1,j-1], B_fcfetree[i,j])
        C_injtree[i,j] = injection(C_cashtree[i-1,j-1], C_fcfetree[i,j])
        D_injtree[i,j] = injection(D_cashtree[i-1,j-1], D_fcfetree[i,j])

###Plot###
fig, axs = plt.subplots(1, 4, figsize = (20, 5))
ax1 = axs[0]
ax2 = axs[1]
ax3 = axs[2]
ax4 = axs[3]

ax1.spy(A_injtree, precision=0.0001, markersize=5)
ax2.spy(B_injtree, precision=0.0001, markersize=5)
ax3.spy(C_injtree, precision=0.0001, markersize=5)
ax4.spy(D_injtree, precision=0.0001, markersize=5)
plt.show()
```

```python
#E.EQUITY
###Functions for initial guess###
def B_proxy(base1, base2, base3, x, beta2):
    """Initial guess for B. Required to solve the system of equations.
    Has a 0.1 floor."""
    B_proxy = (0-(base1*base2-base3))/x**beta2
    return max(B_proxy, 0.1)

def B_proxyB(base1, base3, x, rf, Q, beta2):
    """Initial guess for B. Required to solve the system of equations."""
    B_proxyB = (0-(base1*(min(x, 1.25)/(rf-Q))-base3))/x**beta2
    return B_proxyB

###Solve System of Equations###
#FIRM A
A_BD0 = pd.DataFrame(index = range(2), columns = range(N))
A_BCDxd = pd.DataFrame(index = ['A', 'B', 'C', 'D', 'xd'], columns = range(N))

for j in range(N):
    ##First attempt at conversion
    A_BD0.iloc[0,j] = B_proxy(base1(TR, DSR, CSR, GMA, OSR),
                              base2(A_xtree[-1,j], rf, Q2),
                              base3(TR, cA, fc, rf),
                              A_xtree[-1,j], b2)
    guessing = np.array([0, A_BD0.iloc[0,j], 0, A_BD0.iloc[0,j], x_starA])
    zsol1 = fsolve(system, guessing,
                   args = (x_starA,
                           b1, b2,
                           base1(TR, DSR, CSR, GMA, OSR),
                           base3(TR, cA, fc, rf),
                           rf, Q2, ADC), full_output = True)
    if zsol1[-2] == 1:
        A_BCDxd.iloc[:,j] = zsol1[0]
    ##Second attempt at conversion
    else:
        A_BD0.iloc[1,j] = B_proxyB(base1(TR, DSR, CSR, GMA, OSR),
                                   base3(TR, cA, fc, rf),
                                   A_xtree[-1,j], rf, Q2, b2)
        guessing = np.array([0, A_BD0.iloc[1,j]*0.5, 0, A_BD0.iloc[1,j]*0.5, 1])
        zsol2 = fsolve(system, guessing,
                       args = (x_starA,
                               b1, b2,
                               base1(TR, DSR, CSR, GMA, OSR),
                               base3(TR, cA, fc, rf),
                               rf, Q2, ADC), full_output = True)
        if zsol2[-2] == 1:
            A_BCDxd.iloc[:,j] = zsol2[0]
        ##Third attempt at conversion
        else:
            guessing = np.array([0, A_BD0.iloc[0,j]*0.5, 0, A_BD0.iloc[0,j]*0.5, 1])
            zsol3 = fsolve(system, guessing,
                           args = (x_starA,
                                   b1, b2,
                                   base1(TR, DSR, CSR, GMA, OSR),
                                   base3(TR, cA, fc, rf),
                                   rf, Q2, ADC), full_output = True)
            if zsol3[-2] == 1:
                A_BCDxd.iloc[:,j] = zsol3[0]
```

```python
            ##Fourth attempt at conversion
            else:
                guessing = np.array([0, 0.01, 0.01, 0.01, 0.01])
                zsol4 = fsolve(system, guessing,
                               args = (x_starA,
                                       b1, b2,
                                       base1(TR, DSR, CSR, GMA, OSR),
                                       base3(TR, cA, fc, rf),
                                       rf, Q2, ADC), full_output = True)
                if zsol4[-2] == 1:
                    A_BCDxd.iloc[:,j] = zsol4[0]
                ##Fifth attempt at conversion
                else:
                    guessing = np.array([0, 0.001, 0.001, 0.001, 0.001])
                    zsol5 = fsolve(system, guessing,
                                   args = (x_starA,
                                           b1, b2,
                                           base1(TR, DSR, CSR, GMA, OSR),
                                           base3(TR, cA, fc, rf),
                                           rf, Q2, ADC), full_output = True)
                    if zsol5[-2] == 1:
                        A_BCDxd_1.iloc[:,j] = zsol5[0]
                    ##Sixth attempt at conversion
                    else:
                        guessing = np.array([0, 0.0001, 0.0001, 0.0001, 0.0001])
                        zsol6 = fsolve(system, guessing,
                                       args = (x_starA,
                                               b1, b2,
                                               base1(TR, DSR, CSR, GMA, OSR),
                                               base3(TR, cA, fc, rf),
                                               rf, Q2, ADC), full_output = True)
                        if zsol6[-2] ==1:
                            A_BCDxd.iloc[:,j] = zsol6[0]

display(A_BD0)
display(A_BCDxd)

#FIRM B
B_BD0 = pd.DataFrame(index = range(2), columns = range(N))
B_BCDxd = pd.DataFrame(index = ['A', 'B', 'C', 'D', 'xd'], columns = range(N))

for j in range(N):
    ##First attempt at conversion
    B_BD0.iloc[0,j] = B_proxy(base1(TR, DSR, CSR, GMB, OSR),
                              base2(B_xtree[-1,j], rf, Q2),
                              base3(TR, cB, fc, rf),
                              B_xtree[-1,j], b2)
    guessing = np.array([0, B_BD0.iloc[0,j], 0, B_BD0.iloc[0,j], x_starB])
    zsol1 = fsolve(system, guessing,
                   args = (x_starB,
                           b1, b2,
                           base1(TR, DSR, CSR, GMB, OSR),
                           base3(TR, cB, fc, rf),
                           rf, Q2, ADC), full_output = True)
    if zsol1[-2] == 1:
        B_BCDxd.iloc[:,j] = zsol1[0]
    ##Second attempt at conversion
    else:
        B_BD0.iloc[1,j] = B_proxyB(base1(TR, DSR, CSR, GMB, OSR),
                                   base3(TR, cB, fc, rf),
                                   B_xtree[-1,j], rf, Q2, b2)
        guessing = np.array([0, B_BD0.iloc[1,j]*0.5, 0, B_BD0.iloc[1,j]*0.5, 1])
        zsol2 = fsolve(system, guessing,
                       args = (x_starB,
                               b1, b2,
                               base1(TR, DSR, CSR, GMB, OSR),
                               base3(TR, cB, fc, rf),
                               rf, Q2, ADC), full_output = True)
        if zsol2[-2] == 1:
            B_BCDxd.iloc[:,j] = zsol2[0]
        ##Third attempt at conversion
        else:
            guessing = np.array([0, B_BD0.iloc[0,j]*0.5, 0, B_BD0.iloc[0,j]*0.5, 1])
            zsol3 = fsolve(system, guessing,
                           args = (x_starB,
                                   b1, b2,
                                   base1(TR, DSR, CSR, GMB, OSR),
                                   base3(TR, cB, fc, rf),
                                   rf, Q2, ADC), full_output = True)
            if zsol3[-2] == 1:
                B_BCDxd.iloc[:,j] = zsol3[0]
```

```
            ##Fourth attempt at conversion
            else:
                guessing = np.array([0, 0.01, 0.01, 0.01, 0.01])
                zsol4 = fsolve(system, guessing,
                               args = (x_starB,
                                       b1, b2,
                                       base1(TR, DSR, CSR, GMB, OSR),
                                       base3(TR, cB, fc, rf),
                                       rf, Q2, ADC), full_output = True)
                if zsol4[-2] == 1:
                    B_BCDxd.iloc[:,j] = zsol4[0]
                ##Fifth attempt at conversion
                else:
                    guessing = np.array([0, 0.001, 0.001, 0.001, 0.001])
                    zsol5 = fsolve(system, guessing,
                                   args = (x_starB,
                                           b1, b2,
                                           base1(TR, DSR, CSR, GMB, OSR),
                                           base3(TR, cB, fc, rf),
                                           rf, Q2, ADC), full_output = True)
                    if zsol5[-2] == 1:
                        B_BCDxd.iloc[:,j] = zsol5[0]
                    ##Sixth attempt at conversion
                    else:
                        guessing = np.array([0, 0.0001, 0.0001, 0.0001, 0.0001])
                        zsol6 = fsolve(system, guessing,
                                       args = (x_starB,
                                               b1, b2,
                                               base1(TR, DSR, CSR, GMB, OSR),
                                               base3(TR, cB, fc, rf),
                                               rf, Q2, ADC), full_output = True)
                        if zsol6[-2] ==1:
                            B_BCDxd.iloc[:,j] = zsol6[0]

display(B_BD0)
display(B_BCDxd)

#FIRM C
C_BD0 = pd.DataFrame(index = range(2), columns = range(N))
C_BCDxd = pd.DataFrame(index = ['A', 'B', 'C', 'D', 'xd'], columns = range(N))

for j in range(N):
    ##First attempt at conversion
    C_BD0.iloc[0,j] = B_proxy(base1(TR, DSR, CSR, GMC, OSR),
                              base2(C_xtree[-1,j], rf, Q2),
                              base3(TR, cC, fc, rf),
                              C_xtree[-1,j], b2)
    guessing = np.array([0, C_BD0.iloc[0,j], 0, C_BD0.iloc[0,j], x_starC])
    zsol1 = fsolve(system, guessing,
                   args = (x_starC,
                           b1, b2,
                           base1(TR, DSR, CSR, GMB, OSR),
                           base3(TR, cB, fc, rf),
                           rf, Q2, ADC), full_output = True)
    if zsol1[-2] == 1:
        C_BCDxd.iloc[:,j] = zsol1[0]
    ##Second attempt at conversion
    else:
        C_BD0.iloc[1,j] = B_proxyB(base1(TR, DSR, CSR, GMC, OSR),
                                   base3(TR, cC, fc, rf),
                                   C_xtree[-1,j], rf, Q2, b2)
        guessing = np.array([0, C_BD0.iloc[1,j]*0.5, 0, C_BD0.iloc[1,j]*0.5, 1])
        zsol2 = fsolve(system, guessing,
                       args = (x_starC,
                               b1, b2,
                               base1(TR, DSR, CSR, GMC, OSR),
                               base3(TR, cC, fc, rf),
                               rf, Q2, ADC), full_output = True)
        if zsol2[-2] == 1:
            C_BCDxd.iloc[:,j] = zsol2[0]
        ##Third attempt at conversion
        else:
            guessing = np.array([0, C_BD0.iloc[0,j]*0.5, 0, C_BD0.iloc[0,j]*0.5, 1])
            zsol3 = fsolve(system, guessing,
                           args = (x_starC,
                                   b1, b2,
                                   base1(TR, DSR, CSR, GMC, OSR),
                                   base3(TR, cC, fc, rf),
                                   rf, Q2, ADC), full_output = True)
            if zsol3[-2] == 1:
                C_BCDxd.iloc[:,j] = zsol3[0]
```

```
            ##Fourth attempt at conversion
            else:
                guessing = np.array([0, 0.01, 0.01, 0.01, 0.01])
                zsol4 = fsolve(system, guessing,
                               args = (x_starC,
                                       b1, b2,
                                       base1(TR, DSR, CSR, GMB, OSR),
                                       base3(TR, cC, fc, rf),
                                       rf, Q2, ADC), full_output = True)
                if zsol4[-2] == 1:
                    C_BCDxd.iloc[:,j] = zsol4[0]
                ##Fifth attempt at conversion
                else:
                    guessing = np.array([0, 0.001, 0.001, 0.001, 0.001])
                    zsol5 = fsolve(system, guessing,
                                   args = (x_starC,
                                           b1, b2,
                                           base1(TR, DSR, CSR, GMB, OSR),
                                           base3(TR, cC, fc, rf),
                                           rf, Q2, ADC), full_output = True)
                    if zsol5[-2] == 1:
                        C_BCDxd.iloc[:,j] = zsol5[0]
                    ##Sixth attempt at conversion
                    else:
                        guessing = np.array([0, 0.0001, 0.0001, 0.0001, 0.0001])
                        zsol6 = fsolve(system, guessing,
                                       args = (x_starC,
                                               b1, b2,
                                               base1(TR, DSR, CSR, GMB, OSR),
                                               base3(TR, cC, fc, rf),
                                               rf, Q2, ADC), full_output = True)
                        if zsol6[-2] ==1:
                            C_BCDxd.iloc[:,j] = zsol6[0]

display(C_BD0)
display(C_BCDxd)

#FIRM D
D_BD0 = pd.DataFrame(index = range(2), columns = range(N))
D_BCDxd = pd.DataFrame(index = ['A', 'B', 'C', 'D', 'xd'], columns = range(N))

for j in range(N):
    ##First attempt at conversion
    D_BD0.iloc[0,j] = B_proxy(base1(TR, DSR, CSR, GMD, OSR),
                              base2(D_xtree[-1,j], rf, Q2),
                              base3(TR, cD, fc, rf),
                              D_xtree[-1,j], b2)
    guessing = np.array([0, D_BD0.iloc[0,j], 0, D_BD0.iloc[0,j], x_starD])
    zsol1 = fsolve(system, guessing,
                   args = (x_starD,
                           b1, b2,
                           base1(TR, DSR, CSR, GMD, OSR),
                           base3(TR, cD, fc, rf),
                           rf, Q2, ADC), full_output = True)
    if zsol1[-2] == 1:
        D_BCDxd.iloc[:,j] = zsol1[0]
    ##Second attempt at conversion
    else:
        D_BD0.iloc[1,j] = B_proxyB(base1(TR, DSR, CSR, GMD, OSR),
                                   base3(TR, cD, fc, rf),
                                   D_xtree[-1,j], rf, Q2, b2)
        guessing = np.array([0, D_BD0.iloc[1,j]*0.5, 0, D_BD0.iloc[1,j]*0.5, 1])
        zsol2 = fsolve(system, guessing,
                       args = (x_starD,
                               b1, b2,
                               base1(TR, DSR, CSR, GMD, OSR),
                               base3(TR, cD, fc, rf),
                               rf, Q2, ADC), full_output = True)
        if zsol2[-2] == 1:
            D_BCDxd.iloc[:,j] = zsol2[0]
        ##Third attempt at conversion
        else:
            guessing = np.array([0, D_BD0.iloc[0,j]*0.5, 0, D_BD0.iloc[0,j]*0.5, 1])
            zsol3 = fsolve(system, guessing,
                           args = (x_starD,
                                   b1, b2,
                                   base1(TR, DSR, CSR, GMD, OSR),
                                   base3(TR, cD, fc, rf),
                                   rf, Q2, ADC), full_output = True)
            if zsol3[-2] == 1:
                D_BCDxd.iloc[:,j] = zsol3[0]
```

```python
            ##Fourth attempt at conversion
            else:
                guessing = np.array([0, 0.01, 0.01, 0.01, 0.01])
                zsol4 = fsolve(system, guessing,
                               args = (x_starD,
                                       b1, b2,
                                       base1(TR, DSR, CSR, GMD, OSR),
                                       base3(TR, cD, fc, rf),
                                       rf, Q2, ADC), full_output = True)
                if zsol4[-2] == 1:
                    D_BCDxd.iloc[:,j] = zsol4[0]
                ##Fifth attempt at conversion
                else:
                    guessing = np.array([0, 0.001, 0.001, 0.001, 0.001])
                    zsol5 = fsolve(system, guessing,
                                   args = (x_starD,
                                           b1, b2,
                                           base1(TR, DSR, CSR, GMD, OSR),
                                           base3(TR, cD, fc, rf),
                                           rf, Q2, ADC), full_output = True)
                    if zsol5[-2] == 1:
                        D_BCDxd.iloc[:,j] = zsol5[0]
                    ##Sixth attempt at conversion
                    else:
                        guessing = np.array([0, 0.0001, 0.0001, 0.0001, 0.0001])
                        zsol6 = fsolve(system, guessing,
                                       args = (x_starD,
                                               b1, b2,
                                               base1(TR, DSR, CSR, GMD, OSR),
                                               base3(TR, cD, fc, rf),
                                               rf, Q2, ADC), full_output = True)
                        if zsol6[-2] ==1:
                            D_BCDxd.iloc[:,j] = zsol6[0]

display(D_BD0)
display(D_BCDxd)

###Find the Value of Equity using the variables B, C, D and xd
A_postref = pd.DataFrame(index = ['post_ref'], columns = range(N))
B_postref = pd.DataFrame(index = ['post_ref'], columns = range(N))
C_postref = pd.DataFrame(index = ['post_ref'], columns = range(N))
D_postref = pd.DataFrame(index = ['post_ref'], columns = range(N))

#FIRM A
for j in range(N):
    if A_xtree[-1,j] <= A_BCDxd.iloc[4,j]:
        A_postref.iloc[0,j] = 0
    else:
        A_postref.iloc[0,j] = post_refE(A_xtree[-1,j], x_starA,
                                        b1, b2,
                                        base1(TR, DSR, CSR, GMA, OSR),
                                        base2(A_xtree[-1,j], rf, Q2),
                                        base3(TR, cA, fc, rf),
                                        rf, ADC,
                                        A_BCDxd.iloc[1,j], A_BCDxd.iloc[2,j], A_BCDxd.iloc[3,j])

#FIRM B
for j in range(N):
    if B_xtree[-1,j] <= B_BCDxd.iloc[4,j]:
        B_postref.iloc[0,j] = 0
    else:
        B_postref.iloc[0,j] = post_refE(B_xtree[-1,j], x_starB,
                                        b1, b2,
                                        base1(TR, DSR, CSR, GMB, OSR),
                                        base2(B_xtree[-1,j], rf, Q2),
                                        base3(TR, cB, fc, rf),
                                        rf, ADC,
                                        B_BCDxd.iloc[1,j], B_BCDxd.iloc[2,j], B_BCDxd.iloc[3,j])

#FIRM C
for j in range(N):
    if C_xtree[-1,j] <= C_BCDxd.iloc[4,j]:
        C_postref.iloc[0,j] = 0
    else:
        C_postref.iloc[0,j] = post_refE(C_xtree[-1,j], x_starC,
                                        b1, b2,
                                        base1(TR, DSR, CSR, GMC, OSR),
                                        base2(C_xtree[-1,j], rf, Q2),
                                        base3(TR, cC, fc, rf),
                                        rf, ADC,
                                        C_BCDxd.iloc[1,j], C_BCDxd.iloc[2,j], C_BCDxd.iloc[3,j])
```

```
#FIRM D
for j in range(N):
    if D_xtree[-1,j] <= D_BCDxd.iloc[4,j]:
        D_postref.iloc[0,j] = 0
    else:
        D_postref.iloc[0,j] = post_refE(D_xtree[-1,j], x_starD,
                                        b1, b2,
                                        base1(TR, DSR, CSR, GMD, OSR),
                                        base2(D_xtree[-1,j], rf, Q2),
                                        base3(TR, cD, fc, rf),
                                        rf, ADC,
                                        D_BCDxd.iloc[1,j], D_BCDxd.iloc[2,j], D_BCDxd.iloc[3,j])

display(A_postref, B_postref, C_postref, D_postref)

###Estimate dividend in final year
divA = pd.DataFrame(index = ['div'], columns = range(N))
divB = pd.DataFrame(index = ['div'], columns = range(N))
divC = pd.DataFrame(index = ['div'], columns = range(N))
divD = pd.DataFrame(index = ['div'], columns = range(N))

#FIRM A
for j in range(N):
    if A_xtree[-1,j] <= A_BCDxd.iloc[4,j]:
        divA.iloc[0,j] = 0
    elif A_cashtree[-1,j] >= cash_A:
        divA.iloc[0,j] = A_cashtree[-1,j] - cash_A
    elif A_cashtree[-1,j] < cash_A:
        divA.iloc[0,j] = cash_A - A_cashtree[-1,j]

#FIRM B
for j in range(N):
    if B_xtree[-1,j] <= B_BCDxd.iloc[4,j]:
        divB.iloc[0,j] = 0
    elif B_cashtree[-1,j] >= cash_B:
        divB.iloc[0,j] = B_cashtree[-1,j] - cash_B
    elif B_cashtree[-1,j] < cash_B:
        divB.iloc[0,j] = cash_B - B_cashtree[-1,j]

#FIRM C
for j in range(N):
    if C_xtree[-1,j] <= C_BCDxd.iloc[4,j]:
        divC.iloc[0,j] = 0
    elif C_cashtree[-1,j] >= cash_C:
        divC.iloc[0,j] = C_cashtree[-1,j] - cash_C
    elif C_cashtree[-1,j] < cash_C:
        divC.iloc[0,j] = cash_C - C_cashtree[-1,j]

#FIRM D
for j in range(N):
    if D_xtree[-1,j] <= D_BCDxd.iloc[4,j]:
        divD.iloc[0,j] = 0
    elif D_cashtree[-1,j] >= cash_D:
        divD.iloc[0,j] = D_cashtree[-1,j] - cash_D
    elif D_cashtree[-1,j] < cash_D:
        divD.iloc[0,j] = cash_D - D_cashtree[-1,j]

display(divA, divB, divC, divD)

###Set the last nodes of the tree
A_eqtree[-1,:] = A_postref.iloc[0,:]+divA.iloc[0,:]
B_eqtree[-1,:] = B_postref.iloc[0,:]+divB.iloc[0,:]
C_eqtree[-1,:] = C_postref.iloc[0,:]+divC.iloc[0,:]
D_eqtree[-1,:] = D_postref.iloc[0,:]+divD.iloc[0,:]

###Backpropagate to fill out tree
for i in range(N-2,-1,-1): #start, stop, step
    for j in range(i+1):
        A_eqtree[i,j] = max(0, np.exp(-rf*dt)*
                           (p_d*A_eqtree[i+1,j]+ #down prob
                            p_u*A_eqtree[i+1,j+1]+ #up prob
                            A_injtree[i,j])) #adding the injections

        B_eqtree[i,j] = max(0, np.exp(-rf*dt)*
                           (p_d*B_eqtree[i+1,j]+ #down prob
                            p_u*B_eqtree[i+1,j+1]+ #up prob
                            B_injtree[i,j])) #adding the injections
```

```
        C_eqtree[i,j] = max(0, np.exp(-rf*dt)*
                             (p_d*C_eqtree[i+1,j]+ #down prob
                              p_u*C_eqtree[i+1,j+1]+ #up prob
                              C_injtree[i,j])) #adding the injections

        D_eqtree[i,j] = max(0, np.exp(-rf*dt)*
                             (p_d*D_eqtree[i+1,j]+ #down prob
                              p_u*D_eqtree[i+1,j+1]+ #up prob
                              D_injtree[i,j])) #adding the injections

###Plot###
fig, axs = plt.subplots(1, 4, figsize = (20, 5))
ax1 = axs[0]
ax2 = axs[1]
ax3 = axs[2]
ax4 = axs[3]

ax1.spy(A_eqtree, precision=0.0001, markersize=5)
ax2.spy(B_eqtree, precision=0.0001, markersize=5)
ax3.spy(C_eqtree, precision=0.0001, markersize=5)
ax4.spy(D_eqtree, precision=0.0001, markersize=5)
plt.show()

###Find the value of equity in time zero
print(A_eqtree[0,0], B_eqtree[0,0], C_eqtree[0,0], D_eqtree[0,0])


#3.Credit Risk
#3.1.Monte Carlo simulations
num_reps = 10000
random_df = pd.DataFrame(np.random.standard_normal(size = ((N-1), num_reps,))) #random numbers
drift2 = P2

montecarlo = pd.DataFrame(columns = range(num_reps), index = range(N))
montecarlo.iloc[0,:] = np.log(1)

for i in range(1, N):
    montecarlo.iloc[i,:] = montecarlo.iloc[i-1,:] + drift2*dt + Vol*(dt**0.5)*random_df.iloc[i-1,:]

montecarlo = montecarlo.apply(pd.to_numeric)
montecarlo = np.exp(montecarlo)
display(montecarlo)

#3.2.Default barriers at each moment in time
##Match Equity close to zero to GM tree
A_matched = pd.DataFrame(np.where(A_eqtree < 0.0001, A_xtree, 0))
B_matched = pd.DataFrame(np.where(B_eqtree < 0.0001, B_xtree, 0))
C_matched = pd.DataFrame(np.where(C_eqtree < 0.0001, C_xtree, 0))
D_matched = pd.DataFrame(np.where(D_eqtree < 0.0001, D_xtree, 0))

display(A_matched, B_matched, C_matched, D_matched)

##Find the maximum value of the matched GM in each row --> this is the xd!
A_xds = pd.DataFrame(index = range(N), columns = ['Xd', 'Count', 'PD', 'AdjDD'])
B_xds = pd.DataFrame(index = range(N), columns = ['Xd', 'Count', 'PD', 'AdjDD'])
C_xds = pd.DataFrame(index = range(N), columns = ['Xd', 'Count', 'PD', 'AdjDD'])
D_xds = pd.DataFrame(index = range(N), columns = ['Xd', 'Count', 'PD', 'AdjDD'])

for i in range(N-1):
    A_xds.iloc[i,0] = A_matched.iloc[i,:].max()
    B_xds.iloc[i,0] = B_matched.iloc[i,:].max()
    C_xds.iloc[i,0] = C_matched.iloc[i,:].max()
    D_xds.iloc[i,0] = D_matched.iloc[i,:].max()

display(A_xds, B_xds, C_xds, D_xds)

##Count how many times the simulated values go below the barrier
aux_A = pd.DataFrame(columns = range(num_reps), index = range(N))
aux_B = pd.DataFrame(columns = range(num_reps), index = range(N))
aux_C = pd.DataFrame(columns = range(num_reps), index = range(N))
aux_D = pd.DataFrame(columns = range(num_reps), index = range(N))

for i in range(N):
    for j in range(num_reps):
        if montecarlo.iloc[i,j] < A_xds.iloc[i,0]:
            aux_A.iloc[i,j:] = 1
            break
```

```python
for i in range(N):
    for j in range(num_reps):
        if montecarlo.iloc[i,j] < B_xds.iloc[i,0]:
            aux_B.iloc[i,j:] = 1
            break

for i in range(N):
    for j in range(num_reps):
        if montecarlo.iloc[i,j] < C_xds.iloc[i,0]:
            aux_C.iloc[i,j:] = 1
            break

for i in range(N):
    for j in range(num_reps):
        if montecarlo.iloc[i,j] < D_xds.iloc[i,0]:
            aux_D.iloc[i,j:] = 1
            break

aux_A = aux_A.replace(np.nan, 0)
aux_B = aux_B.replace(np.nan, 0)
aux_C = aux_C.replace(np.nan, 0)
aux_D = aux_D.replace(np.nan, 0)
display(aux_A, aux_B, aux_C, aux_D)


#3.3.Estimate probability of default
#3.4.Estimate Adjusted DD (Inverse PD)
for j in range(N):
    A_xds.iloc[j,1] = aux_A.iloc[:,j].sum()
    A_xds.iloc[j,2] = A_xds.iloc[j,1]/num_reps
    A_xds.iloc[j,3] = -norm.ppf(A_xds.iloc[j,2])

    B_xds.iloc[j,1] = aux_B.iloc[:,j].sum()
    B_xds.iloc[j,2] = B_xds.iloc[j,1]/num_reps
    B_xds.iloc[j,3] = -norm.ppf(B_xds.iloc[j,2])

    C_xds.iloc[j,1] = aux_C.iloc[:,j].sum()
    C_xds.iloc[j,2] = C_xds.iloc[j,1]/num_reps
    C_xds.iloc[j,3] = -norm.ppf(C_xds.iloc[j,2])

    D_xds.iloc[j,1] = aux_D.iloc[:,j].sum()
    D_xds.iloc[j,2] = D_xds.iloc[j,1]/num_reps
    D_xds.iloc[j,3] = -norm.ppf(D_xds.iloc[j,2])
```