FUNDAÇÃO GETULIO VARGAS
ESCOLA DE ECONOMIA DE SÃO PAULO

TIAGO VILAS BOAS CORDEIRO

PREDIÇÃO DE DEFAULT DE EMPRESAS: TÉCNICAS DE MACHINE LEARNING EM DADOS DESBALANCEADOS

SÃO PAULO

2020

TIAGO VILAS BOAS CORDEIRO

PREDIÇÃO DE DEFAULT DE EMPRESAS: TÉCNICAS DE MACHINE LEARNING EM DADOS DESBALANCEADOS

Dissertação apresentada ao Programa de Mestrado Profissional da Escola de Economia de São Paulo da Fundação Getulio Vargas, como requisito para a obtenção do título de Mestre em Economia.

Área de concentração:
Engenharia Financeira.

Orientador:
Prof. Dr. João Luiz Chela

SÃO PAULO
2020

Cordeiro, Tiago Vilas Boas.
Predição de default de empresas : técnicas de machine learning em dados desbalanceados / Tiago Vilas Boas Cordeiro. - 2020.
65 f.

Orientador: João Luiz Chela.
Dissertação (mestrado profissional MPFE) – Fundação Getulio Vargas, Escola de Economia de São Paulo.

1. Aprendizado do computador. 2. Análise de Regressão Logística. 3. Avaliação de risco. I. Chela, João Luiz. II. Dissertação (mestrado profissional MPFE) – Escola de Economia de São Paulo. III. Fundação Getulio Vargas. IV. Título.

CDU 330.115

Ficha Catalográfica elaborada por: Raphael Figueiredo Xavier CRB SP-009987/O

Biblioteca Karl A. Boedecker da Fundação Getulio Vargas - SP

TIAGO VILAS BOAS CORDEIRO

PREDIÇÃO DE DEFAULT DE EMPRESAS: TÉCNICAS DE MACHINE LEARNING EM DADOS DESBALANCEADOS

Dissertação apresentada ao Programa de Mestrado Profissional da Escola de Economia de São Paulo da Fundação Getulio Vargas, como requisito para a obtenção do título de Mestre em Economia.

Área de concentração:
Engenharia Financeira.

Data da Aprovação: 11/11/2020

Banca Examinadora:

Prof. Dr. João Luiz Chela
(Orientador)
EESP-FGV

Prof. Dr. Oswaldo Luiz do Valle Costa
POLI-USP

Profa. Dra. Élia Yathie Matsumoto
EESP-FGV

Este trabalho é dedicado aos meus pais, Miriam e Devanir, por serem um exemplo de dedicação e trabalho e por terem me suportado até aqui, aos meus irmãos, Tamires e Gustavo pela parceria e apoio de sempre, e a minha namorada e futura esposa Bianca.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. João Luiz Chela por se dispor em me orientar ao longo desses meses.

Agradeço aos meus colegas de trabalho e a toda minha equipe, em especial ao Julio Paixão, pela possibilidade de crescimento que me foi proporcionado e ao Enrico Chiavegato, por ser compreensivo e paciente durante esses dois anos.

RESUMO

Dada a importância do gerenciamento do risco de crédito para o setor bancário, modelos de probabilidade de *default* tornaram-se fundamentais. Neste contexto, com o avanço do volume de informações dos clientes e a capacidade computacional, diversas técnicas têm sido estudadas e aplicadas. Neste estudo, utilizamos duas técnicas lineares tradicionais, a Análise Discriminante Linear e a Regressão Logística, e quatro técnicas não-lineares *ensembles*, *Bagging*, *Random Forest*, *Adaboost* e *Stacking*, aplicadas em um problema de predição de *default* de empresas brasileiras utilizando informações de seus demonstrativos financeiros. Os resultados indicam que as transformações nos dados e tratamento de desbalanceamento de classes tem forte impacto no poder preditivo da Regressão Logística. Ainda, o *Random Forest* foi a técnica com melhor desempenho, independente do cenário e da métrica utilizada.

Palavras-chave: Machine Learning. Regressão Logística. Random Forest. Probabilidade de default. Modelos de rating.

ABSTRACT

Given the importance of credit risk management for the banking sector, probability of default models have become fundamental. In this context, with the advances in the volume of information from customers and the computational capacity, several techniques have been studied and applied. In this study, we used two traditional linear techniques, the Linear Discriminant Analysis and Logistic Regression, and four non-linear ensemble techniques, Bagging, Random Forest, Adaboost and Stacking, applied to a problem of probability of default on brazilian companies, using information from their financial statements. The results indicate that the transformations in the data and treatment of class imbalanced have a strong impact on the predictive power of Logistic Regression. Yet, Random Forest was the technique with the best performance regardless of the scenario and the metric used.

Keywords: Machine Learning. Logistic Regression. Random Forest. Probability of default. Risk rating models.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 Introdução

O gerenciamento do risco de crédito e sua correta mensuração é de extrema importância para o setor bancário. Historicamente, observamos que até mesmo bons clientes, com ótimos históricos de pagamento, podem em algum momento inadimplir com suas obrigações financeiras. Modelos de probabilidade de inadimplência, ou probabilidade de *default* (PD), são um dos pilares no gerenciamento do risco de crédito em empréstimos para pessoas físicas e empresas.

Neste contexto, a construção de modelos de probabilidade de *default* de empresas têm sido um tema bastante relevante para o mercado financeiro, dado que a ocorrência do *default* pode resultar em perdas econômicas não só para os bancos, mas para as diversas camadas da sociedade.

Podemos destacar que os principais dificultadores neste tipo de problema são os baixos volumes de *defaults* observados em empréstimos realizados às empresas, o que ocasiona em um alto grau de desbalanceamento entre as classes *default* e *não-default*, além de características intrínsecas de cada empresa e a dificuldade em conseguir dados recentes e confiáveis para esse tipo de estudo.

Nas últimas décadas, devido ao grande avanço computacional, aplicações de técnicas mais sofisticadas de *machine learning* passaram a ser mais estudadas e aplicadas neste tipo de problema. Grande parte destes estudos estão concentrados no uso de fatores específicos das empresas, baseados em seus demonstrativos financeiros, e focados em comparar diferentes técnicas. Destaca-se nesse contexto técnicas de classificação linear, como a análise discriminante e a regressão logística, e mais recentemente métodos não-lineares, como redes neurais artificiais, árvores de classificação e métodos *ensembles*.

Dentro deste contexto, o principal objetivo deste trabalho é o desenvolvimento de modelos, através de técnicas de *machine learning*, para predizer um *score* (ou probabilidade) de empresas entrarem em *default* em até 12 meses após a tomada de um empréstimo, e então analisar os resultados e comparar o desempenho entre as técnicas selecionadas e suas variações. Na classificação de problemas de *machine learning*, este é um problema supervisionado de classificação com dados desbalanceados.

Serão utilizadas seis técnicas de classificação, sendo duas delas modelos lineares (Regressão Logística (LR) e Análise Discriminante Linear (LDA)), e quatro técnicas *Ensembles* (*Bagging* (BAG), *Random Forest* (RF), *Adaboost* (ADA) e *Stacking* (STK)). Para construção dos modelos, serão utilizadas variáveis construídas através dos demonstrativos financeiros anuais das empresas (Ativo, Passivo e DRE), pois além de uma boa confiabilidade, os demonstrativos possuem informações que refletem a saúde financeira das

empresas. As variáveis construídas irão se basear no trabalho de Back, Laitinen e Sere (1996b).

A principal contribuição do estudo está relacionada com a aplicação e comparação de técnicas de *machine learning* em um problema de predição de *default* de empresas brasileiras, através de variáveis dos seus demonstrativos financeiros em dados recentes.

A dissertação está organizada em seis capítulos. Seguindo à introdução, o capítulo dois traz a revisão bibliográfica sobre os principais estudos relacionados à construção de modelos em aplicações de *defaults* de empresas. No capítulo três, a fundamentação teórica é apresentada, aprofundando-se nos conceitos técnicos das metodologias que serão aplicadas e métricas de avaliação utilizadas. No capítulo quatro, a construção da base de dados e das variáveis explicativas são descritas. No capítulo cinco os resultados são expostos e no capítulo seis, as conclusões e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

> Predição de *default* tem sido de grande interesse da comunidade científica e de profissionais do mercado financeiro há quase um século
> (SIGRIST; HIRNSCHALL, 2019).

Dado à relevância do adequado controle do risco de crédito pelos agentes financeiros, muitos estudos foram realizados ao longo dos últimos anos com o objetivo de mensurar a probabilidade de inadimplência associada a uma transação ou um cliente.

Grande parte destes estudos estão concentrados no uso de fatores específicos das empresas, como seus demonstrativos financeiros, e focados em comparar diferentes técnicas. Destaca-se nesse contexto técnicas de classificação linear, como a análise discriminante e a regressão logística [1], e mais recentemente métodos não-lineares, como redes neurais artificiais, árvores de classificação e métodos *ensembles*.

Os primeiros estudos publicados eram focados em índices financeiros, comparando empresas *default* com não-*default*. As principais contribuições destes estudos estavam na identificação dos índices mais significantes (Fitzpatrick (1932), Winakor e Smith (1935), Merwin (1942)).

Dois estudos pioneiros que merecem destaque são os de Beaver e Altman. Beaver (1966) apresentou evidências empíricas utilizando análise univariada, através de uma amostra balanceada de empresas *default* / não-*default*, que determinados índices financeiros são mais importantes na predição do *default*. Altman (1968) combinou um conjunto de índices financeiros usando análise discriminante multivariada, criando um escore de medição de risco de crédito popularmente conhecido como Z *score*. O Z *score* foi um importante passo, porém, limitado à pequenas empresas do setor industrial.

Quase uma década depois, Altman, Haldeman e Narayanan (1977) criaram um novo modelo, o Zeta *score*. Esse novo modelo foi uma evolução do modelo publicado anteriormente por Altman (1968), incrementando novas variáveis nos testes, incluindo empresas de maior porte e outros setores na base de dados. Este modelo foi por muitos anos um dos modelos mais proeminentes para o cálculo do risco de crédito de empresas (FIGINI; FANTAZZINI, 2009).

Um dos primeiros trabalhos publicados utilizando regressão logística como técnica foi o de Ohlson (1980). Além da diferença na técnica, um outro diferencial quando comparado ao estudo de Altman, Haldeman e Narayanan (1977), foi o aumento na

[1] Covariáveis entram no modelo através de uma combinação linear.

quantidade de empresas na base de dados. Foram utilizados dados de empresas americanas no período de 1970 a 1976, aumentando consideravelmente a quantidade de *defaults*.

A regressão logística foi, e continua sendo, uma das principais técnicas utilizadas em problemas de classificação no mercado de crédito. Isso se dá em grande parte, pela facilidade na implementação e interpretabilidade do modelo final. Porém, desde o início dos anos 1990, estudos com novas técnicas começaram a ser cada vez mais publicados, indicando um caminho promissor para técnicas não lineares como as redes neurais artificiais e métodos *ensembles*.

Tam e Kiang (1992) compararam o desempenho das técnicas regressão logística, árvore de decisão, KNN e redes neurais artificiais em um problema de predição de *default* de bancos do Texas (EUA) nos anos de 1985 à 1987. As redes neurais artificiais foram superiores no poder preditivo em relação às demais.

Wilson e Sharda (1994) mostraram que uma rede neural artificial construída utilizando o algoritmo *back propagation* apresentou um poder de discriminação maior quando comparada à análise discriminante [2] em um problema de predição de *default* de empresas, baseado em uma amostra de 129 empresas (sendo 65 empresas que entraram em *default*).

Back, Laitinen e Sere (1996a) avançaram no estudo comparativo de técnicas, construindo uma rede neural combinada com algoritmos genéticos para seleção de variáveis, chegando à um nível de acurácia consideravelmente maior que as demais técnicas testadas.

Uma outra classe de estudos envolvendo a modelagem de *defaults* de empresas, foca em algo bastante característico desse tipo de problema: classes (*default / não-default)* desbalanceadas e portfolios com baixo número de *defaults* (conhecidos como *Low default portfolios - LDP*). Isso ocorre ainda mais quando as empresas na base de modelagem são de grande porte [3], pois em geral se traduz em uma taxa de *default* bastante baixa. Uma das referências nesse tipo de problema, é o trabalho de Pluto e Tasche (2005). Para estimativa da probabilidade de inadimplência, os autores utilizaram o princípio da estimativa mais prudente, baseada em intervalos de confiança entre os *ratings*, dado um grau de conservadorismo.

Muchlinski et al. (2016) explorou o problema de dados desbalanceados em um problema de classificação de guerras civis em países. O evento de interesse a ser modelado era observado em cerca de 1% da base de dados. Para tanto, foram ajustados três tipos de regressão logística (regressão logística clássica, regressão logística ajustada a eventos raros (proposto por Firth (1993)) e regressão logística com regularização L1) e uma Floresta aleatória. Os resultados encontrados mostraram que a Floresta aleatória superava

[2] Foram utilizados os mesmos índices financeiros que Altman (1968) utilizou no modelo Z *score*.

[3] Comumente chamadas no mercado financeiro de *large corporates*. No Brasil, em geral, são empresas com mais de R$500 milhões de faturamento anual bruto.

a regressão logística mesmo com os ajustes e regularizações feitas.

Lessmann et al. (2015) publicou um estudo comparativo de 41 técnicas de classificação aplicados ao problema de *credit score*, avaliados através de 6 métricas de performance. Os resultados mostraram que muitos métodos foram superiores à tradicional regressão logística, destacando os métodos *ensembles*.

Alguns estudos investigaram a importância do cenário macroeconômico na construção de modelos de predição de *default*. Antunes, Ribeiro e Antao (2005) levantaram dados de 1995 à 2004 relativos à empresas cadastradas no sistema de informação de crédito português (Central de Responsabilidades de Crédito) e estimaram um modelo para prever a inadimplência utilizando variáveis relacionadas às operações (como um indicador se outras operações da empresa já entraram em *default*), às empresas (como o tamanho da dívida, o setor de atuação, etc.) e ao cenário macroeconômico (como taxa de juros, desemprego, etc.). As variáveis macroeconômicas taxa de crescimento do PIB e taxa de juros de curto prazo se apresentaram como significativas na modelagem.

Carling et al. (2007) desenvolveram um modelo discreto de previsão do tempo até o *default*. Para isso, utilizaram dados de empresas no período de 1994 a 2000 de um banco sueco. Foram utilizadas informações específicas das empresas, além de variáveis macroeconômicas. A curva de juros, o hiato do produto e a expectativa das famílias em relação à economia foram as variáveis macroeconômicas mais significantes.

Atualmente, existem diversas técnicas de *machine learning* disponíveis e muitas delas são facilmente implementadas através de pacotes em *softwares* estatísticos ou em linguagens de programação como o Python. No entanto, no presente trabalho optamos por focar nos métodos *ensembles* basicamente por dois motivos. Primeiro, são métodos bastante robustos quanto à sensibilidade à *outliers* e, em geral, mostram performances superiores aos métodos lineares clássicos em diversas aplicações. Além disso, são métodos que podem facilmente ser adaptados para amostras desbalanceadas.

Esta dissertação apresenta um estudo comparativo da utilização de métodos lineares e *ensembles* no problema de classificação de *default* de empresas, utilizando índices financeiros.

Os conceitos teóricos e premissas utilizadas nas construções dos modelos são abordados no próximo capítulo.

3 Fundamentação Teórica

Neste capítulo serão descritas as técnicas e conceitos utilizados no processo de modelagem, fornecendo suporte à metodologia proposta neste estudo.

3.1 A importância dos modelos de *default* com probabilidades calibradas

O gerenciamento do risco de crédito é de extrema importância para o setor bancário, sendo necessário uma metodologia adequada para mensuração desse risco. Historicamente, observamos que até mesmo bons clientes podem em algum momento inadimplir com suas obrigações financeiras.

Modelos de probabilidade de *default* (PD) de empresas, podem ser construídos para finalidades específicas, como na avaliação de concessão de crédito ou para cálculo de capital regulatório. Uma característica importante deste tipo de modelo é a necessidade de uma probabilidade associada ao evento de *default*. Independente da finalidade, é bastante importante que esse tipo de modelo produza probabilidades calibradas à real probabilidade de observação dos *defaults*, mapeada em uma escala de *ratings*.

Segundo Wallace e Dahabreh (2012), muitas técnicas de *machine learning* em problemas supervisionados não produzem probabilidades calibradas para dados desbalanceados, mesmo com a adoção de técnicas para tratamento do desbalanceamento dos dados como *oversampling* e *undersampling*. Uma probabilidade não está calibrada quando a mesma superestima ou subestima a real probabilidade de o evento ocorrer. Dado essa necessidade, precisamos selecionar modelos em que seja possível fazer algum ajuste no sentido de que produza probabilidades calibradas, para posteriormente conseguirmos comparar suas performances.

De acordo com Brownlee (2020), modelos que possuem uma base probabilística em sua estimação (como os estimadores de máxima verossimilhança), a exemplo da Regressão Logística (LR) e a Análise discriminante linear (LDA), já produzem probabilidades calibradas desde que o problema dos dados desbalanceados tenha sido tratada.

Neste trabalho, os modelos lineares selecionados (LDA e LR) não precisaram de ajuste para calibração, dado que o problema do desbalanceamento foi tratado como veremos nas próximas seções. Já os métodos não lineares selecionados (BAG, RF, ADA e STK) foram calibrados para utilização como *proxy* de probabilidade. Tal calibração foi feita utilizando a biblioteca `scikit-learn` do Python, através da classe `CalibratedClassifierCV`.

3.2 Preparação de dados

Em problemas de *machine learning*, existem algumas semelhanças de modo que podemos definir uma sequência de etapas que devem ser executadas. Esse processo é comumente conhecido como *data science process*, *machine learning process* ou até mesmo *knowledge discovery in databases - KDD*, e pode ser resumido em quatro etapas: definição do problema, preparação dos dados, treinamento e avaliação de modelos e aplicação. A preparação de dados pode ser uma das etapas mais difíceis em problemas de *machine learning*. Grande parte disso, se deve ao fato de que cada conjunto de dados é diferente e altamente específico ao problema estudado. Nesta seção, detalhamos os principais aspectos teóricos que foram utilizados na preparação dos dados neste trabalho.

3.2.1 Dados ausentes

A identificação e tratamento de dados ausentes, ou *missing values*, é um importante passo na preparação de dados. Isso se deve, em grande parte, ao fato de que muitas das técnicas de *machine learning* não trabalham com dados ausentes. Identificar as razões e os tipos de dados ausentes é determinante para identificar os impactos ao classificador e aplicar abordagens que melhorem a performance do mesmo. Segundo Schafer e Graham (2002), dados ausentes podem ser classificados em três grupos:

- *Dados ausentes completamente aleatórios - MCAR:* Os dados ausentes não possuem relação direta com os dados existentes e possuem origem aleatória.
- *Dados ausentes aleatórios - MAR:* Os dados ausentes possuem alguma relação com algum dado observado do indivíduo.
- *Dados ausentes não aleatórios - MNAR:* Os dados ausentes possuem relação direta com o valores que deveriam ser observados.

Após esta identificação, deve-se selecionar qual estratégia será aplicada, todas elas possuindo alguns prós e contras. Dependendo da natureza do dado ausente, uma estratégia pode ser mais adequada que as demais. Algumas estratégias que podem ser utilizadas são:

- *Remoção das variáveis (ou indivíduos) que possuem dados ausentes:* Deve ser utilizado com cautela, pois mesmo variáveis com alto percentual de dados ausentes podem conter informações relevantes para predição da variável resposta. E no caso da exclusão de indivíduos, pode diminuir significantemente o tamanho da base de dados e afetar o desempenho geral dos modelos.

- *Aplicação de estatísticas da variável para substituir o dado ausente:* Bastante comum e de fácil aplicação. Porém, dependendo da quantidade de dados ausentes, corre-se o risco de uma super adaptação na estimação dos modelos.
- *Substituição dos dados ausentes por uma constante (Ex:0):* Substitui os valores ausentes por um valor que seja padrão para a variável.
- *Aplicação de modelos preditivos para preencher os valores ausentes:* Podemos treinar modelos preditivos para estimar e preencher os dados ausentes. Neste caso, as demais variáveis são utilizadas como variáveis independentes.

Em todos os casos é importante analisar bem a origem dos dados ausentes e sua importância para a característica dos indivíduos, de modo a selecionar a melhor técnica para tratamento, visando não prejudicar a eficácia dos modelos.

3.2.2 Dados discrepantes

Dados discrepantes, ou *outliers*, são valores extremos que estão muito longe das outras observações de uma determinada variável. Por exemplo, se a variável possui uma distribuição normal, os *outliers* podem ser valores na cauda da distribuição.

Muitos algoritmos de *machine learning* são sensíveis ao intervalo e distribuição de valores das variáveis independentes. Valores discrepantes nestas variáveis podem distorcer o processo de treinamento dos modelos, resultando em muito tempo de treinamento, modelos menos precisos e potencialmente com baixo poder de discriminação (em problemas de classificação).

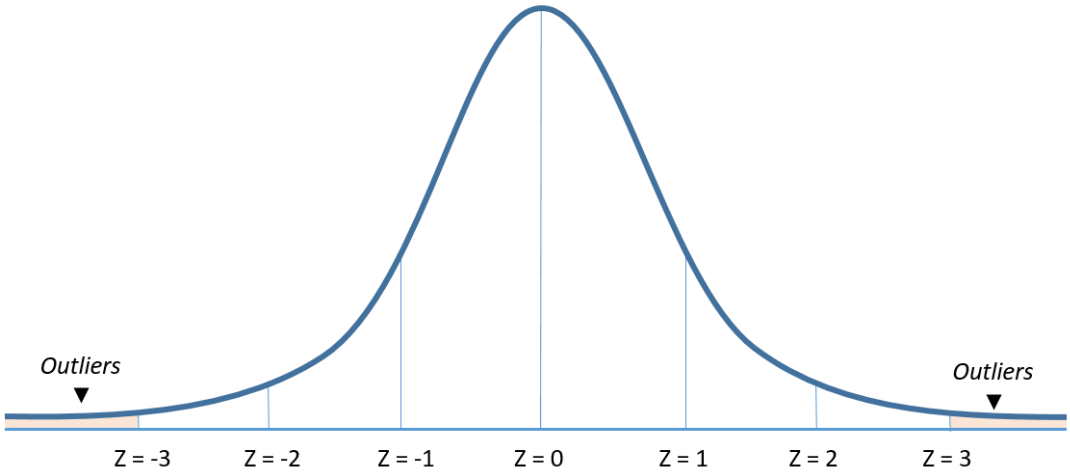


Figura 1 – *Outliers* em uma distribuição Normal

Existem diversas alternativas para identificação e classificação de *outliers* consolidados na literatura. Segundo Aggarwal (2017), as principais técnicas utilizadas são:

- *Análise de valores extremos:* baseia-se na determinação das caudas estatísticas da distribuição subjacente de cada variável. Por exemplo, métodos estatísticos como o *escore-z* em dados univariados.
- *Modelos probabilísticos e estatísticos:* baseia-se na determinação dos *outliers* com modelos probabilísticos ajustados aos dados. Por exemplo, modelos de misturas gaussianas (*Gaussian Mixture Models (GMMs)*).
- *Modelos lineares:* baseia-se em métodos de projeção que ajustam os dados em dimensões inferiores usando correlações lineares. Por exemplo, utilização da a análise de componentes principais, ou *Principal Component Analysis (PCA)*, para essa finalidade.
- *Modelos baseados em proximidade:* baseia-se na identificação de valores que estão isolados da massa de dados determinada por *clusters*, ou determinação do vizinho mais próximo *(K-Nearest Neighbor (KNN))*.
- *Modelos baseados na teoria da informação:* Os *outliers* são identificados como instâncias de dados que aumentam a complexidade do conjunto de dados.

Uma outra forma de lidar com os *outliers*, seria utilizar técnicas de modelagem que são robustas à tais discrepâncias, como o *random forest*. Neste trabalho, como trabalhamos com diversas técnicas, o tratamento dos *outliers* é um passo importante.

3.2.3 Transformações de dados

No início do século XX, o aparecimento dos testes de significância revolucionou a teoria e a prática estatística. Porém, eles se apoiavam na suposição de que os dados eram provenientes de uma amostra aleatória de uma população hipotética com distribuição normal (PINO, 2014). Posteriormente, uma grande quantidade de estudos mostraram os erros que poderiam ocorrer com a aplicação desses testes em casos de não normalidade dos dados.

Grande parte dos modelos probabilísticos assumem que as variáveis independentes utilizadas no processo de estimação, são provenientes de uma distribuição normal. Como trabalhamos com a estimação de alguns modelos probabilísticos neste trabalho, vamos tratar a questão da não normalidade das variáveis independentes através do processo de transformação dos dados, e observar o impacto que tais transformações ocasionam na performance dos modelos.

3.2.3.1 Transformação Box-Cox

Uma das famílias de transformações mais conhecidas e utilizadas é transformação de potência, proposta por Box e Cox (1964), e mais conhecida como *transformação de*

Box-Cox. Muitas outras transformações comuns, como a logarítmica e a raiz quadrada, são apenas casos particulares da *transformação de Box-Cox*. A forma genérica da transformação é dada por:

$$y^{(\lambda)} = \begin{cases} \frac{(y+c)^{\lambda}}{\lambda}, & \text{para } \lambda = 0 \\ \log(y+c), & \text{para } \lambda = 0 \text{ e } y > -c \end{cases} \tag{3.1}$$

onde y representa a observação original, $y(\lambda)$ representa a observação transformada, λ e c são parâmetros desconhecidos e log representa o logaritmo natural.

Supõe-se que para algum valor de λ e algum valor de c as observações transformadas sejam independentes e normalmente distribuídas com variância constante σ^2 e esperança $a\theta$, onde a é uma matriz conhecida de posto completo e θ é um vetor de parâmetros desconhecidos. Os parâmetros podem ser estimados aplicando a teoria de máxima verossimilhança para grandes amostras ou aplicando a teoria de Bayes.

Como a transformação de Box-Cox é definida somente para variáveis positivas, a transformação foi estendida para valores negativos por Yeo e Johnson (2000):

$$y^{(\lambda)} = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda}, para \lambda = 0, y \geq 0 \\ \log(y+1), para \lambda = 0, y \geq 0 \\ -\frac{(1-y)^{2-\lambda}-1}{2-\lambda}, para \lambda = 2, y < 0 \\ -\log(1-y), para \lambda = 2, y < 0 \end{cases} \tag{3.2}$$

Para aplicação das *transformações de dados* neste trabalho, utilizamos a extensão para valores negativos de Yeo-Johnson através da biblioteca `scikit-learn` do Python, e da classe `PowerTransformer` com parâmetro de seleção do método `method='yeo-johnson'`.

3.2.4 Dados desbalanceados

Uma base de dados é dita desbalanceada, quando a variável resposta a ser modelada possui muito mais casos em determinada classe que nas demais. Por exemplo, em uma base de dados há 10.000 empresas e apenas 100 delas (1%) migrou para o estado de default nos próximos 12 meses. Esse tipo de problema compromete tanto a performance como a calibração das probabilidades na maioria dos modelos de predição.

Existem três estratégias bastante consolidadas na literatura para tratamento de bases desbalanceadas: o *undersampling*, o *oversampling* e a combinação entre elas (*over + undersampling*).

- *Undersampling:* Consiste em fazer uma subamostragem na classe prevalente, de modo que os dados a serem modelados estejam equilibrados entre as classes. É mais

adequada quando o conjunto de dados é bastante grade, pois assume-se que existem registros redundantes da classe majoritária. Não é indicado para bases de dados pequenas pois ao realizar o *undersampling* pode-se perder informações relevantes para o ajuste dos modelos. Existem diversos algoritmos para realização do *undersampling*, mas destacamos aqui o ENN (*Edited Nearest Neighbors*). A ideia por trás do ENN é bastante simples: o algoritmo compara três vizinhos mais próximos e se um deles for de classe diferente dos demais, o exemplo é removido da base de treinamento.

- *Oversampling*: Consiste em fazer uma superamostragem da classe minoritária através de variações de algoritmos baseados em similaridade e *bootstraping* (reamostragem). Uma das principais variações é atingida através do algoritmo SMOTE (*Synthetic Minority Oversampling Technique*). O SMOTE encontra um registro que seja semelhante ao registro que está sendo *upsampled* e cria um registro sintético que é uma média aleatoriamente ponderada do registro original e do registro vizinho, em que o peso é gerado separadamente para cada preditora. (BRUCE; BRUCE, 2019)
- *Over + undersampling*: Consiste na combinação das duas estratégias.

Batista, Prati e Monard (2004) exploraram em seu estudo a aplicação de diversas técnicas de *undersampling*, *oversampling* e *over + undersampling* em problemas práticos de dados desbalanceados. Uma das conclusões do estudo foi que a combinação *Over + undersampling* (SMOTE + ENN) apresentou resultados bastante positivos para bases de dados com poucos casos na classe minoritária.

Neste trabalho, vamos utilizar a combinação *Over + undersampling* através das técnicas SMOTE + ENN, baseado nas evidências encontradas na literatura acerca dos bons resultados apresentados.

3.3 Modelos lineares

3.3.1 Regressão Logística (LR)

Para estimação de eventos binários, a Regressão Logística é uma das técnicas mais utilizadas. Segundo Bruce e Bruce (2019), sua popularidade se deve ao fato da sua grande velocidade computacional para estimação dos parâmetros e sua facilidade de implementação para pontuação de novos dados.

Dois conceitos importantes para a Regressão Logística são a *função de resposta logística* e o *logito*. Vamos pensar que pretendemos estimar uma variável resultante com

rótulo binário (valor igual à 1 caso o evento ocorra, e 0 caso contrário). Seja *p* a probabilidade de sua ocorrência, podemos tentar modelar *p* como uma função linear de suas preditoras:

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q \tag{3.3}$$

Porém, ajustar o modelo 3.3. não garante que *p* ficará entre 0 e 1. Para tal, modelamos *p* aplicando uma *função de resposta logística* nas preditoras, garantindo que *p* fique entre 0 e 1:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q)}} \tag{3.4}$$

Para removermos a expressão exponencial do denominador da função 3.4, consideramos a razão de probabilidades (*odds*) que são a probabilidade que o evento ocorra dividida pela probabilidade de que o evento não ocorra, em vez de probabilidades. A *odds* pode ser expressa por:

$$Odds(Y = 1) = \frac{p}{1 - p} \tag{3.5}$$

Podemos isolar *p* a partir da *odds*, e obter:

$$p = \frac{Odds(Y = 1)}{1 + Odds(Y = 1)} \tag{3.6}$$

Combinando 3.4 com 3.6, obtemos:

$$Odds(Y = 1) = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q} \tag{3.7}$$

Finalmente, aplicando logaritmo dos dois lados da expressão 3.7, chegamos em uma expressão linear dos preditores:

$$log(Odds(Y = 1)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q \tag{3.8}$$

A função 3.8, também conhecida como *logito*, é bastante importante na regressão logística pois mapeia a probabilidade *p* de (0,1) para qualquer valor $(-\infty, +\infty)$.

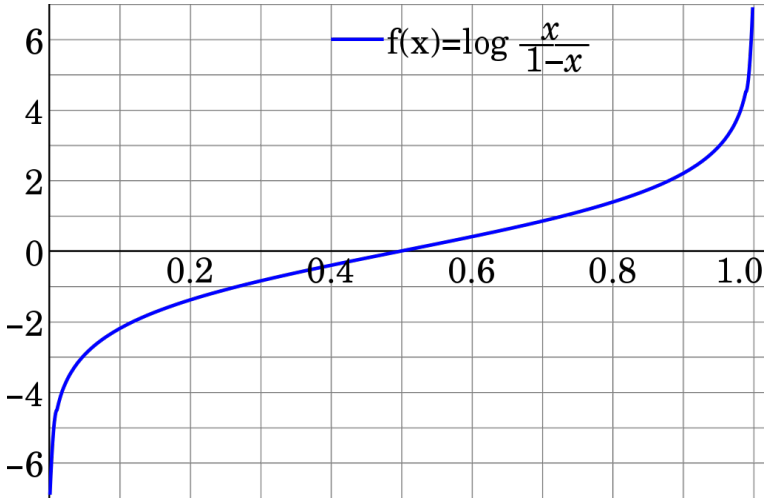


Figura 2 – Função que mapeia uma probabilidade para uma escala adequada para um modelo linear (logito)

Fonte: Baseado em Bruce e Bruce (2019)

Para aplicação da LR neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `LogisticRegression`.

3.3.2 Análise Discriminante Linear (LDA)

A análise discriminante é certamente uma das técnicas de classificação mais antigas difundidas na literatura. A ideia básica por trás da análise discriminante é a classificação de indivíduos em grupos.

Considere n grupos denotados por $\Pi_1, \ldots, \Pi_n$, $n \geq 2$, e para cada grupo Π_j, $j = 1, \ldots, n$, existe uma função densidade de probabilidade $f_j(x)$ em $\mathbb{R}^p$. O objetivo da análise discriminante é alocar um indivíduo em um dos n grupos com base na medida de suas características x (ou nos dizer a qual grupo é mais provável de pertencer).

Uma regra discriminante d corresponde a uma divisão do $\mathbb{R}^p$ em regiões disjuntas $R_1, \ldots, R_n$, ($\bigcup R_j = \mathbb{R}^p$). A regra d é definida por:

$$\text{alocar } x \text{ em } \Pi_j \text{ se } x \quad R_j, \tag{3.9}$$

para $j = 1, \ldots, n$. Existem variações da análise discriminante para casos em que as distribuições são conhecidas, parcialmente conhecidas ou desconhecidas. Dentre os casos em que as distribuições são desconhecidas, a abordagem mais conhecida é a de Fisher (1936), que posteriormente ficou conhecida como Análise Discriminante Linear (LDA).

A ideia apresentada por Fisher é encontrar uma função linear $a'x$ que maximize a razão entre a soma dos quadrados entre os grupos sobre a soma dos quadrados dentro dos

grupos. Seja

$$z = Xa = \begin{bmatrix} X_1 a \\ X_2 a \\ \vdots \\ X_g a \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_g \end{bmatrix}$$

uma combinação linear de X. Então a soma de quadrados totais de z é dada por

$$z\,V z = a\,X\,V\,X a = a\,T a$$

que pode ser particionada como a soma dos quadrados dentro dos grupos

$$\sum_i z_i V_i z_i = \sum_i a\,X_i V_i X_i a = a\,S a$$

mais a soma dos quadrados entre grupos

$$\sum_i n_i\,(\bar{z}_i - \bar{z})^2 = \sum_i n_i\,\{a\,(\bar{x}_i - \bar{x})\}^2 = a\,B a$$

onde $\bar{z}_i$ é a média do i-ésimo sub-vetor de z e V_i (n_i x n_i) é a matriz centro. A razão entre a soma dos quadrados dentro e entre os grupos é então dada por:

$$\frac{a\,B a}{a\,S a} \tag{3.10}$$

Se a maximiza a expressão 3.10, então a função linear $a\,x$ é dita *função discriminante linear de Fisher*. Dado um conjunto de dados, uma vez que a função discriminante linear de Fisher foi calculada, qualquer nova observação pode ser alocada para uma das populações de acordo com seu "escore discriminante".

Para aplicação da LDA neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `LinearDiscriminantAnalysis`.

3.4 Modelos não lineares

Para seleção das metodologias não lineares, nos baseamos no trabalho de Lessmann et al. (2015). Em seu trabalho, foram comparados diversos métodos de classificação em aplicações de *credit scoring*. Uma das conclusões do trabalho foi que dentre os diversos classificadores testados, aqueles baseados em metodologias *ensembles* haviam se destacado, em especial os *ensembles heterogêneos*. A ideia por trás de uma metodologia *ensemble* em um problema de classificação é bastante simples: combinar diversos classificadores bons com o objetivo de gerar um classificador final ainda melhor. Grande parte das soluções vencedoras em competições de *machine learning* envolvem vários métodos *ensembles* (GERON, 2019).

Existem algumas formas de classificarmos os métodos *ensembles*. Primeiramente, podemos pensar se iremos utilizar apenas uma única metodologia para gerar os classificadores individuais (*ensembles homogêneos*) ou diversas metodologias (*ensembles heterogêneos*). O *Random Forest (RF)*, apresentado na seção 3.4.2, é um exemplo de *ensemble homogêneo*, pois utiliza apenas as Árvores de decisão como método de geração dos classificadores individuais.

Uma outra forma de pensarmos sobre os métodos *ensembles*, consiste em como serão combinadas as classificações individuais para uma classificação final. Comumente, utiliza-se a classificação por votos majoritários, também chamado de *hard voting*. O *hard voting* basicamente computa as quantidades de votos dos classificadores individuais, e a saída do *ensemble* será a classe que teve a maioria dos votos. A classificação por votos majoritários geralmente consegue uma capacidade preditiva maior que o melhor classificador individual componente do *ensemble*.(GERON, 2019). Podemos ainda, ao invés utilizar funções triviais para classificação final do *ensemble*, como o *hard voting*, treinarmos um modelo para essa tarefa. Essa metodologia é conhecida como *Stacking*, e é formalmente apresentada na seção 3.4.4.

3.4.1 *Bagging* (BAG)

O *bagging (bootstrap aggregating)*, apresentado por Breiman (1996), é um dos algoritmos de *ensembles* mais famosos e utilizados. Basicamente, o método treina cada classificador individual em uma reamostra *bootstrap*.

Suponha que temos uma variável resposta Y e P variáveis preditoras: $X_1, X_2, ..., X_p$, com N registros. O algoritmo *bagging* é formalmente apresentado abaixo:

1. Inicialize M, o número de modelos a serem treinados, e n, o número de registros a serem escolhidos ($n < N$), sendo N o número total de indivíduos. Ajuste a primeira iteração como $m = 1$.
2. Tire uma reamostra *bootstrap* (com reposição) de n registros dos dados de treinamento para formar uma subamostra Y_m e X_m.
3. Treine um modelo usando Y_m e X_m para criar um conjunto de regras de decisão $\hat{f}_m(X)$.
4. Incremente o contador do modelo $m = m + 1$. Se $m <= M$, volte ao Passo 1.

A estimativa *bagged*, para caso em que $\hat{f}_m$ prevê a probabilidade $Y = 1$, será dada por:

$$\hat{f} = \frac{1}{M}\left[\hat{f}_1(X) + \hat{f}_2(X) + \ldots \hat{f}_M(X)\right] \tag{3.11}$$

Para aplicação do BAG neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `BaggingClassifier`.

3.4.2 *Random Forest* (RF)

O *Random Forest* consiste na aplicação do *bagging* utilizando árvores de decisão como classificadores individuais, porém com uma importante extensão: além de reamostrar os registros, existe também a reamostragem das variáveis o que resulta em uma grande diversidade no processo.

O algoritmo *random forest* é formalmente apresentado abaixo:

1. Tire uma subamostra A através de *bootstrap* com reposição dos indivíduos.
2. Para a primeira divisão, amostre $p < P$ variáveis aleatoriamente sem reposição.
3. Para cada uma das variáveis amostradas $X_{j(1)}, X_{j(2)}, ..., X_{j(p)}$, aplique o algoritmo de divisão:
    a) Para cada valor $s_{j(k)}$ de $X_{j(k)}$:
        i. Divida os registros da subamostra A com $X_{j(k)} < s_{j(k)}$ como uma repartição, e os registros restantes, $X_{j(k)} >= s_{j(k)}$, como outra repartição.
        ii. Meça a homogeneidade de classes dentro de cada repartição de A definida no passo anterior.
    b) Selecione o valor de $s_{j(k)}$ que produza a máxima homogeneidade de classes dentro da subamostra.
4. Selecione a variável $X_{j(k)}$ e o valor de divisão $s_{j(k)}$ que produzam a máxima homogeneidade de classes dentro da repartição.
5. Prossiga à próxima divisão e repita os passos anteriores, começando com o Passo 2.
6. Continue com divisões adicionais seguindo o mesmo procedimento até que a árvore tenha crescido.
7. Volte ao Passo 1, tire outra subamostra *bootstrap*, e recomece todo o processo.

Para aplicação do RF neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `RandomForestClassifier`.

3.4.3 *Boosting* (ADA)

O *Boosting* se refere à qualquer método *ensemble* que combina vários previsores fracos em um forte, treinados de forma sequencial e sempre buscando ajustar aos erros do previsor treinado anteriormente. Dos diversos métodos *boosting* existentes, neste trabalho vamos focar no *AdaBoost*. O algoritmo *AdaBoost* é formalmente apresentado abaixo.

1. Inicialize M, que será o número máximo de modelos a serem ajustados. Ajuste o contador de iterações $m = 1$ e inicialize os pesos de observação $w_i = 1/N$, para $i = 1, 2, ..., N$. Inicialize o modelo de agrupamento $\hat{F}_0 = 0$.
2. Treine um modelo usando $\hat{f}_m$ utilizando os pesos de observação $w_1, w_2, ..., w_N$, que minimiza o erro ponderado e_m definido pela soma dos pesos para as observações classificadas erroneamente.
3. Adicione o modelo ao grupo: $\hat{F}_m = \hat{F}_{m-1} + \alpha_m \hat{f}_m$, onde $\alpha_m = \frac{\log 1 - e_m}{e_m}$.
4. Atualize os pesos $w_1, w_2, ..., w_N$, de forma que os pesos sejam aumentados para as observações que eram mal classificadas. O tamanho do aumento depende de α_m observações mal classificadas. O tamanho do aumento depende de α_m, sendo que valores maiores de α_m leva a maiores pesos.
5. Incremente o contador $m = m + 1$. Caso $m <= M$, retorne ao Passo 1.

Aumentado o peso das observações mal classificadas, o algoritmo força os modelos a treinarem mais nos dados em que teve um mal desempenho. O fator α_m garante que os modelos com menor erro tenham maior peso.

A estimativa *boosted* é dada por:

$$\hat{F} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2 + \cdots + \alpha_M \hat{f}_M \tag{3.12}$$

Para aplicação do ADA neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `AdaBoostClassifier`.

3.4.4 *Stacking* (STK)

O *Stacking*, ou *Stacked generalization*, é uma metodologia *ensemble* apresentada por H.Wolpert (1992). Baseia-se na ideia de acrescentar mais complexidade na camada final do *ensemble* em busca de uma melhora no poder preditivo final. Nem sempre essa melhora esperada é atingida, mas em diversos problemas ela é observada.

A arquitetura do *Stacking* envolve dois conceitos importantes, os *base-models*, que em um problema de classificação são os classificadores individuais, e o *meta-model*, que

é o modelo treinado para melhor combinar as previsões dos *base-models*. Um algoritmo comum do *Stacking* em um problema de classificação é apresentado abaixo.

1. Divida o conjunto de treinamento em dois subconjuntos.
2. O primeiro subconjunto é utilizado para treinar N modelos (*base-models*) previamente definidos. Todos os dados do primeiro subconjunto são utilizados no treinamento de cada um dos N modelos, ou seja, não é feita reamostragem como na metodologia *Bagging*.
3. Todos os N modelos treinados no Passo 2 são utilizados para fazer previsões no segundo subconjunto de dados. Portanto, para cada instância do segundo subconjunto, teremos N previsões, o que forma uma população de previsões.
4. As previsões passam a ser utilizadas como características de entrada e um novo modelo (*meta-model*) é treinado, aprendendo a prever o valor do alvo dadas as previsões dos *base-models* da camada anterior.

Para aplicação do *Stacking* neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `StackingClassifier`.

3.5 Métricas de avaliação

Problemas de classificação binários com dados desbalanceados envolvem duas classes: uma com a maioria dos casos e outra com a minoria. Em problemas desse tipo, a classe majoritária geralmente é referida como *classe negativa* (atribuído valor 0) e a classe minoritária como *classe positiva* (atribuído valor 1). Portanto, em nosso estudo temos:

- Casos não-*default*: classe negativa, valor 0
- Casos *default*: classe positiva, valor 1

Existem diversas métricas propostas na literatura para avaliação de modelos de classificação. Porém, como se trata de um problema com dados desbalanceados, alguns cuidados são essenciais. Medidas em que os pesos das duas classes são os mesmos devem ser descartadas. Um bom exemplo, é a tradicional medida de *Acurácia*, definida em 3.13. Em um exemplo hipotético de problema com dados desbalanceados (suponhamos a proporção de 10% da classe positiva, e 90% da classe negativa), um modelo que prever apenas a classe majoritária, vai possuir uma boa Acurácia.

$$Acurácia = \frac{Previsões\ corretas}{Total\ de\ previsões} \tag{3.13}$$

Para definição de quais métricas seriam utilizadas no trabalho, usamos a taxonomia proposta por Ferri, Hernández e Modroiu (2009). Em seu trabalho, as métricas são resumidas em três classes: Métricas de limite, que habitualmente são fáceis de calcular e de entender, porém com uma limitação importante, que é assumir que a distribuição de classes observadas na base de treinamento se manterá ao longo do tempo nas aplicações. Essa premissa é bastante forte quando se trata de um problema de predição de *default*, pois a proporção de *defaults* observados se altera ao longo do tempo. Métricas probabilísticas, que são habitualmente utilizadas em problemas em que a preocupação maior está no grau de confiança das estimativas e não diretamente na classificação binária correta, e por último as Métricas de ranqueamento, que são aquelas mais focadas no quão efetivo o modelo é na separação e ranqueamento das classes. Métricas de ranqueamento também necessitam que o classificador prediga um *score* ou probabilidade.

> Métricas baseadas em raqueamento são importantes para diversas aplicações [...] onde classificadores são utilizados para selecionar os melhores *n* casos, ou quando uma boa separação de classes é crucial. (FERRI; HERNáNDEZ; MODROIU, 2009).

Dado à natureza do problema proposto neste trabalho, selecionamos duas métricas de ranqueamento bastante utilizadas na literatura: A curva ROC (*Receiver Operating Characteristic*) e sua área (ROC AUC) e a curva PR (*Precision-Recall*) e sua área (PR AUC). Estas métricas são baseadas em conceitos da matriz de confusão binária, ilustrada da figura 3.

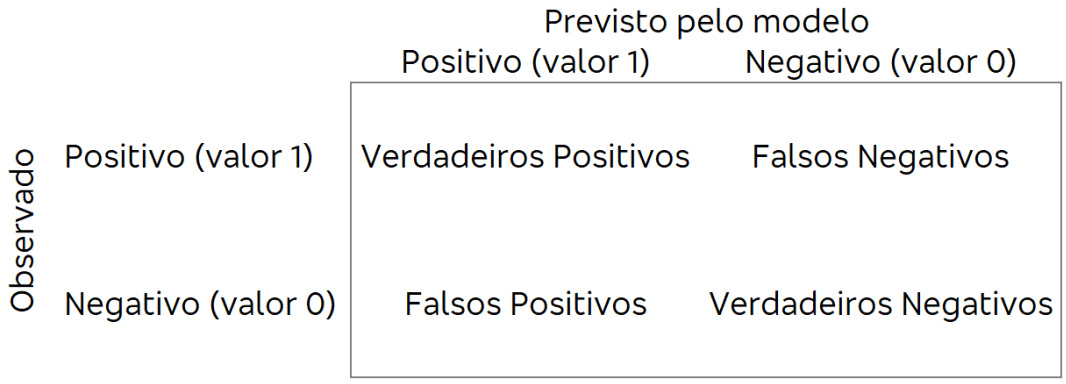


Figura 3 – Matriz de confusão binária

3.5.1 Curva ROC e ROC AUC

A curva ROC é um gráfico que resume o desempenho de um modelo de classificação binário em relação à classe positiva (previsão, pelo modelo, de ocorrência do evento de interesse). O eixo *x* indica a taxa de falsos-positivos (modelo indicou que o evento ocorreria,

mas não ocorreu) e o eixo *y* indica a taxa de verdadeiros-positivos (modelo indicou que o evento ocorreria, e ocorreu).

A taxa de verdadeiros-positivos é a relação calculada entre o número total de verdadeiros-positivos dividido pela soma dos verdadeiros-positivos e falsos-negativos. A taxa de verdadeiros-positivos também é conhecida como *sensibilidade*.

$$VerdadeirosPositivos = \frac{VerdadeirosPositivos}{VerdadeirosPositivos + FalsosNegativos} \tag{3.14}$$

A taxa de falsos-positivos é calculada como a relação entre o número total de falsos-positivos dividido pela soma dos falsos-positivos e verdadeiros-negativos.

$$FalsosPositivos = \frac{FalsosPositivos}{FalsosPositivos + VerdadeirosNegativos} \tag{3.15}$$

Podemos interpretar a curva ROC como a proporção de previsões corretas para a classe positiva (eixo y) versus a proporção de erros da classe negativa (eixo x). Portanto, o melhor modelo possível é aquele que alcança o canto superior esquerdo do gráfico (coordenada 0,1), ou seja, classificaria perfeitamente todos os casos.

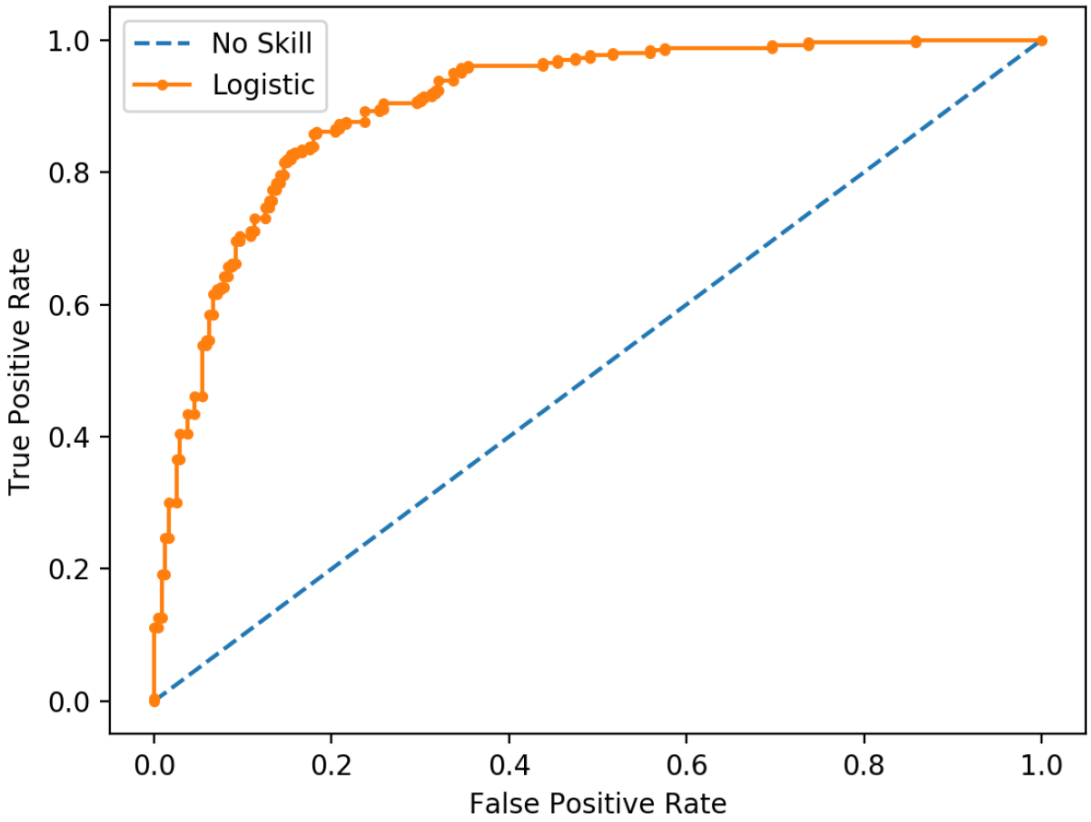


Figura 4 – Exemplo de curva ROC para um modelo logístico (*Logistic*) e um modelo aleatório (*No skill*)

Fonte: Baseado em Brownlee (2020)

A curva ROC é uma ferramenta bastante útil para avaliação de um modelo. Porém, quando se trata de comparação entre modelos, usar a curva é um desafio. Podemos, no entanto, calcular a área abaixo da curva ROC, conhecida como ROC AUC (*Receiver Operating Characteristic Area Under the Curve*). Essa é uma pontuação que varia de 0 a 1, e 1 indica o modelo perfeito.

3.5.2 Curva PR e PR AUC

A *Precisão (Precision)* é uma métrica que relaciona o número de verdadeiros-positivos em relação ao total de previsões positivas (verdadeiras e falsas).

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos} \tag{3.16}$$

O *Recall* é uma métrica que relaciona o o número de verdadeiros-positivos em relação ao total de observações positivas.

$$Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos} \tag{3.17}$$

Ambas as métricas são focadas na classe positiva. Para problemas com dados desbalanceados, métricas com foco específico na classe minoritária podem ser uma característica importante.

A curva PR é gerada sobre o gráfico que une as duas métricas: *Recall* no eixo x e *Precisão* no eixo y. Portanto, um modelo perfeito seria aquele com *Precisão* e *Recall* máximos (coordenada 1,1).

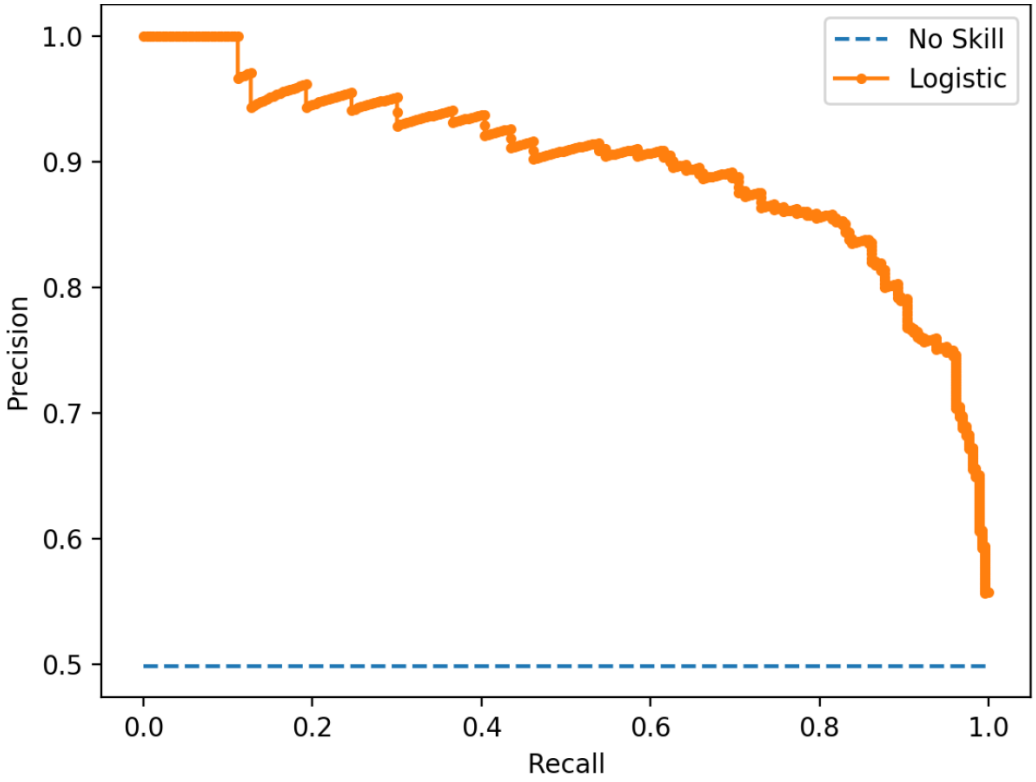


Figura 5 – Exemplo de curva PR para um modelo logístico (*Logistic*) e um modelo aleatório (*No skill*) em um problema com dados balanceados

Fonte: Baseado em Brownlee (2020)

Um modelo aleatório (*No skill*) é aquele que não consegue discriminar bem entre as classes e prevê uma classe aleatória ou uma classe constante em todos os casos. A linha que representa o modelo aleatório muda com base na distribuição das classes positivas e negativas, e é representada pela proporção de casos positivos no conjunto de dados. Para um conjunto de dados balanceado, a linha ficaria em 0,5. Já para um problema com dados desbalanceados, em que o evento de interesse ocorre para apenas 0,02 da população, o modelo aleatório estaria em 0,02.

Assim como com a curva ROC, podemos calcular a área abaixo da curva PR, chamada de PR AUC (*Precision Recall Area Under the Curve*). Essa é uma pontuação que varia de 0 a 1, e 1 indica o modelo perfeito.

3.6 *Cross validation*

No treinamento de modelos, muitas vezes acontece o chamado *overfitting*. O *overfitting* ocorre quando o modelo treinado fica super ajustado aos dados de treinamento, mas possui uma capacidade preditiva significantemente inferior quando aplicado em novos conjuntos de dados.

O *Cross validation* é uma poderosa metodologia que pode servir tanto para identificarmos o *overfitting*, como para simplesmente avaliarmos e compararmos modelos, e consiste basicamente na separação da base de dados em treino, utilizado para estimação do modelo, e teste, utilizado para avaliação de performance.

Dentre as variações do *Cross validation*, neste trabalho iremos utilizar o *k-fold Cross Validation*, que consiste na divisão do conjunto total de dados em k subconjuntos, denominados *folds*. A partir daí, um subconjunto é utilizado para teste, e os demais $k - 1$ restantes para treinamento. O processo é repetido k vezes até que todos os *folds* tenham sido utilizados como base de teste e tenham sido computados as performances. A Figura 6 exemplifica de forma visual o processo.

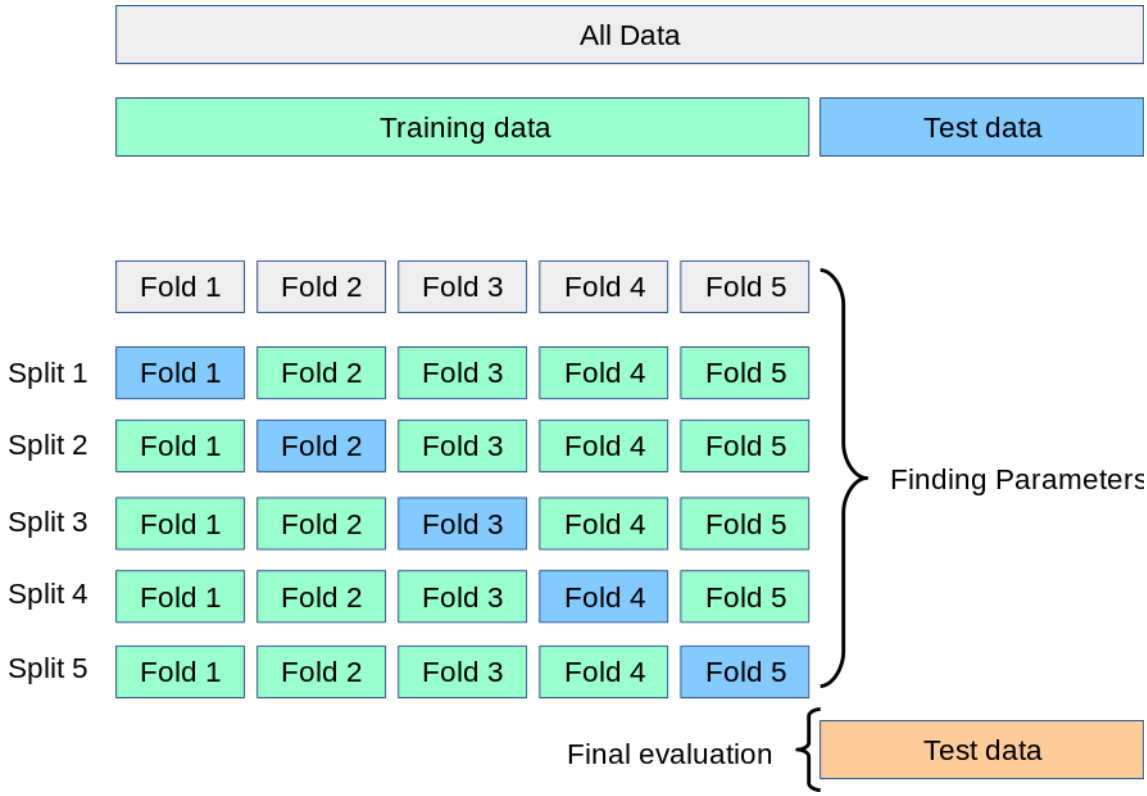


Figura 6 – Exemplo visual da metodologia *k-fold Cross Validation*

Dentre as vantagens da metodologia, podemos destacar que ao realizarmos múltiplos treinamentos e testes, isso nos proporciona uma população de indicadores de performances, possibilitando algumas estatísticas importantes para comparação de modelos como a média que pode nos dar uma ideia do quão bom o modelo pode ser no problema, e o desvio-padrão, que pode nos indicar o quanto a performance do modelo pode variar na prática.

Em problemas com dados muito desbalanceados, precisamos fazer um ajuste para execução do *k-fold Cross Validation*. Isso deve ser feito para que não existam *folds* com muito poucas ou sem observações da classe minoritária, o que pode ser um problema sério

na etapa de treinamento dos modelos. Para evitarmos essa situação, para geração dos *folds* usamos a seleção por amostragem estratificada, para que seja mantida a proporção de classes em cada subconjunto.

Para aplicação do *k-fold Cross Validation com amostragem estratificada* neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através das classes `RepeatedStratifiedKFold` e `cross_val_score`.

3.7 *Wilcoxon signed-ranks* - WSR

Testes estatísticos podem ser utilizados para validar a comparação das medidas de avaliação das técnicas de *machine learning* aplicadas. Pode-se usar os testes para interpretar os resultados das múltiplas execuções de cada uma das técnicas, obtidas através do *Cross-Validation*. Dessa forma, pode-se realizar afirmações sobre os resultados, considerando determinada técnica melhor que a outra no problema estudado.

Os testes estatísticos podem ser divididos entre paramétricos e não-paramétricos. Testes paramétricos têm requisitos sobre a natureza ou a forma das populações envolvidas. Testes não-paramétricos não exigem que as amostras venham de populações com distribuições normais ou qualquer outra distribuição particular, portanto, são chamados de testes livres de distribuição. Isso permite que os testes não-paramétricos se apliquem a uma grande variedade de situações.

Dentre os testes não-paramétricos, o teste de *Wilcoxon signed-ranks* (WSR) apresentado por Wilcoxon (1945), pode ser utilizado para avaliação do desempenho de dois classificadores, comparando-os pelas diferenças da métrica de avaliação utilizada. As diferenças são classificadas de acordo com seus valores absolutos. O objetivo do teste é avaliar:

- H_0 : A diferença entre os pares segue uma distribuição simétrica em torno de zero;
- H_1 : A diferença entre os pares não segue uma distribuição simétrica em torno de zero.

De acordo com Triola (2013), podemos seguir 7 passos para realização do *WSR*:

1. Para cada par de dados X_1 e Y_1, ache a diferença $d_1 = (X_1 - Y_1)$. Descarte quaisquer pares para os quais $d = 0$.
2. Ignore os sinais das diferenças, ordene-as da menor para a maior e as substitua pelo valor do posto correspondente. Quando as correspondências tiverem o mesmo valor numérico, associe a elas a média dos postos envolvidos no empate.

3. Atribua a cada posto o sinal da diferença que o originou. Isto é, insira os sinais que foram ignorados no Passo 2.

4. Ache a soma dos postos positivos. Ache o valor absoluto da soma dos postos negativos.

5. Considere T a menor das duas somas encontradas no Passo 4.

6. Seja n o número de pares de dados para os quais a diferença d não é 0.

7. Determine a estatística do teste e os valores críticos com base no tamanho amostral.

    - Se $n <= 30$, a estatística do teste é T.
    - Se $n > 30$, a estatística do teste é:

$$z = \frac{T - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \tag{3.18}$$

O nível de significância para o teste é no máximo α.

Para aplicação do *Wilcoxon signed-ranks* neste trabalho, utilizamos a biblioteca `scipy.stats` do Python, através da classe `wilcoxon`.

4 Metodologia

Neste capítulo apresentamos os passos referentes ao desenvolvimento dos modelos propostos baseados nos conceitos teóricos vistos no capítulo anterior.

4.1 Definição do evento de interesse - *Default*

Para este trabalho, a variável a ser modelada (variável resposta ou variável dependente) buscou seguir os critérios estabelecidos pelo Banco Central do Brasil, Circular Nº 3.648 (2013).

O descumprimento, ou *default*, é definido como à ocorrência de ao menos uma obrigação da empresa tomadora perante a instituição financeira estar em atraso há mais 90 dias. Além disso, existem algumas situações em que a instituição financeira pode entender que o tomador não irá honrar às obrigações e também classifica-lo em *default*, como solicitação de falência ou concordata por parte da empresa tomadora.

Já a Probabilidade de descumprimento (ou Probabilidade de *default* - PD), é o percentual que corresponde à expectativa de longo prazo das taxas de *default*, para o horizonte temporal de 12 meses.

Tabela 1 – Definição da variável dependente

Classe	Descrição
Não-*Default*	Cliente adimplente por 12 meses após a referência
Default	Cliente migrou para estado de *default* em algum momento dos 12 meses após a referência

4.2 Construção da base de dados

O conjunto de dados utilizado no trabalho foi adquirido junto à um grande banco brasileiro e basicamente consiste na composição e tratamento de duas bases de dados. A primeira, é constituída de demonstrativos financeiros anuais (Ativo, Passivo e DRE) de 14.603 empresas referentes aos anos de 2013 à 2017. A segunda base de dados é referente às marcações de *default* e empréstimos tomados de empresas durante o período de janeiro de 2014 à dezembro de 2019.

A base de dados final utilizada no processo de modelagem é constituída de CNPJs que, em algum momento durante a janela de observações do estudo, possuíram saldos devedores e que possam ser observados em uma janela de 12 meses à frente quanto

a existência ou não de *default*. Por isso, o período de janeiro à dezembro de 2019 foi reservado apenas para observações do evento de *default*, dado que qualquer CNPJ com saldo selecionado durante esse período não seria possível a observação de 12 meses à frente.

Para composição da base de dados final, foram realizados alguns tratamentos que devem ser ressaltados. Primeiramente, empresas dos setores financeiros foram desconsideradas. Isso se deve ao fato de sua contabilidade ser bastante distinta dos demais setores. Além disso, foi considerada uma materialidade de saldo devedor. Empresas com saldos pequenos (abaixo de R$ 12 mil reais) foram descartadas. Por fim, para mantermos a independência entre as observações, bastante importante em diversas técnicas de modelagem, garantimos que apenas uma empresa (CNPJ) de um determinado grupo econômico (conglomerado de empresas com relações interdependentes e que possuem uma mesma matriz (*holding*)), fosse considerada na base final.

Para os casos de empresas componentes de grupos econômicos, existem basicamente três situações que devem ser consideradas para seleção do CNPJ que irá compor a base final. Para grupos econômicos que possuem empresas que não entraram em *default* durante o período de estudo, devemos apenas garantir que um único CNPJ componha a base final, conforme a Figura 7.

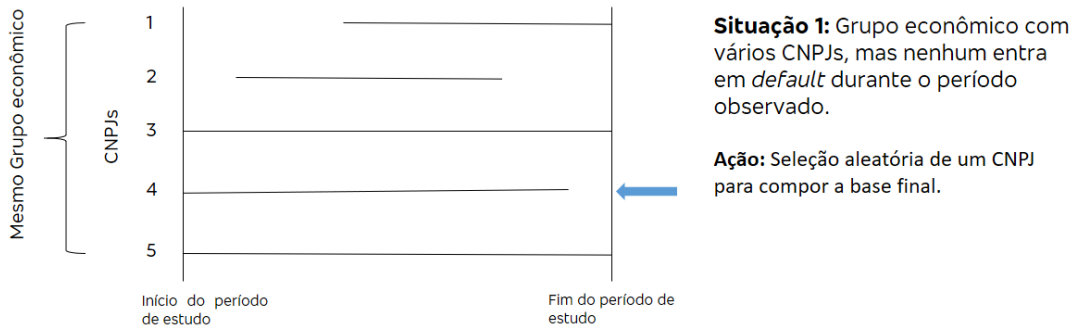


Figura 7 – Nenhum CNPJ do Grupo econômico entra em *default*

Caso algum CNPJ do grupo econômico entre em *default* durante o período do estudo, este CNPJ deverá ser o representante na base final, pois representa o evento de interesse a ser modelado, ilustrado na Figura 8.

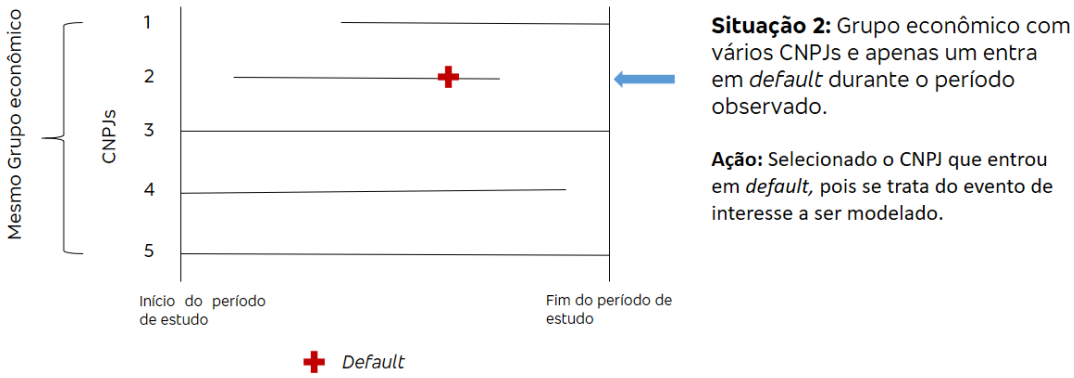


Figura 8 – Apenas um CNPJ do Grupo econômico entra em *default*

Porém, se vários CNPJs do grupo econômico entrarem em *default* durante o período do estudo, será selecionado o CNPJ que primeiro entrou em *default*, como observado na Figura 9 .

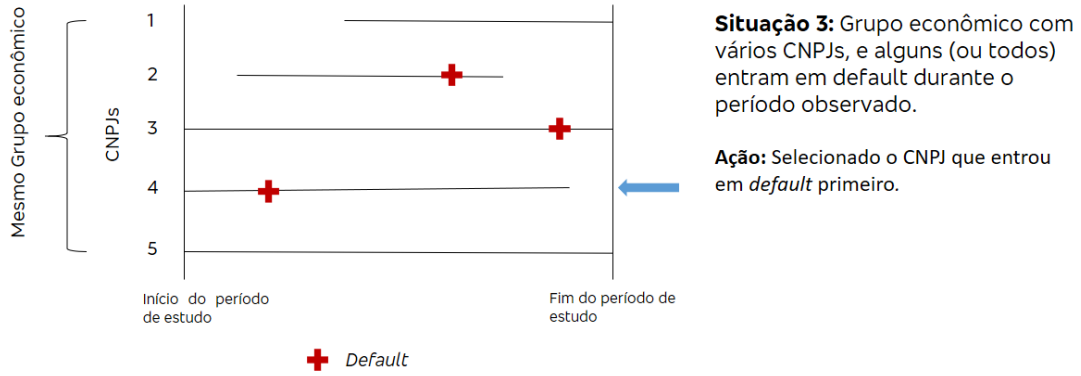


Figura 9 – Vários CNPJs do Grupo econômico entram em *default*

Após selecionados todos os CNPJs que irão compor a base de dados final, o próximo passo é selecionar ao longo do tempo, qual será o corte transversal de cada CNPJ. Essa seleção é importante para definir qual o demonstrativo financeiro será selecionado. Aqui, algumas regras também devem ser cumpridas.

Primeiramente, para os casos que entraram em *default*, o demonstrativo selecionado será imediatamente o do ano anterior ao *default*, pois é a informação mais recente disponível na data do *default*. Na Figura 10, são ilustrados dois exemplos.

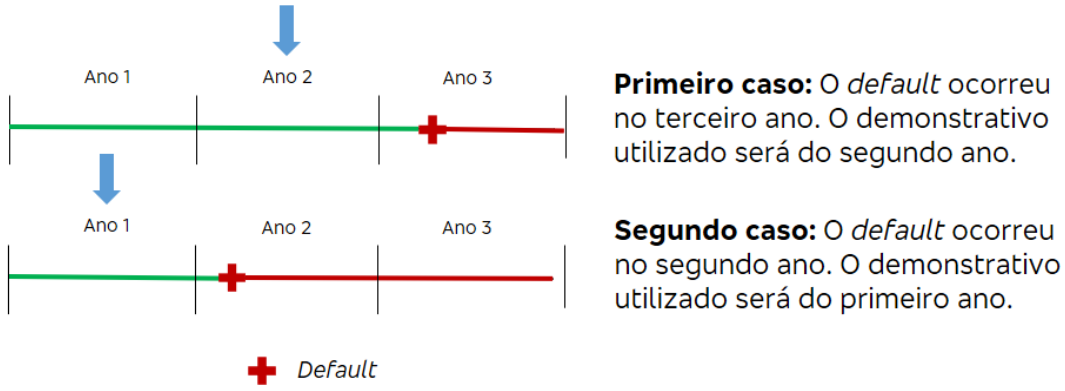


Figura 10 – Seleção do demonstrativo financeiro - Casos *default*

Para os casos que não entraram em *default*, o demonstrativo financeiro será selecionado aleatoriamente, como exemplificado na Figura 11.

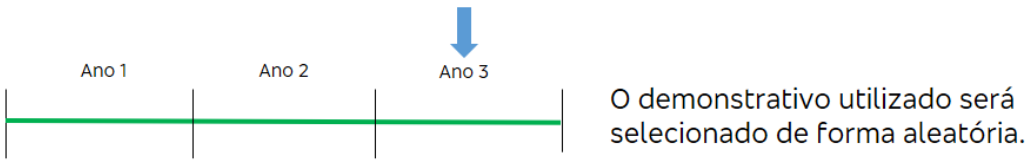


Figura 11 – Seleção do demonstrativo financeiro - Casos não-*default*

Após cumpridos os passos descritos, a base de dados final estará composta e poderá ser utilizada no processo de treinamento de modelos, a ser descrito nas próximas seções.

4.3 Variáveis utilizadas

A análise financeira de uma empresa permite conhecer sua situação e nos ajuda a antecipar situações futuras. Dentro deste contexto, uma das técnicas mais utilizadas pela análise financeira de um negócio são os índices financeiros. Índices financeiros podem ser definidos como instrumentos utilizados para sintetização de uma grande quantidade de informação e permitem acompanhar o desempenho econômico-financeiro das empresas ao longo do tempo. São baseados, essencialmente, em informações presentes nos demonstrativos financeiros das empresas (Balanço Patrimonial, DRE e Fluxo de Caixa).

Os índices financeiros podem sintetizar alguns aspectos da empresa, como sua estrutura (bens e direitos e os meios para os financiar), solvabilidade (capacidade de cumprimento dos compromissos de médio e longo prazo), liquidez (capacidade de cumprimento dos compromissos de curto prazo), lucratividade (capacidade da empresa de gerar caixa), etc.

Neste trabalho, nos baseamos em 25 índices financeiros consolidados na literatura por suas contribuições na predição de *default* de empresas. Os índices utilizados foram extraídos do trabalho de Back, Laitinen e Sere (1996b), que fizeram um levantamento histórico bastante representativo para seleção deste grupo de indicadores. Os índices são apresentados na tabela 2.

Tabela 2 – Índices financeiros utilizados

Índices		Tipo
R1	Caixa / Passivo Circulante	L
R2	Caixa / Faturamento Líquido	L
R3	Caixa / Ativos totais	L
R4	Ativo circulante / Passivo circulante	L
R5	Ativo circulante / Faturamento Líquido	L
R6	Ativo circulante / Ativos totais	L
R7	Ativo circulante / Patrimônio líquido	L
R8	Patrimônio líquido / Ativo permanente	S
R9	Patrimônio líquido / Faturamento líquido	S
R10	Estoques / Faturamento líquido	L
R11	Dívida de LP / Patrimônio líquido	S
R12	Dívida total / Patrimônio líquido	S
R13	Resultado líquido / Ativos totais	P
R14	Disponibilidades / Estoques	L
R15	Faturamento líquido / Ativos totais	P
R16	Resultado operacional / Ativos totais	P
R17	EBIT/ Despesas financeiras	L
R18	Disponibilidades / Passivo Circulante	L
R19	Disponibilidades / Faturamento líquido	L
R20	Disponibilidades / Ativos totais	L
R21	Lucros acumulados / Ativos totais	P
R22	Dívida total / Ativos totais	S
R23	Capital de giro / Faturamento líquido	L
R24	Capital de giro / Patrimônio líquido	L
R25	Capital de giro / Ativos totais	L
Tipo: L=Liquidez, P=Lucratividade, S=Solvabilidade		

Além dos indicadores financeiros, incluímos uma variável setorial: *CNAE (Classificação Nacional de Atividades Econômicas)*. O principal objetivo da inclusão desta variável é que existem divergências de patamares dos índices financeiros entre setores (existem setores que naturalmente trabalham mais alavancados, por exemplo). Então a expectativa é que com a inclusão dessa variável, as técnicas testadas consigam incrementar o poder de discriminação nos resultados.

4.4 Amostragem de dados e medidas de avaliação

A base de treinamento das técnicas de modelagem consiste em 70% do total de dados, selecionados aleatoriamente, sendo os 30% restantes reservados para teste.

Para avaliação dos modelos, foi utilizado o *k-fold Cross Validation com amostragem estratificada* com k igual a 10 e sendo repetido 3 vezes, o que totaliza um total de 30 treinamentos para cada técnica. Como utilizamos amostragem estratificada, a proporção entre as classes *default* e *não-default* foram mantidas entre os *folds*. Os resultados de cada técnica foram plotados em gráficos *Boxplot* para facilitar a visualização dos resultados.

As aplicações das técnicas de balanceamento de classes foram feitas apenas para os dados de treinamento, para não gerar distorções nos dados separados para teste (uma vez que a técnica $SMOTE$ gera dados sintéticos).

A medida de avaliação utilizada para comparação dos modelos foi a ROC AUC, sendo apresentados posteriormente as curvas ROC e PR nos dados separados para teste.

4.5 Pré-processamento

4.5.1 Dados discrepantes

Para identificação e exclusão dos casos discrepantes, utilizamos o método *Interquartile Range Method - IQR*. O método se baseia na diferença entre o primeiro e o terceiro quartil dos dados, chamado IQR, como podemos observar na figura 12.

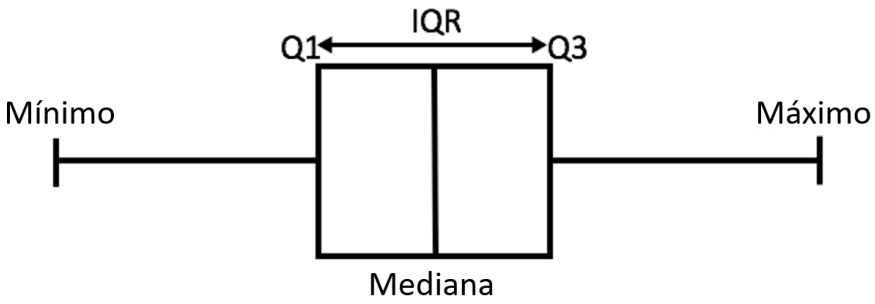


Figura 12 – Boxplot de um conjunto de dados hipotético

Após a identificação da amplitude IQR, definimos uma nova amplitude baseada nela, chamada de amplitude de decisão. Todos os dados que caírem fora da amplitude de decisão são considerados dados discrepantes e são eliminados da base de dados. Os limites inferiores e superiores da amplitude de decisão são:

- Limite inferior: (Q1 - X * IQR)

- Limite superior: (Q3 + X * IQR)

onde X é a constante a ser definida. Em nosso trabalho, calibramos X de modo a não perder mais do que 5% dos dados em cada variável.

Esta metodologia é especialmente útil pois não assume normalidade nos dados e elimina os dados discrepantes com bastante facilidade e eficiência. O contraponto é que pode-se perder mais informação do que o necessário.

4.5.2 Dados ausentes

Como mencionado na seção 3.2.1, grande parte das metodologias de modelagem não aceitam dados ausentes na base de treinamento. Portanto, é essencial a identificação, avaliação e tratamento dos dados ausentes.

Uma das melhores formas de tratarmos dados ausentes é através de metodologias de proximidade, em especial utilizando o *K Nearest Neighbor(KNN) Imputation*. Segundo Troyanskaya et al. (2001), a utilização do *KNN* é mais indicada e robusta que métodos mais simples como substituição dos dados ausentes por zero, ou atribuição de estatísticas como a média da variável.

Para aplicação do *KNN* neste trabalho, utilizamos a biblioteca `scikit-learn` do Python, através da classe `KNNImputer`, configurada com k igual a 5 vizinhos mais próximos e distância euclidiana como métrica.

5 Resultados

Neste capítulo serão apresentados os resultados obtidos pelo estudo e aplicação da metodologia descrita anteriormente. Será apresentada a análise exploratória dos dados, buscando caracterizar melhor a base, e identificar os problemas de dados ausentes e discrepantes para então aplicarmos as técnicas de modelagem apresentadas.

5.1 Pré-processamento

Nesta seção serão mostrados os resultados do pré-processamento da base de dados: tratamento de valores discrepantes e valores ausentes.

5.1.1 Valores discrepantes

Como mencionado na seção 3.2.2, dados discrepantes podem impactar algumas técnicas de modelagem. Em nosso problema, aplicamos a metodologia *Interquartile Range Method - IQR* para identificação e exclusão dos dados discrepantes.

Como podemos observar na tabela 3, houve uma redução de 19% na quantidade de total de casos. Uma preocupação seria se, os casos eliminados não seriam justamente casos do nosso evento de interesse. A proporção de *defaults* se manteve em 12% na base sem *outliers*, o que indica que os casos eliminados foram aleatórios e na mesma proporção.

Tabela 3 – Quantidade de casos - Base com e sem *outliers*

Descrição	# CNPJs	# *Defaults*	% *Default*
Base original	14.603	1.795	12%
Base sem *outliers*	11.821	1.424	12%

5.1.2 Valores ausentes

Após o tratamento dos dados discrepantes, apenas duas variáveis apresentaram valores ausentes, como mostrado na tabela 4.

Tabela 4 – Quantidade de valores ausentes por variável

Variável	# *Missings*	% *Missings*
R1	0	0,0%
R2	0	0,0%
R3	0	0,0%
R4	0	0,0%
R5	0	0,0%
R6	0	0,0%
R7	0	0,0%
R8	0	0,0%
R9	0	0,0%
R10	0	0,0%
R11	0	0,0%
R12	0	0,0%
R13	0	0,0%
R14	2435	20,6%
R15	0	0,0%
R16	0	0,0%
R17	742	6,3%
R18	0	0,0%
R19	0	0,0%
R20	0	0,0%
R21	0	0,0%
R22	0	0,0%
R23	0	0,0%
R24	0	0,0%
R25	0	0,0%

Analisando a causa dos valores ausentes identificados na variável *R14 - Disponibilidades / Estoques*, constatamos que se tratam apenas de casos que não possuem *Estoques*. Estes casos representam uma divisão por zero, causando o o valor ausente na base de dados. Os casos de valores ausentes apresentados na variável *R17 - EBIT/ Despesas financeiras* é parecida. São casos que não possuem *Despesas Financeiras*, o que resulta em uma divisão por zero. Como apresentado na seção 4.5.2, utilizamos a técnica *k nearest neighbor - KNN* para preenchimento dos dados ausentes.

5.2 Análise exploratória dos dados

A base analisada possui 14.603 registros de CNPJs, sendo que 1.795 (12,3%) entraram em *default* e 12.808 (87,7%) não entraram em estado de *default*. As variáveis independentes utilizadas são os 25 índices financeiros listados na seção 4.3 e a Classificação Nacional de Atividades Econômicas - CNAE, totalizando 26 variáveis a serem testadas: CNAE, Caixa / Passivo Circulante (R1), Caixa / Faturamento Líquido (R2), Caixa / Ativos

totais (R3), Ativo circulante / Passivo circulante (R4), Ativo circulante / Faturamento Líquido (R5), Ativo circulante / Ativos totais (R6), Ativo circulante / Patrimônio líquido (R7), Patrimônio líquido / Ativo permanente (R8), Patrimônio líquido / Faturamento líquido (R9), Estoques / Faturamento líquido (R10), Dívida de LP / Patrimônio líquido (R11), Dívida total / Patrimônio líquido (R12), Resultado líquido / Ativos totais (R13), Disponibilidades / Estoques (R14), Faturamento líquido / Ativos totais (R15), Resultado operacional / Ativos totais (R16), EBIT/ Despesas financeiras (R17), Disponibilidades / Passivo Circulante (R18), Disponibilidades / Faturamento líquido (R19), Disponibilidades / Ativos totais (R20), Lucros acumulados / Ativos totais (R21), Dívida total / Ativos totais (R22), Capital de giro / Faturamento líquido (R23), Capital de giro / Patrimônio líquido (R24) e Capital de giro / Ativos totais (R25).

Para entender melhor a distribuição das variáveis, nas tabelas 5, 6 e 7 são mostradas as características das variáveis independentes da base, tais como média, desvio padrão, primeiro quartil (Q25), mediana (Q50) e terceiro quartil (Q75) de todo o conjunto de dados e de dois recortes: um contendo somente CNPJs que entraram em *default* (tabela 6) e outro contendo somente os CNPJs que não entraram em *default* (tabela 7)).

Tabela 5 – Descrição estatística da base por variáveis - Base total

Variável	Média	Dv. Padrão	Q25	Q50	Q75
R1	0,69	3,13	0,04	0,14	0,45
R2	0,17	1,97	0,01	0,03	0,10
R3	0,11	0,15	0,02	0,05	0,14
R4	2,60	5,45	1,02	1,45	2,42
R5	0,76	4,00	0,24	0,40	0,65
R6	0,62	0,25	0,44	0,66	0,84
R7	2,86	34,10	0,27	0,83	2,26
R8	7,45	78,70	0,59	1,31	3,16
R9	0,80	4,78	0,08	0,23	0,53
R10	0,09	0,24	0,00	0,00	0,12
R11	0,63	8,93	-	0,03	0,34
R12	1,28	15,39	0,00	0,26	1,06
R13	0,03	0,18	- 0,01	0,02	0,09
R14	23.794,37	658.682,85	2,12	851,88	8.359,01
R15	1,95	1,93	0,86	1,44	2,36
R16	0,08	0,24	- 0,00	0,05	0,15
R17	54,31	1.261,23	0,61	1,76	5,88
R18	2,02	4,98	0,66	1,05	1,81
R19	0,62	3,98	0,16	0,28	0,47
R20	0,45	0,22	0,28	0,43	0,60
R21	0,14	0,24	-	0,01	0,20
R22	0,20	0,22	0,02	0,14	0,32
R23	0,28	2,27	0,01	0,10	0,29
R24	0,75	20,22	0,09	0,53	0,91
R25	0,19	0,32	0,01	0,18	0,38

Tabela 6 – Descrição estatística da base por variáveis - Só *defaults*

Variável	Média	Dv. Padrão	Q25	Q50	Q75
R1	0,20	0,71	0,02	0,06	0,15
R2	0,08	0,27	0,01	0,02	0,06
R3	0,06	0,09	0,01	0,03	0,07
R4	1,59	1,96	0,85	1,16	1,67
R5	0,78	2,29	0,27	0,44	0,72
R6	0,58	0,24	0,39	0,60	0,78
R7	2,42	33,52	0,42	1,31	3,81
R8	6,26	86,42	0,26	0,79	1,71
R9	0,68	5,93	0,04	0,15	0,41
R10	0,10	0,18	0,00	0,00	0,14
R11	0,74	10,00	-	0,22	0,91
R12	1,50	20,13	0,17	0,91	2,41
R13	- 0,02	0,16	- 0,05	0,00	0,03
R14	24.789,25	197.297,82	2,09	902,68	8.838,28
R15	1,60	1,60	0,76	1,21	1,94
R16	0,00	0,18	- 0,04	0,01	0,05
R17	3,31	19,74	0,20	1,00	1,91
R18	1,22	1,77	0,54	0,82	1,30
R19	0,64	2,28	0,18	0,30	0,52
R20	0,42	0,22	0,26	0,40	0,56
R21	0,09	0,17	-	0,00	0,09
R22	0,33	0,21	0,17	0,31	0,46
R23	0,15	1,79	- 0,06	0,05	0,21
R24	0,77	11,61	- 0,07	0,42	1,02
R25	0,07	0,33	- 0,07	0,08	0,24

Tabela 7 – Descrição estatística da base por variáveis - Só *Não-defaults*

Variável	Média	Dv. Padrão	Q25	Q50	Q75
R1	0,76	3,33	0,05	0,16	0,50
R2	0,18	2,10	0,01	0,04	0,11
R3	0,12	0,15	0,02	0,06	0,15
R4	2,73	5,75	1,04	1,50	2,54
R5	0,75	4,18	0,23	0,39	0,64
R6	0,63	0,25	0,45	0,67	0,84
R7	2,92	34,18	0,26	0,78	2,11
R8	7,61	77,58	0,66	1,40	3,42
R9	0,82	4,60	0,08	0,24	0,55
R10	0,09	0,24	0,00	0,00	0,12
R11	0,61	8,77	-	0,02	0,28
R12	1,25	14,62	0,00	0,22	0,90
R13	0,04	0,18	- 0,01	0,03	0,09
R14	23.658,11	698.543,80	2,12	849,14	8.243,09
R15	2,00	1,97	0,88	1,48	2,40
R16	0,09	0,24	0,00	0,06	0,16
R17	61,30	1.344,67	0,68	2,00	6,82
R18	2,13	5,26	0,69	1,08	1,89
R19	0,62	4,16	0,16	0,27	0,46
R20	0,45	0,22	0,29	0,44	0,61
R21	0,15	0,24	-	0,02	0,22
R22	0,18	0,21	0,01	0,12	0,29
R23	0,30	2,33	0,01	0,11	0,30
R24	0,75	21,13	0,11	0,55	0,90
R25	0,20	0,32	0,02	0,20	0,40

Analisando os dados apresentados nas três tabelas descritivas, o índice financeiro que chama atenção de imediato é o *R14 - Disponibilidades / Estoques*, pelos altos valores de Média, Desvio padrão, Q50 e Q75. Analisando mais a fundo os motivos dessa discrepância, observamos que, por não haver concentração de empresas de um determinado setor, muitos casos não possuem Estoque (ou possuem valores muito baixos, fazendo com que o índice fique alto), a exemplo dos setores de Turismo, Hotelaria, Transportes, Serviços, etc. A inclusão de uma variável como CNAE na base de treinamento tem como objetivo justamente levar essa informação para as técnicas de modelagem testadas na expectativa que algumas delas consigam separar casos em que o índice financeiro seja relevante.

Um outro índice financeiro que chama a atenção é o *R17 - EBIT/ Despesas financeiras* devido ao alto valor do Desvio padrão. Analisando este indicador, observamos que a volatilidade é causada pela diferença natural no perfil das empresas na base de dados. Existe uma parcela da base com empresas chamadas *aplicadoras*. São empresas que não possuem dívida, ou possuem dívidas financeiras muito baixas e possuem recursos aplicados junto aos bancos, o que acaba refletindo no índice financeiro mencionado. Por

outro lado, existem empresas com bastante dívida, que podem ser dívidas consideradas "boas"(natural para financiamento de projetos, investimentos, benefícios fiscais, etc.) ou dívidas consideradas "ruins"(por exemplo, dívidas altas podem causar falta de flexibilidade financeira, fazendo a empresa recusar projetos que possam agregar valor), que poderiam indicar justamente um incremento na probabilidade de *default*.

Aqui, vemos uma das vantagens de trabalhar com poucas variáveis explicativas. Conseguimos ter uma interpretação de cada uma delas, e fazer uma compreensão das causas de algumas discrepâncias, o que muitas vezes não é possível se estivermos trabalhando com grandes bases de dados (*big data*). Os demais índices financeiros utilizados neste estudo possuem comportamento descritivo dentro do esperado.

5.3 Estimação dos modelos

As técnicas apresentadas foram aplicadas em duas bases de dados: Base de dados bruta e Base de dados com transformação de Box-Cox e tratamento de desbalanceamento de classes. Para apuração da performance dos modelos em cada uma das bases, foram separados 70% dos dados para treinamento e 30% para teste. Na base de treinamento, as estimações foram feitas utilizando o *k-fold Cross Validation com amostragem estratificada* com k igual a 10 e sendo repetido 3 vezes, totalizando 30 estimações para cada uma das 6 metodologias, em cada uma das bases de dados. Posteriormente, os modelos com melhor performance foram aplicados na base de teste, e suas performances avaliadas.

5.3.1 Base de dados bruta

Inicialmente, após os tratamentos referentes ao pré-processamento apresentados anteriormente, as seis metodologias de classificação propostas foram estimadas na base bruta (sem transformações de dados e tratamento de desbalanceamento de classes). Os resultados de desempenho ROC AUC são apresentados na tabela 8 e na figura 13. Como podemos observar, todos os modelos apresentaram um desempenho bom (ROC AUC acima de 0,7), com exceção da regressão logística (LR) que apresenta ROC AUC de 0,519, o que é pouca coisa acima do modelo aleatório hipotético (ROC AUC de 0,5).

Tabela 8 – Desempenho ROC AUC na base bruta - Médias e desvios-padrões

Modelo	ROC AUC
LR	0,519 ±0,029
LDA	0,737 ±0,023
BAG	0,755 ±0,019
RF	0,765 ±0,019
ADA	0,714 ±0,024
STK	0,765 ±0,018

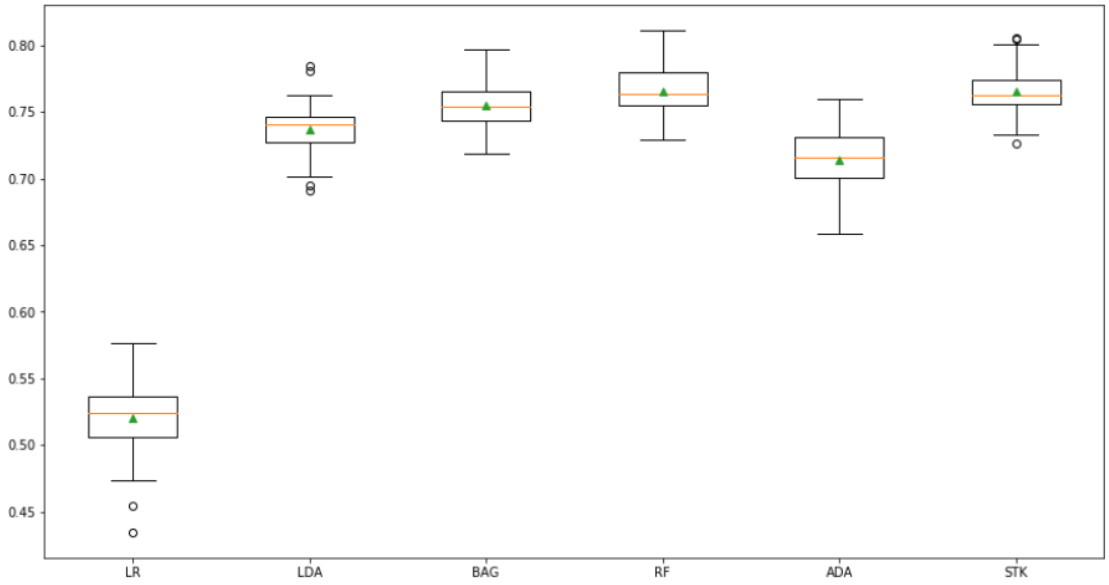


Figura 13 – Boxplot do desempenho ROC AUC dos modelos testados na base bruta

Os modelos que apresentaram os melhores desempenhos foram o Random Forest (RF) e o Stacking (STK), ambos com ROC AUC médio de 0,765. Como foram os modelos campeões, aplicamos o teste *WSR* para compará-los entre si e com as demais técnicas. Com nível de significância $\alpha = 0{,}05$, foi obtido um *p-value* de 0,2133 no comparativo RF x STK, indicando não haver diferenças significativas entre os resultados das duas técnicas. Todas as demais técnicas, LR, LDA, BAG e ADA, apresentaram *p-values* abaixo de 0,05, indicando haver diferenças significativas quando comparadas com as técnicas campeãs RF e STK, no nível $\alpha = 0{,}05$ de significância.

Para ambos os modelos campeões, foram testadas as performances também na base de testes (30% dos dados separados inicialmente). Como podemos observar nas figuras 14 e 15, ambas as técnicas mantiveram um bom poder de discriminação, mantendo os mesmos patamares das estimações na base de treinamento, com o RF atingindo ROC AUC de 0,786 e o STK um ROC AUC de 0,782.

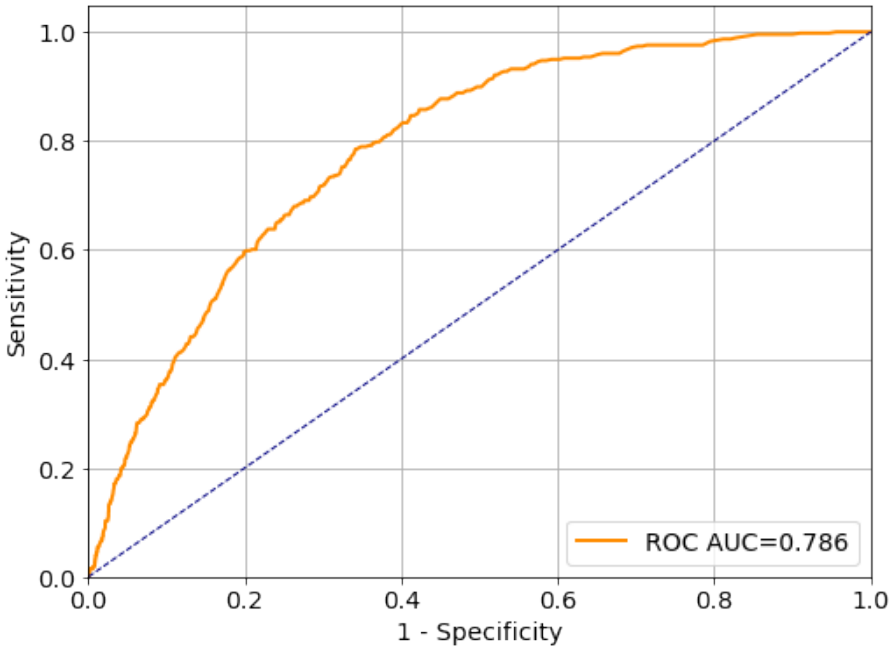


Figura 14 – Curva ROC - RF em dados brutos - Base de teste

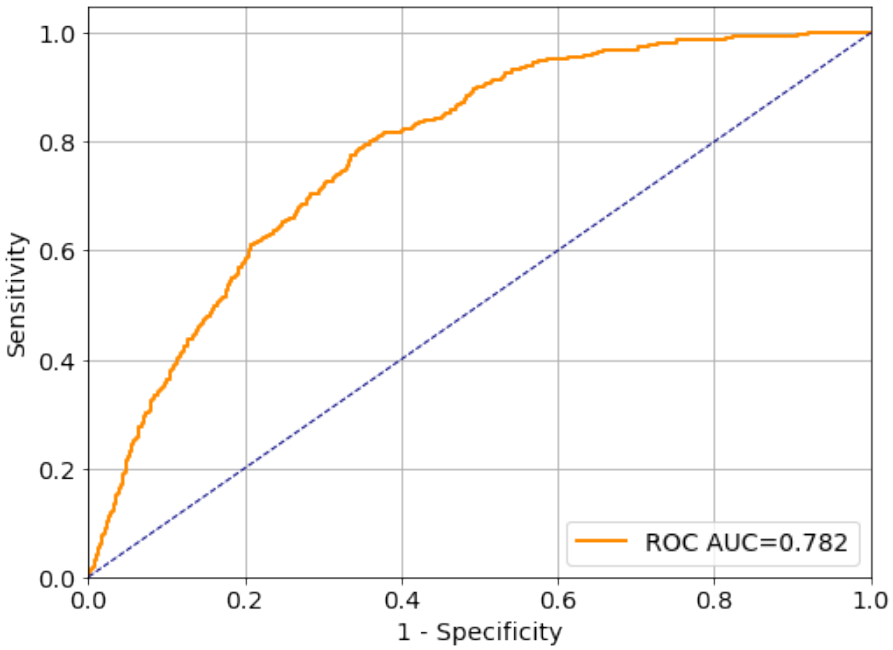


Figura 15 – Curva ROC - STK em dados brutos - Base de teste

Os modelos também foram estimados e avaliados com uma métrica mais focada na classe minoritária (*default*), a Curva PR e a PR AUC. Sabemos que maximizar a Precisão minimizará o número de falsos positivos e maximizar o *Recall* irá minimizar o número de falsos negativos. Essa característica é bastante importante e dependendo do objetivo do problema, o PR AUC é mais adequado que a ROC AUC.

Como podemos observar na tabela 9 e na figura 16, seguindo o que já havia sido observado pela métrica ROC AUC, o modelo LR ficou com performance PR AUC bastante inferior às demais metodologias. Os modelos com melhores desempenhos médios foram o RF e o STK, com PR AUC de 0,296 e 0,292. Pelo teste *WSR*, com nível de significância α = 0,05, foi obtido um *p-value* de 0,1650, indicando não haver diferenças significativas entre as duas técnicas campeãs.

Tabela 9 – Desempenho PR AUC na base bruta - Médias e desvios-padrões

Modelo	PR AUC
LR	0,124 ±0, 010
LDA	0,239 ±0, 025
BAG	0,286 ±0, 038
RF	0,296 ±0, 043
ADA	0,244 ±0, 027
STK	0,292 ±0, 040

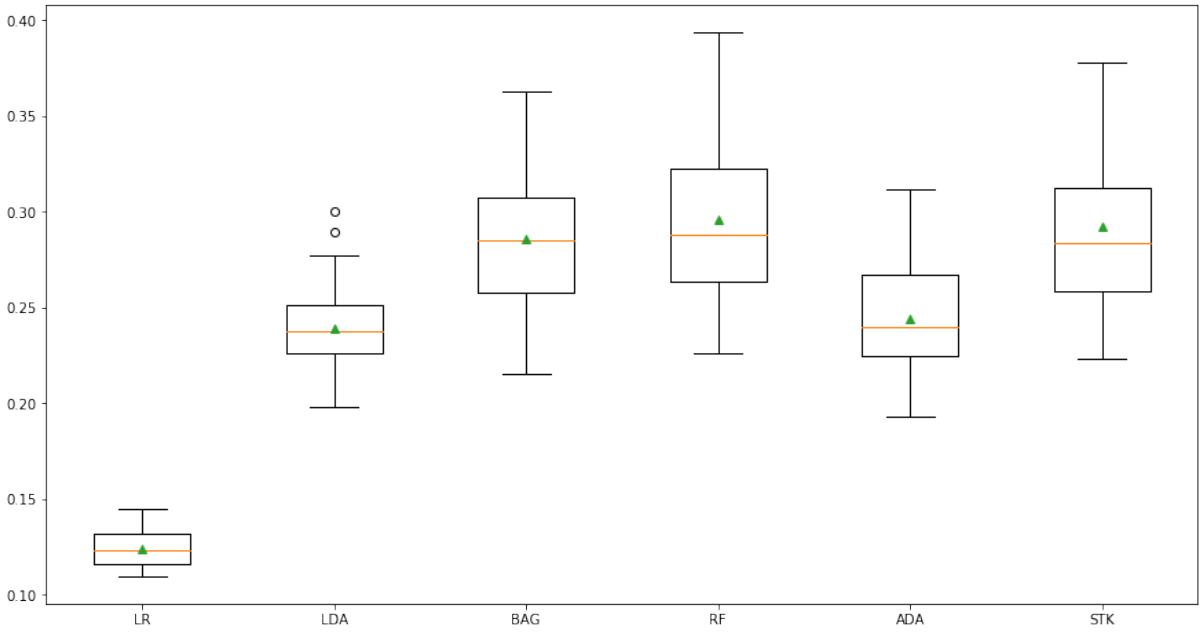


Figura 16 – Boxplot do desempenho PR AUC dos modelos testados na base bruta

5.3.2 Base de dados com transformação de Box-Cox e tratamento de desbalanceamento

Com o objetivo de analisar o impacto das transformações de dados e dos tratamentos de desbalanceamento de classes, as seis técnicas foram reestimadas na base de dados após a transformação de Box-Cox e SMOTE+ENN. Como podemos observar na tabela 11 e na figura 17, houve um grande ganho de performance média da ROC AUC na técnica LR, saindo de 0,519 na base bruta para 0,750 na base com as transformações. As demais técnicas se mantiveram no mesmo patamar de performance atingido com os dados brutos.

Tabela 10 – Desempenho ROC AUC na base com transformação Box-Cox e SMOTE+ENN - Médias e desvios-padrões

Modelo	ROC AUC
LR	0,750 ±0, 021
LDA	0,750 ±0, 020
BAG	0,761 ±0, 020
RF	0,769 ±0, 022
ADA	0,731 ±0, 023
STK	0,746 ±0, 024

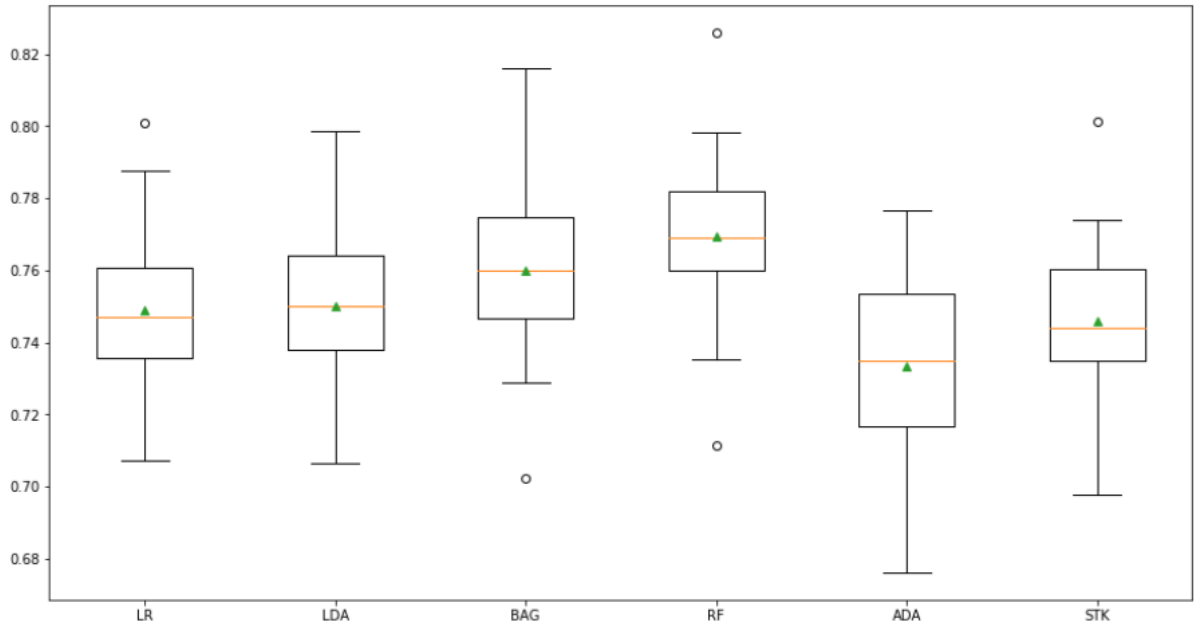


Figura 17 – Boxplot do desempenho (ROC AUC) dos modelos testados com transformação Box-Cox nos dados e SMOTE+ENN (*under + oversampling*)

O modelo que apresentou a melhor performance foi novamente o RF, com ROC AUC médio de 0,769. Aplicamos o teste *WSR* nos resultados encontrados pelo RF na base bruta e na base com as transformações de dados e tratamentos de desbalanceamento. Com nível de significância $\alpha = 0{,}05$, observamos um *p-value* de 0,3600, indicando não haver diferenças significativas entre os resultados. Isso indica que, apesar de ser a técnica com melhor performance na base após os tratamentos, o RF não apresentou ganhos significativos em relação aos resultados atingidos pela mesma técnica na base bruta (0,765).

A performance do RF na base de teste é apresentada na figura 18. Novamente, observamos que a performance na base de testes é bastante boa (ROC AUC de 0,788), se mantendo nos níveis encontrados na base de treinamento.

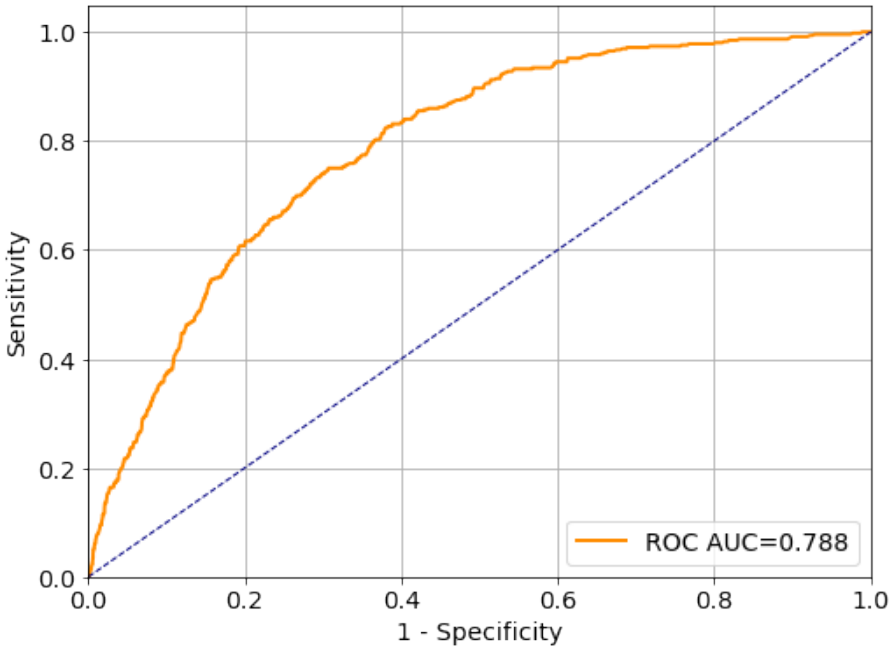


Figura 18 – Curva ROC - RF em dados com transformação Box-Cox e SMOTE+ENN - Base de teste

Os resultados de performance com a métrica PR AUC seguiram o mesmo padrão, com o modelo LR apresentando forte melhora comparado com os resultados da técnica na base bruta (0,261 contra 0,124), e o RF se apresentando como a melhor técnica em desempenho médio. Os resultados podem ser observados nas tabelas 11 e na figura 19.

Tabela 11 – Desempenho PR AUC na base com transformação Box-Cox e SMOTE+ENN - Médias e desvios-padrões

Modelo	PR AUC
LR	0,261 ±0, 025
LDA	0,262 ±0, 025
BAG	0,289 ±0, 039
RF	0,299 ±0, 043
ADA	0,256 ±0, 024
STK	0,285 ±0, 040

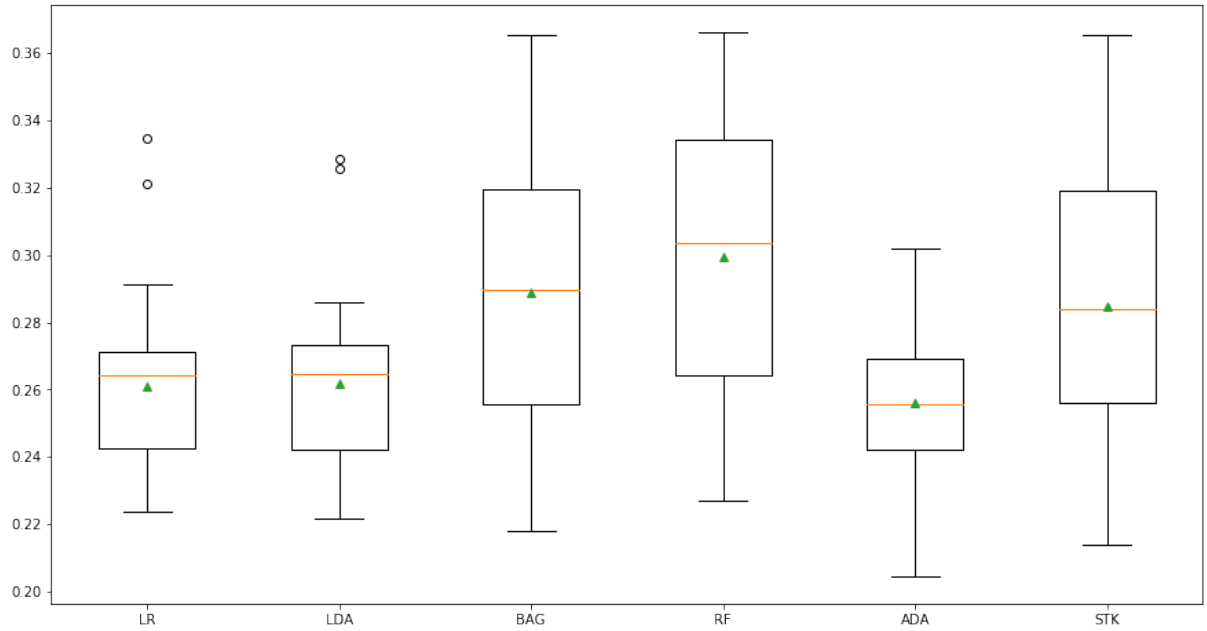


Figura 19 – Boxplot do desempenho PR AUC dos modelos testados na base com transformação Box-Cox e SMOTE+ENN

5.4 Discussão

Dentre as técnicas utilizadas, a que apresentou maior variação nos resultados foi a Regressão Logística (LR). Uma das vantagens desta técnica, é que ela não faz tantas suposições quanto à forma funcional das variáveis independentes, como grande parte dos modelos lineares. Por exemplo, as variáveis independentes não precisam seguir uma distribuição normal multivariada. Porém, diversos estudos mostram que a regressão logística atinge resultados mais estáveis e superiores em dados que seguem uma distribuição normal, ou que possuem forma próximo a ela.

Observamos que, após a transformação de Box-Cox nas variáveis independentes e a aplicação do tratamento de desbalanceamento SMOTE+ENN, a performance da Regressão Logística (LR) aumentou consideravelmente, se colocando em proximidade com as demais técnicas testadas.

Dentre as quatro técnicas *Ensembles* testadas, observamos que o *Random Forest* (RF) se destacou em todos os cenários. Porém, não houveram variações significativas em sua performance após as transformações de dados e tratamentos de desbalanceamento. Isso indica que a metodologia é bastante robusta quando à forma dos dados.

Uma das principais desvantagens do *Random Forest* (RF), é a dificuldade na explicabilidade exata das variáveis componentes finais do modelo. A técnica é considerada um *"Black-box"* quando composta por muitas árvores, pois é bastante difícil uma direta interpretação dos motivos pelo qual o modelo está dando uma alta ou baixa probabilidade

de inadimplência nesse tipo de problema. Em aplicações de decisão de crédito por exemplo, caso ocorra uma divergência entre a percepção do analista de crédito com o apontado pelo modelo, não haveria uma orientação direta e objetiva dos motivos. Esse é um dos principais contrastes em relação às técnicas mais tradicionais, como a Regressão Logística (LR), pois a explicabilidade dos modelos nestas técnicas é um dos grandes pontos fortes.

6 Conclusão

O principal objetivo desse estudo foi a comparação de desempenho entre técnicas selecionadas de *machine learning* em uma aplicação de probabilidade de empresas entrarem em *default* em 12 meses. Esse tipo de problema é bastante importante para instituições financeiras pois pode ter aplicações diretas na mensuração do risco de crédito, como na decisão de empréstimos ou no cálculo de capital regulatório.

Para o desenvolvimento do trabalho, foram utilizadas seis técnicas de classificação, sendo duas delas modelos lineares (Regressão Logística (LR) e Análise Discriminante Linear (LDA)), e quatro técnicas *Ensembles* (*Bagging* (BAG), *Random Forest* (RF), *Adaboost* (ADA) e *Stacking* (STK)). Foram utilizados dados referentes à 14.603 empresas, sendo que 1.795 entraram em *default* no horizonte estudado.

Para construção dos modelos, foram utilizadas variáveis construídas através dos demonstrativos financeiros anuais das empresas (Ativo, Passivo e DRE), pois além de uma boa confiabilidade, os demonstrativos financeiros possuem informações que refletem a saúde financeira das empresas. As variáveis construídas através dos demonstrativos financeiros se basearam no trabalho de Back, Laitinen e Sere (1996b).

Os resultados mostraram que o *Random Forest* (RF) foi a técnica com melhor desempenho, independente da métrica e dos tratamentos utilizados. Porém, após transformações e tratamento no balanceamento dos dados, a Regressão Logística (LR), que é a técnica mais comum em utilização no mercado, atingiu patamares mais próximos de performance, indicando ser uma boa opção em aplicações que ainda dependem de alta explicabilidade dos modelos.

Em estudos futuros, sugere-se a inclusão de uma maior quantidade de variáveis independentes, de diversas naturezas (não só dos demonstrativos financeiros), como variáveis comportamentais de relacionamento com a instituição financeira e variáveis macroeconômicas. A tendência é que a performance dos métodos *Ensembles* se mostrem ainda mais superiores principalmente pela alta capacidade de captação de relações não lineares. Além disso, um processo de otimização de hiperparâmetros também poderiam maximizar os resultados, não explorado nesse trabalho devido ao alto custo computacional.

Referências

AGGARWAL, C. C. *Outlier Analysis*. [S.l.]: Springer, 2017.

ALTMAN, E. I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, Wiley for the American Finance Association, v. 23, n. 4, p. 589–609, 1968.

ALTMAN, E. I.; HALDEMAN, R. G.; NARAYANAN, P. Zeta analysis: A newmodel to identify bankruptcy risk of corporations. *Journal of Banking and Finance*, Elsevier, v. 1, n. 1, p. 29–54, 1977.

ANTUNES, A.; RIBEIRO, N.; ANTAO, P. Estimating probabilities of default under macroeconomic scenarios. *Financial Stability Report*, p. 115–124, 2005.

BACK, B.; LAITINEN, T.; SERE, K. Choosing bankruptcy predictors using discriminant analysis, logit analysis, and genetic algorithms (technical report no 40). *Turku Centre for Computer Science*, 1996.

BACK, B.; LAITINEN, T.; SERE, K. Neural networks and genetic algorithms for bankruptcy predictions. *Expert Systems with Applications*, v. 11, p. 407–413, 1996.

BANCO CENTRAL DO BRASIL, CIRCULAR Nº 3.648. 2013. Disponível em: <https://www.bcb.gov.br/htms/Normativ/CIRCULAR3648.pdf>. Acesso em: 01 mai. 2020.

BATISTA, G.; PRATI, R.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 2004.

BEAVER, W. H. Financial ratios as predictors of failure. *Journal of Accounting Research*, Wiley on behalf of Accounting Research Center, v. 4, p. 71–111, 1966.

BOX, G. E. P.; COX, D. R. An analysis of transformations. *Journal of the Royal Statistical Society*, v. 26, p. 211–252, 1964.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, p. 123–140, 1996.

BROWNLEE, J. *Imbalanced Classification with Python*. v1.2. [S.l.: s.n.], 2020.

BRUCE, P.; BRUCE, A. *Practical Statistics for Data Scientists*: 50 essential concepts. [S.l.]: O'reilly, 2019.

CARLING, K. et al. Corporate credit risk modeling and the macroeconomy. *Journal of Banking Finance*, Elsevier, v. 31, n. 3, p. 845–868, 2007.

FERRI, C.; HERNáNDEZ, J.; MODROIU, R. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, Elsevier, v. 30, p. 27–38, 2009.

FIGINI, S.; FANTAZZINI, D. Random survival forests models for sme credit risk measurement. *Methodology and Computing in Applied Probability*, Elsevier, v. 11, n. 1, p. 29–45, 2009.

FIRTH, D. Bias reduction of maximum likelihood estimates. *Biometrika*, Oxford University Press on behalf of Biometrika Trust, v. 80, n. 1, p. 27–38, 1993.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936.

FITZPATRICK, P. J. A comparison of the ratios of successful industrial enterprises with those of failed companies. *Certif. Public Account*, v. 2, p. 598–605, 1932.

GERON, A. *Hands on machine learning with scikit learn and tensorflow*: Concepts, tools and techniques to build intelligent systems. [S.l.]: O'reilly, 2019.

H.WOLPERT, D. Stacked generalization. *Neural Networks*, v. 5, p. 241–259, 1992.

LESSMANN, S. et al. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, Elsevier, v. 247, n. 1, p. 124–136, 2015.

MERWIN, C. L. Financing small corporations: In five manufacturing industries, 1926-36. National Bureau of Economic Research, 1942.

MUCHLINSKI, D. et al. Comparing random forest with logistic regression for predicting class-imbalanced civilwar onset data. *Political Analysis*, Cambridge University Press on behalf of the Society for PoliticalMethodology, v. 24, n. 1, p. 87–103, 2016.

OHLSON, J. A. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, Wiley on behalf of Accounting Research Center, Booth School of Business, University of Chicago, v. 18, n. 1, p. 109–131, 1980.

PINO, F. A. A questão da não normalidade: uma revisão. *Rev. de Economia Agrícola*, v. 61, p. 17–33, 2014.

PLUTO, K.; TASCHE, D. Estimating probabilities for low default portfolios. 2005. Disponível em: <https://ssrn.com/abstract=635301>.

SCHAFER, J. L.; GRAHAM, J. W. Missing data: our view of the state of the art. *Psychological methods*, v. 7, p. 147–177, 2002.

SIGRIST, F.; HIRNSCHALL, C. Grabit: Gradient tree-boosted tobit models for default prediction. *Journal of Banking Finance*, Elsevier, v. 102, p. 177–192, 2019.

TAM, K. Y.; KIANG, M. Y. Managerial applications of neural networks: The case of bank failure predictions. *Management Science*, INFORMS, v. 38, n. 7, p. 926–947, 1992.

TRIOLA, M. F. *Elementary Statistics*. 11th. ed. [S.l.]: Pearson, 2013.

TROYANSKAYA, O. et al. Missing value estimation methods for dna microarrays. *Bioinformatics*, v. 17, p. 520–525, 2001.

WALLACE, B. C.; DAHABREH, I. J. Class probability estimates are unreliable for imbalanced data (and how to fix them). *IEEE 12th International Conference on Data Mining*, 2012.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, v. 1, n. 6, p. 80–83, 1945.

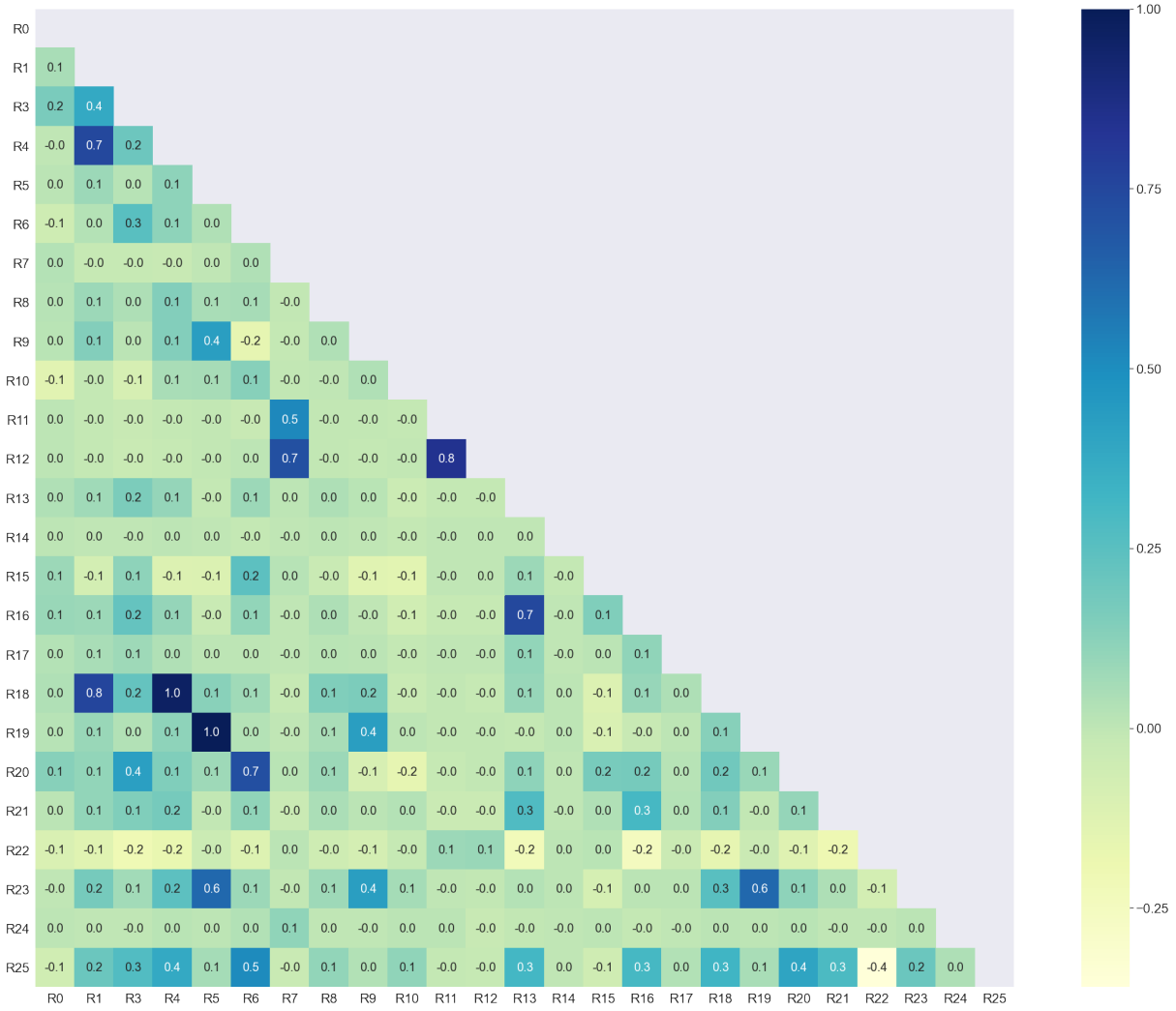
WILSON, R. L.; SHARDA, R. Bankruptcy prediction using neural networks. *Decision Support Systems*, v. 11, p. 545–557, 1994.

WINAKOR, A.; SMITH, R. Changes in the financial structure of unsuccessful in- dustrial corporations. *Bureau of Business Research*, Bull 51, 1935.

YEO, I.-K.; JOHNSON, R. A new family of power transformations to improve normality or symmetry. *Biometrika*, v. 87, 12 2000.

Apêndices

APÊNDICE A – Correlações de Pearson entre as variáveis independentes



CNAE (R0), Caixa / Passivo Circulante (R1), Caixa / Faturamento Líquido (R2), Caixa / Ativos totais (R3), Ativo circulante / Passivo circulante (R4), Ativo circulante / Faturamento Líquido (R5), Ativo circulante / Ativos totais (R6), Ativo circulante / Patrimônio líquido (R7), Patrimônio líquido / Ativo permanente (R8), Patrimônio líquido / Faturamento líquido (R9), Estoques / Faturamento líquido (R10), Dívida de LP / Patrimônio líquido (R11), Dívida total / Patrimônio líquido (R12), Resultado líquido / Ativos totais (R13), Disponibilidades / Estoques (R14), Faturamento líquido / Ativos totais (R15), Resultado operacional / Ativos totais (R16), EBIT/ Despesas financeiras (R17), Disponibilidades / Passivo Circulante (R18), Disponibilidades / Faturamento líquido (R19), Disponibilidades / Ativos totais (R20), Lucros acumulados / Ativos totais (R21),

Dívida total / Ativos totais (R22), Capital de giro / Faturamento líquido (R23), Capital de giro / Patrimônio líquido (R24) e Capital de giro / Ativos totais (R25).

APÊNDICE B – Modelos estimados - Parâmetros

B.1 Regressão Logística (LR)

```
LogisticRegression(
    C=1.0, class_weight=None, dual=False, fit_intercept=True,
    intercept_scaling=1, l1_ratio=None, max_iter=100,
    multi_class='auto', n_jobs=None, penalty='l2',
    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
    warm_start=False))
```

B.2 Análise Discriminante Linear (LDA)

```
LinearDiscriminantAnalysis(
    n_components=None, priors=None, shrinkage=None,
    solver='svd', store_covariance=False, tol=0.0001))
```

B.3 Bagging (BAG)

```
BaggingClassifier(
    base_estimator=None, bootstrap=True, bootstrap_features=False,
    max_features=1.0, max_samples=1.0, n_estimators=1000,
    n_jobs=None, oob_score=False, random_state=None, verbose=0,
    warm_start=False))
```

B.4 Random Forest (RF)

```
RandomForestClassifier(
    bootstrap=True, ccp_alpha=0.0, class_weight=None,
    criterion='gini', max_depth=None, max_features='auto',
    max_leaf_nodes=None, max_samples=None,
    min_impurity_decrease=0.0, min_impurity_split=None,
    min_samples_leaf=1, min_samples_split=2,
    min_weight_fraction_leaf=0.0, n_estimators=1000,
    n_jobs=None, oob_score=False, random_state=None,
    verbose=0, warm_start=False))
```

B.5 Boosting (ADA)

```
AdaBoostClassifier(
    algorithm='SAMME.R', base_estimator=None, learning_rate=1.0,
    n_estimators=1000, random_state=None))
```

B.6 Stacking (STK)

Para estimação do Stacking (STK), os cinco modelos utilizados nesse trabalho, Regressão Logística (LR), Análise Discriminante Linear (LDA), Bagging (BAG), Random Forest (RF) e Boosting (ADA), foram utilizados como *base-models*. Como *meta-model*, foi utilizada a Regressão Logística (LR).